

UNIVERSIDAD DE CANTABRIA



**DEPARTAMENTO DE INGENIERÍA DE
COMUNICACIONES**

TESIS DOCTORAL

**TÉCNICAS DE MODELADO NO LINEAL CON
INCORPORACIÓN DE RESTRICCIONES EN LAS
DERIVADAS Y SU APLICACIÓN A
COMUNICACIONES**

Marcelino Lázaro Teja

2001

UNIVERSIDAD DE CANTABRIA



**DEPARTAMENTO DE INGENIERÍA DE
COMUNICACIONES**

TESIS DOCTORAL

**TÉCNICAS DE MODELADO NO LINEAL CON
INCORPORACIÓN DE RESTRICCIONES EN LAS
DERIVADAS Y SU APLICACIÓN A
COMUNICACIONES**

Autor : Marcelino Lázaro Teja

**Directores : Ignacio Santamaría Caballero
Carlos Pantaleón Prieto**

Grupo de Tratamiento Avanzado de Señal

2001

Tesis Doctoral: Técnicas de modelado no lineal con incorporación de restricciones en las derivadas y su aplicación a comunicaciones

Autor: Marcelino Lázaro Teja

Directores: Ignacio Santamaría Caballero
Carlos Pantaleón Prieto

El tribunal nombrado para juzgar la tesis doctoral citada, compuesto por los señores

Presidente:

Vocales:

Secretario:

Acuerda otorgarle la calificación de

Santander, a de de 2001

*A los días de lluvia, redentores de la tierra y de mis penas,
y a la luna, dueña de mis sueños.*

Agradecimientos

Me gustaría, en primer lugar, agradecer a mis directores de Tesis, Ignacio Santamaría Caballero y Carlos Pantaleón Prieto, su constante aportación de ideas, su paciencia, y su acertada labor de corrección a lo largo de la realización de esta Tesis, así como pedirles disculpas por algunos de los dolores de cabeza que, a buen seguro, les habré ocasionado.

Igualmente, tengo que agradecer la colaboración de los compañeros del Grupo de Microondas, su amabilidad al facilitarnos medidas reales de transistores y su disponibilidad para aclararnos en cualquier momento todas las dudas que sobre el modelado de este tipo de dispositivos nos han podido surgir.

Y especialmente, quisiera agradecer a todos mis compañeros del Grupo de Tratamiento Avanzado de Señal (GTAS todos, perrazos y jefazos) el maravilloso ambiente en el que he tenido la enorme fortuna de poder trabajar durante estos años, y todas las facilidades que me han puesto para realizar esta Tesis.

Prólogo

La reconstrucción de una función, sometida a una serie de restricciones en sus derivadas, es un problema que aparece en un gran número de aplicaciones, como el modelado de dispositivos de microondas cuando se desea representar de forma adecuada el fenómeno de la intermodulación, la economía, el análisis de sistemas caóticos, etc.

Tras realizar una revisión de las técnicas existentes para resolver este problema, en esta Tesis se proponen varias soluciones al mismo dentro de los ámbitos de interpolación, aproximación funcional y modelado.

En primer lugar, se propone una técnica para la interpolación simultánea de un conjunto de muestras de una función y sus derivadas. La solución que se obtiene pertenece al subespacio de funciones splines. La utilización de las muestras de las derivadas permite obtener una reconstrucción local de la función, utilizando un banco de filtros FIR, en lugar de la reconstrucción global propia de los splines convencionales, donde el valor de la función en un punto depende del valor de las muestras en todos los instantes de muestreo. Como todas las técnicas de interpolación, el modelo propuesto presenta una elevada sensibilidad al ruido, especialmente cuando las señales están muy sobremuestreadas. Para reducir esta sensibilidad, se propone una técnica de regularización del conjunto de muestras. Con esta alternativa se mejoran de forma sensible los resultados obtenidos. Los modelos de interpolación y de regularización propuestos se extienden también a espacios de entrada bidimensionales, donde se obtienen soluciones diferentes si se dispone o no de las derivadas cruzadas.

Los métodos de interpolación/regularización propuestos precisan tantos parámetros como número de medidas, que en general puede ser un número elevado, lo que dificulta su utilización en problemas donde se tienen restricciones en el número de parámetros. Para este tipo de problemas se propone un modelo suave y derivable, obtenido a partir de un modelo lineal a tramos, utilizando una función suave para implementar las transiciones entre tramos lineales. Este es un modelo parsimonioso, con un número reducido de parámetros, que se muestra muy eficaz para aproximar una función y generalizar de forma adecuada las derivadas de la misma gracias a sus condiciones de suavidad y derivabilidad. La suavidad de la solución se controla mediante un único parámetro, lo que establece una relación entre este modelo y la teoría de la regularización. En el entrenamiento del modelo, es posible utilizar la información de las derivadas, relacionadas con la suavidad, para adaptar dicho parámetro.

Otra vía que se explora es la utilización de redes neuronales en este problema. En este caso, se proponen dos alternativas para introducir la información de las derivadas

en la construcción de la red. Por un lado, se considera una nueva función de coste en el algoritmo de entrenamiento. Esta función de coste, que introduce la información de las derivadas, permite obtener una mejor aproximación de las mismas. Además, la introducción de la información de las derivadas en el entrenamiento de la red también mejora, en algunos casos, la reconstrucción de la función. Por otro lado, se propone introducir la información de las derivadas a través de la arquitectura de la red; se considera una arquitectura modular en la que módulos distintos se encargan de aproximar la función y las derivadas. En esta línea se presenta una arquitectura específica para el modelado global de transistores de microondas, válido tanto para gran señal como para pequeña señal, en el que se tienen módulos distintos para caracterizar el comportamiento pequeña señal, donde las derivadas resultan de gran importancia, y el comportamiento gran señal de los mismos. Los dos módulos se combinan utilizando funciones de pertenencia de lógica borrosa. Para el entrenamiento del módulo pequeña señal, implementado mediante una red GRBF, se propone un nuevo algoritmo de entrenamiento para redes RBF, basado en el algoritmo EM, que permite acelerar la convergencia y reducir la sensibilidad a los mínimos locales del entrenamiento.

Finalmente, los modelos propuestos se aplican al modelado de transistores de microondas. Se obtienen modelos gran señal, pequeña señal, válidos para análisis de intermodulación, y un modelo global (pequeña señal + gran señal).

Esta Tesis supone, en definitiva, una aportación a las técnicas de modelado no lineal de una función con la introducción de restricciones en las derivadas de la misma, que presentan numerosas aplicaciones en los campos de tratamiento digital de señales y de comunicaciones.

Preface

The reconstruction of a function, subject to a set of constraints in its derivatives, is a common problem in a number of applications, such as the nonlinear modeling of microwave devices to predict the intermodulation behavior, the economy, the analysis of chaotic systems, etc.

After reviewing the known techniques to solve this problem, in this Thesis several interpolation and modeling solutions are proposed.

First of all, a technique for the simultaneous interpolation of a set of samples of the function and its derivatives is proposed. The obtained solution belongs to the subspace of spline functions. The information of the derivatives allows a local reconstruction of the function, using a FIR filter bank instead of the global reconstruction, based on IIR filtering, typical of the conventional splines. Like other interpolation techniques, the proposed model presents a high noise sensitivity, especially when the signals are very oversampled. In order to reduce this sensitivity, a technique for the regularization of the samples is proposed. With this approach the obtained results improve notably. The proposed interpolation and regularization techniques are extended to bidimensional input spaces.

The proposed interpolation/regularization techniques requires as much parameters as the number of available samples; this makes them unsuitable in modeling problems where typically a small number of parameters are required. For this kind of problems, we propose a smooth and derivable model, extended from a piecewise linear model by using a smooth function to implement the transition between linear regions. This is a parsimonious model, with a reduced number of parameters, which is able of fitting a function and approximate its derivatives because of its characteristics of smoothness and derivability. A single parameter controls the smoothness of the model. This fact provides a link with the regularization theory; in particular, the parameter controlling the smoothness can be seen as a regularization parameter. In the model training, it is possible to make use of the information about de derivatives to adjust this parameter.

Another explored alternative is the utilization of neural networks in this problem. In this case, two alternatives are proposed to introduce the information of the derivatives in the construction of the network. In one hand, a new cost function for the training algorithm is considered. This cost function, which includes the information of the derivatives, allows to obtain a better approximation of the derivatives. Moreover, in some cases, the information of the derivatives can help to improve the fit of the function. In the other hand, we propose to introduce the information of the derivatives

through the network architecture; a modular architecture is considered in which a module approximates the function and other one the derivatives. A specific architecture is presented to the global modeling of microwave transistors; the model is valid to represent both the large and the small signal behavior of the transistors by using two different modules to characterize each kind of behavior. The two modules are combined, to provide the global solution, by using functions of fuzzy logic. For the training of the small signal module, implemented using a GRBF network, a new training algorithm, based in the EM algorithm, is proposed. This algorithm accelerates the convergence and reduces the sensitivity to local minima of the training process of RBF networks.

Finally, the proposed models are applied to the nonlinear modeling of microwave transistors. Large-signal, small-signal and global models (valid for both large and small-signal) are obtained.

To conclude, this Thesis represents a contribution to the nonlinear modeling of a function with the introduction of constraints in its derivatives, which is a problem that arises in a number of applications of digital signal processing and communications.

Índice General

1	Introducción	1
1.1	Reconstrucción de una función con restricciones en las derivadas: Planteamiento y motivación del problema	1
1.2	Planteamiento y objetivos de la Tesis	5
1.3	Organización y contribuciones de la Tesis	7
2	Interpolación y modelado con la incorporación de restricciones en las derivadas: Revisión de técnicas existentes	11
2.1	Interpolación en el subespacio de señales limitadas en banda	11
2.1.1	Generalización del Teorema del Muestreo de Shannon	12
2.1.2	Particularización para el caso de interpolación de secuencias discretas periódicas	14
2.1.3	Extensión a muestreo irregular	17
2.1.4	Método basado en proyecciones	19
2.2	Interpolación en subespacios generados por funciones de escala	21
2.2.1	Subespacios multiresolución generados por funciones de escala	22
2.2.2	Incorporación de la información de las derivadas	25
2.3	Interpolación en el subespacio de funciones splines	28
2.3.1	Introducción	28
2.3.2	Incorporación de la información de las derivadas	36
2.4	Interpolación regularizada	41
2.4.1	Teoría de la Regularización	41
2.4.2	Splines Regularizados	42
2.5	Redes Neuronales	42
2.5.1	Introducción	43
2.5.2	El Perceptrón Multicapa	44
2.5.3	La Red de Funciones de Base Radial	45
2.5.4	Capacidad de aproximación	47
2.5.5	Incorporación de la información de las derivadas	48
3	Una nueva técnica de interpolación/regularización para la reconstrucción simultánea de una función y sus derivadas	51
3.1	El Modelo Local de Interpolación (MLI) en espacios unidimensionales	52

3.1.1	Interpolación en el espacio de los splines	53
3.1.2	Representación mediante un banco de filtros	57
3.1.3	Sensibilidad al ruido	60
3.2	El Modelo Local Regularizado (MLR) en espacios unidimensionales . .	60
3.2.1	Modelo regularizado de grado 2	61
3.2.2	Selección de los parámetros de regularización	63
3.3	Extensión a espacios de entrada bidimensionales	64
3.3.1	Modelos de interpolación en espacios bidimensionales	64
3.3.2	Modelos regularizados en espacios bidimensionales	70
3.3.3	Extensión a D derivadas	73
3.4	Resultados	73
3.4.1	Resultados en espacios de entrada unidimensionales	73
3.4.2	Resultados en espacios de entrada bidimensionales	79
3.5	Discusión	87
4	El Modelo Lineal a Tramos Suavizado (MLTS)	89
4.1	Introducción	89
4.2	El Modelo Canónico Lineal a Tramos (MLT)	90
4.2.1	Presentación del modelo	90
4.2.2	Redes canónicas lineales a tramos	92
4.3	Suavizado de la función valor absoluto	94
4.4	El Modelo Lineal a Tramos Suavizado (MLTS)	99
4.4.1	Análisis de suavidad: relación con la teoría de la regularización .	100
4.4.2	Entrenamiento del modelo	103
4.5	Análisis de prestaciones	109
4.5.1	Resultados en espacios de entrada unidimensionales	109
4.5.2	Resultados en espacios de entrada bidimensionales	116
4.6	Discusión	120
5	Alternativas neuronales para el modelado de una función y sus derivadas	123
5.1	Introducción de la información de las derivadas en la función de coste .	124
5.1.1	Perceptrón Multicapa (MLP)	125
5.1.2	Red de Funciones de Base Radial Generalizada (GRBF)	125
5.1.3	Modelo Lineal a Tramos Suavizado (MLTS)	126
5.1.4	Resultados	127
5.2	Una arquitectura modular para la aproximación de una función y sus derivadas	130
5.2.1	Arquitectura propuesta	131
5.2.2	Arquitectura específica para el problema de modelado de transistores	133
5.2.3	Entrenamiento de la GRBF mediante un algoritmo EM	137
5.2.4	Análisis de prestaciones del modelo global	146

5.3	Discusión	148
6	Aplicación al modelado de transistores de microondas	151
6.1	Modelo gran señal de transistores HEMT bajo iluminación óptica utilizando un MLTS	151
6.1.1	Introducción	151
6.1.2	Resultados de modelado con el MLTS	153
6.2	Modelo pequeña señal de transistores MESFET y HEMT para análisis de intermodulación empleando una GRBF	156
6.2.1	Introducción	156
6.2.2	Caracterización no lineal de MESFET/HEMTs considerando el efecto de la intermodulación	157
6.2.3	Resultados de modelado de MESFETs y HEMTs	158
6.3	Modelo global de un MESFET	162
6.3.1	Módulo gran señal	162
6.3.2	Módulo pequeña señal	165
6.3.3	Modelo global	165
6.4	Modelo gran señal de un transistor con el MLR	166
7	Conclusiones y líneas futuras de trabajo	169
7.1	Aportaciones	169
7.2	Líneas futuras	172
7.2.1	SVM's en problemas de regresión con información de las derivadas	173
7.2.2	Otras líneas	176
A	Notación y Abreviaturas	179
A.1	Notación	179
A.1.1	Reglas generales	179
A.1.2	Aspectos específicos de la Tesis	179
A.2	Abreviaturas	181
B	Descomposiciones de funciones trigonométricas en fracciones parciales	183
C	Modelos MLI de grados 2 y 3	185
C.1	Modelo de grado 3	185
C.1.1	Expresiones de interpolación y reconstrucción	185
C.1.2	Representación mediante un banco de filtros	186
C.2	Modelo de grado 4	188
C.2.1	Expresiones de interpolación y de reconstrucción	189
C.2.2	Representación mediante un banco de filtros	191

D Modelos MLR de grados 3 y 4	193
D.1 Modelo de grado 3	193
D.2 Modelo de grado 4	195
E Demostración de la continuidad del modelo MSI	199
F Funciones de prueba en espacios bidimensionales	203
G Publicaciones a las que ha dado lugar la realización de la Tesis	207

Capítulo 1

Introducción

En este primer capítulo se hace una breve introducción del problema de la reconstrucción (interpolación), aproximación y modelado de una función sometida a una serie de restricciones en sus derivadas. En primer lugar se presenta el planteamiento y la motivación del problema, para a continuación describir los principales objetivos que se han perseguido con la realización de esta Tesis y las diferentes líneas de investigación en las que se ha trabajado para la consecución de tales objetivos. Para finalizar, se describe la organización que sigue esta Tesis, perfilando, además, las principales contribuciones que se han aportado en la misma.

1.1 Reconstrucción de una función con restricciones en las derivadas: Planteamiento y motivación del problema

El problema general de la reconstrucción de una determinada función, $f(x)$, a partir de una serie de valores de la misma, $y[n] = f(x[n])$, es un problema muy común en un gran número de disciplinas, que ha recibido una gran atención tanto en la literatura matemática [Whittaker35, Hoskins94], como en la de tratamiento de señal y comunicaciones [Shannon49, Jerri77, Marvasti93].

Este problema, planteado sin restricciones, pertenece a la categoría de problemas inversos mal condicionados en el sentido de que existe un conjunto infinito de posibles soluciones del mismo. Por lo tanto, es preciso imponer algún tipo de restricción o incorporar alguna información *a priori* que permita transformar el problema en bien condicionado. Algunas de las restricciones más comunes son, por ejemplo, las restricciones de suavidad [Poggio90], o de limitación en banda de la función [Shannon49]. En particular, en esta Tesis se aborda este problema cuando la información *a priori* la proporcionan las derivadas de la función. En algunos casos es posible disponer de esta información de forma explícita, mediante un conjunto de valores de las derivadas de la función, y en otros puede estar contenida de forma implícita en algún tipo de restricción que se impone a las derivadas de la solución: por ejemplo en forma de restricciones de

suavidad de la solución.

Existen diferentes formas de formular el problema que, en general, se pueden clasificar dentro de tres categorías:

- Interpolación o reconstrucción
- Aproximación funcional
- Modelado

En un problema de interpolación o reconstrucción funcional se fuerza que la solución ajuste de forma exacta el conjunto de muestras disponibles. Cuando se relaja esta restricción, bien porque se considera que puede ser beneficioso, como en presencia de ruido, o bien porque se establece alguna limitación en el número de parámetros de la solución, que hace imposible interpolar todas las muestras, consideraremos un problema de modelado.

Existen multitud de aplicaciones en las que realizar una aproximación lo más fiel posible de la función no resulta suficiente, sino que también es preciso obtener una aproximación razonable de las derivadas de la misma. En particular, la aplicación que originó el interés en este problema, y que puede considerarse la motivación de esta Tesis, es el modelado de dispositivos de microondas, y más concretamente el de transistores MESFET (*“Metal Semiconductor Field Effect Transistor”*) y HEMT (*“High Electron Mobility Transistor”*). Estos dos tipos de transistores son, sin duda, los más utilizados en aplicaciones de microondas y de ondas milimétricas, razón por la que han concentrado, en gran parte, las actividades de modelado durante años [Golio91, Anholt95]. En particular, en sistemas multiportadora el comportamiento no lineal de este tipo de transistores produce distorsión de intermodulación (IMD). Esto hace que la predicción y el modelado de los efectos de intermodulación sea un aspecto sumamente importante en el diseño de sistemas de comunicaciones de banda ancha.

La Figura 1.1 muestra el circuito equivalente no lineal más extendido de un MESFET en la región de saturación. El elemento no lineal predominante en los transistores MESFET y HEMT es la corriente drenador-fuente, I_{ds} , que depende de las tensiones drenador-fuente, V_{ds} , y puerta-fuente, V_{gs} . Esta dependencia se suele denominar característica corriente/tensión (I/V) del transistor. Cuando únicamente se desea reproducir el comportamiento gran señal de los transistores, en general, basta con modelar de forma adecuada la característica no lineal del dispositivo, es decir, la característica I/V [Fernández96]. Pero cuando además se desea modelar el comportamiento pequeña señal, siendo capaces de reproducir los efectos de la intermodulación, no basta con tener una buena aproximación de la característica I/V del dispositivo, sino que además es necesario caracterizar de forma adecuada las derivadas de la misma hasta el orden de los productos de intermodulación que se quiera modelar [Crosmun89, Maas91]. De esta manera, para modelar de forma adecuada los efectos de la intermodulación de hasta tercer orden, que es lo más habitual para mezcladores y amplificadores en aplicaciones de comunicaciones [Maas90], es necesario aproximar hasta la tercera derivada de la característica I/V del dispositivo. Así pues, en este tipo de aplicación va a ser

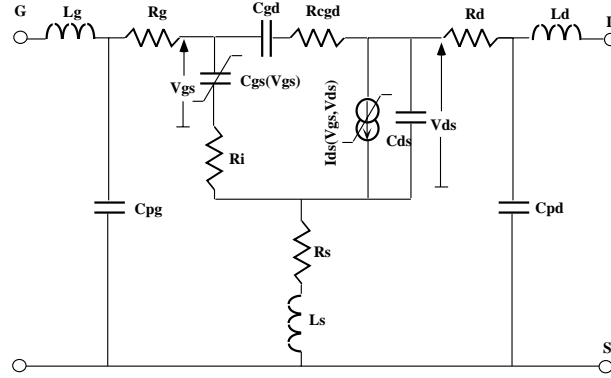


Figura 1.1: Circuito equivalente de un transistor MESFET

necesario obtener modelos que aproximen de forma adecuada hasta la tercera derivada de la característica I/V en la zona de operación de los dispositivos. Sin embargo, las técnicas no lineales convencionales aplicadas al modelado de transistores, tales como las ampliamente utilizadas funciones analíticas [Curtice85, McCamant90], los modelos lineales a tramos [Chua86], o el uso de tablas de datos [Root91], fallan a la hora de ajustar simultáneamente la característica I/V y sus derivadas.

En el modo de operación normal de los transistores MESFET y HEMT, cuando se aplica una pequeña señal de RF sobre un punto de polarización, la corriente I_{ds} depende de las tensiones de polarización, (V_{ds0}, V_{gs0}) , y de las tensiones instantáneas de pequeña señal, (v_{ds}, v_{gs}) . Partiendo de estas premisas, el problema general se puede plantear como el problema de encontrar una solución

$$\hat{I}_{ds} = f(V_{ds0}, V_{gs0}, v_{ds}, v_{gs}), \quad (1.1)$$

capaz de modelar la corriente I_{ds} , como función de las tensiones de polarización y de pequeña señal. Para reproducir el fenómeno de la intermodulación, el modelo debe también aproximar de forma adecuada hasta la tercera derivada de la función en el punto de polarización.

Las derivadas de la característica I/V , evaluadas en un determinado punto de polarización, pueden ser obtenidas experimentalmente a partir de la medida de potencias de intermodulación para una señal de excitación compuesta por dos tonos [Pedro94]. Así pues, es posible disponer de esta información a la hora de reconstruir la función. En esta línea, se han obtenido algunas soluciones: por ejemplo, en [Pedro94] se propone modelar la corriente I_{ds} en un intervalo alrededor del punto de polarización utilizando el desarrollo de Taylor en dos dimensiones truncado

$$\begin{aligned} I_{ds} = & I_{ds0} + G_m v_{gs} + G_{ds} v_{ds} + G_{m2} v_{gs}^2 + \\ & G_{md} v_{gs} v_{ds} + G_{d2} v_{ds}^2 + G_{m3} v_{gs}^3 + \\ & G_{m2d} v_{gs}^2 v_{ds} + G_{md2} v_{gs} v_{ds}^2 + G_{d3} v_{ds}^3, \end{aligned} \quad (1.2)$$

donde I_{ds0} es la corriente de polarización, v_{ds} y v_{gs} son los voltajes incrementales de drenador-fuente y puerta-fuente, respectivamente, y los parámetros (G_m, \dots, G_{d3}) son los coeficientes relacionados con las derivadas de la característica I/V evaluadas en el punto de polarización. Por ejemplo, G_{md2} vendría dado por

$$G_{md2} = \frac{1}{2} \frac{\partial^3 I_{ds}}{\partial v_{gs} \partial v_{ds}^2} \Bigg|_{(V_{gs0}, V_{ds0})}, \text{ en } v_{ds} = v_{gs} = 0 \quad (1.3)$$

En este caso, disponiendo de un modelo adecuado de las derivadas en distintos puntos de polarización, es posible obtener una reconstrucción local, en torno al punto de polarización, de la característica del dispositivo que sea válida para reproducir los efectos de intermodulación [García99, Cairon2000, Rodríguez2000]. Sin embargo, en todas estas contribuciones, la principal limitación está, precisamente, en la obtención de un modelo de las derivadas que reproduzca adecuadamente la forma de las mismas. Conviene finalmente destacar que, hasta la fecha, no existe ninguna solución conocida al problema global, esto es, la obtención de un modelo de la característica I/V (gran señal), válido cuando aplicamos un pequeño voltaje incremental sobre las tensiones de polarización (pequeña señal) y que, además, reproduzca los efectos de intermodulación (es decir, que modele adecuadamente las derivadas). Proporcionar y estudiar soluciones a este problema es la principal motivación de esta Tesis.

Además del modelado de transistores de microondas que, como ya se ha dicho, ha sido la principal motivación de este trabajo, existe un gran número de aplicaciones en las que las derivadas de la función a modelar resultan de gran importancia. Este es el caso, por ejemplo, de los trabajos realizados por Jordan [Jordan89] en el campo de la robótica, más concretamente en el aprendizaje de movimientos suaves. En este caso, la matriz que define el *Jacobiano*, que incluye las derivadas de la función, es la matriz que establece la relación entre pequeños cambios en la salida del controlador y pequeños cambios en las variables del espacio de las tareas, y en la mayoría de los casos no se puede obtener un conocimiento *a priori* de dicha matriz, ni tampoco puede ser proporcionada por el entorno. Sin embargo, todas las derivadas del Jacobiano son derivadas directas: es posible obtenerlas de forma sencilla mediante diferenciación si existe un modelo de la función. En este caso, las derivadas del modelo proporcionan el operador de aprendizaje que permite al sistema utilizar los errores sobre el espacio de las tareas para adaptar el controlador. De esta forma, el aprendizaje de una aproximación del Jacobiano a partir de una transformación desconocida definida por un conjunto de datos es un componente clave en el método de Jordan para el aprendizaje de movimientos suaves.

Este problema aparece también en aplicaciones dentro del campo de la economía: por ejemplo, en la teoría de la empresa y el consumo aparecen ciertas funciones, como funciones de producción, de costes, de venta, etc [Varian78], en las que consideraciones teóricas conducen a hipótesis acerca de las propiedades de las derivadas, por ejemplo las *elasticidades*, definidas como $\partial \ln f / \partial \ln x_i = (\partial f / \partial x_i)(x_i / f)$. De esta forma, la aproximación de estas funciones y sus derivadas puede ayudar en la confirmación o

refutación de hipótesis. Este tipo de análisis han sido llevados a cabo, por ejemplo, por Elbadawi, Gallant y Souza [Elbadawi83] empleando series de Fourier en el campo del análisis de demanda, o por Vinod y Ullah [Vinod85].

La aproximación de las derivadas también permite llevar a cabo análisis de sensibilidad en los que se analizan los efectos en la salida de un sistema ante pequeños cambios en las variables de entrada en diferentes regiones del espacio. Gilstrap y Dominny [Gilstrap89] han propuesto este tipo de análisis como la base que permite explicar el comportamiento de cierto tipo de redes neuronales.

Otro ejemplo de aplicación que requiere la información de las derivadas es la dinámica de sistemas caóticos. A partir de la aproximación de las derivadas de la función de transferencia de un determinado sistema caótico es posible, por ejemplo, obtener información acerca de sus exponentes de Lyapunov [Katok95].

Finalmente, existen una serie de aplicaciones en las que de forma natural se puede disponer de un conjunto de muestras de la función y de las derivadas a partir del cual se pretende reconstruir la función, como por ejemplo la telemetría o la simulación de control de tráfico aéreo [Fogel55]. En estos casos, la principal motivación que puede conducir a muestrear las derivadas es la esperanza de poder muestrear con menor frecuencia que la necesaria cuando sólo se muestrea la función y seguir siendo capaces de recuperar la señal. En el caso de las dos aplicaciones mencionadas, por ejemplo, se pretende reconstruir las trayectorias completas a partir de las medidas de la posición y de la velocidad (derivada), que en principio se van a poder muestrear a la mitad de velocidad que si se muestreara sólo la posición. Asimismo, es razonable pensar que la reconstrucción utilizando la información de las derivadas pueda proporcionar algunas ventajas adicionales como, por ejemplo, una mayor inmunidad del modelo frente al ruido en las muestras.

En todos estos casos es preciso desarrollar modelos que sean capaces de interpolar, aproximar o modelar de forma adecuada las derivadas de una función, bien a partir únicamente de las muestras de la función, o bien disponiendo también de muestras de las derivadas. Todos estos problemas muestran la importancia de encontrar soluciones para la interpolación y modelado de una determinada función cuando se imponen restricciones en las derivadas de la misma, bien de forma explícita, en forma de muestras de las propias derivadas, o bien de forma implícita considerando restricciones más generales, como la suavidad, que permitan una adecuada generalización de las derivadas de la función que se reconstruye o modela.

1.2 Planteamiento y objetivos de la Tesis

El objetivo de esta Tesis es el estudio de técnicas de interpolación y de modelado no lineales que permitan la incorporación de restricciones (información *a priori*) en la(s) derivada(s). En particular, en esta Tesis se han considerado cuatro enfoques del problema de aproximación de una función y sus derivadas que han dado lugar a varios tipos de modelos novedosos. Las líneas de investigación seguidas y sus objetivos han

sido:

1. La búsqueda de modelos que interpolen de forma exacta un conjunto de muestras de la función y sus derivadas. Dentro de esta línea, el problema está completamente resuelto en espacios de entrada unidimensionales para varios subespacios de señales pertenecientes al espacio de señales L_2 : el subespacio de señales limitadas en banda, y el conjunto de subespacios de Hilbert generados a partir de funciones de escala que formen una base de Riezs y que cumplan las condiciones de multiresolución. Estas soluciones se estudian con más detalle en el Capítulo 2, en el que se realiza una revisión de las aportaciones que existen en este contexto. En el caso de espacios de entrada bidimensionales, el problema está resuelto en el caso de señales limitadas en banda y en el subespacio de funciones splines. En esta línea se han explorado soluciones alternativas eficientes para señales no limitadas en banda en espacios de entrada unidimensionales y, especialmente, la extensión de estas técnicas de interpolación a espacios de entrada bidimensionales.
2. La búsqueda de modelos que realicen una aproximación regularizada de una función a partir de un conjunto de muestras de la misma y de sus derivadas. En esta segunda línea de investigación se elimina la restricción de interpolación para imponer ciertas restricciones de suavidad sobre las derivadas de la solución. En ciertos casos, como en presencia de ruido, esta alternativa puede resultar más conveniente que la interpolación. En esta línea, no existen soluciones conocidas. El objetivo, por tanto, es el desarrollo de técnicas de modelado novedosas tanto en espacios de entrada unidimensionales, como en espacios de entrada multidimensionales.
3. Tal como han sido planteadas en esta Tesis, las dos vías anteriores (interpolación y aproximación regularizada) requieren un conocimiento explícito de las derivadas en ciertos puntos y dan lugar a modelos con un elevado número de parámetros. En esta tercera línea de investigación, el objetivo ha sido la obtención de modelos con un reducido número de parámetros (parsimoniosos) y que permitan una aproximación simultánea de una función y sus derivadas. Se han considerado diferentes posibilidades de introducción de la información adicional proporcionada por las derivadas en la construcción del modelo.
4. La utilización de redes neuronales para la aproximación de una función y sus derivadas. Es conocida la capacidad teórica de las redes neuronales para aproximar una función y sus derivadas [Hornik90]. Sin embargo, en la práctica no existen alternativas constructivas en este campo. En general, el ajuste de las derivadas se confía a la capacidad de generalización de este tipo de redes a partir de una correcta aproximación de la función, pero casi nunca se suele utilizar la información disponible de las derivadas. En esta línea se ha investigado sobre varias alternativas para introducir esta información en diferentes tipos de redes neuronales, actuando sobre aspectos diferentes, tanto sobre los algoritmos de entrenamiento,

como sobre la arquitectura de la red. Por otro lado, se ha abordado el problema tanto desde una perspectiva de aproximación global, como de una aproximación localizada, en la que la reconstrucción de la función sólo es necesaria en una región localizada del espacio de entrada, como es el caso de la aplicación principal considerada en la Tesis: el modelado de transistores de microondas.

1.3 Organización y contribuciones de la Tesis

Esta Tesis está organizada de la siguiente forma:

En el Capítulo 2 se realiza una revisión de las técnicas y modelos existentes en la literatura para resolver el problema planteado en esta Tesis: la reconstrucción, modelado y aproximación de una función con restricciones en las derivadas. En este sentido se describen diferentes extensiones del Teorema de Muestreo de Shannon que permiten la inclusión de la información de las derivadas en la reconstrucción de señales limitadas en banda, así como algunas soluciones iterativas en problemas de muestreo irregular. Asimismo, se describe el método de reconstrucción de una función y sus derivadas en subespacios multiresolución generados por funciones de escala. Como caso particular de este tipo de subespacios, se describe el subespacio de funciones splines, que fueron una de las primeras alternativas de modelado en las que se utilizaron restricciones sobre las derivadas de una función y que, dada su importancia en esta Tesis, se revisa de forma independiente. A continuación se describen las técnicas de regularización, que permiten introducir restricciones de suavidad en la solución. Para finalizar, se describen los resultados existentes en el campo de las redes neuronales en relación con el problema abordado en esta Tesis. Se describe la conocida capacidad teórica para la aproximación de una función y sus derivadas, que posteriormente se aprovechará en esta Tesis.

En el Capítulo 3 se presenta un nuevo método de interpolación de una función y sus derivadas sobre el subespacio de funciones splines en espacios unidimensionales que hemos denominado Modelo Local de Interpolación de una función y sus derivadas (MLI). Este método de interpolación es local, lo que permite, por un lado, una reducida carga computacional y, por otro lado, su aplicación a problemas tanto de muestreo regular como de muestreo irregular. Además, admite una representación mediante bancos de filtros que facilita su utilización dentro del ámbito de procesado de señal. Sin embargo, como todos los métodos de interpolación, presenta una elevada sensibilidad al ruido. Por ello, a partir de este método, se plantea su extensión regularizada, obteniendo el que hemos denominado Modelo Local Regularizado para la reconstrucción de una función y sus derivadas (MLR). Este método ya no interpola el conjunto de muestras, sino que introduce restricciones de suavidad en la solución para reducir en lo posible el efecto del ruido en las muestras. A continuación, se ha planteado la extensión de ambos modelos a espacios de entrada bidimensionales, obteniendo soluciones novedosas de interpolación y modelado.

En el Capítulo 4 se presenta el Modelo Lineal a Tramos Suavizado (MLTS). Este

modelo es una extensión del Modelo Canónico Lineal a Tramos [Chua77, Kang78], en el que se incorpora una función para suavizar las transiciones entre diferentes regiones lineales. Se ha realizado un análisis de suavidad del modelo a partir del cual se muestra que dicha suavidad se controla con un único parámetro. Este parámetro puede entenderse como un parámetro de regularización. Este modelo admite una interpretación como una red neuronal, y se ha demostrado que resulta especialmente adecuado para problemas de modelado en los que se buscan modelos parsimoniosos, con un número reducido de parámetros, que sean capaces de generalizar de forma adecuada las derivadas de la función que modelan.

En el Capítulo 5 se presentan varias alternativas para la introducción de la información de las derivadas en el modelado, mediante redes neuronales, de una función. Por un lado, se ha abordado el problema a través de la modificación de la función de coste de las redes. Se ha estudiado la posibilidad de introducir la información de las derivadas en el entrenamiento de redes neuronales modificando la función de coste a minimizar para incluir esta información. Esta estrategia se ha aplicado a los dos tipos de redes neuronales más conocidas y ampliamente utilizadas, el *Perceptrón Multicapa* (MLP) y la *Red de Funciones de Base Radial* (RBF), así como al modelo MLTS presentado en el Capítulo 4. Los resultados obtenidos muestran que, como era de esperar, la introducción de la información de las derivadas en el entrenamiento de la red mejora de forma sensible la aproximación de las mismas. Además, se ha comprobado que la incorporación de información sobre las derivadas en la función de coste puede mejorar también la aproximación de la función original, lo que sugiere el interés de esta técnica aun cuando no se esté explícitamente interesado en las reconstrucción de las derivadas.

Por otro lado, se ha abordado el problema trabajando sobre la arquitectura de la red. En este sentido se presenta un modelo modular para la reconstrucción de la función y sus derivadas en el que un módulo diferente se encarga de capturar la información de la función y otro la de las derivadas. A partir de estos módulos, el modelo global es capaz de realizar una reconstrucción localizada de la función de interés. Este modelo está especialmente enfocado al modelado de transistores de microondas, donde el problema principal se ajusta a estas características: se precisa una reconstrucción localizada, en torno al punto de polarización, en la que la aproximación de las derivadas es importante. Para el módulo de gran señal, en el que se captura únicamente la información de la función, se ha empleado el modelo MLTS propuesto en esta Tesis. Para el módulo de pequeña señal, que captura la información de las derivadas, se utiliza una Red de Funciones de Base Radial Generalizada (GRBF). Para este tipo de red, se ha desarrollado un nuevo algoritmo de entrenamiento, basado en el conocido algoritmo EM (*Expectation-Maximization*) [Dempster77], que permite la aceleración de la convergencia y una reducción del efecto de los mínimos locales en este tipo de redes.

En el Capítulo 6 se presentan algunas de las aplicaciones en las que se han empleado los modelos desarrollados en esta Tesis, que especialmente son aplicaciones de modelado de dispositivos de microondas, en particular transistores MESFET y HEMT. Se han desarrollado modelos de gran señal, de pequeña señal, y un modelo global para este tipo de transistores. Estos modelos ofrecen unas prestaciones superiores a las obtenidas

con los métodos convencionales de modelado para estos dispositivos.

Finalmente, en el Capítulo 7 se resumen las principales conclusiones de esta Tesis y se presentan algunas de las líneas futuras de investigación que se pueden plantear a partir de la misma.

Capítulo 2

Interpolación y modelado con la incorporación de restricciones en las derivadas: Revisión de técnicas existentes

Con objeto de centrar el ámbito de este trabajo, en este capítulo se realiza una revisión de las técnicas de interpolación (reconstrucción) y de modelado o de aproximación funcional que permiten la incorporación de restricciones en las derivadas.

2.1 Interpolación en el subespacio de señales limitadas en banda

En este apartado se revisan las técnicas de interpolación de un conjunto de muestras de una función y sus derivadas en el subespacio de funciones limitadas en banda, pertenecientes al espacio de Hilbert L_2 , compuesto por todas aquellas funciones que son de cuadrado integrable en el sentido de Lebesgue (señales de energía finita). La correspondiente norma L_2 viene dada por

$$\|f(x)\| = \left(\int_{-\infty}^{\infty} |f(x)|^2 dx \right)^{\frac{1}{2}} = \sqrt{\langle f(x), f(x) \rangle}, \quad (2.1)$$

inducida por el producto escalar, definido como

$$\langle f(x), g(x) \rangle = \int_{-\infty}^{\infty} f^*(x)g(x)dx. \quad (2.2)$$

El subespacio de señales limitadas en banda resulta de una gran importancia en los campos de procesado de señal y de comunicaciones. Shannon demostró en 1949 el denominado *Teorema de Muestreo* [Shannon49] sobre este espacio de señales, estableciendo las bases del procesado de señal, al demostrar que toda señal limitada en

banda a una frecuencia W puede ser recuperada a partir de sus muestras adquiridas, al menos, a una frecuencia doble de la frecuencia W . Esta frecuencia mínima, $2W$, se denomina frecuencia de Nyquist. Se trata, sin duda alguna, de uno de los trabajos teóricos que ha tenido un mayor impacto en el campo del tratamiento de señal. Por otra parte, la inmensa mayoría de sistemas de comunicaciones trabajan con señales limitadas en banda. Existen infinidad de trabajos desarrollando las técnicas de muestreo y reconstrucción en este subespacio. Como referencia se puede consultar, por ejemplo, [Higgins96, Jerri77].

Centrándonos ya en el tema de esta Tesis, existen distintas técnicas para la reconstrucción de señales limitadas en banda a partir de las muestras de la misma y de sus derivadas. A continuación se revisan los principales resultados que existen en esta línea.

2.1.1 Generalización del Teorema del Muestreo de Shannon

La principal aportación teórica que permite la interpolación (reconstrucción) exacta de una función limitada en banda a partir de las muestras de la misma y de sus derivadas es la basada en la generalización para el muestreo de las derivadas del teorema del muestreo de Shannon. Para empezar recordemos este teorema cuyo planteamiento original es el siguiente [Shannon49]

Teorema 2.1 (Shannon) *Si una función $f(x)$ no contiene frecuencias superiores a W Hz, entonces queda completamente determinada a partir de sus muestras en la serie de puntos espaciados por $T = (1/2W)$, $y[n] = f(nT)$.*

Demostración 2.1 *La demostración de este teorema se puede encontrar en [Shannon49].*

La expresión que nos da la reconstrucción de la función a partir de tales muestras, $y[n]$, es

$$f(x) = \sum_{n=-\infty}^{\infty} y[n] \operatorname{sinc} \left(\frac{1}{T}(x - nT) \right). \quad (2.3)$$

Shannon obtuvo un enorme crédito por la formalización de este resultado y por ser consciente de su potencial dentro de la teoría de la comunicación y del procesamiento de señal. No obstante, el propio Shannon era conocedor de formulaciones previas de este teorema en la literatura matemática, en particular del trabajo de Whittaker [Whittaker29]. Además, en la literatura rusa, el teorema fue demostrado con anterioridad a los trabajos de Shannon por Kotelnikov [Kotelnikov33]. Por esta razón este teorema es a veces conocido como *Teorema de Shannon, Whittaker y Kotelnikov*.

Cuando Shannon introdujo este teorema, remarcó que el valor de la función $f(x)$ puede ser reconstruido a partir de muestras de la función y su derivada tomadas a mitad de velocidad, y extendió el resultado a derivadas de orden superior. Con posterioridad, Fogel [Fogel55] también consideró esta cuestión de forma independiente y dedujo el siguiente teorema

Teorema 2.2 (Fogel) *Si una función $f(x)$ no contiene frecuencias superiores a W Hz, entonces queda completamente determinada a partir de las medidas de D derivadas en la serie de puntos espaciados por $T = (D/2W)$.*

Demostración 2.2 *La demostración de este teorema (utilizando polinomios de interpolación de Lagrange e integración de contornos), se puede encontrar en [Fogel55].*

Como generalización de estos resultados, y como respuesta a la cuestión suscitada por Shannon referente a la reconstrucción a partir de muestras la función y de sus primeras D derivadas, medidas en la serie de puntos equiespaciados $T = (D+1)/(2W)$, Linden [Linden59] y Linden y Abramson [Linden60], obtuvieron la siguiente fórmula de reconstrucción

$$f(x) = \sum_{n=-\infty}^{\infty} \left[\phi(nT) + (x - nT)\phi^{(1)}(nT) + \cdots + \frac{(x - nT)^D}{D!} \phi^{(D)}(nT) \right] \operatorname{sinc} \left(\frac{1}{T}(x - nT) \right)^{D+1} \quad (2.4)$$

donde las funciones $\phi^{(j)}(nT)$ son combinaciones lineales de las muestras de las derivadas de la función, es decir, de $f^{(j)}(nT)$

$$\phi^{(j)}(nT) = \sum_{i=0}^j \binom{j}{i} \left(\frac{\pi}{T} \right)^{j-i} \Gamma_{R+1}^{(j-i)} f^{(i)}(nT), \quad (2.5)$$

y

$$\Gamma_{\alpha}^{(\beta)} = \frac{d^{\beta}}{dx^{\beta}} \left[\left(\frac{x}{\sin(x)} \right)^{\alpha} \right] \Big|_{x=0}. \quad (2.6)$$

Los valores $\Gamma_{\alpha}^{(\beta)}$ se pueden expresar como números generalizados de Bernouilli, algunos de estos valores son

$$\Gamma_{\alpha}^{(0)} = 1; \quad \Gamma_{\alpha}^{(2)} = \frac{\alpha}{3}; \quad \Gamma_{\alpha}^{(4)} = \frac{\alpha(5\alpha + 2)}{15}$$

$$\Gamma_{\alpha}^{(6)} = \frac{\alpha(35\alpha^2 + 42\alpha + 16)}{63}; \quad \Gamma_{\alpha}^{(\beta)} = 0, \quad \text{para } \beta \text{ impar}$$

Se puede ver que, mediante estas expresiones, es posible reconstruir una función unidimensional limitada en banda a partir de muestras de la función y de sus D derivadas si se muestrea con una velocidad $(D+1)$ veces menor que la que marca el límite del Teorema de Muestreo para el muestreo únicamente de la función. Una interesante discusión acerca de la naturaleza y significado de estas expresiones es planteada en la completa revisión realizada por Jerri [Jerri77] sobre el teorema de muestreo de Shannon y sus múltiples extensiones.

Este método de reconstrucción presenta las mismas ventajas y desventajas que el teorema de muestreo de Shannon. Aunque en principio es posible una reconstrucción perfecta de la señal, esto sólo va a ser cierto cuando se dispone de los infinitos términos de la serie. Si sólo se dispone de un conjunto finito de muestras se va a producir un error, debido al truncamiento, que además será mayor en los extremos de la secuencia. La naturaleza de la función base, $\text{sinc}(x)$, cuya caída no es demasiado rápida, hace que el error debido al truncamiento de la serie sea un problema a tener en cuenta. Además, se trata de un método de reconstrucción global, ya que para calcular el valor de la función en un punto se necesitan los valores de las muestras en todos los instantes, lo que hace que el método sea costoso computacionalmente. Además, como técnica de modelado, tampoco se trata de un método parsimonioso, ya que es preciso almacenar todas las muestras para llevar a cabo la reconstrucción de la función.

2.1.2 Particularización para el caso de interpolación de secuencias discretas periódicas

Un interesante caso particular de la aplicación del teorema de muestreo de Shannon es el caso de secuencias discretas periódicas. En este caso, la convolución del kernel $\text{sinc}(x)$ con la secuencia infinita de una función periódica se puede reescribir como una suma finita. La idea es que, a partir de las muestras que forman un periodo, la contribución del resto de muestras puede ser calculada gracias a la condición de periodicidad. A partir de esta idea, expuesta originalmente por Schanze [Schanze95], dada una secuencia de muestras, tomadas a una velocidad superior a la de Nyquist, de una función limitada en banda, $y[n] = f(nT)$, periódica de periodo N , es posible reconstruir la función $f(x)$ mediante una suma finita.

Partiendo de la expresión (2.3), particularizada para $T = 1$, para simplificar las expresiones resultantes, y utilizando la relación

$$\sin(\pi(x - n)) = (-1)^n \sin(\pi x), \quad (2.7)$$

se puede escribir la expresión de reconstrucción como

$$f(x) = \frac{\sin(\pi x)}{\pi} \sum_{n=-\infty}^{\infty} y[n] \frac{(-1)^n}{x - n}. \quad (2.8)$$

Reordenando los términos se puede obtener

$$\begin{aligned} f(x) &= \frac{\sin(\pi x)}{\pi} \sum_{j=-\infty}^{\infty} \sum_{n=-L+jN}^{M-1+jN} y[n] \frac{(-1)^n}{x - n} \\ &= \frac{\sin(\pi x)}{\pi} \sum_{j=-\infty}^{\infty} \sum_{n=-L}^{M-1} y[n + jN] \frac{(-1)^{n+jN}}{x - n - jN}, \end{aligned} \quad (2.9)$$

donde L y M son enteros arbitrarios que satisfagan la relación $N = L + M$. Los sumatorios interiores se pueden separar para el caso de que N sea par o impar, y teniendo en cuenta la periodicidad de la secuencia de muestras $y[n]$, se puede escribir

$$\begin{aligned} f(x) &= \frac{\sin(\pi x)}{\pi} \sum_{j=-\infty}^{\infty} \sum_{n=-L}^{M-1} y[n] \frac{(-1)^n}{x - n - jN}, \quad \text{para } N \text{ par,} \\ f(x) &= \frac{\sin(\pi x)}{\pi} \sum_{j=-\infty}^{\infty} \sum_{n=-L}^{M-1} y[n] \frac{(-1)^n (-1)^j}{x - n - jN}, \quad \text{para } N \text{ impar.} \end{aligned} \quad (2.10)$$

Intercambiando el orden de los sumatorios

$$\begin{aligned} f(x) &= \frac{\sin(\pi x)}{\pi} \sum_{n=-L}^{M-1} y[n] (-1)^n \sum_{j=-\infty}^{\infty} \frac{1}{x - n - jN}, \quad \text{para } N \text{ par,} \\ f(x) &= \frac{\sin(\pi x)}{\pi} \sum_{n=-L}^{M-1} y[n] (-1)^n \sum_{j=-\infty}^{\infty} \frac{(-1)^j}{x - n - jN}, \quad \text{para } N \text{ impar.} \end{aligned} \quad (2.11)$$

Los sumatorios interiores en (2.11) corresponden a las descomposiciones en fracciones de las funciones cotangente y cosecante, como se puede ver en el Apéndice B. Teniendo en cuenta estas relaciones, es fácil comprobar que las dos ecuaciones, para N par y para N impar, se pueden agrupar en la siguiente expresión de reconstrucción [Schanze95]

$$f(x) = \frac{\sin(\pi x)}{2N} \sum_{n=0}^{N-1} (-1)^n y[n] \left[(-1)^{N+1} \tan\left(\pi \frac{x-n}{2N}\right) + \cot\left(\pi \frac{x-n}{2N}\right) \right]. \quad (2.12)$$

Esta expresión, aunque plantea la reconstrucción exacta de este tipo de funciones, presenta ciertos problemas de estabilidad numérica cuando se evalúa en instantes próximos a n . Recientemente, dos métodos equivalentes, y que vienen a solventar este problema de estabilidad numérica, han sido presentados en [Candocia98] y en [Dooley2000]. El primero resuelve el problema en el dominio de la frecuencia mediante la transformada discreta de Fourier, y el segundo mediante la utilización de filtros de retardo fraccional.

2.1.2.1 Interpolación de secuencias periódicas con muestras de las derivadas

Cuando se dispone de muestras de la función y las derivadas de secuencias periódicas, siguiendo la idea básica expresada en [Schanze95], es posible obtener fórmulas de reconstrucción mediante sumas finitas equivalentes a (2.12). Por ejemplo, a continuación se presentan las expresiones que hemos obtenido para los casos concretos de 1 y de 2 derivadas. En el primer caso, se dispone de muestras de la función, $y[n]$, y de la derivada primera, $y^1[n]$. La expresión (2.4) particularizada para $D = 1$ resulta

$$f(x) = \sum_{n=-\infty}^{\infty} (y[n] + (x - nT)y^1[n]) \left(\frac{\sin(\pi(x - nT)/T)}{\pi(x - nT)/T} \right)^2. \quad (2.13)$$

En este caso se pueden recuperar señales cuyas componentes frecuenciales sean menores o iguales que $1/T$. Tomando de nuevo $T = 1$ para simplificar las expresiones, y teniendo en cuenta la relación (2.7), se puede reconstruir la función mediante

$$f(x) = \left(\frac{\sin(\pi x)}{\pi} \right)^2 \sum_{n=-\infty}^{\infty} (y[n] + (x - n)y^1[n]) \frac{(-1)^{2n}}{(x - n)^2}, \quad (2.14)$$

que se puede desarrollar como

$$f(x) = \left(\frac{\sin(\pi x)}{\pi} \right)^2 \left\{ \sum_{j=-\infty}^{\infty} \sum_{n=-L+jN}^{M-1+jN} y[n + jN] \frac{(-1)^{2n+2jN}}{(x - n - jN)^2} + \sum_{j=-\infty}^{\infty} \sum_{n=-L+jN}^{M-1+jN} (x - n - jN)y^1[n + jN] \frac{(-1)^{2n+2jN}}{(x - n - jN)^2} \right\}. \quad (2.15)$$

Aprovechando la periodicidad de las secuencias de muestras y reordenando las expresiones se llega a

$$f(x) = \left(\frac{\sin(\pi x)}{\pi} \right)^2 \left\{ \sum_{n=-L}^{M-1} y[n] \sum_{j=-\infty}^{\infty} \frac{1}{(x - n - jN)^2} + \sum_{n=-L}^{M-1} y^1[n] \sum_{j=-\infty}^{\infty} \frac{1}{(x - n - jN)} \right\}. \quad (2.16)$$

Nuevamente los sumatorios interiores corresponden a la descomposición en fracciones de funciones trigonométricas (Apéndice B); finalmente, se obtiene la fórmula de reconstrucción

$$f(x) = \left(\frac{\sin(\pi x)}{\pi} \right)^2 \sum_{n=0}^{N-1} \left\{ y[n] \left(\frac{\pi}{N} \right)^2 \csc^2 \left(\pi \frac{x - n}{N} \right) + y^1[n] \left(\frac{\pi}{N} \right) \cot \left(\pi \frac{x - n}{N} \right) \right\}. \quad (2.17)$$

Para el caso en el que se dispone de muestras de la primera y segunda derivadas ($D = 2$), la expresión (2.4) toma la forma

$$f(x) = \sum_{n=-\infty}^{\infty} \left\{ y[n] + (x - nT)y^1[n] + \frac{(x - nT)^2}{2} \left(y^2[n] + \left(\frac{\pi}{T}\right)^2 y[n] \right) \right\} \left(\frac{\sin(\pi(x - nT)/T)}{\pi(x - nT)/T} \right)^3. \quad (2.18)$$

Para secuencias periódicas de período N , cuando se dispone de muestras de la función, $y[n]$, y de las dos primeras derivadas, $y^1[n]$ e $y^2[n]$, de nuevo es posible realizar la reconstrucción utilizando una suma de N términos. Siguiendo el mismo procedimiento que en los casos anteriores, y teniendo en cuenta los desarrollos en serie del Apéndice B, se obtienen las siguientes expresiones de reconstrucción

$$f(x) = \left(\frac{\sin(\pi x)}{\pi} \right)^3 \sum_{n=0}^{N-1} (-1)^n \left\{ y[n] \left(\frac{\pi}{N} \right)^3 \left[\cot^3 \left(\pi \frac{x-n}{N} \right) + \cot \left(\pi \frac{x-n}{N} \right) \right] + y^1[n] \left(\frac{\pi}{N} \right)^2 \csc^2 \left(\pi \frac{x-n}{N} \right) + \frac{y^2[n] + \pi^2 y^2[n]}{2} \left(\frac{\pi}{N} \right) \cot \left(\pi \frac{x-n}{N} \right) \right\}, \quad (2.19)$$

cuando N es par y

$$f(x) = \left(\frac{\sin(\pi x)}{\pi} \right)^3 \sum_{n=0}^{N-1} (-1)^n \left\{ y[n] \left(\frac{\pi}{N} \right)^3 \left[\csc^3 \left(\pi \frac{x-n}{N} \right) - \frac{\csc \left(\pi \frac{x-n}{N} \right)}{2} \right] + y^1[n] \left(\frac{\pi}{N} \right)^2 \left[\frac{\csc^2 \left(\pi \frac{x-n}{2N} \right)}{2} - \csc^2 \left(\pi \frac{x-n}{N} \right) \right] + \frac{y^2[n] + \pi^2 y^2[n]}{2} \left(\frac{\pi}{N} \right) \csc \left(\pi \frac{x-n}{N} \right) \right\}, \quad (2.20)$$

para N impar.

2.1.3 Extensión a muestreo irregular

En las secciones anteriores se ha considerado la reconstrucción de una función limitada en banda, a partir de las medidas de la misma y de sus derivadas, para el caso de muestreo regular; es decir, cuando las diferentes muestras están equiespaciadas. Un problema alternativo aparece cuando el muestreo es irregular, donde la distancia entre muestras es, en general, arbitraria. Una generalización del método presentado en la Sección 2.1.1 para este problema de muestreo irregular fue propuesta por Rawn [Rawn89].

De forma específica, [Rawn89] demuestra que es posible la reconstrucción de funciones limitadas en la banda $[-\pi, \pi]$, a partir de medidas de la función y de D derivadas, para el caso en que la secuencia de muestreo irregular t_n está restringida al conjunto de secuencias de la forma $S = (\{x[n]\} : |x[n] - nD| \leq d < 1/(4(D + 1)), \forall n \in Z)$. En este caso, la función se puede reconstruir mediante la siguiente expresión

$$f(x) = \sum_{n=-\infty}^{\infty} \left[\phi(x[n]) + \left(1 - \frac{x}{x[n]}\right) \phi^{(1)}(x[n]) + \dots + \left(1 - \frac{x}{x[n]}\right)^D \phi^{(D)}(x[n]) \right] \Psi_n(x), \quad (2.21)$$

donde

$$\phi^{(k)}(x[n]) = \frac{(-1)^k (x[n])^k}{k!} \frac{d^k}{dx^k} \left[\frac{f(x)}{\Psi_n(x)} \right] \Big|_{x=x[n]}, \quad (2.22)$$

siendo

$$\Psi_n(x) = \left[\frac{G(x)}{(x - x[n])G'(x[n])} \right]^{D+1}, \quad (2.23)$$

$$G(x) = xF(x), \quad (2.24)$$

y

$$F(x) = \prod_{n=1}^{\infty} \left(1 - \frac{x}{x[n]}\right) \left(1 - \frac{x}{x[-n]}\right). \quad (2.25)$$

Las funciones $\Psi_n(x[n])$ son funciones interpoladoras del tipo de Lagrange. Este método supone una generalización del caso particular de muestreo uniforme, ya que se puede demostrar que cuando $\{x[n]\} = \{n(D + 1), \forall n \in Z\}$, la expresión (2.21) se reduce a la expresión (2.4).

Asimismo, Rawn demuestra la convergencia uniforme de la serie definida por la ecuación (2.21) tanto en $(-\infty, \infty)$ como en el sentido de la norma L^2 y, además, la estabilidad en sentido de la norma L^2 con respecto a las secuencias de muestreo de la forma S para las señales limitadas en la banda $[-\pi, \pi]$.

La demostración de la convergencia de la serie (2.21) es válida, igualmente, para demostrar la convergencia del método propuesto por Linden y Abramson para muestreo regular. La demostración dada por estos autores era intuitivamente correcta desde el punto de vista de teoría de sistemas, pero no resulta adecuada desde un punto de vista rigurosamente matemático, ya que el espacio de funciones para las que la expresión (2.4) es válida no estaba claramente definido, y por otro lado, en [Linden60] no se abordaba la cuestión de la naturaleza de la convergencia. De hecho, de las aportaciones de Linden y Abramson, sólo se puede concluir una convergencia uniforme de la expresión (2.4) para el espacio de funciones C^∞ (derivables infinitamente) y con soporte compacto. Rawn lleva este estudio un poco más allá abordándolo desde un punto de

vista matemático, estudiando la convergencia en un espacio cualquiera de Hilbert, demostrando que el método es válido para todas las señales limitadas en banda de energía finita, probando una convergencia uniforme tanto en $(-\infty, \infty)$ como en el sentido de la norma L^2 .

Nótese que las series de muestreo para las que este método es válido corresponden al caso particular de perturbaciones sobre secuencias de muestreo regular, y no a un caso general de muestreo irregular. Esta extensión, aunque interesante desde un punto de vista teórico, no resulta muy adecuada para su utilización práctica, ya que implica el cálculo de las derivadas de la función respecto de las funciones interpoladoras, y, en principio, no se tiene porqué conocer la función. Además los cálculos de las distintas funciones intermedias involucran series infinitas, lo que conlleva errores al truncar las series de muestras.

2.1.4 Método basado en proyecciones

Una de las alternativas más comúnmente utilizadas a la hora de resolver el problema de la reconstrucción de una función limitada en banda, a partir de un conjunto de muestras no uniformes, es la utilización de métodos iterativos. Existe una gran cantidad de variantes de este tipo de métodos, pero en esta sección se describirá únicamente un método que permite la incorporación de la información de las derivadas en la reconstrucción de la función. Antes de pasar a describir este método, haremos una introducción de este tipo de algoritmos iterativos, basados en proyecciones, para el caso de muestreo irregular de la función. Para ello se presenta el más básico de los mismos, el denominado *método de Sauer-Allebach* [Sauer87]. El planteamiento general del problema es el siguiente: dada una secuencia de muestras de una función, $y[n] = f(x[n])$, obtenida en unos instantes arbitrarios $x[n]$, reconstruir la función original $f(x)$ perteneciente al conjunto de funciones limitadas en banda. Para empezar se introducen las siguientes definiciones

Definición 2.1 *Se define el espacio de funciones limitadas en banda, B_W , como el subespacio, perteneciente al espacio de Hilbert L_2 , compuesto por aquellas señales cuyas componentes frecuenciales son menores que la frecuencia W Hz.*

Definición 2.2 *Se define la función escalón de Voronoid, $V_n(x)$, a partir de una secuencia de instantes de muestreo $S = \{x[n]\}_{n \in \mathbb{Z}}$ tal que $x[n-1] < x[n]$ y con $\lim_{n \rightarrow \pm\infty} x[n] = \pm\infty$, como aquella función que toma el valor unidad en el intervalo $[z[n-1], z[n])$, donde la secuencia $z[n] = (x[n] + x[n+1])/2$, es la secuencia de los instantes intermedios de muestreo. Se puede comprobar que el conjunto de funciones escalón de Voronoid forman una partición de la unidad, es decir, que $\sum_{n=-\infty}^{\infty} V_n(x) = 1, \forall x$.*

Partiendo de estas definiciones, el método de Sauer-Allebach se resume en el siguiente teorema

Teorema 2.3 Dado $\delta = \sup_{n \in \mathbb{Z}} (x[n+1] - x[n]) < 1/2W$, entonces cualquier función $f(x) \in B_W$ queda determinada de forma única por sus muestras $y[n] = f(x[n])$ y puede ser reconstruida iterativamente de la forma siguiente

$$\begin{aligned} f_0(x) &= P_W \left(\sum_{n \in \mathbb{Z}} f(x[n]) V_n(x) \right), \\ f_m(x) &= f_{m-1}(x) + P_W \left(\sum_{n \in \mathbb{Z}} (f(x[n]) - f_{m-1}(x[n])) V_n(x) \right), \\ f(x) &= \lim_{m \rightarrow \infty} f_m(x). \end{aligned} \quad (2.26)$$

El operador P_W realiza la proyección ortogonal de la señal sobre el espacio de señales limitadas en banda B_W . Bajo las condiciones establecidas en el teorema, el método converge de forma geométrica en el sentido de la norma L^2 , y el error en cada iteración está acotado de la forma siguiente

$$\|f(x) - f_m(x)\| \leq (2\delta W)^{m+1} \|f(x)\|. \quad (2.27)$$

Demostración 2.3 La demostración de este teorema se puede encontrar en [Sauer87].

La idea básica de este método es la siguiente siguiente: la primera etapa de cada iteración consiste en la construcción de una función error mediante la interpolación del error que se tiene en la iteración anterior en los instantes de muestreo conocidos. En este caso la interpolación se realiza empleando la función escalón de Voronoid, por lo que en ocasiones a este método se le denomina también *Método de Voronoid*. Esta función interpolada se proyecta paso bajo mediante el operador P_W , y el resultado se añade a la reconstrucción obtenida en la iteración anterior.

Existen otras alternativas para llevar a cabo la interpolación de la función error, como por ejemplo la interpolación lineal. Asimismo, esta función error puede ser multiplicada por un parámetro de relajación para garantizar la convergencia del algoritmo. En general, existe una gran cantidad de variantes de este tipo de métodos, con diferentes restricciones sobre la secuencia de instantes de muestreo, $x[n]$, en general menos restrictivas que la del Teorema 2.3. Para una revisión más exhaustiva de este tipo de métodos se puede consultar [Benedetto95] y, en particular, el Capítulo 8 [Feichtinger95].

Empleando ideas similares a las expuestas, Razafinjatovo [Razafinjatovo95] propuso una extensión para la reconstrucción de la función incluyendo las muestras de las derivadas para muestreo irregular, que se resume en el siguiente teorema

Teorema 2.4 (Razafinjatovo) Sea una función $f(x) \in B_W$, de la que se dispone de muestras de la función y sus primeras D derivadas, $f^1(x), \dots, f^D(x)$, en unos instantes dados por la secuencia de muestreo $S = (x[n])_{n \in \mathbb{Z}}$, que satisface que

$\sup_n(x[n+1] - x[n]) \leq \delta$. Entonces, si $\delta < C(D)/W$, la función $f(x)$ queda completamente determinada a partir de las muestras $\{f^j(x[n]), n \in Z, j = 0, \dots, D\}$, y se puede reconstruir empleando el siguiente algoritmo iterativo

$$f_0(x) = 0,$$

$$f_m(x) = f_{m-1}(x) + P_W \left(\sum_{n \in Z} \left(\sum_{j=0}^D (f^j(x[n]) - f_{m-1}^j(x[n])) \frac{(x - x[n])^j}{j!} \right) V_n(x) \right),$$

$$f(x) = \lim_{m \rightarrow \infty} f_m(x),$$
(2.28)

donde P_W es la proyección ortogonal sobre el espacio de señales limitadas en la banda B_W . La convergencia del método viene caracterizada por el error en la iteración m -ésima

$$\|f(x) - f_m(x)\| \leq \left(\frac{\delta W}{C(D)} \right)^{(m+1)(D+1)} \|f(x)\|,$$
(2.29)

donde el parámetro $C(D)$, que limita la máxima distancia entre instantes de muestreo, viene dado por

$$C(D) = 2 \left(D! \sqrt{2D+1} (2D+2) \right)^{\frac{1}{D+1}}.$$
(2.30)

Demostración 2.4 La demostración de este Teorema se encuentra en [Razafinjatovo95].

En este caso, la aportación principal consiste en que la señal de error en cada iteración se construye, de una forma simple, a partir de la información de las derivadas.

2.2 Interpolación en subespacios generados por funciones de escala

En la Sección 2.1 se ha hecho una revisión de las alternativas para la interpolación de una función y sus derivadas en el subespacio de señales limitadas en banda. En esta sección se hace una revisión de las técnicas de interpolación existentes cuando la función pertenece a un subespacio más amplio como es el subespacio generado por funciones de escala (subespacios multiresolución).

En primer lugar, se hará una introducción sobre el tipo de subespacios de multiresolución en los que se está interesado, para a continuación describir las técnicas de interpolación de una función y sus derivadas proyectando las secuencias de muestras sobre este tipo de espacios.

2.2.1 Subespacios multiresolución generados por funciones de escala

Para la caracterización de los subespacios que vamos a tratar en esta sección resultan necesarias las siguientes definiciones

Definición 2.3 Se define el conjunto de secuencias cuadráticamente sumables, l_2 , como el conjunto de secuencias $c[n]$ para las que $\sum_n |c[n]|^2 < \infty$.

Definición 2.4 Un conjunto φ de funciones, contenido en un espacio V , es completo, si la expansión lineal cerrada de φ es todo el espacio V .

Definición 2.5 Un conjunto de funciones $\{\varphi_n(x) = \varphi(x - n)\}_{n \in \mathbb{Z}}$, en un espacio de Hilbert H , forma una base de Riezs para H si $\{\varphi(x)\}$ es completo y si existen dos constantes estrictamente positivas A y B , $0 < A \leq B < \infty$ tal que

$$A\|c[n]\|^2 \leq \left\| \sum_n a[n]\varphi_n(x) \right\|^2 \leq B\|c[n]\|^2, \quad (2.31)$$

para cualquier $\{c[n]\} \in l_2$ (ver [Young80, Higgins96] para más detalles y propiedades).

Definición 2.6 Un conjunto de funciones $\{\varphi_n(x) = \varphi(x - n)\}_{n \in \mathbb{Z}}$, satisface la condición de partición de la unidad si

$$\sum_{n \in \mathbb{Z}} \varphi(x + n) = 1, \forall x \in \mathbb{R}. \quad (2.32)$$

2.2.1.1 Formulación multiresolución: función de escala

El tipo de subespacios a los que hacemos referencia en esta sección pertenecen al conjunto de espacios invariantes con el desplazamiento, que se generan a partir de versiones desplazadas de una determinada *función generadora* $\varphi(x)$. En particular, este tipo de subespacios tienen la forma

$$V(\varphi) = \left\{ f(x) = \sum_{n \in \mathbb{Z}} c[n]\varphi(x - n) : c[n] \in l_2 \right\}. \quad (2.33)$$

Esto significa que cualquier función $f(x) \in V(\varphi)$ está caracterizada por una secuencia de coeficientes $c[n]$; esta es la representación discreta de la señal que será utilizada en el procesamiento digital de la misma. Para que esta representación tenga sentido, la función $\varphi(x)$ ha de ser una función generadora admisible, para lo que ha de cumplirse

1. $c[n] \in l_2$,
2. $\{\varphi(x - n)\}$ forma una base de Riezs de $V(\varphi)$,

3. $\{\varphi(x - n)\}$ satisface la condición de partición unidad.

En particular, se puede demostrar que las funciones de escala que aparecen en la Transformada Wavelet [Burrus98] y en el análisis multiresolución [Mallat89], cumplen las condiciones anteriores y son, por tanto, funciones generadoras admisibles. Para una revisión detallada de estos aspectos se puede consultar [Mallat89, Daubechies92, Burrus98].

Para presentar la idea de multiresolución, comenzaremos definiendo la función de escala y el espacio que genera. Empleando la notación

$$\varphi_n(x) = \varphi(x - n), \quad (2.34)$$

el subespacio de L_2 compuesto por las funciones que genera esta función base se va a denominar V_0 . Esto significa que cualquier función perteneciente a V_0 se va a poder representar mediante una combinación lineal de las funciones $\varphi_n(x)$ de acuerdo a (2.33). El tamaño del subespacio generado se puede cambiar variando la escala temporal de las funciones de escala. De esta forma se genera una familia de funciones

$$\varphi_{i,n}(x) = 2^{i/2}\varphi(2^i x - n), \quad (2.35)$$

que generan el subespacio de funciones V_i . A partir de estas definiciones, los requerimientos básicos para llevar a cabo un *análisis multiresolución* o MRA (*Multiresolution Analysis*) fueron formulados por Mallat [Mallat89] exigiendo que los subespacios generados cumplan

$$\cdots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots \subset L^2, \quad (2.36)$$

o lo que es lo mismo

$$V_i \subset V_{i+1}, \quad \text{para todo } i \in Z, \quad (2.37)$$

con

$$V_{-\infty} = \{0\}, \quad V_{\infty} = L^2. \quad (2.38)$$

Teniendo en cuenta la definición de V_i , los espacios deben satisfacer una condición natural de escalado

$$f(x) \in V_i \Leftrightarrow f(2x) \in V_{i+1}, \quad (2.39)$$

lo que asegura que los elementos en un espacio son simplemente versiones escaladas de los elementos del siguiente espacio. El anidamiento de los espacios se puede conseguir exigiendo que $\varphi(x) \in V_0$, lo que significa que también pertenece a V_1 , el espacio generado por $\varphi(2x)$, con lo que $\varphi(x)$ puede expresarse como una combinación lineal de funciones $\varphi(2x)$ desplazadas, es decir

$$\varphi(x) = \sum_n h[n]\sqrt{2}\varphi(2x - n), \quad n \in Z, \quad (2.40)$$

donde los coeficientes $h[n]$ son una secuencia de números denominados coeficientes de la función de escala, o también filtro de escala. Se puede ver, por ejemplo, que para la

función de escala más sencilla, la llamada función de Haar [Haar10], que es un simple pulso, se cumple

$$\varphi(x) = \varphi(2x) + \varphi(2x - 1), \quad (2.41)$$

lo que significa que $h[0] = 1/\sqrt{2}$ y $h[1] = 1/\sqrt{2}$. Otro ejemplo sencillo sería la función de escala triangular, para la que se tiene que $h[0] = \frac{1}{2\sqrt{2}}$, $h[1] = 1/\sqrt{2}$, $h[2] = \frac{1}{2\sqrt{2}}$. En la Figura 2.1 se muestran estas relaciones para las dos funciones, y se puede apreciar con claridad la idea del concepto de multiresolución.

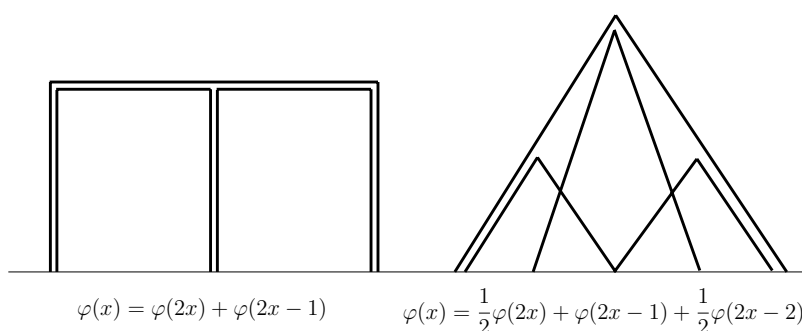


Figura 2.1: Funciones de escala de Haar y triangular

Con estas premisas, una función de escala, $\varphi(x)$, es válida como función generadora de un espacio multiresolución, en el sentido definido por Mallat [Mallat89], si y sólo si el conjunto $\varphi_n(x)$

1. Forma una base de Riezs.
2. Satisface la condición de partición de la unidad.
3. Satisface la condición de escalado (2.40).

Las dos primeras condiciones garantizan que $\varphi(x)$ es una función generadora admisible. Para garantizar la estabilidad y la no ambigüedad de la representación, se impone la condición de que el conjunto de funciones $\{\varphi_n(x) = \varphi(x - n)\}_{n \in \mathbb{Z}}$ debe formar una base de Riezs para $V(\varphi)$, que es lo mejor después de una base ortogonal [Daubechies92]. Como consecuencia directa de esta condición, debido a la desigualdad inferior en (2.31), se puede comprobar que $f(x) = \sum_n c[n]\varphi_n(x) = 0$ implica que $c[n] = 0, \forall n$. Por lo tanto, las funciones base son linealmente independientes, lo que significa que cada señal $f(x) \in V(\varphi)$ es especificada, de forma única, por su secuencia de coeficientes, $c[n] \in l_2$. La desigualdad superior implica que la norma L_2 es finita, de tal modo que $V(\varphi)$ es un subespacio válido de L_2 , asegurando también la estabilidad de la representación [Young80].

La condición de partición de la unidad asegura que es posible aproximar cualquier función con una precisión arbitraria seleccionando un subespacio V_i adecuado, lo que,

en términos de muestreo en este tipo de subespacios, supone muestrear con un intervalo suficientemente pequeño [Unser2000]. La tercera condición asegura la posibilidad de realizar un análisis multiresolución a partir de tales funciones. En general, esta es la condición más restrictiva.

2.2.2 Incorporación de la información de las derivadas

Una de las técnicas de reconstrucción de una función a partir de una serie de muestras de la misma y de sus derivadas es la que recientemente han propuesto Djokovic y Vaidyanathan [Djokovic97] dentro del marco teórico de los espacios multiresolución definidos por las funciones de escala que se han presentado en el apartado anterior. Es un hecho conocido que, bajo una serie de condiciones poco restrictivas en la función de escala $\varphi(x)$, los subespacios de multiresolución generados a partir de la misma y definidos en la Sección (2.2.1), son espacios de Hilbert con kernel de reproducción (RKHS) [Walter92, Janssen93]. Esto permite el desarrollo de una teoría de muestreo en dichos espacios. Djokovic y Vaidyanathan emplean dicha teoría para resolver problemas tales como muestreo no uniforme periódico, sobremuestreo, muestreo multibanda y muestreo de una función y sus derivadas en subespacios multiresolución. Todos estos problemas son tratados de una forma común empleando la teoría de bancos de filtros de reconstrucción perfecta, a través de la que se obtienen expresiones de las funciones de síntesis, $S_n(x)$, con soporte compacto, a partir de las cuales puede reconstruir la función. A continuación se presenta, de forma simplificada, el método propuesto en [Djokovic97] para el problema que nos interesa, que es el de reconstrucción de una función a partir de muestras de la misma y de sus derivadas. Supongamos que se dispone de N muestras de la función $f(x)$ en los instantes $x[n]$, $n = 0, 1, \dots, N$, que denotaremos como $y[n] = f(x[n])$. Se asume igualmente que la función de escala $\varphi(x)$ genera una base de Riezs para el subespacio $V_0 \subset L_2$. Para demostrar que V_0 es un RKHS, Walter [Walter92] únicamente impone la condición de que dicha función decaiga más rápidamente que $1/|x|$, es decir, que exista una constante $C > 0$ tal que

$$|\varphi(x)| < \frac{C}{(1 + |x|)^{1+\epsilon}}, \quad \text{para algún } \epsilon > 0. \quad (2.42)$$

Janssen [Janssen93] llega al mismo resultado imponiendo condiciones menos restrictivas: que $\varphi(x)$ esté acotada y que

$$\sum_n |\varphi(x - n)| < C, \quad (2.43)$$

converja de forma uniforme en $[0, 1]$. Esto asegura que la transformada de Fourier de la función de escala está bien definida y que es una función continua [Chui92]. Como $\varphi(x)$ es una base de Riezs para V_0 , cualquier función en este espacio va a poder definirse por una secuencia única de coeficientes $c[n]$ tal que

$$f(x) = \sum_n c[n]\varphi(x - n). \quad (2.44)$$

Asumiendo que la función de escala $\varphi(x)$ tiene soporte compacto, que su derivada es $\varphi^1(x)$, y que tanto la función de escala como su derivada satisfacen las condiciones de Janssen anteriormente mencionadas, entonces es posible diferenciar la ecuación (2.44) y llegar a la expresión

$$f^1(x) = \sum_n c[n] \varphi^1(x - n). \quad (2.45)$$

Con esto es posible demostrar [Djokovic97] que la función y la derivada se pueden escribir como

$$f(x[n]) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{jn\omega} C(e^{j\omega}) \Phi_0(e^{j\omega}) d\omega, \quad (2.46)$$

$$f^1(x[n]) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{jn\omega} C(e^{j\omega}) \Phi_1(e^{j\omega}) d\omega, \quad (2.47)$$

donde

$$\Phi_0(e^{j\omega}) = \sum_n \varphi(n) e^{-jn\omega}, \quad (2.48)$$

$$\Phi_1(e^{j\omega}) = \sum_n \varphi^1(n) e^{-jn\omega}, \quad (2.49)$$

son las transformadas de Fourier (o transformadas z) de la función de escala y su derivada respectivamente, y $C(e^{j\omega})$ es la transformada de la secuencia de coeficientes $c[n]$. El mismo proceso se puede extender a un mayor número de derivadas. Además, es conocido que $f(x)$ puede ser reconstruida a partir de las muestras $f(x[n])$, con lo que la información de la derivada es redundante. Por ello, es posible utilizar esta redundancia para recuperar la función a partir de las muestras de función y derivada en la secuencia de instantes $x[2n]$, es decir, muestreando a la mitad de velocidad que si sólo se muestreara la función, o en general a partir de $x[Dn]$ cuando se dispone de muestras de las primeras $D - 1$ derivadas. En este caso, la interpretación mediante bancos de filtros del problema cuando se dispone de muestras de las $D - 1$ primeras derivadas es la que aparece en la Figura 2.2 [Djokovic97].

Es posible establecer una representación equivalente mediante matrices polifase $E(z)$ y $R(z)$ de la forma que se muestra en la Figura 2.3. Este esquema corresponde a un banco de filtros máximamente diezmado de D canales, con $\Phi_n(e^{j\omega})$, $n = 0, \dots, D-1$, como filtros de análisis. La reconstrucción estable de este sistema es posible si la matriz polifase definida por las funciones $\Phi_n(e^{j\omega})$ es no-singular para todo $\omega \in [-\pi, \pi]$.

El procedimiento a seguir para la reconstrucción de la función y las derivadas es el siguiente:

- Construir la matriz de polifase $E(z)$ a partir de las funciones $\Phi_n(z)$. Esta matriz se obtiene de forma sencilla muestreando las funciones $\varphi^d(x)$ y tomando su transformada z .

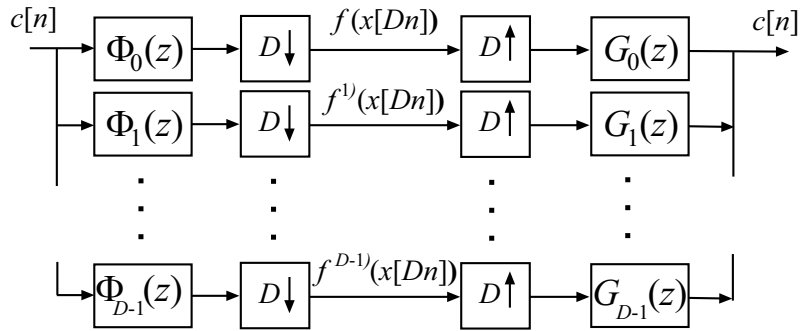


Figura 2.2: Interpretación mediante banco de filtros de reconstrucción perfecta del problema de reconstrucción de una función a partir de un conjunto de muestras de la misma y de sus derivadas

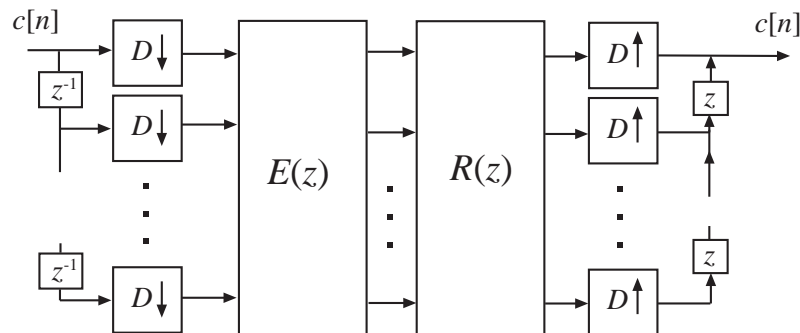


Figura 2.3: Interpretación polifase del problema de reconstrucción de una función a partir de las muestras de la misma y de sus derivadas

- Invertir la matriz $E(z)$ para obtener la matriz $R(z)$.
- Obtener las expresiones de las funciones de síntesis. Estas funciones se obtienen a partir de una combinación de la función de escala desplazada. La combinación viene definida por la matriz $R(z)$.

Un ejemplo de este método se presenta en la siguiente sección, en la que se aplica para el espacio multiresolución particular generado por las funciones splines [Vaidyanathan2001].

2.3 Interpolación en el subespacio de funciones splines

El subespacio formado por las funciones splines es un caso particular de los subespacios multiresolución generados mediante funciones de escala que se han visto en el Apartado 2.2. Sin embargo, dada la importancia de las funciones splines en esta Tesis, se tratan de forma independiente y se hace una introducción general de las mismas, de sus propiedades y representaciones, y de los diferentes algoritmos utilizados para su implementación. Además, se presentan las alternativas para la interpolación de una función y sus derivadas en este tipo de espacios: la denominada interpolación *oscultoria* o de Hermite, y la particularización del método presentado en la Sección 2.2.2 para el caso de funciones splines.

2.3.1 Introducción

Los splines polinómicos son una de las primeras alternativas de modelado o de aproximación funcional que emplearon una serie de restricciones sobre las derivadas del modelo, en este caso restricciones de continuidad, para tratar de mejorar la aproximación obtenida con el mismo. Los splines han obtenido una gran aceptación y se han empleado con éxito en un gran número de aplicaciones. A continuación se presenta una introducción sobre las funciones splines y se muestran sus principales características.

La aproximación utilizando polinomios es una alternativa para la aproximación funcional que resulta interesante cuando se desea aproximar una función suave de forma local. Por ejemplo, un desarrollo en serie de Taylor truncado

$$\hat{f}(x) = \sum_{d=0}^D \frac{(x - x_0)^d}{d!} \left. \frac{d^d f(x)}{dx^d} \right|_{x=x_0}, \quad (2.50)$$

proporciona una aproximación satisfactoria de $f(x)$ si la función es suficientemente suave y si x está suficientemente cerca de x_0 . Pero si la función ha de ser aproximada en un intervalo relativamente largo, el grado del polinomio ha de ser generalmente excesivamente grande para resultar práctico. Lo mismo ocurre con una aproximación polinómica convencional. Además, los polinomios de grado elevado presentan una serie

de problemas añadidos, como la inestabilidad o la posibilidad de oscilaciones. Una alternativa es subdividir el intervalo de aproximación, $[a, b]$, en intervalos suficientemente pequeños $[\xi_j, \xi_{j+1}]$ (con $a = \xi_1 < \dots < \xi_{l+1} = b$), de forma que en dichos intervalos un polinomio p_j de un grado relativamente pequeño pueda proporcionar una aproximación razonable de la función. Esto puede realizarse de formas diversas. Una de ellas es uniendo los tramos de polinomios de forma suave, es decir, que la función compuesta

$$\hat{f}(x) = p_j(x), \quad \text{para } x \in [\xi_j, \xi_{j+1}), \quad (2.51)$$

sea continua, y que, además, un cierto número de sus derivadas lo sea también. Este tipo de funciones polinómicas a tramos con un número de derivadas continuas es lo que se denomina normamente una función *spline* y su formulación inicial se debe al trabajo de Schoenberg [Schoenberg46]. En un spline de grado k , cada segmento es un polinomio de grado k . Los puntos de unión de los distintos polinomios se denominan generalmente *knots* o *puntos de ruptura*. Schoenberg introduce el fundamento matemático del problema y específicamente muestra como se pueden utilizar los splines para interpolar una secuencia de muestras equiespaciadas de una función. Además introduce las funciones B-splines, que son los *átomos* básicos a partir de los cuales se construyen los splines.

Curiosamente, a pesar de su temprano descubrimiento (el trabajo de Schoenberg data de 1946), los splines no fueron prácticamente utilizados hasta principios de los años 60, cuando los matemáticos se dieron cuenta de que estas funciones podían ser utilizadas para modelar el proceso físico de dibujar una curva suave. Desde entonces los splines han sido ampliamente utilizados en aproximación funcional debido a sus características de suavidad. De hecho, los más populares y más comúnmente utilizados, los splines cúbicos, deben en gran parte su aceptación a su conocida propiedad de máxima suavidad [DeBoor78] (también conocida como propiedad de mínima curvatura), que significa que un spline cúbico es la función que interpola un determinado número de puntos con una integral de la derivada segunda al cuadrado (que es una medida habitual de suavidad) mínima. Este descubrimiento generó un enorme interés en este campo y rápidamente los splines comenzaron a utilizarse en aplicaciones como teoría de la aproximación, [DeBoor72], [Schumaker81], análisis numérico [Prenter75] y otras diversas ramas de matemáticas aplicadas [Ahlberg67]. Más tarde, con la llegada de los ordenadores, los splines atrajeron la atención del mundo ingenieril y causaron un enorme impacto en el diseño asistido por ordenador [Diercks95], [Lancaster86] y en los gráficos por ordenador [Bartels87].

Entre las diferentes representaciones que admite una función spline, las dos más habituales son la polinómica y la B-spline. La primera consiste en una descripción en términos de los puntos de ruptura o *knots* ξ_1, \dots, ξ_{l+1} , y de los coeficientes c_{ij} de cada uno de los polinomios locales p_j . Al ser cada polinomio de grado k , se necesitan $k + 1$ coeficientes para describir cada segmento (por eso se suele hablar de orden $k + 1$). Hay que tener en cuenta, sin embargo, las restricciones adicionales de suavidad que imponen la continuidad de las derivadas de hasta orden $k - 1$ en los puntos de ruptura, que limitan los grados de libertad efectivos del sistema. Esta es la forma más natural

de representación y la más conveniente para la evaluación del modelo spline.

Sin embargo, la representación más comúnmente empleada es la representación en términos de B-splines. Para simplificar en lo posible la notación, se considera una secuencia normalizada de puntos de ruptura equidistantes (distancia 1). En este caso, la representación en términos de B-splines tiene la forma

$$\hat{f}(x) = \sum_{n \in \mathbb{Z}} c[n] \beta^k(x - n). \quad (2.52)$$

El prefijo “B” significa básico o base, y se puede ver que la representación B-spline es una combinación lineal de una función base desplazada. Esa función base, $\beta^k(x)$, es la función B-spline de grado k . Los parámetros del modelo son los coeficientes B-spline, $c[n]$. Los B-splines son funciones simétricas, con forma de campana y con soporte compacto, lo que les confiere una gran utilidad. De hecho, Schoenberg mostró que son los splines polinómicos más simples [Schoenberg73]. Su expresión analítica es la siguiente

$$\beta^k(x) = \frac{1}{k!} \sum_{n=0}^{k+1} \binom{k+1}{n} (-1)^n \left(x - n + \frac{k+1}{2} \right)_+^k, \quad (2.53)$$

donde

$$(x)_+^k = \begin{cases} x^k, & x \geq 0 \\ 0, & x < 0 \end{cases}, \quad (2.54)$$

es la función potencia positiva. Se puede ver con claridad que $\beta^k(x)$ es una función polinómica a tramos de orden k . Esto también implica que la derivada de orden $(k+1)$ es una serie de deltas de Dirac, lo que indica que $\beta^k(x)$ es diferenciable hasta el orden k . Para k impar los puntos de discontinuidad de la derivada de orden k son los enteros, mientras que para k par lo son los medio enteros.

Los B-splines también se pueden construir de una forma más simple convolucionando $(k+1)$ veces un pulso rectangular (B-spline de orden 0). Es decir, si

$$\beta^0(x) = \begin{cases} 1, & -\frac{1}{2} < x < \frac{1}{2} \\ \frac{1}{2}, & |x| = \frac{1}{2} \\ 0, & \text{en otro caso} \end{cases}, \quad (2.55)$$

entonces

$$\beta^k(x) = \underbrace{\beta^0(x) * \beta^0(x) * \dots * \beta^0(x)}_{k+1 \text{ veces}}. \quad (2.56)$$

En la Figura 2.4 se muestran los B-splines de grados 0 hasta 3. El spline de grado 3, o spline cúbico, es el más popular y el más comúnmente empleado. Si se calcula la expresión de su B-spline, usando (2.56), se llega a

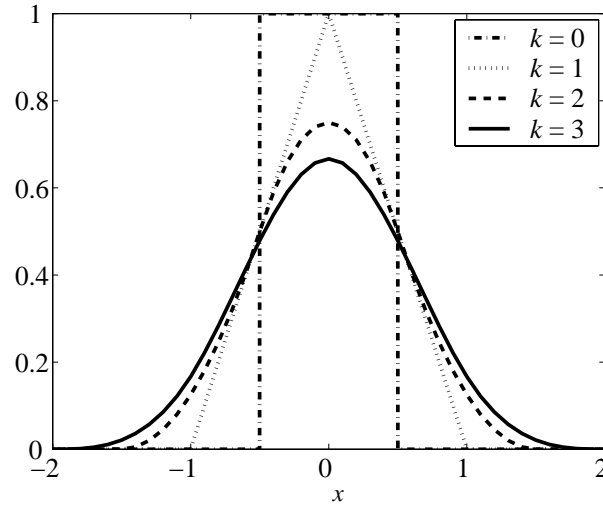


Figura 2.4: B-splines, $\beta^k(x)$, de grados 0 hasta 3

$$\beta^3(x) = \begin{cases} \frac{2}{3} - |x|^2 + \frac{|x|^3}{2}, & 0 \leq |x| < 1, \\ \frac{(2 - |x|)^3}{6}, & 1 \leq |x| < 2, \\ 0, & 2 \leq |x|. \end{cases} \quad (2.57)$$

Gracias a la representación B-spline, cada spline es caracterizado de forma completa y única por su secuencia de coeficientes B-spline, $c[n]$. La gran ventaja de los B-splines es que son muy fáciles de manipular. Por ejemplo, sus derivadas se pueden obtener mediante

$$\frac{d\beta^k(x)}{dx} = \beta^{k-1}\left(x + \frac{1}{2}\right) - \beta^{k-1}\left(x - \frac{1}{2}\right), \quad (2.58)$$

que reduce el grado del spline en 1. De forma similar, la integral de un B-spline de grado k se puede evaluar como

$$\int_{-\infty}^{\infty} \beta^k(x) dx = \sum_{n=0}^{\infty} \beta^{k+1}\left(x - \frac{1}{2} - n\right). \quad (2.59)$$

Una vez conocido el efecto de operadores lineales como (2.58) o (2.59) en las funciones base, resulta trivial aplicarlos a cualquier función spline a través de la representación B-spline (2.52).

2.3.1.1 Propiedades de suavidad

Los splines tienen varias propiedades muy interesantes. Una de ellas es la que se denomina *relación de la primera integral* [Ahlberg67], que establece que para cualquier

función $f(x)$ cuya derivada de orden m es de cuadrado integrable, y de la que se dispone de un conjunto de medidas $y[n] = f(x[n])$, se cumple que

$$\int_{-\infty}^{\infty} \left(\frac{d^m f(x)}{dx^m} \right)^2 dx = \int_{-\infty}^{\infty} \left(\frac{d^m \hat{f}(x)}{dx^m} \right)^2 dx + \int_{-\infty}^{\infty} \left(\frac{d^m (f(x) - \hat{f}(x))}{dx^m} \right)^2 dx \quad (2.60)$$

donde $\hat{f}(x)$ es el spline interpolador (es decir, $\hat{f}(x[n]) = y[n]$) de grado $k = 2m - 1$. De esta relación se desprende que el spline interpolante $\hat{f}(x)$ es la única función interpolante que minimiza la norma de la derivada de orden m , lo que es un resultado importante [Schoenberg73], y que viene a decir que el spline es la función interpolante que menos oscila. Para el caso $m = 2$, correspondiente a los splines cúbicos, la función de energía correspondiente es una aproximación de la medida de curvatura (o de suavidad) de la función $f(x)$. De esta forma, el spline cúbico interpolante exhibe la propiedad de *mínima curvatura* (o de *máxima suavidad*), lo que constituye una de las razones que ha impulsado su desarrollo en todo tipo de aplicaciones. Como curiosidad, esto justifica la analogía con la denominada *curva francesa* o “*draftman’s spline*”, que es un haz elástico obligado a pasar por una serie de puntos, y que es de donde proviene el nombre de las funciones splines.

2.3.1.2 Obtención de los coeficientes B-spline mediante filtrado digital

Hasta hace poco los splines y el mundo del procesado de señal, con excepción, quizá, del tratamiento de imagen, han estado en campos totalmente diferenciados. Básicamente porque los primeros se enmarcaban dentro del mundo matemático y porque el mundo del procesado de señal estaba fundamentalmente centrado en la teoría de muestreo de Shannon para funciones limitadas en banda. Sin embargo, recientemente los splines han recibido un renovado interés por parte de la comunidad científica de procesado de señal, debido fundamentalmente a los trabajos de M. Unser en este campo [Unser91, Unser93a, Unser93b, Unser99, Unser2000]. En particular, como se comenta en [Unser2000], los splines pueden solventar algunos de los problemas que presenta la teoría de Shannon; los más típicos son

- Precisa del uso de filtros ideales.
- La idea de limitación en banda está en oposición con la idea de señales de duración finita.
- La operación de limitar en banda puede producir oscilaciones de Gibbs, que en el caso de procesado de imagen puede ser visualmente molesto.
- La función base para el espacio de las señales limitadas en banda, la función $\text{sinc}(x)$, es una función con una caída muy lenta, lo que hace que la computación en el dominio del tiempo sea bastante ineficiente.

Una de las principales razones detrás del renovado interés en el modelado mediante splines es que los coeficientes B-spline pueden calcularse mediante filtrado digital. Como ya se ha comentado, el problema de interpolar un conjunto de muestras, $y[n]$, con una función spline se puede reducir a determinar el modelo B-spline (2.52) para dichas muestras, lo que significa determinar los coeficientes B-spline, $c[n]$. Para splines de grado 0 (constantes a tramos) y de grado 1 (lineales a tramos), esto es trivial, ya que los coeficientes coinciden con los valores de las muestras: $c[n] = y[n]$. Sin embargo, para splines de un orden mayor el problema es más complejo. Tradicionalmente el problema se solucionaba utilizando teoría matricial: se obtenía un sistema matricial de ecuaciones y se resolvía empleando técnicas numéricas (sustitución hacia delante y hacia atrás o descomposición LU) [DeBoor78, Prenter75]. Pero a principios de los 90 se vio que este problema puede también ser solucionado utilizando sencillas técnicas de filtrado digital [Goshtasby90, Unser91, Unser93a, Unser93b]. Para derivar este tipo de algoritmos de procesamiento de señal se introduce el kernel discreto de los B-splines, $b^k(n)$ y su transformada z , $B^k(z)$, que se obtiene muestreando el B-spline de grado n

$$b^k(n) = \beta^k(x) \Big|_{x=n} \quad \xleftrightarrow{z} \quad B^k(z) = \sum_{n \in \mathbb{Z}} b^k(n) z^{-n}. \quad (2.61)$$

Para interpolar la secuencia de muestras, es decir, forzar $\hat{f}(x[n]) = y[n]$, hay que determinar los coeficientes $c[n]$ del modelo (2.52) que ajusten las medidas. Usando los B-splines discretos esta restricción se puede escribir como

$$y[n] = b^k(n) * c[n], \quad (2.62)$$

y, por lo tanto

$$(b^k)^{-1}[n] \xleftrightarrow{z} 1/B^k(z), \quad (2.63)$$

la solución se encuentra mediante filtrado inverso (ver [Unser93a])

$$c[n] = (b_1^k)^{-1}[n] * y[n]. \quad (2.64)$$

Dado que b^k es un filtro FIR (respuesta impulsiva finita) simétrico, el denominado filtro B-spline directo, $(b^k)^{-1}$ es un filtro todo polos que se puede implementar eficientemente utilizando una cascada de dos filtros recursivos, uno causal y otro anticausal [Unser91, Unser93b]. Este algoritmo es numéricamente estable, rápido y fácil de implementar.

A la hora de establecer una relación entre la interpolación mediante splines y el tradicional marco teórico para funciones limitadas en banda, es útil introducir las funciones base “*cardinal spline*”, que serían el análogo a la función $\text{sinc}(x)$. Combinando (2.52) y (2.64) se tiene que

$$\begin{aligned}
\hat{f}(x) &= \sum_{n \in \mathbb{Z}} ((b^k)^{-1}[n] * y[n]) \beta^k(x - n) \\
&= \sum_{n \in \mathbb{Z}} y[n] \sum_{l \in \mathbb{Z}} (b^k)^{-1}[l] \beta^k(x - l - n) \\
&= \sum_{n \in \mathbb{Z}} y[n] \phi^n(x - n),
\end{aligned} \tag{2.65}$$

de donde se obtiene el cardinal spline de grado k

$$\phi^k(x) = \sum_{n \in \mathbb{Z}} (b^k)^{-1}[n] \beta^k(x - n). \tag{2.66}$$

De esta forma, la expresión (2.65) proporciona una fórmula de interpolación mediante splines que utiliza los valores de la señal, $y[n]$, como coeficientes. Esto es posible porque la función cardinal spline tiene la misma propiedad de interpolación que la función $\text{sinc}(x)$: tiene nulos en todos los valores enteros excepto en el origen, donde vale uno. El cardinal spline representa la respuesta al impulso del correspondiente spline interpolador. Aunque para $k \geq 2$ estas funciones no tienen soporte compacto, tienen una caída de tipo exponencial, mucho más rápida que la de la función $\text{sinc}(x)$. Por otra parte, la respuesta en frecuencia para un spline interpolador de grado k es

$$H^k(\omega) = \left(\frac{\sin(\omega/2)}{\omega/2} \right)^{k+1} \frac{1}{B_1^k(e^{j\omega})}. \tag{2.67}$$

Las respuestas al impulso y en frecuencia del cardinal spline cúbico aparecen en la Figura 2.5, en la que se aprecia su similitud con la función $\text{sinc}(x)$. De hecho, se puede demostrar que la función $\phi^k(x)$ converge a la función $\text{sinc}(x)$ cuando k tiende a infinito. Es un tipo de convergencia fuerte (en el sentido de la norma L_p) que se produce tanto en el tiempo como en la frecuencia [Aldroubi92].

Es interesante hacer notar que la relación entre los splines de orden infinito y las funciones limitadas en banda ya era conocida por Schoenberg o De Boor. Sin embargo, estos resultados matemáticos no han llegado a la comunidad de procesamiento de señal hasta hace bien poco, debido fundamentalmente a las importantes diferencias de contexto y terminología. Es también importante resaltar que la principal utilidad de los cardinal splines es conceptual, ya que proporcionan una mejor comprensión del algoritmo. No obstante, desde un punto de vista práctico resulta mucho más conveniente trabajar con la expansión en B-splines (2.52). Si se tiene en cuenta que, generalmente, la parte más costosa computacionalmente es la evaluación del modelo, dicha representación tiene la ventaja de que sus funciones base son de soporte compacto (los B-splines), de manera que el número de términos que contribuye en un determinado punto, x , es limitado (en concreto $k + 1$). La función $\text{sinc}(x)$, por el contrario, decae de forma lenta (como $1/|x|$), por lo que para computar un determinado valor con un error razonable es preciso tener en cuenta la contribución de un elevado número de muestras, lo que supone un

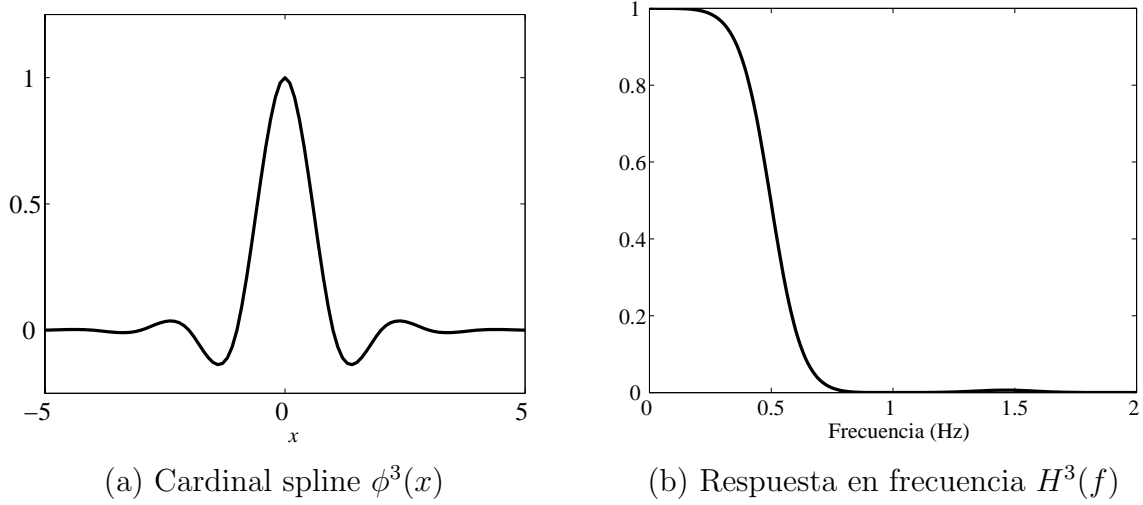


Figura 2.5: Respuestas al impulso y en frecuencia de un spline cúbico interpolador

coste computacional mucho más elevado. Si se trabaja en espacios N dimensionales el problema se agrava, lo que justifica la popularidad de los splines en procesamiento de imágenes, por ejemplo.

2.3.1.3 Extensión a espacios de entrada 2D

La teoría de los splines unidimensionales ha sido extendida de formas diferentes a espacios de entrada multidimensionales. De entre todas estas generalizaciones, la más extendida y ampliamente utilizada es la denominada *splines mediante producto de tensores* [DeBoor78]. Aunque presenta algunas limitaciones frente a otras alternativas, la popularidad de este tipo de splines se debe a su facilidad de manejo y la eficiencia de cálculo que permite su formulación.

Definición 2.7 Si se consideran las secuencias de puntos de ruptura estrictamente crecientes

$$a = \xi_0 < \xi_1 < \dots < \xi_{N_1-1} = b, \quad (2.68)$$

y

$$c = \mu_0 < \mu_1 < \dots < \mu_{N_2-1} = d, \quad (2.69)$$

la función $f(x_1, x_2)$ se denomina un spline bidimensional mediante producto de tensores en $R = [a, b] \times [c, d]$, de grado k en x_1 y l en x_2 , con puntos de ruptura ξ_i , $i = 0, 1, \dots, N_1$, en la dirección x_1 , y μ_j , $j = 0, 1, \dots, N_2$, en la dirección x_2 , si se cumplen las siguientes condiciones:

1. En cada rectángulo $R_{i,j} = [\xi_i, \xi_{i+1}] \times [\mu_j, \mu_{j+1}]$, $f(x_1, x_2)$ es un polinomio de grado k en x_1 y l en x_2 , lo que se suele denotar como

$$f(x_1, x_2)|_{R_{i,j}} \in P_k \otimes P_l, \quad (2.70)$$

donde P_k es el espacio de los polinomios de grado k .

2. La función $f(x_1, x_2)$ y sus derivadas parciales

$$\frac{\partial^{i+j} f(x_1, x_2)}{\partial x_1^i \partial x_2^j}, \quad i = 0, \dots, k-1; j = 0, \dots, l-1, \quad (2.71)$$

son continuas en R .

La extensión a espacios de entrada de mayor dimensión es inmediata. Un aspecto interesante de este tipo de splines bidimensionales es que, en términos de las funciones B-splines, admiten una representación única mediante productos de B-splines unidimensionales. En el caso de tener una secuencia normalizada de puntos de ruptura equidistantes (distancia 1), esta representación tiene la forma

$$f(x_1, x_2) = \sum_{n_1} \sum_{n_2} c[n_1, n_2] \beta^k(x_1 - n_1) \beta^l(x_2 - n_2). \quad (2.72)$$

El cálculo de los coeficientes del desarrollo se puede hacer de forma eficiente a partir de los algoritmos desarrollados para splines unidimensionales [DeBoor78, Diercks95]. Esta es la razón por la que, a pesar de algunas de sus limitaciones (como su limitación a espacios rectangulares), los splines mediante producto de tensores son los más utilizados en espacios de entrada multidimensionales.

2.3.2 Incorporación de la información de las derivadas

Como se vió en la Sección 2.3.1, los splines proporcionan una aproximación polinómica a tramos en la que, además de las condiciones de interpolación de la función, se añaden restricciones de suavidad en la solución, forzando para ello un determinado número de derivadas continuas. Dentro de las técnicas de interpolación mediante splines, hay distintas alternativas para introducir la información de las derivadas e interpolar un conjunto de muestras de la función y de una o varias de sus derivadas. A continuación se describen brevemente estas alternativas.

2.3.2.1 Interpolación de Hermite

Una función spline de grado k convencional interpola las muestras de la función y garantiza la continuidad de hasta la derivada de orden $k-1$. Si, además de muestras de la función, se dispone de muestras de la derivada, se pueden utilizar los grados de libertad adicionales del sistema, en lugar de para forzar continuidad de las derivadas, para interpolar también las mismas. Por ejemplo, un spline cúbico convencional interpola las medidas de la función y garantiza la continuidad de la primera y segunda derivadas. Sin embargo, si se dispone de medidas de la derivada es posible eliminar la restricción de continuidad de la derivada segunda e imponer las restricciones asociadas

a la interpolación de la derivada primera. Esta solución se denomina interpolación de Hermite [DeBoor78].

La misma idea se puede extender a splines de mayor grado cuando se dispone de medidas de más de una derivada. Cuando se dispone de muestras de D derivadas, se deben liberar tantas condiciones de continuidad en las derivadas de orden inferior como condiciones de interpolación se añaden. Por tanto, la solución se obtiene en el espacio de splines de grado $2D + 1$. En este caso, la interpolación produce un *contacto* de orden mayor entre la función y su aproximación. Este tipo de interpolación es muy fácil de llevar a cabo utilizando la representación B-spline. A grandes rasgos, en este caso la suavidad en los puntos de ruptura, o lo que es lo mismo, el número de derivadas continuas en tales puntos, está gobernada por la multiplicidad de la secuencia de puntos de ruptura. Replicando un determinado punto de ruptura se elimina una condición de continuidad en ese punto y el grado de libertad liberado se puede emplear para ajustar una de las derivadas [DeBoor78].

Siguiendo la misma idea que en el caso unidimensional, la interpolación de Hermite se ha extendido también a espacios de entrada bidimensionales. Se trata de liberar condiciones de continuidad de las derivadas de orden superior para añadir las condiciones de interpolación de las derivadas. Por ejemplo, con un spline cúbico bidimensional es posible interpolar las muestras de la función y de las derivadas

$$\frac{\partial f(x_1, x_2)}{\partial x_1}, \quad \frac{\partial f(x_1, x_2)}{\partial x_2} \quad \text{y} \quad \frac{\partial^2 f(x_1, x_2)}{\partial x_1 x_2}. \quad (2.73)$$

Los grados de libertad necesarios para añadir las condiciones de interpolación de las tres derivadas se liberan eliminando las condiciones de continuidad de las derivadas de orden superior. La misma idea se puede extender a un número mayor de derivadas.

De nuevo, como en el caso unidimensional, este procedimiento de interpolación se puede llevar a cabo de forma eficiente, utilizando la notación B-spline, mediante la replica de puntos de ruptura, que indica la condición de interpolación de las derivadas y la liberación de las condiciones de continuidad de las derivadas de orden superior.

Esta alternativa tiene el inconveniente de que genera soluciones con polinomios de grado elevado, especialmente a medida que aumenta el número de derivadas.

2.3.2.2 Interpolación mediante filtros de reconstrucción perfecta

Además de la técnica de interpolación de Hermite, que era conocida desde los primeros tiempos de la utilización de los splines, recientemente se ha desarrollado otro método de interpolación en subespacios multiresolución, de los que los splines son un caso particular, como ya se ha descrito en la Sección 2.2.2. A partir de la función de escala generadora de este espacio, que corresponde a las bases B-spline, es posible encontrar un conjunto de funciones de síntesis con las que filtrar las respectivas secuencias de muestras de función y derivadas para obtener la proyección sobre el espacio de los splines correspondientes. Vamos a ilustrar esta técnica con dos ejemplos. Consideraremos, en primer lugar, el problema de interpolar las muestras de la función y la primera derivada, $y^{(d)}[n]$, con $d = 0, 1$ y $n = 0, \dots, N - 1$. En este caso, la solución que se obtiene

es una solución sobre el espacio de los splines cuadráticos. La expresión de su función de escala es [Chui92]

$$\varphi(x) = \begin{cases} \frac{x^2}{2}, & 0 \leq x < 1 \\ -(x - 3/2)^2 + 3/4, & 1 \leq x < 2 \\ \frac{(x - 3)^2}{2}, & 2 \leq x < 3 \\ 0, & \text{en otro caso} \end{cases}, \quad (2.74)$$

que, obviamente, coincide con la función base B-spline de grado 2, $\beta^2(x)$, pero desplazada.

En primer lugar, se construye la matriz polifase, $E(z)$, de acuerdo a la estructura de banco de filtros de reconstrucción perfecta mostrada en las Figuras 2.2 y 2.3. La función de escala, su derivada y las muestras enteras de las mismas se muestran en la Figura 2.6.

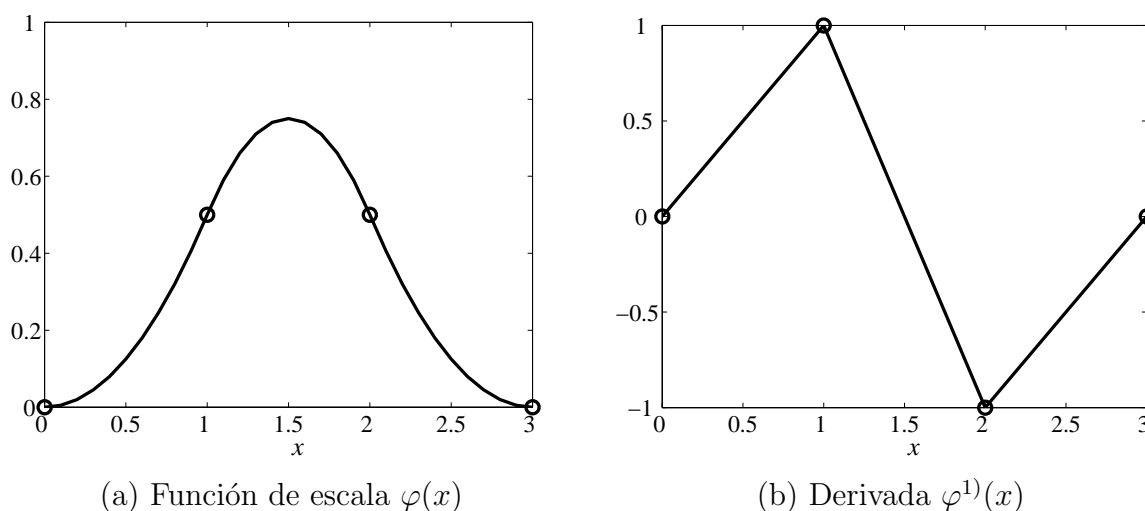


Figura 2.6: Funciones $\varphi(x)$ y $\varphi^{(1)}(x)$ y sus muestras en puntos enteros

A partir de las mismas es sencillo comprobar que

$$\Phi_0(z) = \sum_n \varphi(n)z^{-n} = \frac{1}{2}(z^{-1} + z^{-2}), \quad (2.75)$$

y

$$\Phi_1(z) = \sum_n \varphi^{(1)}(n)z^{-n} = z^{-1} - z^{-2}, \quad (2.76)$$

con lo que la matriz polifase tiene la forma

$$E(z) = \begin{pmatrix} z^{-1}/2 & 1/2 \\ -z^{-1} & 1 \end{pmatrix}, \quad (2.77)$$

y su inversa

$$R(z) = \begin{pmatrix} z & -z/2 \\ 1 & 1/2 \end{pmatrix}. \quad (2.78)$$

A partir de $R(z)$ es fácil comprobar que las funciones de síntesis tienen las siguientes expresiones

$$S_0(x) = \varphi(x+1) + \varphi(x+2), \quad (2.79)$$

y

$$S_1(x) = \frac{1}{2}\varphi(x+1) - \frac{1}{2}\varphi(x+2). \quad (2.80)$$

La función se reconstruye, empleando estas funciones de síntesis, utilizando la fórmula de reconstrucción

$$\hat{f}(x) = \sum_{n=0}^{N-1} y[n]S_0(x-2n) + y^{(1)}[n]S_1(x-2n). \quad (2.81)$$

Como segundo ejemplo se considera la reconstrucción de una función y sus dos primeras derivadas, $y^{(d)}[n]$, con $d = 0, 1, 2$ y $n = 0, \dots, N-1$, con lo que la proyección se realiza sobre el espacio de splines cúbicos. En este caso la función de escala vale

$$\varphi(x) = \begin{cases} \frac{x^3}{6}, & 0 \leq x < 1 \\ \frac{2}{3} - (x-2)^2 + \frac{(x-2)^3}{2}, & 1 \leq x < 2 \\ \frac{2}{3} - (x-2)^2 + \frac{(2-x)^3}{2}, & 2 \leq x < 3 \\ \frac{(4-x)^3}{6}, & 3 \leq x < 4 \\ 0, & \text{en otro caso} \end{cases}, \quad (2.82)$$

que coincide con la función B-spline $\beta^3(x)$ (2.57), pero desplazada. Es sencillo comprobar que, en este caso

$$\Phi_0(z) = \sum_n \varphi(n)z^{-n} = \frac{1}{6}(z^{-1} + 4z^{-2} + z^{-3}), \quad (2.83)$$

$$\Phi_1(z) = \sum_n \varphi^{(1)}(n)z^{-n} = \frac{1}{2}(z^{-1} - z^{-3}), \quad (2.84)$$

$$\Phi_2(z) = \sum_n \varphi^{(2)}(n)z^{-n} = z^{-1} - 2z^{-2} + z^{-3}, \quad (2.85)$$

con lo que la matriz polifase tiene la forma

$$E(z) = \begin{pmatrix} z^{-2}/6 & 2z^{-1}/3 & 1/6 \\ -z^{-2}/2 & 0 & 1/2 \\ z^{-2} & -2z^{-1} & 1 \end{pmatrix}, \quad (2.86)$$

y su inversa

$$R(z) = \begin{pmatrix} z^2 & -z^2 & z^2/3 \\ z & 0 & -z/6 \\ 1 & 1 & 1/3 \end{pmatrix}. \quad (2.87)$$

A partir de $R(z)$ es fácil obtener las funciones de síntesis, que tienen las siguientes expresiones

$$S_0(x) = \varphi(x+1) + \varphi(x+2) + \varphi(x+3), \quad (2.88)$$

$$S_1(x) = \varphi(x+1) - \varphi(x+3), \quad (2.89)$$

$$S_2(x) = \frac{1}{3}\varphi(x+1) - \frac{1}{6}\varphi(x+2) + \frac{1}{3}\varphi(x+3). \quad (2.90)$$

La función se reconstruye, empleando estas funciones de síntesis, utilizando la fórmula de reconstrucción

$$\hat{f}(x) = \sum_{n=0}^{N-1} y[n]S_0(x-3n) + y^1[n]S_1(x-3n) + y^2[n]S_2(x-3n). \quad (2.91)$$

Esta alternativa es bastante diferente a la interpolación de Hermite. En este caso, cuando se tienen muestras de D derivadas, la proyección se realiza sobre el espacio de los splines de grado $D+1$, mientras que en el caso de la interpolación de Hermite, la solución pertenece al espacio de splines de grado $2D+1$. Por otro lado, en la interpolación de Hermite, los puntos de ruptura son los instantes de muestreo, mientras que en este caso se insertan entre cada dos instantes de muestreo otros D instantes de ruptura, con lo que se produce, de forma implícita, una interpolación discreta de las secuencias de muestras. De hecho, como se verá en el Capítulo 3 (Sección 3.1), en el que se presenta una formulación alternativa para este problema, es posible interpolar las secuencias de muestras por un factor $D+1$, obteniendo así un conjunto de muestras interpoladas de la función sobre la que, aplicando un spline convencional, se llega a la misma solución. De alguna manera, muestrear las derivadas resulta equivalente a sobremuestrear la función.

Recientemente, en [Vaidyanathan2001], se han presentado estas expresiones de reconstrucción, haciendo referencia al hecho de que el muestreo de las derivadas permite obtener la reconstrucción con simples filtros FIR, en lugar de necesitar filtros IIR, como en el caso de disponer únicamente de muestras de la función.

2.4 Interpolación regularizada

En las secciones anteriores de este capítulo se han presentado varias alternativas para la interpolación de una función y sus derivadas. Sin embargo, en algunos casos la interpolación no es la solución más conveniente para llevar a cabo la reconstrucción de una función. Por ejemplo, cuando las muestras de que se dispone son ruidosas, una solución interpoladora introduce el error debido al ruido en la solución, pudiendo incluso, en determinadas circunstancias, amplificar el efecto del ruido en la solución final. En estos casos, resulta conveniente relajar la condición de interpolación e imponer ciertas restricciones a la solución que puedan minimizar, dentro de lo posible, el efecto del ruido. Una de las alternativas más comúnmente empleadas es la regularización del conjunto de muestras, que habitualmente consiste en imponer restricciones de suavidad en la solución.

En esta sección se introducen algunos aspectos de la teoría de regularización, y se considera el ejemplo particular de la interpolación mediante splines, dado su interés en esta Tesis.

2.4.1 Teoría de la Regularización

Es un hecho bien conocido que el problema de encontrar una función que interpole un determinado conjunto de medidas, sin ninguna restricción adicional, está mal condicionado. Existe un conjunto infinito de funciones que pueden ser una solución del problema. Por esta razón es preciso hacer una serie de suposiciones *a priori* que permitan transformar el problema en bien condicionado [Poggio90]. Una de las suposiciones más simples y razonables es la de suavidad: un modelo se puede considerar suave cuando pequeños cambios en algún parámetro de entrada del sistema determina pequeños cambios en la salida del mismo. La mayoría de los sistemas reales se comportan de este modo: es, por tanto, una de las restricciones más generales que es posible imponer. El marco teórico en el que se engloba un amplio grupo de técnicas de aproximación que utilizan este tipo de restricciones es la teoría de la regularización [Tikhonov77].

Las técnicas convencionales de regularización se basan en la minimización de una función de coste que consta de dos términos: el primero se encarga de medir la fidelidad en el ajuste del conjunto de medidas que se tratan de aproximar, y el segundo término tiene en cuenta el coste asociado con un funcional que mide la suavidad de la solución

$$E = \sum_n (\hat{f}(x[n]) - y[n])^2 + \lambda \|P\hat{f}(x)\|^2, \quad (2.92)$$

donde $y[n]$ son las medidas de la función en los puntos $x[n]$, λ es el parámetro de regularización, y P es un funcional (estabilizador) que se encarga de medir la suavidad.

La suavidad puede ser evaluada de varias formas diferentes. Generalmente, el estabilizador P está relacionado con alguna derivada de la función. Una de las más extendidas clases de estabilizadores es la considerada por Duchon [Duchon77] y Meinguet [Meinguet79]

$$\|P^m f(x)\|^2 = \sum_{i_1 \dots i_m}^n \int_{R^n} dx (\partial_{i_1 \dots i_m} f(x))^2, \quad (2.93)$$

donde $\partial_{i_1 \dots i_m} = \partial^m / \partial x_{i_1} \dots \partial x_{i_m}$ y $m \geq 1$. Este tipo de estabilizadores tienen la interesante propiedad de ser invariantes frente a la rotación y a la traslación. En este trabajo, para evaluar la suavidad, se utilizarán este tipo de estabilizadores.

Las técnicas de regularización se aplican tanto en el ámbito de la interpolación, cuando el modelo subyacente permite el ajuste exacto de las muestras, pero se prefiere regularizar la solución, como en el ámbito del modelado, en el que no se pretende en ningún caso un ajuste exacto de las muestras disponibles, sino la minimización de algún tipo de criterio de error de la solución, en general el error cuadrático.

2.4.2 Splines Regularizados

Uno de los ejemplos más conocidos de interpolación regularizada, son los splines regularizados. Aunque las funciones splines permiten una interpolación exacta del conjunto de muestras de una función, en casos en los que las muestras son ruidosas, resulta habitual pasar de un problema de interpolación a un problema de regularización y buscar la función $\hat{f}(x)$ que minimiza un funcional del tipo

$$\sum_{n \in Z} (f(x[n]) - \hat{f}(x[n]))^2 + \lambda \int_{-\infty}^{\infty} \left(\frac{d^m \hat{f}(x)}{dx^m} \right)^2 dx. \quad (2.94)$$

Formamos así un funcional regularizado, como (2.92), en el que el primer término pondera el error entre el modelo y las medidas y el segundo impone una restricción en la suavidad del modelo. La elección del parámetro λ se hace en función de algún conocimiento a priori del problema, bien de la potencia del ruido o del grado de suavidad de la función que se trata de reconstruir. Se puede demostrar que la solución óptima es un spline de grado $k = 2m - 1$ [Schoenberg64, Reinsh67], que se denomina *spline regularizado o suavizado* porque es equivalente a regularizar o suavizar los datos antes de reconstruir la función. Para $m = 2$, que corresponde al caso más habitual de medida de la suavidad, la solución al problema es un spline cúbico. En general, bajo una adecuada selección del parámetro de regularización, λ , esta solución proporciona, en situaciones de ruido, mejores resultados que una solución interpoladora. La generalización de este problema regularizado a espacios de entrada bidimensionales conduce a los llamados *thin plate splines* [Poggio90].

2.5 Redes Neuronales

A diferencia de las alternativas presentadas en las secciones previas de este capítulo, encuadradas en el ámbito de la interpolación, las redes neuronales, aunque también

pueden ser utilizadas en este ámbito, se encuentran principalmente enfocadas a aplicaciones de modelado o aproximación funcional donde, en general, se requieren modelos parsimoniosos, con un número reducido de parámetros. En este caso, ya no se intenta realizar un ajuste exacto del conjunto de medidas disponible, sino que se trata de minimizar algún criterio de error sobre las mismas, como, por ejemplo, el error cuadrático.

En lo que a esta Tesis respecta, el interés en las redes neuronales reside en que esta es una alternativa de modelado no lineal que ha experimentado un enorme auge en los últimos años y en que, como veremos con posterioridad, presentan una reconocida capacidad para aproximar una función y sus derivadas.

En esta sección se hace una breve introducción sobre las redes neuronales, con especial interés en los tipos de redes utilizadas en esta Tesis, y se discuten las aportaciones que aparecen en la literatura sobre la utilización de las mismas para el modelado simultáneo de una función y sus derivadas.

2.5.1 Introducción

“Una red neuronal es un procesador distribuido masivamente paralelo que presenta una natural propensión para almacenar conocimiento experimental y hacerlo disponible para su uso. Las redes neuronales se asemejan al cerebro en dos aspectos:

- 1. El conocimiento se adquiere por la red a través de un proceso de aprendizaje.*
- 2. Las conexiones entre los diferentes elementos (neuronas) de la red, conocidas también como uniones sinápticas, se emplean para almacenar este conocimiento.”*

Esta es la definición que hace Simon Haykin [Haykin94] de una red neuronal. Las redes neuronales nacieron a principios de los años 50 dentro del campo de la inteligencia artificial (IA) con la idea de modelar los aspectos físicos de la mente. Es interesante contrastar esta idea con los hasta entonces convencionales sistemas de IA que se basaban en que la mente puede ser modelada de forma independiente de los mecanismos físicos en los que está basada. Los investigadores de este campo asumen que la consciencia es la causa del comportamiento inteligente, y por lo tanto tratan de modelar la consciencia a través de procesos como la resolución de problemas o el razonamiento. En cambio, la visión convencional de las redes neuronales es que la consciencia es sólo un fenómeno emergente de la compleja interacción de los millones de neuronas que forman nuestro cerebro. Por esta razón, se propone la alternativa de intentar modelar ese tipo de fenómenos físicos del cerebro. Bajo estas premisas aparecen una serie de estructuras, más o menos complejas, que tratan de modelar el funcionamiento de las neuronas y su agrupación utilizando diferentes arquitecturas de red. Se han definido diferentes modelos de neuronas, diferentes tipos de arquitecturas de red, para la conexión de las mismas, y diferentes tipos de algoritmos de entrenamiento. Para un estudio detallado se puede consultar, por ejemplo [Hecht90, Haykin94, Bishop97, Hush92, Hush92a].

Centrándonos ya en el ámbito de esta Tesis, en problemas de modelado o aproximación funcional, existen dos tipos de redes neuronales que son especialmente conocidas

y empleadas: el *Perceptrón Multicapa* o MLP (*“Multilayer Perceptron”*) y la *Red de Funciones de Base Radial* o RBF (*“Radial Basis Function network”*). A continuación se presentan las principales características de estos dos tipos de redes.

2.5.2 El Perceptrón Multicapa

El Perceptrón Multicapa (MLP), es, sin lugar a dudas, la red neuronal más conocida y utilizada. Se trata de una red neuronal con arquitectura multicapa hacia delante (*“feedforward”*), tal y como se muestra en la Figura 2.7. Las neuronas se agrupan en una capa de entrada, una o varias capas ocultas, y una capa de salida, y las entradas de las neuronas de cada capa son las salidas de las neuronas de la capa anterior.

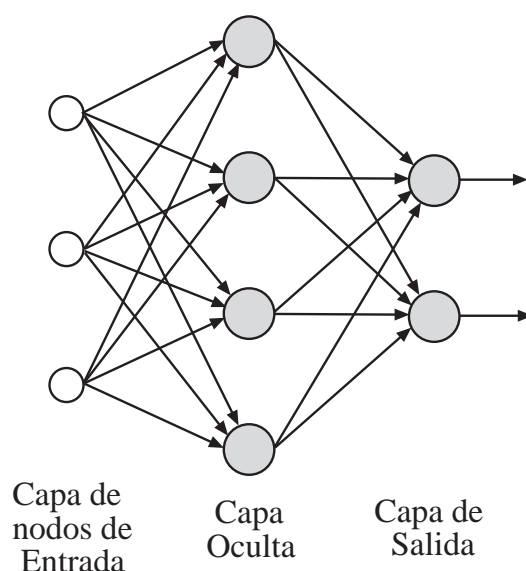


Figura 2.7: Arquitectura de red neuronal hacia delante típica del Perceptrón Multicapa

Como función no lineal de activación, las neuronas de la capa oculta suelen utilizar la función sigmoideal o la función tangente hiperbólica [Haykin94], mientras que en la capa de salida, en problemas de aproximación funcional, suele ser lineal. El entrenamiento de estas redes se realiza mediante el conocido algoritmo de *Retropropagación* (*“Backpropagation”*), descrito inicialmente por Werbos en su Tesis Doctoral [Werbos74] y posteriormente popularizado por Rumelhart y sus colaboradores [Rumelhart86, Rumelhart86b]. El descubrimiento de este algoritmo supuso una revolución en el desarrollo de las redes neuronales. Básicamente, el algoritmo de retropropagación consiste en dos pasos: en el primer paso, hacia delante, los patrones de entrenamiento se aplican a la red y se propagan por las capas hasta llegar a la capa de salida para producir una salida de la red. Durante este paso, los pesos o parámetros de la red permanecen fijos. En el segundo paso, hacia atrás, se obtiene una señal de error como la diferencia entre la salida esperada y la obtenida al propagar los patrones. Ese error se propaga hacia

atrás en la red, definiendo una dirección de variación de los parámetros de la misma para minimizar dicho error. De aquí procede el nombre de Retropropagación.

2.5.3 La Red de Funciones de Base Radial

La Red de Funciones de Base Radial (RBF) debe su nombre al tipo de funciones de activación de sus neuronas. Las funciones de base radial fueron inicialmente concebidas para resolver los problemas de interpolación en espacios de entrada de varias dimensiones y, posteriormente, se extendió su utilización en problemas de modelado o aproximación funcional. Desde este punto de vista, los primeros trabajos fueron llevados a cabo por Powell [Powell85]. Más tarde, Broomhead y Lowe fueron los primeros en aprovechar el uso de este tipo de funciones en el mundo de las redes neuronales [Broomhead88]. Otras contribuciones importantes en este campo han sido llevadas a cabo por Moody y Darken [Moody89], presentando un eficiente algoritmo de entrenamiento, y por Poggio y Girosi [Poggio90], demostrando que la RBF surge como solución a un problema de interpolación regularizada.

En su forma más general, una RBF implementa una relación entrada-salida de la forma siguiente

$$f(\mathbf{x}) = \sum_{i=1}^N \lambda_i G(\mathbf{x}, \mathbf{c}_i), \quad (2.95)$$

donde las funciones base son radiales, es decir

$$G(\mathbf{x}, \mathbf{c}_i) = G(\|\mathbf{x} - \mathbf{c}_i\|), \quad (2.96)$$

siendo \mathbf{c}_i los *centroides* de las distintas funciones base. Se observa que el valor de la función depende únicamente de la distancia (norma) al centroide. Entre las distintas funciones de activación algunas de las más habituales son, la *gaussiana*

$$G(x) = \exp\left(-\frac{x^2}{\sigma^2}\right), \quad (2.97)$$

que es, sin duda alguna, la más utilizada, y los *thin plate splines*

$$G(x) = x^2 \ln(x), \quad (2.98)$$

que resultan interesantes porque una RBF interpoladora, con tantas neuronas como puntos de medida, con esta función de activación, es la solución que minimiza la función de coste regularizada con el estabilizador

$$\|P^2 f\|^2 = \int_R \left[\left(\frac{\partial^2 f}{\partial x_1^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x_1 \partial x_2} \right)^2 + \left(\frac{\partial^2 f}{\partial x_2^2} \right)^2 \right] dx_1 dx_2, \quad (2.99)$$

en un espacio de entrada 2D [Poggio90], de forma similar a como un spline cúbico lo hacía para el mismo tipo de estabilizador en 1D (ver Sección 2.4.2).

En cuanto al entrenamiento de la red, el algoritmo de entrenamiento debe estimar los centroides y las varianzas de cada función base y los pesos de la capa de salida. Normalmente, el proceso de entrenamiento se separa en dos fases:

1. Mediante un proceso de optimización no lineal, se seleccionan los centroides y varianzas de las funciones base.
2. Los pesos de la capa de salida se estiman mediante un proceso de optimización lineal.

Las alternativas más simples para el proceso de optimización no lineal de centroides y varianzas consisten en fijar inicialmente las varianzas y seleccionar los centroides de forma aleatoria [Broomhead88] o aplicando algoritmos de cuantificación vectorial (*clustering*) [Moody89]. Sin embargo, cuando se emplean funciones base gaussianas, una alternativa más eficaz es utilizar el método OLS (*“Orthogonal Least Squares”*) [Chen91], en el que se van seleccionando uno a uno los centroides más idóneos en el sentido de que maximicen la varianza o energía de la salida, y que proporciona una inicialización bastante adecuada en la mayoría de los casos, aunque es dependiente de los parámetros de la función base, en este caso de σ^2 . Otras alternativas tratan de solucionar este problema de optimización utilizando procedimientos de entrenando supervisado, basados en gradiente, para estimar los parámetros de la red que minimicen una función de coste, generalmente el error cuadrático medio de la aproximación [Lowe89, Karayanis97, Santamaría99].

Uno de los problemas más conocidos de las RBF es la denominada *maldición de la dimensionalidad* (*“curse of dimensionality”*), que significa que, a medida que crece la dimensión del espacio de entrada, el número de neuronas necesario para poder obtener una buena aproximación crece exponencialmente con la dimensión del espacio. Una forma de paliar en cierto modo este problema es utilizar las denominadas funciones de base radial generalizadas (GRBF). Estas suponen una extensión de las RBF en el sentido de que ahora las funciones base van a poder tener una varianza diferente en diferentes direcciones del espacio de entrada, lo que formalmente se plantea a través de la utilización de una norma ponderada, es decir

$$f(\mathbf{x}) = \sum_{i=1}^N \lambda_i G(\|\mathbf{x} - \mathbf{c}_i\|_{\mathbf{P}_i}), \quad (2.100)$$

donde la expresión general de la norma ponderada es

$$\|\mathbf{x}\|_{\mathbf{P}}^2 = \mathbf{x}^T \mathbf{P}^T \mathbf{P} \mathbf{x}, \quad (2.101)$$

siendo \mathbf{P} una matriz cuadrada positiva definida de la dimensión del espacio de entrada. Para las GRBF gaussianas, que son las más comúnmente empleadas, esto se traduce en que la forma de las funciones base es

$$G(\|\mathbf{x} - \mathbf{c}_i\|_{\mathbf{P}_i}) = \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{c}_i)^T \Sigma_i^{-1}(\mathbf{x} - \mathbf{c}_i)\right), \quad (2.102)$$

siendo la matriz inversa Σ_i^{-1} dada por

$$\frac{1}{2}\Sigma_i^{-1} = \mathbf{P}_i^T \mathbf{P}_i. \quad (2.103)$$

Dentro de esta familia, la más habitual es aquella en la que las matrices de ponderación \mathbf{P}_i son diagonales, lo que significa que la función base tiene una varianza distinta en cada una de las direcciones que define el espacio de entrada, ya que esto simplifica el número de parámetros y las técnicas de entrenamiento. La más habitual es inicializar la posición de los centroides mediante OLS y luego ajustar su posición y varianzas mediante técnicas de gradiente, ajustando finalmente las amplitudes λ_i por mínimos cuadrados.

2.5.4 Capacidad de aproximación

Una de las más destacables cualidades de las redes neuronales es su conocida capacidad de aproximación. Existen en la bibliografía un gran número de trabajos mostrando la capacidad universal de aproximación de las redes multicapa hacia delante con funciones de tipo sigmoideal en las capas ocultas. Los primeros trabajos en esta línea fueron llevados a cabo, por Cybenko [Cybenko89] y Funahashi [Funahashi89], que demostraron que las funciones sigmoideales eran densas en varios sentidos, llegando a la conclusión de que cualquier función continua en un intervalo compacto puede ser aproximada con una precisión arbitraria mediante una red neuronal de este tipo, con una única capa oculta, y con un número suficiente de neuronas en la capa oculta.

Trabajos similares en la misma línea han sido llevados a cabo por Hecht-Nielsen [Hecht89], Leshno y sus colaboradores [Leshno93], o Hornik, Stinchcombe y White [Hornik89], en los que se discute qué restricciones sobre las funciones de activación hacen posible obtener una aproximación tan exacta como se desee. Stinchcombe y White hacen lo mismo para el caso de funciones de activación que no son de tipo sigmoideal [Stinchcombe89]. También existen resultados en la misma línea para funciones de base radial, como los presentados por Park y Sandberg [Park91], o Powell [Powell92], donde se demuestra que las RBF son también aproximadores universales.

Aunque este tipo de demostraciones son valiosas y han potenciado la utilización de las redes neuronales (especialmente del MLP), no existen estudios constructivos en el sentido de encontrar la dimensión adecuada de la red para un cierto problema, o la forma de calcular los parámetros óptimos de la misma. Es cierto que se han llevado a cabo diversos estudios para tratar de determinar el tamaño de la red, es decir, el número de neuronas necesario para aproximar un determinado tipo de funciones con una determinada precisión, como el clásico resultado de Barron [Barron93], en el que se muestra que para una función que satisfaga ciertas condiciones, expresadas en términos de su transformada de Fourier, y usando la función de activación sigmoideal, el tamaño de la red es del orden de $O(\epsilon^{-2})$ neuronas para conseguir una aproximación con un error inferior a ϵ . Un aspecto interesante de este resultado es que indica que el tamaño de la red es independiente de la dimensión del espacio de entrada. Resultados similares

se han obtenido por otros autores para determinadas condiciones de la función de activación.

Todos estos estudios demuestran la capacidad teórica de la red, pero no proporcionan mecanismos para obtener los parámetros de la misma, con lo que, en la práctica, encontrar la dimensión de la red es una cuestión laboriosa, agravada por el hecho de que los algoritmos de entrenamiento (especialmente para el MLP) tienen mínimos locales, es decir, no garantizan la convergencia al mínimo global de la función de coste, lo que hace preciso llevar a cabo repetidos experimentos hasta encontrar una solución cuyos resultados sean adecuados. A pesar de estas dificultades, hay que decir que las redes neuronales proporcionan, en la mayoría de las aplicaciones que necesitan de aproximación funcional a partir de un conjunto de muestras del problema a modelar, unos resultados satisfactorios.

2.5.5 Incorporación de la información de las derivadas

Como ya se mencionó en la Sección 2.5.4, las redes neuronales presentan la interesante propiedad de ser *aproximadores universales* en el sentido de que son capaces de aproximar una determinada función, bajo ciertas sencillas condiciones, con una precisión arbitraria. En un momento en el que la utilización de este tipo de redes alcanzó una gran aceptación, comenzaron también a utilizarse para la aproximación de determinadas funciones y de sus derivadas a partir de las medidas de la función. En esta línea se obtuvieron resultados aceptables, por ejemplo en los experimentos de Jordan [Jordan89], pero sin ninguna garantía teórica de que este tipo de redes tuviera, en general, la capacidad de aproximar una función y sus derivadas simultáneamente. Es cierto que la intuición sugiere que estas redes, con unas funciones de activación suaves en la capa oculta deberían funcionar relativamente bien en la aproximación de las derivadas. Sin embargo, la justificación de esta intuición no es obvia. Por un lado, la naturaleza de las funciones base y de sus derivadas no es, en general, la misma. Por ejemplo, considerando el caso de la típica función sigmoideal, es conocido que una red con tal función de activación es un aproximador universal. Sin embargo, sus derivadas ya no son sigmoideales, y en cualquier caso, aunque se pudiera demostrar que dichas funciones dan lugar también a un aproximador universal, no está claro que los pesos que hacen que la combinación de funciones base aproxime bien la función coincidan con los que aproximan bien la derivada.

De cara a solventar esta carencia a nivel teórico, surgieron varios estudios sobre la capacidad de las redes neuronales para aproximar también las derivadas. Sin duda, uno de los más relevantes es el llevado a cabo por Hornik, Stinchcombe y White [Hornik90]. En este estudio se dan las condiciones que aseguran que las redes multicapa hacia delante, con una única capa oculta y una apropiada función de activación, son capaces de aproximar de forma arbitrariamente exacta una determinada función y sus derivadas. En particular, demostraron que las funciones base que forman la red neuronal son densas en varios tipos de espacios de Sobolev. En general, las condiciones impuestas en cada caso a las funciones de activación son relativamente *suaves*, demostrándose

que las funciones de activación más comunes cumplen tales condiciones. Además, se ha demostrado también que este tipo de redes, con una única capa oculta, puede también aproximar funciones que no son diferenciables en el sentido clásico, sino que poseen una derivada generalizada, como es el caso de ciertas funciones diferenciables a tramos. Este trabajo de Hornik proporciona pues la justificación teórica que permite el uso de redes multicapa hacia adelante en aplicaciones que requieren la aproximación simultánea de una función y sus derivadas.

Existen varias aportaciones en la misma dirección; por ejemplo Gallant y White [Gallant92], que mostraron que las estima de mínimos cuadrados o similares son consistentes cuando el número de neuronas y el tamaño del conjunto de entrenamiento de la red aumentan de forma conjunta, o Ito [Ito93], que demostró la capacidad de las redes neuronales para llevar a cabo la aproximación de una función de tipo polinómico empleando un número acotado de neuronas en la capa oculta. Asimismo, aprovechando el teorema de aproximación polinomial de Nachbin [Nachbin67], este resultado se extiende para cualquier tipo de función.

En cuanto a la complejidad de la red también aparecen estudios, como el realizado por Mhaskar [Mhaskar96] en el que se obtienen cotas para la aproximación de funciones pertenecientes a clases definidas en términos del número de derivadas que poseen tales funciones, y las propiedades que deben cumplir las funciones de activación para obtener tales capacidades.

De nuevo, la mayoría de estos resultados tienen un elevado interés teórico por cuanto demuestran la capacidad de aproximación de este tipo de arquitecturas, pero no aparecen alternativas constructivas en las que realmente se utilice la información de las derivadas a la hora de llevar a cabo el entrenamiento o el diseño de la red. La mayoría de los intentos de aproximar las derivadas confían en un ajuste adecuado de la función para, a través de ese ajuste, tener una estima razonable de las derivadas [Jordan89, Li96, Nguyen99]. La única alternativa constructiva es la propuesta por Cardaliaguet y Euvrard [Cardaliaguet92] en la que se presenta una aproximación neuronal con fórmulas explícitas de los coeficientes para el caso de aproximación de una función y su derivada en un intervalo compacto utilizando funciones sigmoideas con derivadas con característica de campana. En concreto demostraron que, en el intervalo compacto $[-1, 1]$, la función $f(x)$ se puede aproximar mediante

$$f_n(x) = f(0) + \sum_{k=-n^2}^{k=n^2} \frac{f'(\frac{k}{n})}{In} S\left(n^{1-\alpha}\left(x - \frac{k}{n}\right)\right) \quad (2.104)$$

donde $I = \int_{-\infty}^{\infty} S(x)dx$, y $S(x)$ es una función de tipo sigmoideal que debe cumplir $S(0) = 0$ y cuya derivada debe tener característica de campana. En [Cardaliaguet92] se demuestra que las funciones $f'_n(x)$ convergen de forma uniforme a $f'(x)$ cuando n tiende a infinito.

Esta aproximación requiere conocer los valores de la derivada en ciertos puntos y además presenta problemas para extender su capacidad a espacios de mayor dimensión. Para espacios 2D, Cardaliaguet y Euvrard, obtuvieron una solución, pero sólo para

redes que permitan *neuronas producto*, es decir, neuronas cuya expresión 2-D sea el producto de dos neuronas 1D. Para espacios de mayor dimensión, no existen soluciones explícitas. Por eso, en este mismo artículo, los autores proponen como alternativa para estos casos la utilización de fórmulas de retropropagación del error extendiéndolo para incluir el error de las derivadas.

La capacidad teórica para la solución del problema que se aborda en la Tesis y la falta de alternativas constructivas en este ámbito de trabajo fueron la principal motivación para realizar un estudio de alternativas que permitan introducir la información de las derivadas en las redes neuronales. Parece lógico esperar que, si una red neuronal es capaz de generalizar de forma adecuada las derivadas de una función, tal y como se ha podido comprobar en diferentes aplicaciones [Jordan89, Li96, Nguyen99], la posibilidad de introducir de algún modo la información de las derivadas en la construcción de la misma pueda contribuir a mejorar esa capacidad de aproximación. Además, tampoco es descartable la posibilidad de que la introducción de dicha información pueda ser útil en otros sentidos tales como la inmunidad frente al ruido. Por esta razón, en la Tesis se ha realizado un estudio para evaluar estos factores. Este estudio se presenta en el Capítulo 5.

Capítulo 3

Una nueva técnica de interpolación/regularización para la reconstrucción simultánea de una función y sus derivadas

En este capítulo se presenta una nueva técnica de interpolación/regularización para la reconstrucción simultánea de una función y sus derivadas a partir de un conjunto de medidas de las mismas. Inicialmente consideramos el caso unidimensional; en este caso, la técnica propuesta, que hemos denominado *Modelo Local de Interpolación* (MLI), da lugar a una solución que proyecta la interpolación de las secuencias de muestras de función y derivadas sobre el espacio de las funciones splines (el orden del spline depende del número de derivadas que se pretende interpolar). Este tipo de solución, como la mayoría de técnicas de interpolación de función y derivadas, tiene como principal inconveniente que presenta una gran sensibilidad frente al ruido en las derivadas. Por esta razón, en este capítulo se propone y analiza una solución regularizada, que se ha denominado *Modelo Local Regularizado* (MLR). Como demostraremos, relajando las condiciones de interpolación e imponiendo restricciones para forzar la suavidad de la solución final, es posible llegar a una solución regularizada que también pertenece al espacio de las funciones splines, pero que presenta una menor sensibilidad al ruido.

El problema de la interpolación/regularización simultánea de una función y sus derivadas en un espacio bidimensional presenta, respecto al caso unidimensional, una notable complejidad adicional. Además, el problema es diferente si se dispone únicamente de las derivadas direccionales¹ de la función, o si, además, es posible disponer de información de las derivadas cruzadas, con respecto a ambas variables de entrada. Mediante un procedimiento novedoso, en este capítulo se propone una extensión de las técnicas de interpolación y regularización a espacios bidimensionales. Este procedimiento asegura

¹Aunque por definición todas las derivadas en un espacio 2D son direccionales, en este contexto denominamos así a las derivadas, de cualquier orden, con respecto a una única variable del espacio de entrada

la congruencia y continuidad de las derivadas.

El capítulo está organizado de la forma siguiente. En primer lugar, en la Sección 3.1 se presenta una nueva técnica para la proyección de las muestras de una función unidimensional y su(s) derivada(s) sobre el espacio de funciones splines, que hemos denominado *Modelo Local de Interpolación* (MLI) en espacios 1D. A continuación, en la Sección 3.2 se presenta la solución regularizada para la reconstrucción de la función partiendo del modelo MLI, que hemos denominado *Modelo Local Regularizado* (MLR) en espacios 1D. A partir de estos modelos unidimensionales, se presenta la extensión a espacios de entrada bidimensionales. En este caso, se proponen soluciones diferentes en función de la información de que se dispone:

- Cuando sólo se dispone de las derivadas direccionales de la función, se han propuesto un método semilocal para la interpolación de las muestras de la función y sus derivadas, que hemos denominado *Modelo Semilocal de Interpolación* (MSI), y su extensión regularizada, el *Modelo Semilocal Regularizado* (MSR).
- Cuando se dispone también de las derivadas cruzadas, se ha propuesto un método local de interpolación, denominado *Modelo Local de Interpolación* (MLI)². Para la extensión regularizada, se han propuesto dos alternativas: una técnica de regularización estricta, denominada *Modelo Local de Regularización* (MLR)², y una técnica de regularización simplificada, que reduce la carga computacional, que hemos llamado *Modelo Local Regularizado Simple* (MLRs).

Para finalizar, se presentan algunos de los resultados que se han obtenido con los modelos propuestos.

3.1 El Modelo Local de Interpolación (MLI) en espacios unidimensionales

Una de las aproximaciones más sencillas que se puede plantear, cuando se dispone de un conjunto de muestras de una función, es la aproximación lineal a tramos entre dichas muestras. Cuando se tiene un conjunto de medidas de la función y de sus D primeras derivadas, parece lógico plantear una solución en la que la derivada de orden D sea lineal a tramos y, a partir de ella, reconstruir el resto de derivadas, hasta llegar a la función, mediante integración. Cuando se plantea esta solución lineal a tramos entre las muestras, la reconstrucción de la función a partir de la derivada de orden D presenta problemas de discontinuidad en los instantes de muestreo. Si se desea forzar la continuidad de la solución propuesta, tanto en la función como en las D derivadas, hay que añadir una serie de grados de libertad al sistema. Una posibilidad para introducir

²En general, siempre va a estar claramente definido el tipo de espacio en el que se trabaja, por lo que no hay posibilidad de confundir las versiones unidimensional y bidimensional. Si el contexto lo requiere, se añadirá el sufijo 1D o 2D para diferenciarlos (MLI1D, MLI2D, MLR1D, MLR2D).

estos grados de libertad consiste en insertar entre cada dos muestras consecutivas una serie de puntos de ruptura adicionales. Estos puntos de ruptura van a ser los que permitan satisfacer las condiciones de continuidad de función y derivadas.

A continuación se presentan las expresiones concretas para el caso particular del método de interpolación de las muestras de una función y su derivada primera (modelo de grado 2) que, al ser al caso más simple, permite ver con claridad la idea general del método. Además, se presenta una interesante representación, mediante un banco de filtros, del método de interpolación propuesto, y se estudia el efecto del ruido sobre este tipo de reconstrucción. A partir del modelo de grado 2, se perfila la extensión del método para un número mayor de derivadas, cuyas expresiones particulares se incluyen en el Apéndice C.

3.1.1 Interpolación en el espacio de los splines

El problema general que se pretende resolver se plantea de la siguiente manera: dado un conjunto de N muestras de una función y de sus primeras D derivadas

$$y^d[n] = f^d(x[n]), \quad n = \{0, 1, \dots, N-1\}, \quad d = \{0, 1, \dots, D\}, \quad (3.1)$$

se desea reconstruir una función, $f(x)$, que cumpla las condiciones de interpolación dadas por (3.1).

Por simplicidad se ha considerado el caso de muestreo uniforme, con periodo de muestreo T , es decir, $x[n] = nT$, pero, como se verá más adelante, el método propuesto se puede extender a problemas de muestreo irregular de forma trivial. También por simplicidad, se van a presentar las expresiones resultantes del modelo para el caso particular $D = 1$, es decir, cuando se dispone únicamente de muestras de la función y de la derivada primera, pero la misma idea general del modelo se puede extender a un número mayor de derivadas. Asimismo, en lo sucesivo se va a obviar el superíndice $d = 0$ para simplificar en lo posible la notación.

En el caso particular que nos interesa, $D = 1$, se dispone de un conjunto de muestras de la función y de la derivada de la forma (3.1) con $d = 0, 1$. El objetivo es encontrar una solución en la que la última derivada disponible, en este caso la derivada primera, sea una función lineal a tramos y, además, se cumplan las condiciones de interpolación, que definen las muestras, y de continuidad, tanto para la función como para la derivada. Como resultado de este procedimiento, la señal reconstruida pertenece al espacio de las funciones splines (splines cuadráticos en este caso particular, por lo que se ha denominado modelo de grado 2).

Para cumplir con las condiciones de interpolación y de continuidad es preciso disponer de un número suficiente de grados de libertad en el sistema. En este caso los grados de libertad necesarios se obtienen a través de la inserción de puntos de ruptura que se añaden a los definidos inicialmente por los instantes de muestreo. De esta forma, entre cada dos instantes de muestreo consecutivos se inserta un nuevo punto de ruptura, en principio equidistante de ambos instantes de muestreo, dando lugar a unas secuencias

interpoladas por un factor 2

$$y_i^d[n], \quad d = \{0, 1\}, \quad n = \{0, 1, \dots, 2N - 2\}. \quad (3.2)$$

Planteando ahora un sistema en el que la derivada es lineal a tramos sobre estas secuencias interpoladas, con las restricciones impuestas por las condiciones de interpolación y de continuidad definidas por las muestras de la función y de la derivada, se pueden obtener los valores de la función y la derivada en los instantes de ruptura insertados. En particular, se obtiene que las secuencias interpoladas tienen las siguientes expresiones

$$y_i^d[2n] = y^d[n], \quad n = \{0, \dots, N - 1\}, \quad d = \{0, 1\}, \quad (3.3)$$

$$y_i^1[2n + 1] = \frac{2(y[n + 1] - y[n])}{T} - \frac{y^1[n] + y^1[n + 1]}{2}, \quad n = \{0, 1, \dots, N - 2\}, \quad (3.4)$$

$$y_i[2n + 1] = \frac{y[n] + y[n + 1]}{2} + T \frac{(y^1[n] - y^1[n + 1])}{8}, \quad n = \{0, 1, \dots, N - 2\}, \quad (3.5)$$

donde la secuencia de instantes de muestreo asociada a estas secuencias interpoladas es ahora

$$x_i[n] = \frac{nT}{2}, \quad n = \{0, 1, \dots, 2N - 2\}. \quad (3.6)$$

A partir de las secuencias interpoladas de la función y la derivada, el modelo fuerza una derivada primera lineal a tramos. Una vez que la derivada está determinada, la reconstrucción de la función se puede llevar a cabo mediante la integración de la misma. De esta forma, para un instante x situado entre las muestras n y $n + 1$ de la secuencia interpolada, es decir $x_i[n] < x < x_i[n + 1]$, la reconstrucción de la función se calcula como

$$f(x) = y_i[n] + y_i^1[n]\Delta x + \frac{y_i^1[n + 1] - y_i^1[n]}{T}(\Delta x)^2 \quad (3.7)$$

donde $\Delta x = x - x_i[n]$ es la diferencia entre x y el instante de tiempo asociado a la muestra n de la secuencia interpolada.

El proceso de reconstrucción es totalmente local: es decir, el valor de las secuencias interpoladas en los puntos de ruptura insertados sólo depende del valor de las dos muestras entre las que está intercalado, y la reconstrucción de la función mediante (3.7) sólo depende del valor de las secuencias en los dos extremos del intervalo. Este hecho supone, por un lado, un bajo coste computacional para realizar la reconstrucción de la función, y por otro lado, facilita la aplicación directa de este método de reconstrucción a problemas de muestreo irregular, en los que la secuencia de instantes de muestreo, $x[n]$, sea arbitraria.

La solución presentada es equivalente a la propuesta por Djokovic y Vaidyanathan [Djokovic97, Vaidyanathan2001], dentro del marco teórico de los espacios multiresolución, utilizando un banco de filtros de reconstrucción perfecta, como se ha visto con cierto detalle en la Sección 2.2.2 y en la Sección 2.3.2.2, donde se particulariza para el espacio de funciones splines. Esta formulación, tal y como se presenta, elegantemente encuadrada dentro de un marco de teoremas de muestreo en subespacios de multiresolución, no es, sin embargo, la más adecuada para llevar a cabo la extensión regularizada. Por esta razón, dicha extensión se realiza a partir de la formulación propuesta en esta sección, más adecuada para este problema particular.

3.1.1.1 Posición de los puntos de ruptura insertados

En el método propuesto en el apartado anterior, los puntos de ruptura se insertan en el punto medio entre dos muestras. La elección de este punto parece razonable desde un punto de vista intuitivo y, además, simplifica cuestiones notacionales. Asimismo permite, cuando se trabaja en situaciones de muestreo regular, que las secuencias interpoladas tengan asociados unos instantes de muestreo equiespaciados, con lo que se sigue trabajando en un entorno de muestreo regular. Pero además de estas razones, la elección de este punto se justifica también desde el punto de vista de la suavidad de la aproximación. Para comprobar esta afirmación, supongamos que disponemos de muestras de una función $f(x)$ y su derivada únicamente en dos puntos, x_0 y x_1 separados por el tiempo de muestreo T . Los valores en estos puntos, para simplificar la notación, son y_0 y y_1 para la función y $y_0^{(1)}$ y $y_1^{(1)}$ para la derivada, tal y como se muestra en la Figura 3.1.

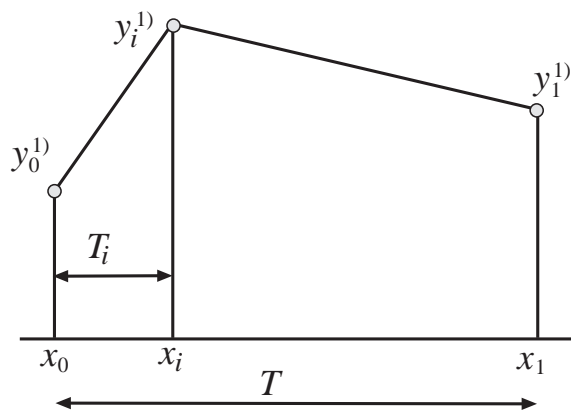


Figura 3.1: Posición de los puntos de ruptura insertados

A continuación se estudia como varía la suavidad de la solución obtenida en función de la elección del instante de ruptura, x_i , insertado, o lo que es lo mismo, en función de T_i .

El valor de la función en el punto x_1 es

$$y_1 = y_0 + \left(\frac{y_0^{(1)} + y_i^{(1)}}{2} \right) T_i + \left(\frac{y_i^{(1)} + y_1^{(1)}}{2} \right) (T - T_i), \quad (3.8)$$

a partir del cual se obtiene

$$y_i^{(1)} = \frac{2(y_1 - y_0) + T_i(y_1^{(1)} - y_0^{(1)} - T y_1^{(1)})}{T}. \quad (3.9)$$

Por otra parte, la derivada segunda en el intervalo $[0, T]$ viene dada por

$$f^{(2)}(x) = \begin{cases} \frac{y_i^{(1)} - y_0^{(1)}}{T_i}, & \text{para } 0 \leq x \leq T_i \\ \frac{y_1^{(1)} - y_i^{(1)}}{T - T_i}, & \text{para } T_i \leq x \leq T \end{cases}. \quad (3.10)$$

La suavidad de la aproximación se puede evaluar mediante la integral de la derivada segunda al cuadrado en el intervalo de interés, que en este caso toma la forma

$$S = \int_0^T (f^{(2)}(x))^2 dx = \left(\frac{y_i^{(1)} - y_0^{(1)}}{T_i} \right)^2 T_i + \left(\frac{y_1^{(1)} - y_i^{(1)}}{T - T_i} \right)^2 (T - T_i). \quad (3.11)$$

La derivada de (3.11) con respecto a T_i es

$$\frac{dS}{dT_i} = \frac{[T(y_0^{(1)} + y_1^{(1)}) + 2(y_0 - y_1)]^2 (2T_i - T)}{TT_i^2(T_i - T)^2}, \quad (3.12)$$

expresión que se anula para el valor $T_i = T/2$, con lo que éste es el valor para el que la solución presenta un valor mínimo de S o, lo que es lo mismo, para el que se obtiene la solución más suave.

3.1.1.2 Extensión a D derivadas

La extensión de este modelo para el caso en el que se dispone de D derivadas se basa en la misma idea desarrollada para el caso $D = 1$. En este caso es necesario introducir D nuevos instantes de ruptura entre cada dos muestras consecutivas de la función, con lo que las secuencias $y_i^{(d)}[n]$ están ahora interpoladas por un factor $D + 1$. La solución pertenecerá al espacio de los splines de grado $D + 1$; por ejemplo, para dos derivadas la solución pertenece al espacio de los splines cúbicos.

El proceso de reconstrucción es siempre local. El valor de las secuencias interpoladas en los puntos de ruptura insertados depende, únicamente, de los valores de la función y derivadas en los instantes de muestreo entre los que se inserta. La posterior reconstrucción en un determinado punto, a partir de las secuencias interpoladas, depende únicamente de los valores de dichas secuencias en los extremos del intervalo.

Las expresiones particulares para la obtención de las secuencias interpoladas $y_i^{(d)}[n]$ y para la reconstrucción local a partir de las mismas para los casos $D = 2$ y $D = 3$ se encuentran en el Apéndice C.

3.1.2 Representación mediante un banco de filtros

El modelo MLI que se ha descrito anteriormente admite una representación sencilla utilizando un banco de filtros digitales MIMO (“Multiple Input Multiple Output”), seguido de un banco de filtros analógicos de reconstrucción. La Figura 3.2 muestra la representación mediante este tipo de banco de filtros del modelo MLI de grado 2 ($D = 1$).

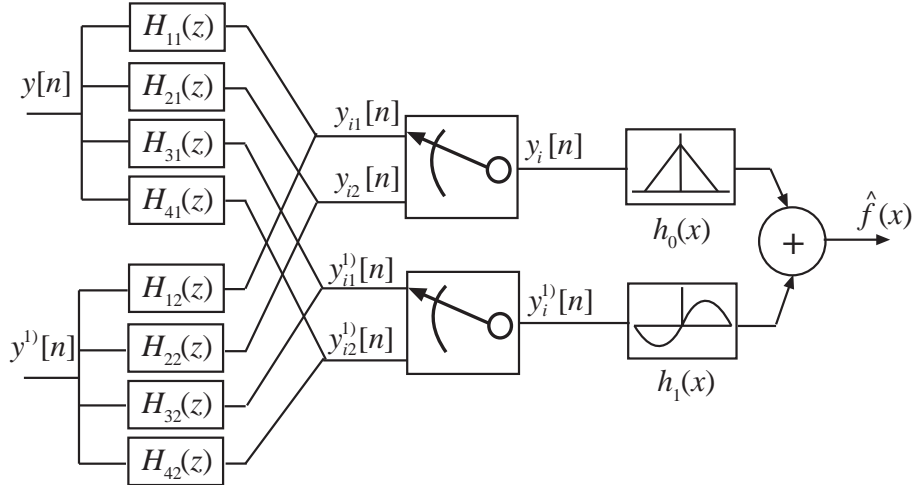


Figura 3.2: Representación mediante un banco de filtros MIMO del modelo MLI de grado 2 ($D = 1$)

El banco de filtros digitales MIMO realiza la interpolación de las secuencias de muestras mediante la utilización de filtros polifase. Se ha empleado la notación $y_{ij}^d[n]$ para definir la componente polifase j -ésima de la secuencia interpolada $y_i^d[n]$. A partir de las distintas componentes polifase, se obtienen las secuencias interpoladas $y_i[n]$ y $y_i^1[n]$, cada una de las cuales pasa a través de un filtro reconstructor analógico y finalmente se obtiene la señal reconstruida, $\hat{f}(x)$, mediante la suma de las salidas de estos filtros analógicos.

La respuesta de los filtros MIMO se puede obtener de forma sencilla a partir de las expresiones para el cálculo de las secuencias interpoladas (3.3)- (3.5)

$$\begin{aligned}
 H_{11}(z) = H_{32}(z) = 1, & \quad H_{12}(z) = H_{31}(z) = 0, \\
 H_{21}(z) = \frac{z+1}{2}, & \quad H_{41}(z) = \frac{2z-2}{T}, \\
 H_{22}(z) = \frac{1-z}{8}T, & \quad H_{42}(z) = -\frac{z+1}{2}.
 \end{aligned} \tag{3.13}$$

En cuanto a los filtros reconstructores, su respuesta impulsiva tiene la siguiente forma

$$h_0(x) = 1 - \frac{|x|}{T_i}, \quad \text{para } |x| \leq T_i. \tag{3.14}$$

y

$$h_1(x) = \frac{x}{2} \left(1 - \frac{|x|}{T_i} \right), \quad \text{para } |x| \leq T_i. \quad (3.15)$$

donde T_i es el período de muestreo asociado a las secuencias interpoladas $y_i^{(d)}$, que en este caso vale $T_i = T/2$. Las respuestas impulsionales de los filtros se muestran en la Figura 3.3.

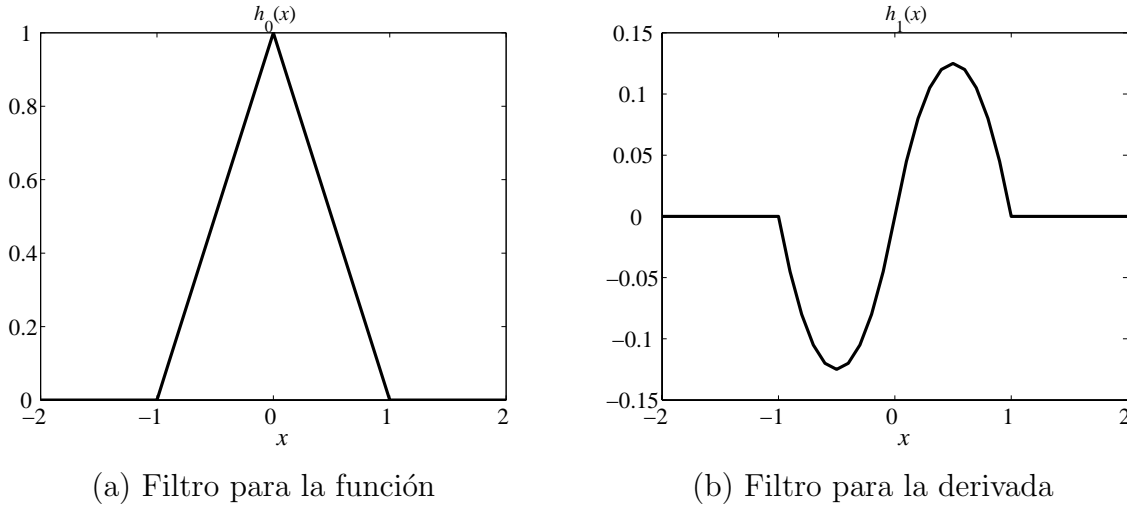


Figura 3.3: Respuestas al impulso de los filtros reconstructores

El filtro $h_0(x)$ tiene la respuesta triangular característica de un interpolador lineal a tramos. Así pues, su contribución es la interpolación lineal a tramos de la secuencia $y_i[n]$. La respuesta del filtro $h_1(x)$ se ha obtenido teniendo en cuenta la diferencia entre una interpolación lineal, cuya derivada es constante entre cada dos muestras, y la solución proporcionada por el modelo MLI, en cuyo caso la derivada es lineal entre cada dos muestras. Para un “impulso” en la derivada, se plantea la situación que muestra la Figura 3.4.

De las condiciones de continuidad en la función se puede deducir que la integral entre 0 y x de la derivada del modelo MLI, que es lineal, y la de la derivada de la interpolación lineal, que es constante, han de coincidir, lo que lleva a determinar el valor de la derivada: en este caso, $1/2$. De esta forma, la contribución del filtro a la salida sería la diferencia entre la integral de la derivada del modelo lineal a tramos (línea B) y la integral de la contribución del modelo MLI (línea A). Esta diferencia nos permite calcular la parte positiva del filtro reconstructor, que vendría dada por la siguiente expresión

$$h_1^+(x) = \int_0^x \left[\left(1 - \frac{x}{T_i} \right) - \frac{1}{2} \right] dx = \frac{x}{2} - \frac{x^2}{2T_i}. \quad (3.16)$$

Teniendo en cuenta que el filtro ha de ser simétrico, finalmente se llega a la ecuación (3.15). Las correspondientes respuestas en frecuencia de estos filtros son

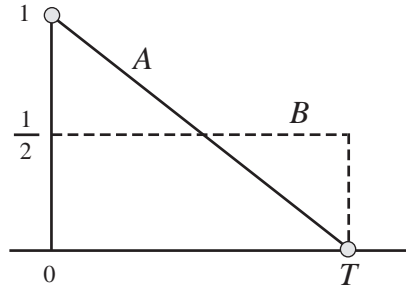


Figura 3.4: Derivadas del modelo para el cálculo de la respuesta del filtro reconstructor $h_1(x)$

$$H_0(\Omega) = T_i \operatorname{sinc}^2\left(\frac{\Omega T_i}{2\pi}\right), \quad (3.17)$$

y

$$H_1(\Omega) = -\frac{2j}{\Omega^3 T_i} \left(2 \sin^2\left(\frac{\Omega T_i}{2}\right) - \Omega T_i \cos\left(\frac{\Omega T_i}{2}\right) \sin\left(\frac{\Omega T_i}{2}\right) \right). \quad (3.18)$$

La Figura 3.5 muestra estas respuestas en frecuencia para un valor de $T_i = 1$.

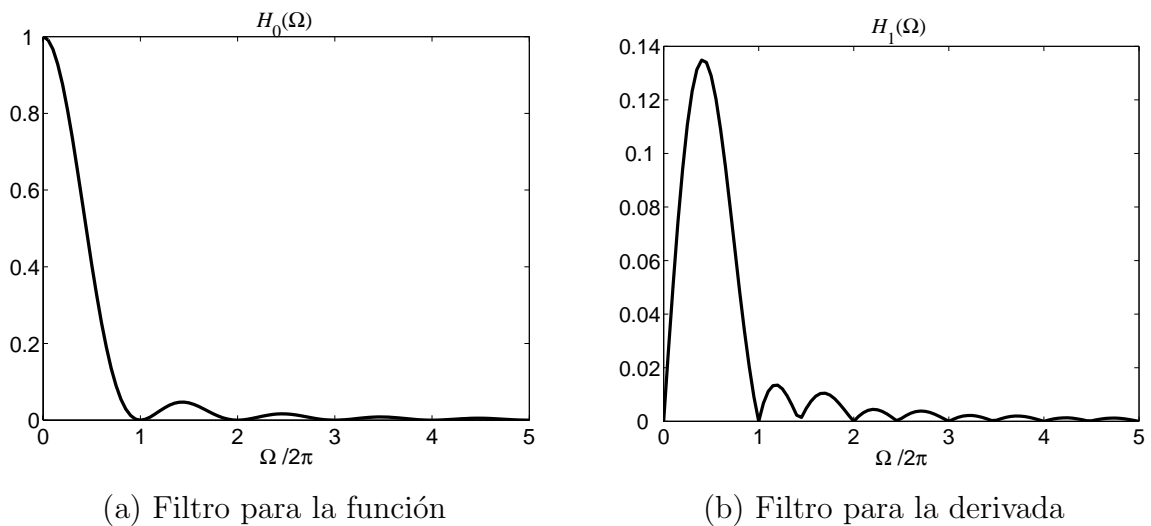


Figura 3.5: Respuesta en frecuencia de los filtros reestructores

Se puede ver que todos los filtros utilizados son filtros FIR; esto hace posible obtener una reconstrucción local de la función. Las técnicas de filtrado utilizadas con splines convencionales se basan en filtros IIR anticausales [Unser93a]. Esta es una de las ventajas que proporciona la utilización de la información de las derivadas.

Los diagramas de bloques y las expresiones de los filtros polifase y de los filtros analógicos de reconstrucción para los casos particulares de $D = 2$ y $D = 3$ se muestran en el Apéndice C.

3.1.3 Sensibilidad al ruido

Uno de los principales inconvenientes que presenta el método de interpolación propuesto es su sensibilidad al ruido. Esta es una característica que comparte con la mayoría de los métodos de interpolación. Por ello, vamos a realizar un análisis detallado del comportamiento del modelo en condiciones de ruido.

3.1.3.1 Modelo de grado 2

Consideramos nuevamente el caso más sencillo, que corresponde al modelo de grado 2. Se asume que el ruido en las muestras de la función y de la derivada se puede modelar como ruido blanco gaussiano de media nula y varianzas σ_0^2 y σ_1^2 , respectivamente. Teniendo en cuenta las ecuaciones (3.3)-(3.7) se comprueba que, para $0 \leq x \leq T/2$, la varianza de ruido en la función reconstruida $\sigma_{0r}^2(x)$ y en su derivada $\sigma_{1r}^2(x)$ tienen la forma

$$\sigma_{0r}^2(x) = \sigma_0^2 \left[\left(1 - \frac{2x^2}{T^2}\right)^2 + \frac{4x^4}{T^4} \right] + \sigma_1^2 \left[\left(1 - \frac{3x^2}{2T}\right)^2 + \frac{x^4}{4T^2} \right], \quad (3.19)$$

$$\sigma_{1r}^2(x) = \sigma_1^2 \left[\left(1 - \frac{3x}{T}\right)^2 + \frac{x^2}{T^2} \right] + \frac{32x^2}{T^4} \sigma_0^2. \quad (3.20)$$

Es interesante apuntar que la potencia de ruido en la derivada de la función reconstruida varía como $32x^2\sigma_0^2/T^4$. Por ejemplo, en $x = T/2$, es decir, en los instantes en los que se interpolan las secuencias de muestras, la varianza de ruido de la derivada varía como $8\sigma_0^2/T^2$.

Este hecho muestra claramente la sensibilidad frente al ruido de este método de reconstrucción, que se agrava a medida que la señal se muestrea más rápido (el ruido en la derivada varía como $1/T^2$). Esta sensibilidad es compartida por la mayoría de métodos comúnmente empleados para resolver problemas de interpolación, y es el principal motivo para buscar un método de regularización que pueda reducir en lo posible esta sensibilidad.

3.2 El Modelo Local Regularizado (MLR) en espacios unidimensionales

Cuando las muestras de la función y sus derivadas son ruidosas, en lugar de exigir una interpolación exacta del conjunto de muestras, una solución más conveniente es relajar esa condición e imponer ciertas restricciones de suavidad en la solución. Una forma de

forzar la suavidad de la solución consiste en minimizar un funcional de regularización como el siguiente

$$J(y_r^d) = \sum_{d=0}^D \left(\lambda_d \sum_{n=0}^{N-1} (y_r^d[n] - y^d[n])^2 \right) + \lambda_r J_r(y_r^d[n]), \quad (3.21)$$

donde $y_r^d[n]$ son las secuencias de muestras regularizadas, que ahora ya no cumplen las condiciones de interpolación, es decir, ahora en general

$$y_r^d[n] \neq f^d(x[n]). \quad (3.22)$$

$J_r(y_r^d[n])$ es un término que mide la suavidad de la solución, y los parámetros λ_d y λ_r se encargan de ponderar la contribución al funcional del error de la derivada de orden d y del término de suavidad $J_r(y_r^d[n])$ respectivamente.

De esta forma, el primer término en (3.21) mide el error de la solución con respecto a las muestras, y el segundo es un término de regularización que fuerza la suavidad de la solución. Como medida de suavidad se utiliza la integral de la derivada segunda al cuadrado. Esta medida corresponde al estabilizador (2.93) para un espacio de entrada unidimensional. El hecho de forzar la suavidad en la solución se traduce en una disminución de la sensibilidad frente al ruido. A continuación se detalla esta técnica de regularización para el modelo de grado 2 ($D = 1$).

3.2.1 Modelo regularizado de grado 2

En este caso particular, el término de regularización tiene la forma siguiente

$$J_r = \sum_{n=0}^{2N-3} (y_i^1[n] - y_i^1[n+1])^2. \quad (3.23)$$

donde la secuencia $y_i^1[n]$ es nuevamente la secuencia interpolada de la derivada. De forma similar a (3.3) y (3.4), esta secuencia puede escribirse, en función de las secuencias regularizadas, como

$$y_i^1[2n] = y_r^1[n], \quad n = 0, \dots, N-1, \quad (3.24)$$

y

$$y_i^1[2n+1] = \frac{2(y_r[n+1] - y_r[n])}{T} - \frac{y_r^1[n] + y_r^1[n+1]}{2}, \quad n = 0, 1, \dots, N-2. \quad (3.25)$$

Por lo tanto, sustituyendo (3.24) y (3.25) en (3.23), se observa que el funcional (3.21) sólo depende de las secuencias regularizadas $y_r^d[n]$. Para obtener el mínimo de este funcional se evalúan sus derivadas con respecto a $y_r^d[n]$

$$\frac{\partial J}{\partial y_r[n]} = 2\lambda_0(y_r[n] - y[n]) + \lambda_r \frac{\partial J_r}{\partial y_r[n]}, \quad (3.26)$$

$$\frac{\partial J}{\partial y_r^1[n]} = 2\lambda_1(y_r^1[n] - y^1[n]) + \lambda_r \frac{\partial J_r}{\partial y_r^1[n]}. \quad (3.27)$$

Teniendo en cuenta (3.23), (3.24) y (3.25) se comprueba que, para $n = 0$, la derivada del término de regularización con respecto a $y_r[n]$ es

$$\begin{cases} \frac{\partial J_r}{\partial y_r[n]} = \frac{16}{T^2}(y_r[n] - y_r[n+1]) + \frac{8}{T}(y_r^1[n] + y_r^1[n+1]), \\ \frac{\partial J_r}{\partial y_r^1[n]} = \frac{8}{T}(y_r[n] - y_r[n+1]) + 5y_r^1[n] + 3y_r^1[n+1], \end{cases} \quad (3.28)$$

mientras que para $0 < n < N - 1$

$$\begin{cases} \frac{\partial J_r}{\partial y_r[n]} = \frac{16}{T^2}(2y_r[n] - y_r[n-1] - y_r[n+1]) + \frac{8}{T}(y_r^1[n+1] - y_r^1[n-1]), \\ \frac{\partial J_r}{\partial y_r^1[n]} = \frac{8}{T}(2y_r[n-1] - y_r[n+1]) + 10y_r^1[n] + 3y_r^1[n+1] + 3y_r^1[n-1], \end{cases} \quad (3.29)$$

y, finalmente, para $n = N - 1$

$$\begin{cases} \frac{\partial J_r}{\partial y_r[n]} = \frac{16}{T^2}(y_r[n] - y_r[n-1]) - \frac{8}{T}(y_r^1[n] + y_r^1[n-1]), \\ \frac{\partial J_r}{\partial y_r^1[n]} = \frac{8}{T}(y_r[n-1] - y_r[n]) + 5y_r^1[n] + 3y_r^1[n-1]. \end{cases} \quad (3.30)$$

Igualando las derivadas (3.26) y (3.27) a cero se obtiene un sistema lineal de ecuaciones que se puede expresar en forma matricial como

$$\mathbf{b} = \mathbf{A}\mathbf{x} \quad (3.31)$$

donde

$$\mathbf{b} = [\lambda_0(x[0], \dots, x[N-1]), \lambda_1(x^1[0], \dots, x^1[N-1])]^T, \quad (3.32)$$

$$\mathbf{x} = [x_r[0], \dots, x_r[N-1], x_r^1[0], \dots, x_r^1[N-1]]^T, \quad (3.33)$$

y \mathbf{A} es una matriz $(2N \times 2N)$ generalmente con $\text{rango}(\mathbf{A}) = 2N$. Por lo tanto, la solución del sistema se puede obtener mediante

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}. \quad (3.34)$$

La ecuación (3.34) proporciona de forma directa la solución regularizada; sin embargo, cuando el número de muestras N es elevado, la inversión de la matriz puede

ser computacionalmente costosa. En este caso, el problema se puede resolver de una forma más eficiente empleando un algoritmo iterativo basado en gradiente. Es importante tener en cuenta que el comportamiento local de las ecuaciones (3.28)-(3.30) hace que la carga computacional de la alternativa basada en gradiente sólo crezca de forma lineal con N , lo que hace que sea una alternativa más económica cuando el número de muestras es elevado.

Una vez obtenidas las secuencias regularizadas $y_r^d[n]$, la reconstrucción de la función se lleva a cabo proyectando estas secuencias sobre el espacio de los splines cuadráticos, tal y como se ha descrito en la Sección 3.1.

La extensión al caso de D derivadas es inmediata; una vez calculado el funcional de regularización para un caso particular, se sigue un procedimiento análogo al mostrado para $D = 1$. En el Apéndice D se encuentran las expresiones para la regularización de las muestras para los casos particulares $D = 2$ y $D = 3$.

3.2.2 Selección de los parámetros de regularización

Un aspecto importante de este método es la selección de los parámetros de ponderación λ_d ($d = 0, 1, \dots, D$) y el parámetro de regularización λ_r . Para esta selección existen varias alternativas. Si no se tiene ninguna información sobre la potencia de ruido en las muestras de la función (σ_0^2) y de las derivadas (σ_d^2), la alternativa que parece más lógica sería fijar los valores de λ_d y entonces seleccionar el valor del parámetro λ_r empleando técnicas de validación cruzada [Perry93, Reeves90]. Para los valores de λ_d se puede seleccionar $\lambda_0 = 1$ como referencia, y el resto bien a uno o bien a un valor que tenga en cuenta aspectos como la energía de las distintas derivadas con respecto a la energía de la función, o la importancia relativa que se da a cada derivada en la aplicación específica que se trate de abordar.

Cuando es posible tener una estima de las potencias de ruido en las muestras de la función y de las derivadas, una alternativa simple y que ofrece unos resultados razonables consiste en fijar $\lambda_0 = 1$ y seleccionar

$$\lambda_d = \frac{\sigma_0^2}{\sigma_d^2}, \quad (3.35)$$

y

$$\lambda_r = \frac{P_0 \sigma_0^2}{P_1^2 T^2} \quad (3.36)$$

donde P_0 y P_1 son, respectivamente, la potencia de la función y de la derivada primera, que pueden ser estimadas de forma sencilla. En este caso los valores de λ_d tratan de ponderar la fidelidad de las muestras de la función y de la derivada para que la contribución de los errores en las diferentes derivadas en la función de coste $J(y_r^d[n])$ sea del mismo orden. En cuanto a la expresión de λ_r , su valor se ha obtenido de forma heurística a partir de los resultados que se han podido observar en diferentes simulaciones. Hay que resaltar que la expresión (3.36) es genérica, para cualquier valor de D ; aunque sólo dependa de parámetros relacionados con la función y la primera

derivada, se ha comprobado que esta elección proporciona resultados satisfactorios, y que complicar la expresión introduciendo más parámetros no conduce a una mejora significativa en los resultados.

3.3 Extensión a espacios de entrada bidimensionales

A continuación se presenta la extensión de los modelos MLI y MLR, desarrollados con anterioridad para espacios de entrada unidimensionales, a espacios de entrada bidimensionales.

3.3.1 Modelos de interpolación en espacios bidimensionales

Existen distintas alternativas para la interpolación en espacios de entrada bidimensionales en función de la información de que se disponga. En esta sección se presentan dos alternativas:

- Una alternativa de interpolación semilocal, cuando se dispone únicamente de la información de las derivadas direccionales (derivadas con respecto a x_1 y con respecto a x_2). Se ha denominado al modelo resultante *Modelo Semilocal de Interpolación* (MSI).
- Una alternativa de interpolación local, cuando se dispone de información de las derivadas direccionales y de las derivadas cruzadas. El modelo resultante se ha denominado *Modelo Local de Interpolación* (MLI).

3.3.1.1 Modelo Semilocal de Interpolación (MSI)

El planteamiento del problema es el siguiente: dado un conjunto de muestras de una función, $f(x_1, x_2)$, y de sus D primeras derivadas, con respecto a cada una de las variables del espacio de entrada, de la forma

$$y[n_1, n_2] = f(x_1, x_2) \Big|_{x_1=x_1[n_1]; x_2=x_2[n_2]}, \quad (3.37)$$

y

$$y_{x_k}^{(d)}[n_1, n_2] = \frac{\partial^d f(x_1, x_2)}{\partial x_k^d} \Big|_{x_1=x_1[n_1]; x_2=x_2[n_2]}, \quad (3.38)$$

con $k = 1, 2$ y $d = 1, \dots, D$, y donde

$$x[n_k] = n_k T_k, \quad k = 1, 2; \quad n_k = 0, \dots, N_k - 1, \quad (3.39)$$

siendo T_k el período de muestreo en la dirección k del espacio de entrada, encontrar una función, $\hat{f}(x_1, x_2)$, que interpole las muestras (3.37) y (3.38) de la función $f(x_1, x_2)$ y de sus derivadas.

Así pues, se ha restringido el problema al caso particular en el que las muestras están distribuidas de forma uniforme en el espacio de entrada, formando una rejilla de $N_1 \times N_2$ instantes de muestreo, en los que se dispone de muestras de la función y de las derivadas con respecto a cada una de las direcciones del espacio. De nuevo, como en el caso unidimensional, por simplicidad, se considera el caso particular $D = 1$. Posteriormente, se presenta la extensión del modelo a un número mayor de derivadas.

Con estas premisas, para llevar a cabo la interpolación del conjunto de muestras de la función y las derivadas primeras respecto a x_1 y x_2 , la alternativa de reconstrucción propuesta se basa en la utilización del modelo MLI unidimensional (MLI1D). Con un conjunto de instantes de muestreo como el mostrado en la Figura 3.6, si se considera cada rectángulo de manera independiente, la reconstrucción de la función en el perímetro del mismo se puede realizar con un modelo MLI1D. En la dirección de x_1 (líneas horizontales en la figura), se reconstruye utilizando los valores de la función y de la derivada respecto a x_1 . En la dirección de x_2 (líneas verticales en la figura), utilizando los valores de la función y de la derivada respecto a x_2 . El problema se plantea en la reconstrucción en el interior del rectángulo, donde se busca una solución continua, tanto en la función como en las derivadas con respecto a x_1 y x_2 . La alternativa que se propone es la siguiente: reconstruir la función utilizando un modelo MLI1D en una de las direcciones del espacio de entrada, x_k . Para ello, es preciso conocer los valores de la función y de la derivada con respecto a x_k en los instantes de muestreo de esa variable, $x_k[n_k]$, para cualquier valor de la otra variable de entrada. La Figura 3.6 ilustra esta idea cuando se selecciona x_1 como dirección de reconstrucción.

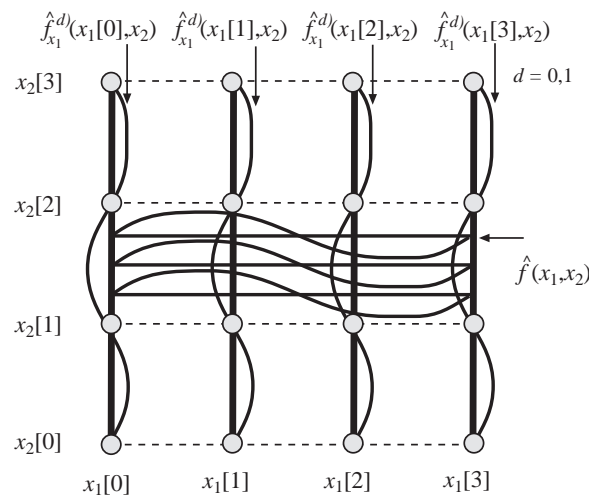


Figura 3.6: Esquema de reconstrucción en la dirección x_1

En este caso, si se dispone de los valores de la función y la derivada con respecto

a x_1 en los instantes $x_1[n_1]$ para cualquier x_2 (líneas verticales en la figura), que se pueden denotar como

$$\hat{f}(x_1[n_1], x_2), \quad (3.40)$$

y

$$\hat{f}_{x_1}^1(x_1[n_1], x_2), \quad (3.41)$$

la reconstrucción de $\hat{f}(x_1, x_2)$, para cualquier valor de x_2 , se reduce a un problema de reconstrucción unidimensional a lo largo de la dirección x_1 (línea horizontal en la figura). Mediante esta alternativa, el problema se reduce a encontrar un modelo para (3.40) y (3.41) que garantice las condiciones de interpolación y de continuidad de la solución.

Para reconstruir $\hat{f}(x_1[n_1], x_2)$, la solución más sencilla, siguiendo la filosofía del modelo unidimensional, es utilizar el modelo MLI1D en la dirección x_2 (líneas verticales). Es decir, se reconstruye la función a partir de las muestras de la función y de la derivada respecto a x_2 en los instantes $x_1[n_1]$, lo que asegura la interpolación de ambas. Definiendo la secuencia de instantes de muestreo interpolada en la dirección x_2

$$x_{2i}[n] = \frac{nT_2}{2}, \quad n = \{0, 1, \dots, 2N_2 - 2\}, \quad (3.42)$$

y aplicando el modelo MLI1D, es decir, utilizando las expresiones (3.3)-(3.5) para las muestras correspondientes de la función y de la derivada con respecto a x_2 , se obtiene

$$\hat{f}(x_1[n_1], x_{2i}[2n_2]) = y[n_1, n_2], \quad n_2 = \{0, \dots, N_2 - 1\}, \quad (3.43)$$

$$\hat{f}_{x_2}^1(x_1[n_1], x_{2i}[2n_2]) = y_{x_2}^1[n_1, n_2], \quad n_2 = \{0, \dots, N_2 - 1\}, \quad (3.44)$$

y

$$\begin{aligned} \hat{f}(x_1[n_1], x_{2i}[2n_2 + 1]) &= \frac{y[n_1, n_2] + y[n_1, n_2 + 1]}{2} \\ &+ T_2 \frac{(y_{x_2}^1[n_1, n_2] - y_{x_2}^1[n_1, n_2 + 1])}{8}, \end{aligned} \quad (3.45)$$

$$\begin{aligned} \hat{f}_{x_2}^1(x_1[n_1], x_{2i}[2n_2 + 1]) &= \frac{2(y[n_1, n_2 + 1] - y[n_1, n_2])}{T_2} \\ &- \frac{y_{x_2}^1[n_1, n_2] + y_{x_2}^1[n_1, n_2 + 1]}{2}, \end{aligned} \quad (3.46)$$

con $n_2 = \{0, 1, \dots, N_2 - 2\}$ en ambos casos.

La función $\hat{f}(x_1[n_1], x_2)$ se reconstruye, para un instante $x_{2i}[n] \leq x_2 \leq x_{2i}[n + 1]$, utilizando la expresión de reconstrucción del modelo unidimensional

$$\begin{aligned} \hat{f}(x_1[n_1], x_2) &= \hat{f}(x_1[n_1], x_{2i}[n]) + \hat{f}_{x_2}^1(x_1[n_1], x_{2i}[n])\Delta x_2 \\ &+ \frac{\hat{f}_{x_2}^1(x_1[n_1], x_{2i}[n+1]) - \hat{f}_{x_2}^1(x_1[n_1], x_{2i}[n])}{T_2}(\Delta x_2)^2 \end{aligned} \quad (3.47)$$

donde $\Delta x_2 = x_2 - x_{2i}[n]$ es la diferencia entre x_2 y el instante de tiempo asociado a la muestra n de la secuencia interpolada.

Para reconstruir $\hat{f}_{x_1}^1(x_1[n_1], x_2)$, no es posible utilizar el modelo unidimensional, ya que no se dispone de su derivada en la dirección x_2 , por lo que es necesario buscar otras alternativas. La más sencilla es plantear un modelo en el que esta derivada sea lineal a tramos entre cada dos muestras en la dirección de x_2 . Con este modelo, se obtiene una solución que interpola el conjunto de muestras de función y derivadas, que es continuo dentro de cada rectángulo formado por la rejilla de muestras, pero cuya derivada con respecto de x_2 es discontinua en la dirección de x_2 en los instantes de muestreo $x_2[n_2]$.

Una segunda alternativa consiste en emplear un spline cuadrático. En este caso, se obtiene una solución que interpola las muestras de la función y las derivadas y que es continua. La continuidad, tanto de la función como de las derivadas, dentro de cada uno de los “rectángulos” de la rejilla de instantes de muestreo, se demuestra en el Apéndice E. La demostración de la continuidad en todo el espacio de entrada, al perderse la localidad, es más compleja y no se presenta de forma analítica.

Una vez definidos los modelos para $\hat{f}(x_1[n_1], x_2)$ y $\hat{f}_{x_1}^1(x_1[n_1], x_2)$, la reconstrucción de $\hat{f}(x_1, x_2)$ se reduce a obtener el modelo MLI1D para cada valor de x_2 . Definiendo la secuencia de instantes de muestreo interpolada

$$x_{1i}[n] = \frac{nT_1}{2}, \quad n = \{0, 1, \dots, 2N - 2\}, \quad (3.48)$$

el modelo MLI1D permite obtener los valores de función y derivada con respecto a x_1 particularizados en esos instantes

$$\hat{f}(x_{1i}[2n_1], x_2) = \hat{f}(x_1[n_1], x_2), \quad n_1 = \{0, \dots, N_1 - 1\}, \quad (3.49)$$

$$\hat{f}_{x_1}^1(x_{1i}[2n_1], x_2) = \hat{f}_{x_1}^1(x_1[n_1], x_2), \quad n_1 = \{0, \dots, N_1 - 1\}, \quad (3.50)$$

y

$$\begin{aligned} \hat{f}(x_{1i}[2n_1 + 1], x_2) &= \frac{\hat{f}(x_1[n_1], x_2) + \hat{f}(x_1[n_1 + 1], x_2)}{2} \\ &+ T_1 \frac{(\hat{f}_{x_1}^1(x_1[n_1], x_2) - \hat{f}_{x_1}^1(x_1[n_1 + 1], x_2))}{8}, \end{aligned} \quad (3.51)$$

$$\begin{aligned} \hat{f}_{x_1}^1(x_{1i}[2n_1 + 1], x_2) &= \frac{2(\hat{f}(x_1[n_1 + 1], x_2) - \hat{f}(x_1[n_1], x_2))}{T_1} \\ &- \frac{\hat{f}_{x_1}^1(x_1[n_1], x_2) + \hat{f}_{x_1}^1(x_1[n_1 + 1], x_2)}{2}, \end{aligned} \quad (3.52)$$

con $n_1 = \{0, 1, \dots, N_1 - 2\}$, en ambos casos.

De esta forma, para un instante $x_{1i}[n] \leq x_1 \leq x_{1i}[n+1]$, la reconstrucción de la función se calcula empleando la expresión

$$\begin{aligned} \hat{f}(x_1, x_2) = & \hat{f}(x_{1i}[n], x_2) + \hat{f}_{x_1}^{(1)}(x_{1i}[n], x_2) \Delta x_1 \\ & + \frac{\hat{f}_{x_1}^{(1)}(x_{1i}[n+1], x_2) - \hat{f}_{x_1}^{(1)}(x_{1i}[n], x_2)}{T_1} (\Delta x_1)^2 \end{aligned} \quad (3.53)$$

donde $\Delta x_1 = x_1 - x_{1i}[n]$ es la diferencia entre x_1 y el instante de tiempo asociado a la muestra n de la secuencia $x_{1i}[n]$.

El procedimiento de reconstrucción se puede realizar en cualquiera de las dos direcciones del espacio de entrada; aunque las soluciones obtenidas son en general diferentes, esta es una alternativa de reconstrucción computacionalmente sencilla que, como se mostrará con posterioridad, presenta unos resultados satisfactorios y permite una sencilla extensión regularizada.

Comentar por último que el método de reconstrucción propuesto es semilocal. En el ejemplo de reconstrucción en la dirección x_1 , el método no es local en la dirección de x_2 , ya que para obtener $\hat{f}_{x_1}^{(1)}(x_1[n_1], x_2)$ es preciso estimar un spline cuadrático para cada valor de n_1 . Una vez estimadas estas derivadas, el resto del procedimiento de reconstrucción es un procedimiento basado en el MLI1D, que es local.

3.3.1.2 Modelo Local de Interpolación (MLI) con las derivadas cruzadas

El planteamiento del problema ahora es el siguiente: dado un conjunto de muestras de una función, $f(x_1, x_2)$, y de sus derivadas, direccionales y cruzadas, de hasta orden D en cada una de las direcciones del espacio, de la forma

$$y_{x_1, x_2}^{d_1, d_2}[n_1, n_2] = \left. \frac{\partial^{(d_1+d_2)} f(x_1, x_2)}{\partial x_1^{d_1} \partial x_2^{d_2}} \right|_{x_1=x_1[n_1]; x_2=x_2[n_2]}, \quad (3.54)$$

con $d_1 = 0, \dots, D$, y $d_2 = 0, \dots, D$, y donde

$$x[n_k] = n_k T_k, \quad k = 1, 2; \quad n_k = 0, \dots, N_k - 1, \quad (3.55)$$

siendo T_k el período de muestreo en la dirección k del espacio de entrada, encontrar una función, $\hat{f}(x_1, x_2)$, que interpole las muestras (3.54) de la función $f(x_1, x_2)$ y de sus derivadas.

De nuevo se ha restringido el problema al caso particular en el que las muestras están distribuidas de forma uniforme en el espacio de entrada, formando una rejilla de $N_1 \times N_2$ instantes de muestreo.

Por simplicidad, se considera el caso particular $D = 1$. En este caso, para simplificar la notación, se omiten los superíndices en (3.54): se dispone así de un conjunto de muestras de la función, $y[n_1, n_2]$, de sus derivadas primeras respecto a x_1 y x_2 , $y_{x_1}[n_1, n_2]$ e $y_{x_2}[n_1, n_2]$, y de la derivada cruzada respecto a ambas variables, $y_{x_1, x_2}[n_1, n_2]$.

El método de interpolación se basa nuevamente en el modelo MLI1D. Siguiendo la misma idea expresada en la Sección 3.3.1.1, se reconstruye a lo largo de una dirección del espacio de entrada utilizando el modelo MLI1D. Si, por ejemplo, se selecciona x_1 como dirección de reconstrucción, de nuevo es necesario conocer la función y la derivada con respecto a x_1 en los instantes $x_1[n_1]$, para cualquier x_2 (líneas verticales en la Figura 3.6), que se pueden denotar como

$$\hat{f}(x_1[n_1], x_2), \quad (3.56)$$

y

$$\hat{f}_{x_1}^1(x_1[n_1], x_2). \quad (3.57)$$

De nuevo, para la reconstrucción de $\hat{f}(x_1[n_1], x_2)$ se utiliza el modelo MLI1D en la dirección x_2 , utilizando las muestras de la función y de la derivada con respecto a x_2 , tal y como se hace en la Sección 3.3.1.1.

La diferencia con el caso anterior aparece en la reconstrucción de $\hat{f}_{x_1}^1(x_1[n_1], x_2)$: en este caso se puede utilizar el modelo MLI1D, ya que se dispone de las muestras de dicha función y de su derivada con respecto a x_2 , que es, en este caso, la derivada cruzada. Dada la secuencia interpolada x_{2i} (3.42), utilizando el modelo MLI1D se obtienen los siguientes valores

$$\hat{f}_{x_1}(x_1[n_1], x_{2i}[2n_2]) = y_{x_1}[n_1, n_2], \quad n_2 = \{0, \dots, N_2 - 1\}, \quad (3.58)$$

$$\hat{f}_{x_1, x_2}(x_1[n_1], x_{2i}[2n_2]) = y_{x_1, x_2}[n_1, n_2], \quad n_2 = \{0, \dots, N_2 - 1\}, \quad (3.59)$$

y

$$\begin{aligned} \hat{f}_{x_1}(x_1[n_1], x_{2i}[2n_2 + 1]) &= \frac{y_{x_1}[n_1, n_2] + y_{x_1}[n_1, n_2 + 1]}{2} \\ &+ T_2 \frac{y_{x_1, x_2}[n_1, n_2] - y_{x_1, x_2}[n_1, n_2 + 1]}{8}, \end{aligned} \quad (3.60)$$

$$\begin{aligned} \hat{f}_{x_1, x_2}(x_1[n_1], x_{2i}[2n_2 + 1]) &= \frac{2(y_{x_1}[n_1, n_2 + 1] - y_{x_1}[n_1, n_2])}{T_2} \\ &- \frac{y_{x_1, x_2}[n_1, n_2] + y_{x_1, x_2}[n_1, n_2 + 1]}{2}, \end{aligned} \quad (3.61)$$

con $n_2 = \{0, 1, \dots, N_2 - 2\}$ en ambos casos. La función $\hat{f}_{x_1}(x_1[n_1], x_2)$ se reconstruye, para un instante $x_{2i}[n] \leq x_2 \leq x_{2i}[n + 1]$, utilizando la expresión del modelo unidimensional

$$\begin{aligned} \hat{f}_{x_1}(x_1[n_1], x_2) &= \hat{f}_{x_1}(x_1[n_1], x_{2i}[n]) + \hat{f}_{x_1, x_2}(x_1[n_1], x_{2i}[n])\Delta x_2 \\ &+ \frac{\hat{f}_{x_1, x_2}(x_1[n_1], x_{2i}[n + 1]) - \hat{f}_{x_1, x_2}(x_1[n_1], x_{2i}[n])}{T_2}(\Delta x_2)^2 \end{aligned} \quad (3.62)$$

donde $\Delta x_2 = x_2 - x_{2_i}[n]$ es la diferencia entre x_2 y el instante de tiempo asociado a la muestra n de la secuencia interpolada.

Una vez que se han obtenido $\hat{f}(x_1[n_1], x_2)$ y $\hat{f}_{x_1}(x_1[n_1], x_2)$, la función se reconstruye de igual modo que en la Sección 3.3.1.1. Con este método se obtiene una solución local, que es continua en todo el espacio de entrada, tanto para la función como para las derivadas interpoladas. Además, la solución es única, independientemente de la dirección seleccionada para la reconstrucción. Esta solución pertenece al espacio de los splines cuadráticos en dos dimensiones mediante producto de tensores. Como en el caso unidimensional, la solución puede formularse mediante un spline convencional en 2D, obtenido a partir de las secuencias interpoladas.

3.3.2 Modelos regularizados en espacios bidimensionales

En esta sección se presentan varias alternativas para la regularización de los modelos de interpolación 2D propuestos. Para el modelo MSI se propone una alternativa simple, basada en el modelo MLR unidimensional (en adelante MLR1D), que se ha denominado *Modelo Semilocal Regularizado*(MSR). Para el modelo MLI, se proponen dos alternativas: una regularización estricta en un espacio de entrada 2D, utilizando un funcional de regularización como (2.93), que se ha denominado *Modelo Local Regularizado* (MLR), y una regularización simple basada en el modelo MLR1D, que se ha denominado *Modelo Local Regularizado simple* (MLRs).

3.3.2.1 Modelo Semilocal de Regularización (MSR)

Cuando sólo se dispone de las derivadas direccionales, plantear la minimización de un funcional regularizado convencional en espacios 2D, a partir del modelo MSI, resulta complejo debido a la no localidad, en una de las direcciones, de la solución propuesta. Sin embargo, es posible plantear una alternativa sencilla de regularización empleando el modelo MLR1D.

Dado un conjunto de instantes de muestreo, situados sobre el espacio de entrada como se muestra en la Figura 3.6, un procedimiento sencillo consiste en regularizar de forma independiente en las direcciones x_1 y x_2 , aplicando el modelo MLR1D sobre las muestras de la función y la derivada correspondiente a cada dirección. De esta forma, para las derivadas se tiene un único valor regularizado en cada instante de muestreo, mientras que para las muestras de la función se obtienen dos valores regularizados diferentes, uno por cada dirección del espacio. Una alternativa razonable es estimar el valor de las muestras regularizadas de la función como el promedio de estos dos valores. Aunque esta alternativa no puede considerarse estrictamente una técnica de regularización convencional, en el sentido de que no se utiliza un funcional de regularización en espacios 2D, tiene la ventaja de su sencillez de implementación, y, como se muestra con posterioridad, en la práctica proporciona resultados satisfactorios. No obstante, el hecho de regularizar las secuencias de muestras de forma independiente en cada una de las direcciones del espacio afecta a la reconstrucción de las derivadas, especialmente

las de mayor orden, como se mostrará con posterioridad.

En esta alternativa de regularización, los parámetros de regularización se seleccionan mediante las expresiones (3.35) y (3.36) aplicadas en cada una de las direcciones del espacio.

Como en el caso unidimensional, una vez obtenidas las secuencias de muestras regularizadas, el modelo MSR reconstruye la función aplicando el modelo de interpolación MSI.

3.3.2.2 Regularización con las derivadas cruzadas (MLR y MLRs)

Cuando además de las derivadas direccionales, $f_{x_1}^d(x_1, x_2)$ y $f_{x_2}^d(x_1, x_2)$, se conocen las derivadas cruzadas, a partir del modelo MLI, es posible plantear una regularización estricta en espacios de entrada bidimensionales, o utilizar una alternativa de regularización más sencilla, basada en el modelo MLR1D, similar a la descrita en el modelo MSR. A continuación se presentan ambas alternativas.

Regularización estricta en espacios 2D (MLR): En este caso se emplea un funcional de regularización de la forma

$$J_r = \int_R \left[\left(\frac{\partial^2 \hat{f}(x_1, x_2)}{\partial x_1^2} \right)^2 + 2 \left(\frac{\partial^2 \hat{f}(x_1, x_2)}{\partial x_1 \partial x_2} \right)^2 + \left(\frac{\partial^2 \hat{f}(x_1, x_2)}{\partial x_2^2} \right)^2 \right] dx_1 dx_2. \quad (3.63)$$

Al igual que en el caso unidimensional, el funcional completo de regularización, J , consiste en la suma ponderada de los errores cuadráticos de las muestras de la función y sus derivadas, más el término de regularización J_r . Para el caso $D = 1$, el funcional completo de regularización toma la expresión

$$\begin{aligned} J = & \lambda_0 \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (y_r[n_1, n_2] - y[n_1, n_2])^2 \\ & + \lambda_1 \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (y_{x_1r}[n_1, n_2] - y_{x_1}[n_1, n_2])^2 \\ & + \lambda_2 \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (y_{x_2r}[n_1, n_2] - y_{x_2}[n_1, n_2])^2 \\ & + \lambda_{1,2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (y_{x_1, x_2r}[n_1, n_2] - y_{x_1, x_2}[n_1, n_2])^2 \\ & + \lambda_r J_r, \end{aligned} \quad (3.64)$$

donde $y_r[n_1, n_2]$, $y_{x_1r}[n_1, n_2]$, $y_{x_2r}[n_1, n_2]$ y $y_{x_1, x_2r}[n_1, n_2]$, son las muestras regularizadas de la función, derivada con respecto a x_1 , derivada con respecto a x_2 y derivada cruzada respectivamente. Los parámetros λ_i son los parámetros de regularización.

Derivando el funcional J con respecto a las muestras regularizadas, e igualando a cero el conjunto de ecuaciones resultantes, se obtiene un sistema matricial como en el modelo MLR1D, a partir del cual se obtiene la solución regularizada. Las muestras regularizadas se aplican al modelo MLI para obtener la reconstrucción de la función. Obviamente, los requerimientos computacionales del modelo son elevados, especialmente a medida que crece el orden del mismo. Por ejemplo, en el caso $D = 2$, se tienen como incógnitas del sistema las muestras regularizadas de la función y de 8 derivadas. Si se considera una rejilla de $N_1 \times N_2$ instantes de muestreo, se tienen $9 \times N_1 \times N_2$ incógnitas, lo que significa la inversión de una matriz cuadrada del mismo tamaño. Aunque también es posible utilizar una alternativa mediante descenso de gradiente para resolver el sistema, la carga computacional es, en cualquier caso, importante.

Para la selección de los parámetros de regularización asociados a las muestras de las derivadas, se utiliza la relación de potencias de ruido entre la derivada correspondiente y la función, como en la expresión (3.35). Para λ_r se evalúa la expresión (3.36) para cada una de las direcciones del espacio de entrada y se toma el valor promedio.

Regularización simplificada (MLRs) Para evitar la elevada carga computacional asociada a la regularización global del modelo 2D, es posible emplear nuevamente un modelo basado en el modelo de regularización unidimensional, similar a la propuesta MSR. Para el caso $D = 1$, teniendo en cuenta la Figura 3.6, la alternativa propuesta, utilizando el modelo MLR1D

1. Regulariza la función y la derivada respecto a x_1 , en la dirección de x_1 para todo $x_2[n_2]$ (líneas horizontales).
2. Regulariza la función y la derivada respecto a x_2 , en la dirección de x_2 para todo $x_1[n_1]$ (líneas verticales).
3. Regulariza la derivada respecto a x_1 , y la derivada cruzada, en la dirección de x_2 para todo $x_1[n_1]$ (líneas verticales).
4. Regulariza la derivada respecto a x_2 , y la derivada cruzada, en la dirección de x_1 para todo $x_2[n_2]$ (líneas horizontales).

De esta forma se han obtenido dos valores regularizados para cada una de las secuencias de muestras, tanto de la función como de sus derivadas con respecto a x_1 y x_2 y su derivada cruzada, cada uno de ellos regularizando en una dirección. La solución propuesta consiste en promediar estos dos valores para cada secuencia.

Esta alternativa simplificada, aunque de nuevo no se puede considerar una técnica de regularización estricta, y aunque, como se mostrará con posterioridad, proporciona resultados inferiores al modelo MLR, implica una carga computacional considerablemente menor, por lo que en algunas aplicaciones puede resultar más interesante.

Por último, comentar que los parámetros de regularización se obtienen en cada caso mediante las expresiones (3.35) y (3.36) aplicadas al problema de regularización unidimensional específico.

3.3.3 Extensión a D derivadas

La extensión de los modelos MLI y MLR propuestos para un número $D > 1$ de derivadas resulta evidente. En el caso de los modelo MSI y MLI, se sigue el mismo procedimiento presentado para el caso $D = 1$, empleando los modelos unidimensionales del grado correspondiente. El único detalle a considerar es que, en el modelo MSI, las derivadas con respecto a la variable en cuya dirección se pretende reconstruir se modelan mediante un spline del grado correspondiente al espacio sobre el que se proyecta la solución, en la dirección opuesta a la reconstrucción ($\hat{f}_{x_1}^1(x_1[n_1], x_2)$ en el ejemplo mostrado en el Apartado 3.3.1). Por ejemplo, en el caso $D = 2$, cuando se reconstruye en la dirección x_1 , se emplea un spline cúbico para modelar $\hat{f}_{x_1}^1(x_1[n_1], x_2)$ y $\hat{f}_{x_1}^2(x_1[n_1], x_2)$.

En el caso de los modelo MSR, MLR y MLRs, la extensión resulta también evidente.

3.4 Resultados

En esta sección se presentan algunos resultados obtenidos con los modelos de interpolación y regularización propuestos en espacios de entrada 1D y 2D.

3.4.1 Resultados en espacios de entrada unidimensionales

3.4.1.1 Resultados del modelo MLI en espacios de entrada 1D

En primer lugar, vamos a mostrar que muestrear la función y D derivadas resulta en general ventajoso, cuando se trabaja en situaciones de ruido, frente a la alternativa de muestrear únicamente la función a una velocidad D veces superior. A continuación se comparan los resultados obtenidos con el modelo MLI de grado 3 ($D = 2$), y con un spline cúbico convencional obtenido interpolando únicamente las muestras de la función a una velocidad 3 veces superior.

Como señales de prueba se han utilizado señales limitadas en banda generadas mediante una combinación de 100 tonos de amplitudes, frecuencias (dentro de la banda de interés) y fases aleatorias con una distribución uniforme. Para evaluar los resultados se han generado en cada caso 1000 experimentos con este tipo de señales. Los resultados se evalúan mediante la relación señal a error (SER) en dB de la función reconstruida y de sus derivadas, para una determinada relación señal a ruido (SNR) en las muestras de la función, y en función de la SNR en las muestras de las derivadas.

En primer lugar se han muestreado las funciones de prueba al doble de la frecuencia de Nyquist ($T = 1/4BW$ en el caso del spline convencional, y $T = 3/4BW$ para el MLI, siendo BW el ancho de banda de la señal). La Figura 3.7 muestra los resultados obtenidos con las dos alternativas para una SNR en las muestras de la función de 20 dB. La reconstrucción de las derivadas (primera y segunda) es notablemente mejor con el MLI que con el spline. Además, si la SNR en las derivadas es mejor o igual que en la función, el MLI aprovecha eficientemente esa información ya que proporciona mejores

resultados en la reconstrucción de la función que el spline convencional. Obviamente, si la SNR en las derivadas es menor, la reconstrucción de la función empeora, aunque en las derivadas de la señal reconstruida se observa una mejora incluso con un margen de unos 6 dB de SNR para la derivada primera y casi 10 dB para la derivada segunda.

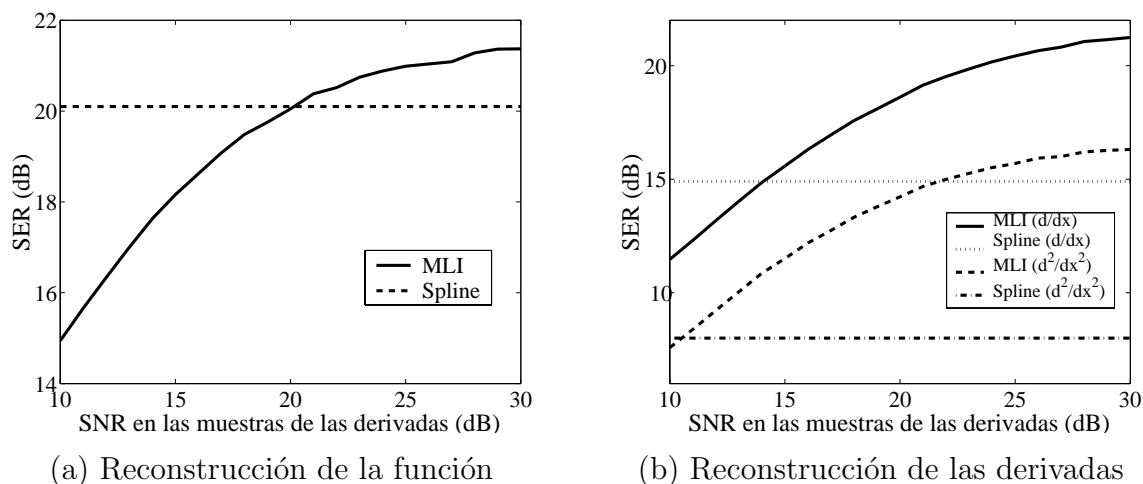


Figura 3.7: Reconstrucción de una señal limitada en banda muestreada a 2 veces la velocidad de Nyquist con el modelo MLI y con un spline cúbico. La SNR en las muestras de la función es de 20 dB, y los resultados se presentan en función de la SNR común en las muestras de las dos derivadas

Este comportamiento se hace más evidente a medida que se sobremuestran más las señales. La Figura 3.8 muestra los resultados obtenidos con ambas alternativas cuando las señales se muestrean a 3 veces la frecuencia de Nyquist de nuevo con una relación SNR de 20 dB en las muestras de la función. En este caso se aprecia que la mejora muestreando las derivadas es mucho más significativa. Se observa una mejora en la reconstrucción de la función con un margen de unos 3 dB en la SNR de las derivadas. En la reconstrucción de las derivadas, incluso con 10 dB menos de SNR en las muestras de las mismas, se siguen obteniendo mejores resultados utilizando el modelo MLI.

Estos resultados indican que, en general, utilizar las muestras de las derivadas en la reconstrucción de la función proporciona algunas ventajas, especialmente en aplicaciones en las que la aproximación de las derivadas sea también de interés. A esta ventaja hay que añadir el hecho de que el modelo MLI es un método de reconstrucción local, con lo que la carga computacional para la reconstrucción localizada de la función es sensiblemente menor que la de un spline cúbico convencional.

3.4.1.2 Resultados del modelo MLR en espacios de entrada 1D

A continuación se presentan algunos resultados en espacios de entrada unidimensionales que demuestran que, en situaciones de ruido, la regularización de las muestras de

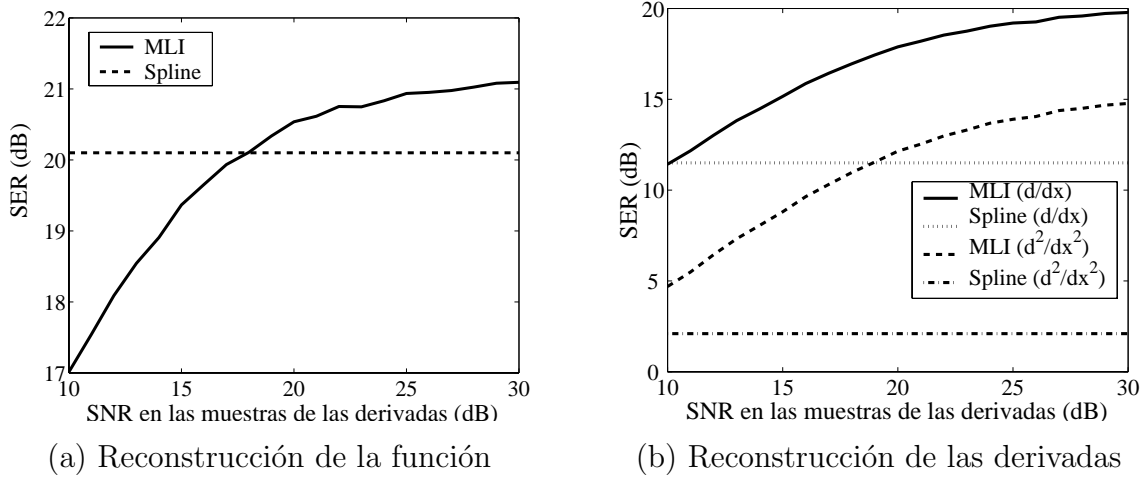


Figura 3.8: Reconstrucción de una señal limitada en banda muestreada a 3 veces la velocidad de Nyquist con el modelo MLI y con un spline cúbico con una SNR en las muestras de la función de 20 dB

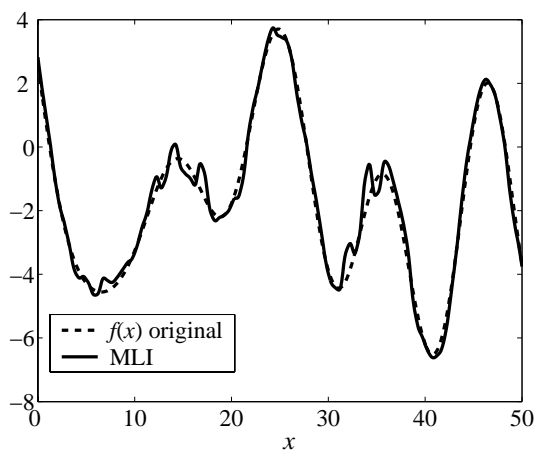
función y derivadas (modelo MLR) proporciona mejores resultados que la interpolación (modelo MLI).

La Figura 3.9 compara la reconstrucción de una señal limitada en banda a 0.1 Hz, con un período de muestreo $T = 1$, mediante los modelos MLI y MLR de grado 3 ($D = 2$). Las muestras de la función y de las dos derivadas tienen una SNR de 20 dB. Se observa con claridad como la reconstrucción mediante el MLR es mucho mejor que la obtenida utilizando el MLI, especialmente en la reconstrucción de la derivada. En este caso, se trata de una función muy sobremuestreada, donde la sensibilidad al ruido en la derivada es elevada.

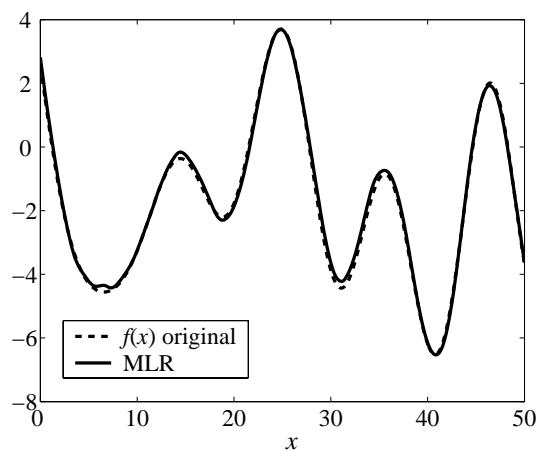
A continuación se comparan las prestaciones del modelo MLR de grado 3 ($D = 2$), con las de dos modelos de interpolación: el modelo MLI y el modelo derivado de la extensión del teorema de Shannon para el muestreo de las derivadas [Linden59, Linden60], que se ha descrito en la Sección 2.1.1. Los parámetros de regularización se han seleccionado de acuerdo a las expresiones (3.35) y (3.36).

Como señales de prueba se han generado de nuevo funciones limitadas en banda utilizando una combinación de 100 tonos de amplitudes, frecuencias (dentro de la banda de interés) y fases aleatorias con una distribución uniforme. En cada ejemplo se han generado 1000 experimentos con estas señales. Se presentan los resultados obtenidos en la reconstrucción de la función y de la derivada a través de la SER, en dB, para una determinada SNR en las muestras de la función, en función de la SNR en dB en las muestras de las derivadas (para ambas derivadas tomamos la misma SNR).

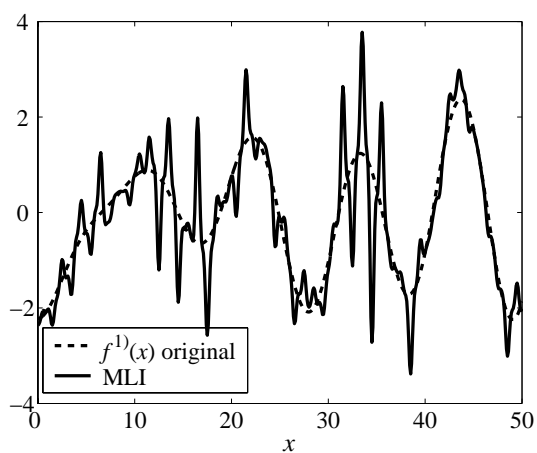
En el primer experimento se considera un conjunto de muestras de la función y de las derivadas primera y segunda, muestreadas cada una a $2/3$ de su frecuencia de Nyquist (el número de muestras es así equivalente a muestrear únicamente la función a 2 veces su frecuencia de Nyquist). La SNR en las muestras de la función es de 10



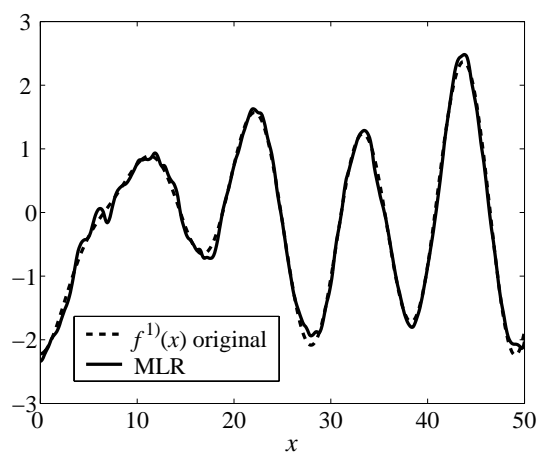
(a) Función reconstruida con el MLI



(b) Función reconstruida con el MLR



(c) Derivada reconstruida con el MLI



(d) Derivada reconstruida con el MLR

Figura 3.9: Reconstrucción de una función y su derivada primera con los modelos MLI y MLR de grado 3

dB. Los parámetros de regularización se han seleccionado de nuevo de acuerdo a las expresiones (3.35) y (3.36).

La Figura 3.10 presenta los resultados obtenidos con los tres modelos. En este caso, el modelo MLR obtiene mejores resultados respecto al MLI y al método de Shannon, en la reconstrucción tanto de la función como de la derivada.

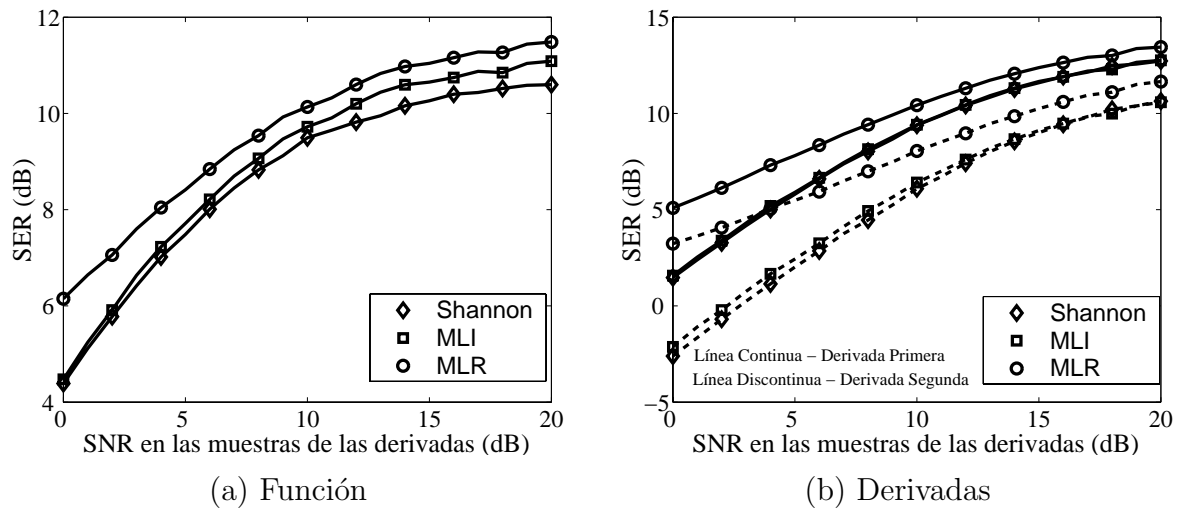


Figura 3.10: Reconstrucción de una función a partir de las muestras de la misma y de las derivadas primera y segunda muestreadas cada una a $2/3$ de la frecuencia de Nyquist y con una SNR en las muestras de la función de 10 dB, en función de la SNR en las muestras de las derivadas.

Para estudiar la influencia del factor de sobremuestreo, consideramos ahora que tanto la función como las derivadas han sido muestreadas cada una a su frecuencia de Nyquist, de nuevo con una relación señal a ruido (SNR) en las muestras de la función de 10 dB. En la Figura 3.11 se muestran los resultados obtenidos en este caso con cada modelo. La ganancia proporcionada por el MLR frente al MLI es sensiblemente mayor que en el caso anterior. Además, cuando la SNR en las muestras de las derivadas disminuye, la mejora es más significativa, mientras que cuando la SNR en las derivadas es mayor que en la función, el MLR es capaz de utilizar esta información para mejorar en la reconstrucción de la función.

La mejora en los resultados obtenidos con el MLR se explica debido a la sensibilidad al ruido de los métodos de interpolación, que se incrementa al aumentar la velocidad de muestreo. Este aspecto, que para el MLI se mostró en la Sección 3.1.3, se ilustra en la Figura 3.12; en ella se muestran los resultados en la reconstrucción de una función de ancho de banda 1, con una SNR en las muestras de la función y de las derivadas de 10 dB, en función del período de muestreo. Se comprueba que a medida que disminuye el período de muestreo, con los métodos de interpolación se produce una degradación en la reconstrucción de las derivadas, aunque el error en la aproximación de la función sólo disminuya ligeramente. Sin embargo, el MLR no presenta esta sensibilidad tan

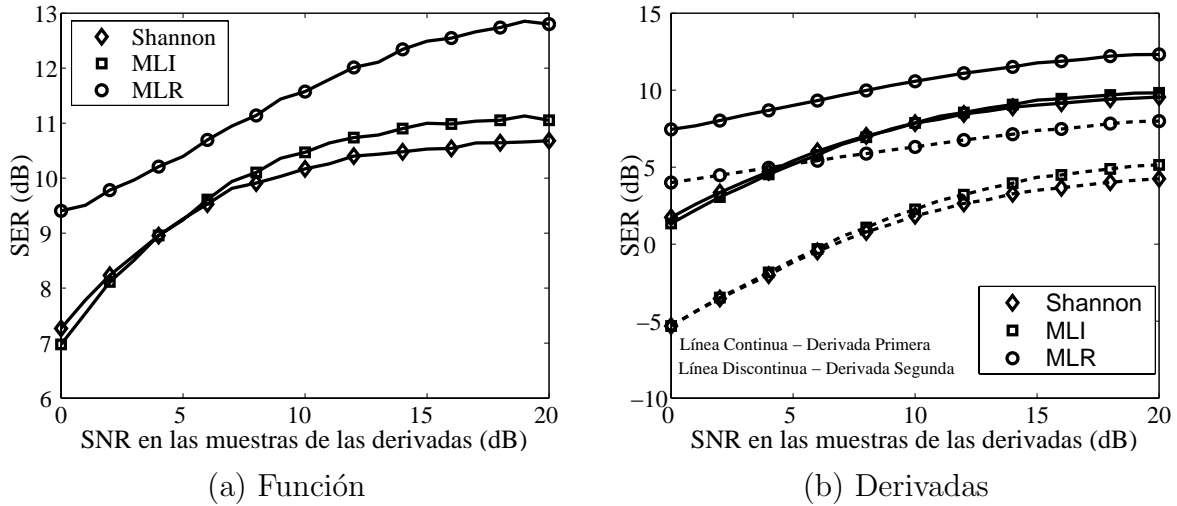


Figura 3.11: Reconstrucción de una función a partir de las muestras de la misma y de sus derivadas primera y segunda muestreadas a la frecuencia de Nyquist y con una SNR en las muestras de la función de 10 dB, en función de la SNR en las muestras de las derivadas.

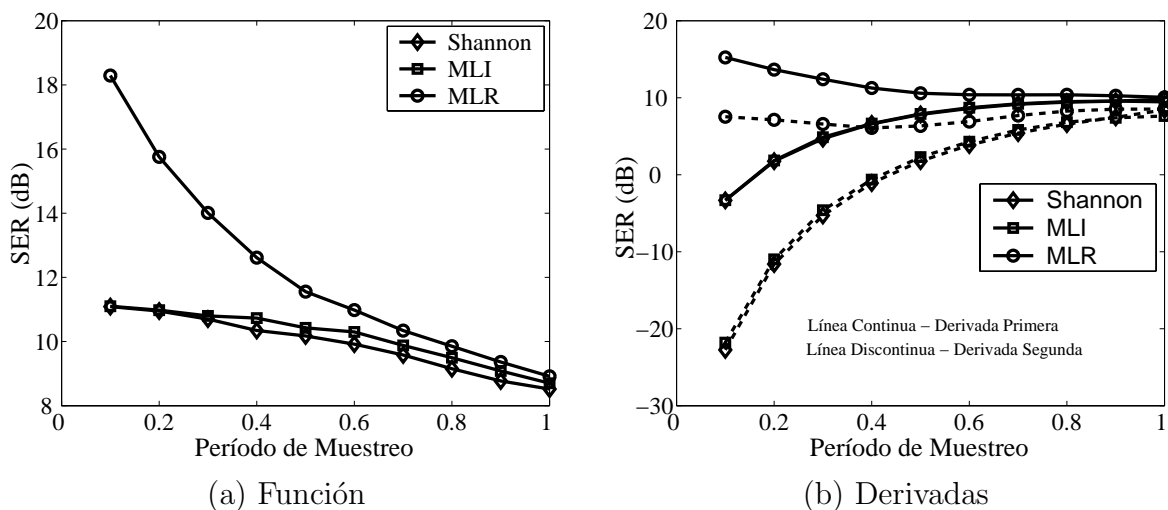


Figura 3.12: Reconstrucción de una función de ancho de banda 1, con una SNR en las muestras de función y derivadas de 10 dB, en función del período de muestreo.

pronunciada al ruido. Se observa como, a medida que disminuye el período de muestreo se obtiene una apreciable mejora en la aproximación de la función.

Todos los resultados que se han presentado corresponden a señales limitadas en banda. Se ha utilizado este conjunto de señales porque se considera un conjunto suficientemente representativo, por ser el tipo de señales más habituales con las que se trabaja en procesado de señal, y porque permite la comparación del modelo propuesto con uno de los métodos más conocidos de interpolación de una función y sus derivadas: la extensión del teorema de muestreo de Shannon. No obstante, los mismos resultados se han obtenido para otras señales no limitadas en banda (splines, gaussianas, etc).

3.4.1.3 Conclusiones

En resumen, las principales conclusiones obtenidas para los modelos MLI y MLR propuestos son:

- El MLI aprovecha la información de las derivadas para obtener una aproximación más fiel de las mismas. Además, cuando el ruido en las derivadas es menor que en la función, o cuando las señales están muy sobremuestreadas, la información de las derivadas permite mejorar la reconstrucción de la función.
- El MLR mejora la reconstrucción obtenida con el MLI en situaciones de ruido.
- El factor de sobremuestreo de las funciones afecta a la sensibilidad al ruido en la reconstrucción de las derivadas del MLI. El MLR permite reducir esta sensibilidad: la mejora obtenida con el MLR es mayor a medida que crece la frecuencia de muestreo.

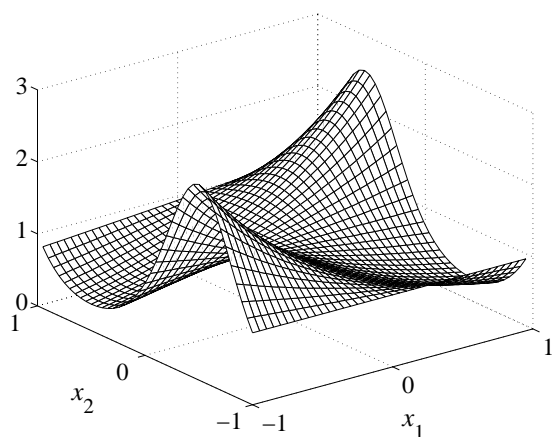
3.4.2 Resultados en espacios de entrada bidimensionales

Para evaluar los resultados obtenidos con los modelos 2D propuestos, se ha seleccionado un conjunto de prueba formado por 8 funciones previamente utilizadas por Cherkassky, Gehring y Mulier [Cherkassky96] para llevar a cabo una comparación entre diferentes alternativas de modelado adaptativo. Se trata de un conjunto de funciones obtenidas de diferentes referencias bibliográficas, con distintas características, por lo que forman un conjunto de prueba suficientemente representativo. Las expresiones y la representación de estas funciones se muestran en el Apéndice F. Los resultados de reconstrucción se presentan mediante la SER en la reconstrucción de la función y de las derivadas. Para un espacio de entrada bidimensional, la reconstrucción de las derivadas de orden d se evalúa mediante el promedio de la derivada con respecto a las dos direcciones del espacio de entrada.

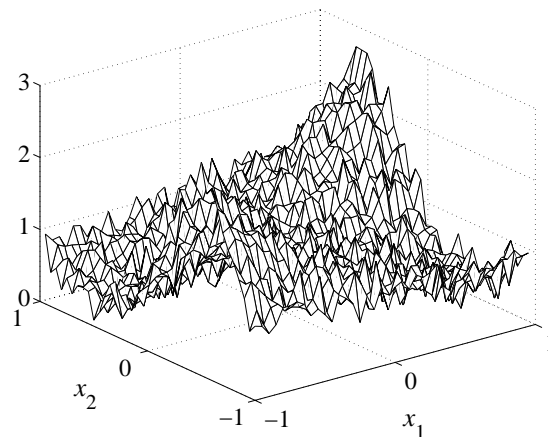
3.4.2.1 Modelo MSR (Derivadas direccionales)

En primer lugar se presentan los resultados obtenidos cuando se dispone únicamente de las derivadas direccionales, aplicando el modelo MSR. Se pretende con este primer

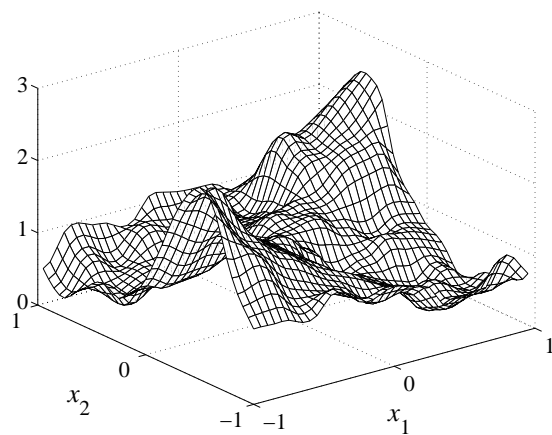
ejemplo resaltar las ventajas del modelo regularizado MSR frente al modelo de interpolación MSI. La Figura 3.13 muestra la reconstrucción de la Función 2 empleando el modelo de grado 3 ($D = 2$), con 10 instantes de muestreo en cada dirección del espacio de entrada y con una SNR de 10 dB tanto en las muestras de la función como de las derivadas. Se observa con claridad la mejora obtenida por la regularización de las secuencias de muestras.



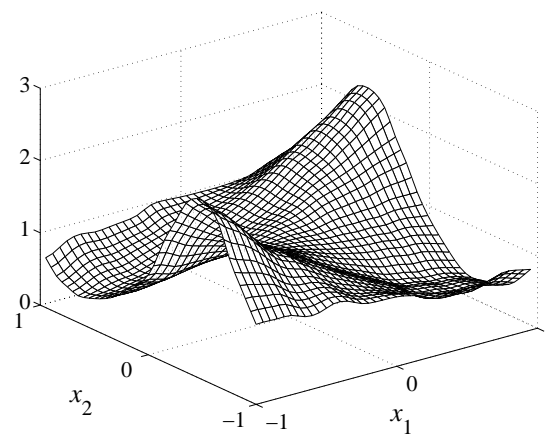
(a) Función 2



(b) Función ruidosa (SNR=10dB)



(c) MSI



(d) MSR

Figura 3.13: Reconstrucción de la Función 2 mediante los modelos MSI y MSR con una SNR de 10 dB en las muestras de función y derivadas, y con 10 instantes de muestreo equiespaciados en cada dirección del espacio de entrada

Esta mejora depende de la SNR en las muestras de función y derivadas, así como del período de muestreo considerado. Para ilustrar este comportamiento, la Figura 3.14 muestra los resultados obtenidos con los modelos de grado 2 ($D = 1$) en la reconstrucción de la Función 1 (línea continua) y sus derivadas primeras (línea discontinua),

considerando 15 instantes de muestreo equiespaciados en cada dirección del espacio de entrada y una SNR en las muestras de la función de 20 dB. Se observa la importante mejora obtenida en este caso al regularizar la secuencia de muestras.

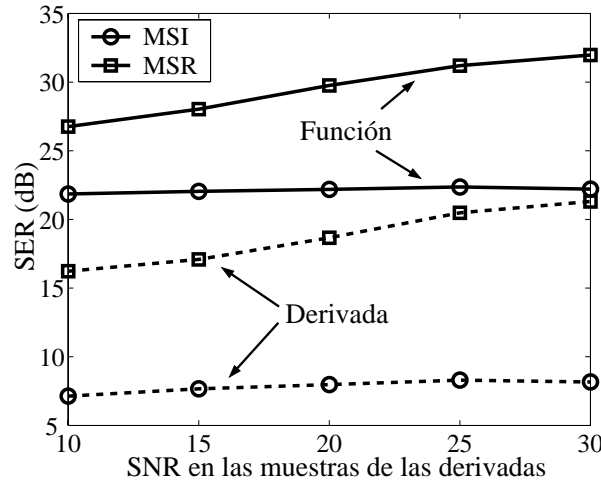


Figura 3.14: Resultados en la reconstrucción de la Función 1 con una SNR de 20 dB en las muestras de la función y 15 instantes de muestreo en cada dirección del espacio de entrada

Cuando se disminuye el número de instantes de muestreo, la mejora se reduce de forma apreciable. En la Figura 3.15 se representan los resultados obtenidos considerando ahora 5 instantes de muestreo equiespaciados en cada dirección del espacio de entrada y una SNR en las muestras de la función de 20 dB: aunque se sigue observando una mejora al regularizar, esta mejora se ha reducido de forma significativa.

El mismo comportamiento se aprecia cuando se tiene un número mayor de derivadas. En concreto, la Figura 3.16 presenta los resultados obtenidos con los modelos de grado 3 ($D = 2$) en la reconstrucción de la Función 1 (línea continua) y sus derivadas primera (línea discontinua) y segunda (línea de puntos), considerando 5 y 15 instantes de muestreo en cada dirección del espacio de entrada y una SNR en las muestras de la función de 20 dB. De nuevo, la mejora obtenida con el modelo MLR es superior cuando aumenta la velocidad de muestreo.

Resulta de especial interés comentar los resultados obtenidos en la reconstrucción de la derivada segunda. Por un lado, se observa que el error de reconstrucción aumenta al aumentar el número de instantes de muestreo. Esto se debe al proceso de regularización empleado, que solamente suaviza las derivadas teniendo en cuenta independientemente cada una de las direcciones del espacio. Como la sensibilidad al ruido es mayor a medida que crece el orden de las derivadas, la deficiencia del método de regularización se manifiesta en las derivadas de mayor orden. Se puede apreciar que al aumentar la velocidad de muestreo, aunque se mejora en la reconstrucción de la función y de la derivada primera, los resultados en la derivada segunda son sensiblemente peores. A

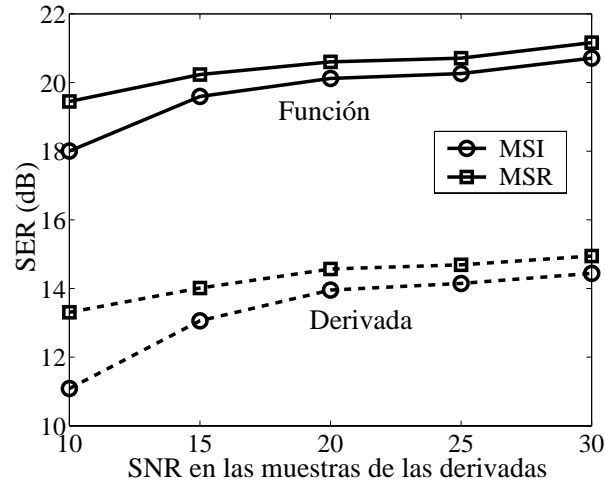
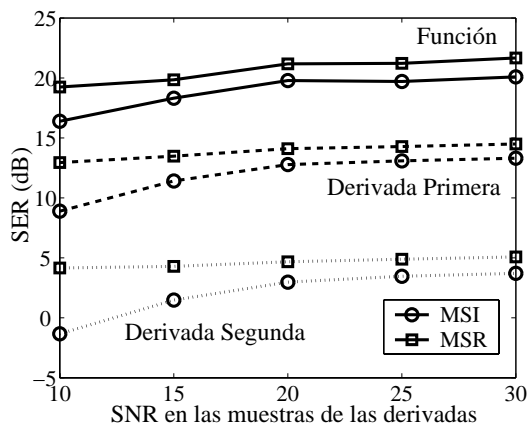
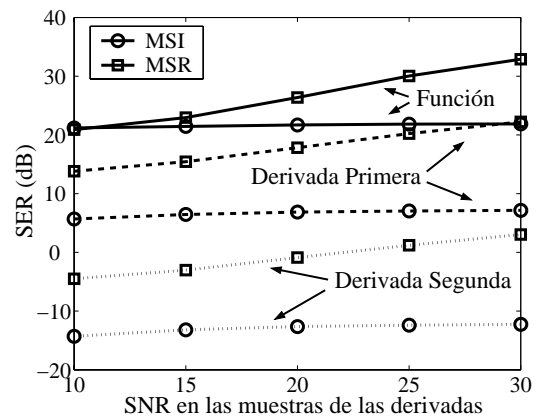


Figura 3.15: Resultados en la reconstrucción de la Función 1 con una SNR de 20 dB en las muestras de la función y 5 instantes de muestreo en cada dirección del espacio de entrada



(a) 5 instantes de muestreo



(b) 15 instantes de muestreo

Figura 3.16: Resultados en la reconstrucción, mediante los modelos de grado 3, de la Función 1 con una SNR de 20 dB en las muestras de la función

pesar de esta limitación, el modelo MSR sigue obteniendo unos resultados notablemente superiores a los obtenidos mediante el modelo MSI. Por otro lado, cuando la frecuencia de muestreo es baja, hay que tener en cuenta que la aproximación de las derivadas segundas se lleva a cabo mediante funciones lineales a tramos, lo que limita la capacidad de aproximación cuando el período de muestreo es grande.

El mismo comportamiento mostrado para la Función 1 en los resultados previos se ha observado al reconstruir el resto de las 8 funciones de prueba. Las Tablas 3.1 y 3.2 resumen los resultados obtenidos en la reconstrucción de cada función (y el valor medio en las 8 funciones) utilizando, respectivamente, 5 y 15 instantes de muestreo en cada dirección del espacio de entrada. Los resultados corresponden a una SNR de 20 dB tanto en las muestras de la función como de las derivadas.

Aproximación de la función									
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8	V. Medio
MLI	19.8	19.3	20.0	17.7	13.5	19.5	19.9	8.1	17.2
MLR	21.7	21.5	21.9	17.8	14.7	20.3	20.5	9.0	18.5

Aproximación de la derivada primera									
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8	V. Medio
MLI	12.8	13.6	11.8	15.0	12.2	16.4	16.3	5.8	13.0
MLR	14.1	16.7	13.3	15.4	13.3	17.3	16.6	6.7	14.2

Aproximación de la derivada segunda									
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8	V. Medio
MLI	2.9	-4.0	1.8	8.1	5.0	10.6	10.1	0.1	4.3
MLR	4.6	-0.5	3.5	9.3	6.5	11.9	10.8	1.1	5.9

Tabla 3.1: Resultados obtenidos en la reconstrucción de las funciones de prueba, utilizando únicamente las derivadas direccionales, con 5 instantes de muestreo en cada dirección del espacio de entrada y una SNR de 20 dB en las muestras de la función y las derivadas

El modelo MLR proporciona resultados superiores al modelo MLI en todos los casos, y la mejora es apreciablemente superior a medida que el tiempo de muestreo disminuye. Un caso especial es la Función 4, para cual, con 15 instantes de muestreo, el modelo MLI proporciona unos mejores resultados de SER en la función, aunque obtiene resultados similares en la derivada primera e inferiores en la derivada segunda. También se aprecia que, al aumentar el número de instantes de muestreo, aunque se obtiene una reconstrucción con menor error en la función, y en general se mejora en la derivada primera, la reconstrucción de la derivada segunda sufre una importante degradación en todos los casos.

3.4.2.2 Reconstrucción con las derivadas cruzadas

Cuando se conocen tanto las derivadas direccionales como las cruzadas, se pueden utilizar las dos alternativas de regularización propuestas, MLR y MLRs. Utilizando la

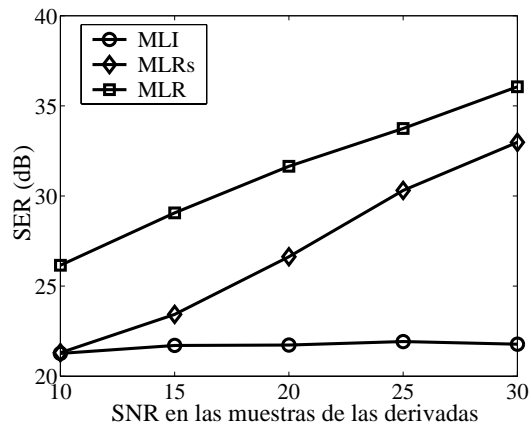
Aproximación de la función									
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8	V. Medio
MLI	21.7	21.2	21.6	21.5	19.2	21.3	21.4	21.0	21.1
MLR	26.9	23.5	31.2	19.6	25.5	22.1	21.9	22.1	24.1
Aproximación de la derivada primera									
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8	V. Medio
MLI	6.8	6.2	3.4	11.9	8.7	8.6	9.1	10.7	8.2
MLR	17.8	15.3	16.7	11.7	18.3	16.2	15.9	16.1	16.0
Aproximación de la derivada segunda									
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8	V. Medio
MLI	-12.6	-18.6	-17.1	-3.4	-8.6	-6.6	-6.3	-4.2	-9.7
MLR	-0.8	-9.2	-3.4	3.0	1.7	1.8	2.6	3.6	-0.1

Tabla 3.2: Resultados obtenidos en la reconstrucción de las funciones de prueba, utilizando únicamente las derivadas direccionales, con 15 instantes de muestreo en cada dirección del espacio de entrada y una SNR de 20 dB en las muestras de la función y las derivadas

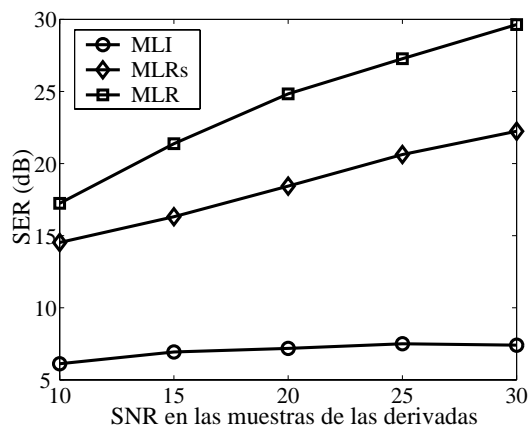
regularización global del problema 2D (modelo MLR) se obtienen resultados superiores, especialmente al aumentar el número de instantes de muestreo, donde la sensibilidad al ruido del modelo interpolador es mayor. Para ilustrar este aspecto, la Figura 3.17 presenta los resultados obtenidos con ambas alternativas en la reconstrucción de la Función 1, con una SNR en las muestras de la función de 20 dB y 15 instantes de muestreo en cada una de las direcciones del espacio. Se observa que los resultados obtenidos con el MLR son superiores a los obtenidos con la regularización simple del modelo MLRs. La mejora es más sensible a medida que aumenta el orden de las derivadas.

El mismo comportamiento se ha observado también en el resto de funciones de prueba consideradas. Las Tablas 3.3 y 3.4 comparan los resultados obtenidos con ambas soluciones regularizadas en las 8 funciones de prueba. Las tablas presentan los resultados de SER en dB para una SNR de 20 dB en las muestras de función y derivadas, y con 5 y 15 instantes de muestreo, respectivamente, en cada una de las direcciones del espacio. Se comprueba que el modelo MLR presenta resultados sensiblemente superiores al MLRs para el caso de 15 instantes de muestreo. Sin embargo, cuando se tienen únicamente 5 instantes de muestreo en cada dirección del espacio, los resultados son muy similares. En este caso, la ganancia sobre el modelo MLI es notablemente menor.

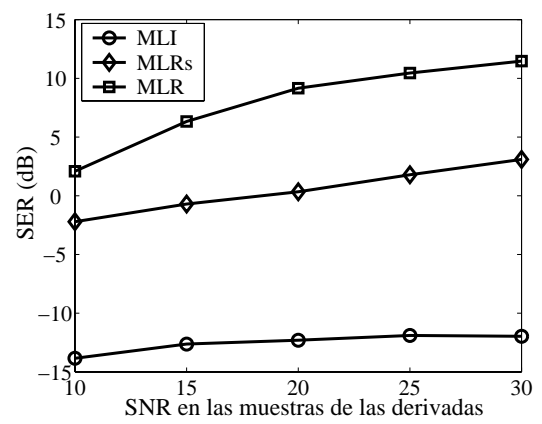
Aunque el modelo MLR resulta muy eficaz, presenta el inconveniente de su elevada carga computacional. Por esta razón, en algunas aplicaciones puede resultar más interesante utilizar uno de los modelos simplificados de regularización: el MSR o el MLRs. Estos dos modelos, cuando la velocidad de muestreo es elevada, proporcionan unos resultados similares en la reconstrucción de las derivadas direccionales. Sin embargo, en las derivadas cruzadas, como era de esperar, el modelo MLRs proporciona unos



(a) Función



(b) Derivada primera



(c) Derivada segunda

Figura 3.17: Resultados en la reconstrucción, mediante los modelos de grado 3, de la Función 1 con una SNR de 20 dB en las muestras de la función y 15 instantes de muestreo en cada dirección del espacio de entrada

Aproximación de la función									
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8	V. Medio
MLI	20.9	19.4	20.8	19.5	14.3	19.7	19.9	12.1	18.3
MLRs	22.8	22.0	22.8	18.4	15.5	21.1	20.7	13.2	19.6
MLR	22.9	22.2	22.9	18.3	15.7	21.3	20.5	13.7	19.7

Aproximación de la derivada primera									
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8	V. Medio
MLI	16.1	13.4	13.1	18.4	13.12	16.5	16.0	11.0	14.7
MLRc	18.1	16.9	15.3	17.5	14.1	18.0	17.1	11.7	16.1
MLR	18.1	17.0	15.4	17.3	14.3	18.1	17.0	11.9	16.1

Aproximación de la derivada segunda									
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8	V. Medio
MLI	6.8	-4.5	3.1	12.4	5.5	11.5	8.8	3.9	5.9
MLRs	9.3	-0.5	5.5	13.8	6.6	13.1	10.6	4.7	7.9
MLR	9.5	-0.5	5.4	13.6	6.7	13.3	10.0	4.8	7.9

Tabla 3.3: Resultados obtenidos en la reconstrucción de las funciones de prueba, utilizando las derivadas direccionales y cruzadas, con 5 instantes de muestreo en cada dirección del espacio de entrada y una SNR de 20 dB en las muestras de la función y las derivadas

Aproximación de la función									
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8	V. Medio
MLI	21.7	21.5	21.6	21.6	19.9	21.5	21.7	21.4	21.3
MLRs	27.2	23.9	31.4	19.6	26.6	23.2	22.6	22.9	24.7
MLR	32.2	29.5	35.3	20.9	31.7	27.5	30.2	27.1	29.3

Aproximación de la derivada primera									
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8	V. Medio
MLI	7.1	6.5	3.5	12.1	9.4	8.9	9.4	11.4	8.5
MLRs	18.7	16.3	16.8	12.1	19.6	17.7	17.0	17.8	17.0
MLR	24.8	23.0	21.3	18.9	24.7	22.3	20.9	18.9	21.9

Aproximación de la derivada segunda									
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8	V. Medio
MLI	-12.3	-18.3	-17.1	-3.3	-8.1	-5.7	-5.6	-3.4	-9.2
MLRs	0.6	-7.6	-3.2	3.9	3.1	6.0	5.2	6.4	1.8
MLR	9.2	-1.5	1.6	8.8	14.7	16.4	7.5	7.0	8.0

Tabla 3.4: Resultados obtenidos en la reconstrucción de las funciones de prueba, utilizando las derivadas direccionales y cruzadas, con 15 instantes de muestreo en cada dirección del espacio de entrada y una SNR de 20 dB en las muestras de la función y las derivadas

resultados claramente superiores. La Figura 3.18 ilustra este aspecto: se representan, por un lado, la reconstrucción de la función y de las derivadas direccionales, y por otro, la reconstrucción de las derivadas cruzadas de diferentes órdenes. Se comprueba que si bien en la función y en las derivadas el modelo MLRs es sólo ligeramente superior al MSR, en la aproximación de las derivadas cruzadas la diferencia es importante, especialmente a medida que crece el orden de las derivadas.

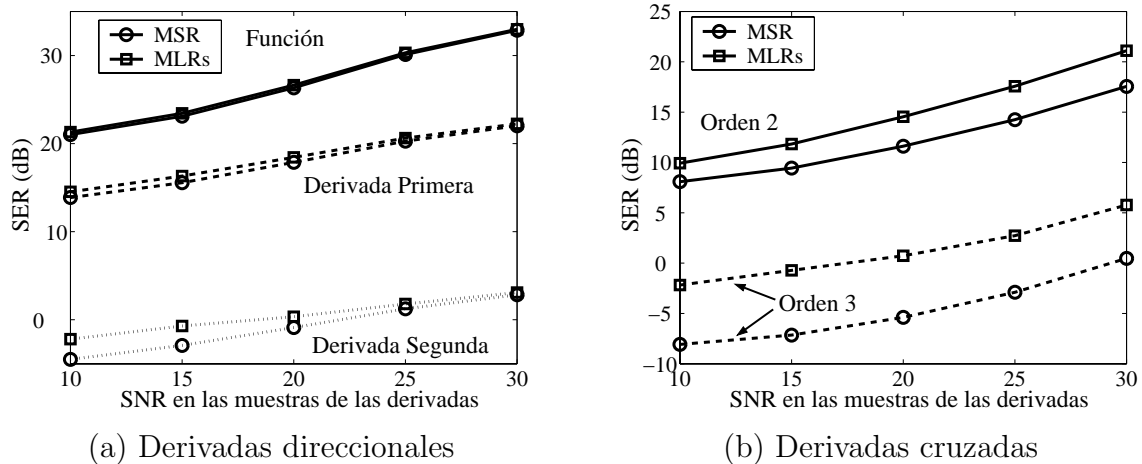


Figura 3.18: Reconstrucción de las derivadas direccionales y de las derivadas cruzadas empleando los modelos MSR y MLRs, con 15 instantes de muestreo en cada dirección del espacio de entrada

Cuando la frecuencia de muestreo disminuye, el modelo MLRs proporciona resultados claramente superiores incluso en la reconstrucción de la función y de las derivadas direccionales. La Figura 3.19 muestra los resultados obtenidos con ambos modelos cuando el número de instantes de muestreo en cada dirección del espacio de entrada es 5. En este caso, se comprueba que la información adicional que introducen las derivadas cruzadas en la reconstrucción resulta determinante.

3.5 Discusión

A lo largo de este capítulo se ha mostrado que en problemas de interpolación, la utilización de las derivadas en la reconstrucción de una función puede, en ciertas situaciones, mejorar la aproximación de la misma y, especialmente, la de sus derivadas.

Además, los modelos regularizados propuestos disminuyen de forma apreciable la sensibilidad al ruido de los modelos de interpolación. Este efecto es más apreciable a medida que disminuye el intervalo de muestreo, pues entonces aumenta la sensibilidad al ruido de los métodos de interpolación, especialmente en la reconstrucción de las derivadas.

Los modelos de regularización MSR y MLRs son métodos de regularización sub-óptimos, ya que no plantean la minimización de un funcional 2D de regularización.

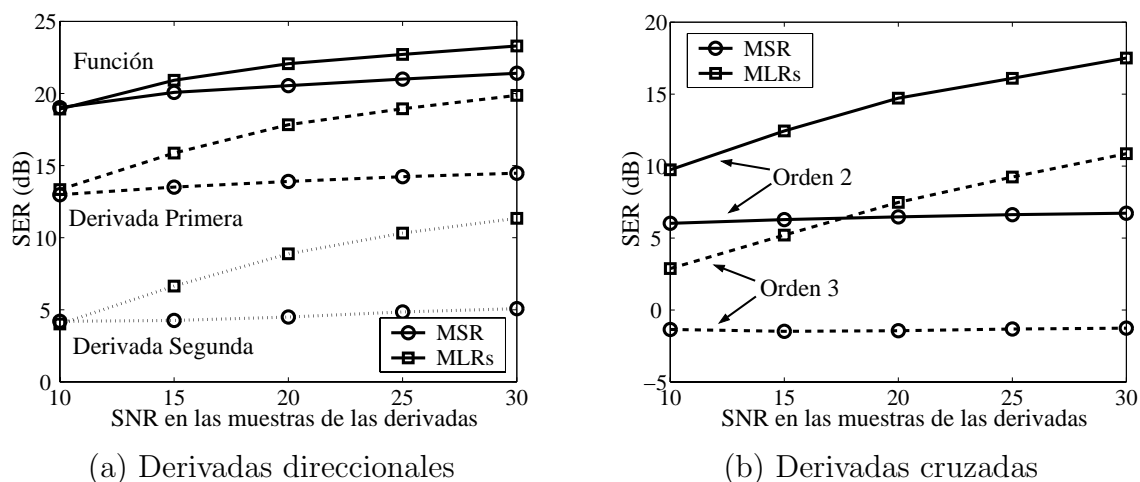


Figura 3.19: Reconstrucción de las derivadas direccionales y de las derivadas cruzadas empleando los modelos MSR y MLRs, con 5 instantes de muestreo en cada dirección del espacio de entrada

Las secuencias de muestras se regularizan de forma independiente en cada dirección del espacio. Esto hace que sus prestaciones, especialmente a medida que aumenta la frecuencia de muestreo, sean peores que las del modelo MLR. Sin embargo, este último modelo presenta el inconveniente de su elevada carga computacional.

En espacios de entrada bidimensionales, los métodos de interpolación/regularización propuestos tienen el inconveniente de que involucran una gran cantidad de derivadas a medida que aumenta el grado del modelo. Por ejemplo, para D derivadas el modelo sin derivadas cruzadas MSI necesita $1 + 2D$ secuencias de muestras, mientras que el modelo con derivadas cruzadas MLI necesita $(D + 1)^2$ secuencias de muestras. Esto limita la extensión del modelo a un número elevado de derivadas, especialmente para el modelo MLI (y sus extensiones regularizadas). La extensión a espacios de entrada de mayor dimensión presenta el mismo problema, incluso de forma más acusada.

Capítulo 4

El Modelo Lineal a Tramos Suavizado (MLTS)

4.1 Introducción

En el capítulo anterior se ha presentado un método para la interpolación/regularización de un conjunto de muestras de una función y sus derivadas. Desde el punto de vista de modelado, esta técnica precisa tantos parámetros como puntos de medida (en la función y las derivadas), lo que le resta utilidad en aquellas aplicaciones en que sea imperativo disponer de un modelo parsimonioso. Para solventar este inconveniente, en este capítulo se presenta una extensión de los modelos lineales a tramos que hemos denominado *Modelo Lineal a Tramos Suavizado* (MLTS). El problema considerado en este capítulo pertenece al ámbito del modelado, donde generalmente se tienen restricciones en el número de parámetros de la solución y ya no se pretende interpolar todo el conjunto de muestras disponible.

Los modelos lineales a tramos se han mostrado como herramientas muy útiles en una gran cantidad de problemas, como, por ejemplo, en el modelado de circuitos. Esto se debe a que son un tipo de modelos muy simple, con unos requerimientos computacionales reducidos y a que sus parámetros son relativamente fáciles de estimar. En cuanto a sus limitaciones, quizás la más importante sea la que deriva de la naturaleza abrupta de las transiciones entre regiones lineales, que hace que el modelo no sea derivable en esos puntos y que supone una importante limitación en un gran número de aplicaciones.

En este capítulo se presenta un nuevo método que trata de solventar este problema manteniendo las ventajas de los modelos lineales a tramos. La idea básica del mismo consiste en suavizar la transición entre regiones lineales de tal modo que el modelo final que se obtiene sea suave y derivable. Esto puede ayudar a tener un modelo más realista del problema que se aborda y además permite la utilización del método en entornos en los que la aproximación de las derivadas sea también importante. El método propuesto resulta una extensión del Modelo Lineal a Tramos Canónico inicialmente introducido por Chua y Kang [Chua77, Kang78]. A partir de este modelo, se propone

el suavizado de las transiciones entre las diferentes regiones lineales, para lo que se han estudiado distintas funciones de suavizado. Asimismo, se ha estudiado la suavidad del modelo, demostrando que se controla mediante un único parámetro que, por tanto, puede entenderse como un parámetro de regularización. Para el entrenamiento de este parámetro, se proponen distintas técnicas que permiten introducir la información de las derivadas en el modelo.

La estructura que sigue este capítulo es la siguiente. Se comienza presentando el Modelo Lineal a Tramos Canónico. A partir de este modelo, se analiza el problema de suavizar transiciones lineales a tramos y se introduce la función de suavizado que se va a utilizar en el modelo propuesto. A continuación se presenta el Modelo Lineal a Tramos Suavizado (MLTS), se realiza un análisis de suavidad del mismo y, para finalizar, se presentan algunos resultados que se han obtenido con dicho modelo.

4.2 El Modelo Canónico Lineal a Tramos (MLT)

4.2.1 Presentación del modelo

El Modelo Canónico Lineal a Tramos (MLT), propuesto por Chua y Kang [Chua77, Kang78], consiste en una representación analítica compacta compuesta de funciones lineales a tramos, en regiones delimitadas por particiones lineales. El adjetivo *canónico* hace referencia al hecho de que la representación canónica puede ser considerada como la que emplea la notación *mínima*, en el sentido de que utiliza de forma completa los grados de libertad paramétricos; es decir, que no utiliza más parámetros de los estrictamente necesarios. De esta forma, el modelo canónico lineal a tramos aproxima una función $f : R^M \rightarrow R^N$ de la forma siguiente

$$f(\mathbf{x}) = \mathbf{a} + \mathbf{B}\mathbf{x} + \sum_{i=1}^{\theta} \mathbf{c}_i | \langle \boldsymbol{\alpha}_i, \mathbf{x} \rangle - \beta_i |, \quad (4.1)$$

donde \mathbf{x} y $\boldsymbol{\alpha}_i$ son vectores de la misma dimensión, M , que el espacio de entrada, \mathbf{a} y \mathbf{c}_i son vectores de dimensión N , \mathbf{B} es una matriz $N \times M$, β_i es un escalar y \langle, \rangle representa el producto escalar. La expresión que aparece dentro de la función valor absoluto define un hiperplano de dimensión $M - 1$, que realiza la partición del espacio de entrada en las diferentes regiones lineales. La función valor absoluto determina la naturaleza de la transición entre las diferentes regiones lineales, dotando al modelo con las limitaciones propias de esta función como, por ejemplo, la falta de derivabilidad en la transición.

A efectos prácticos, la construcción de una función $f(\mathbf{x})$ con este modelo se va a llevar a cabo mediante la suma de una combinación lineal de las variables que definen el espacio de entrada, definida por los parámetros \mathbf{a} y \mathbf{B} , y de una combinación lineal de un conjunto de *hiperplanos bisagra* (“*hinging hyperplanes*”) [Breiman93]. Cada uno de estos hiperplanos bisagra se construye mediante dos hiperplanos lineales que se unen en un hiperplano de una dimensión menor situado sobre el espacio de entrada. Por

ejemplo, en el caso de un espacio de entrada unidimensional, serían dos rectas que se unen en un punto, en el caso de un espacio de entrada bidimensional serían dos planos que se unen sobre una recta, y así sucesivamente (La Figura 4.1 muestra un ejemplo de estos hiperplanos para un espacio de entrada de una y de dos dimensiones). Para un análisis más detallado de esta representación y de sus características y propiedades, se puede consultar [Chua88].

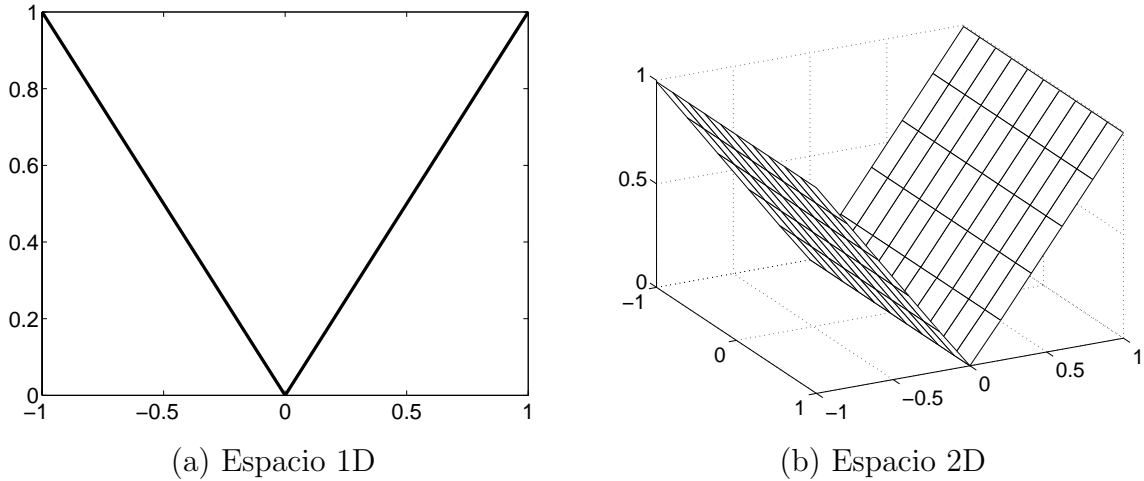


Figura 4.1: Hiperplanos bisagra del modelo MLT

La representación canónica tiene varias ventajas cuando se compara con otro tipo de descripciones de funciones lineales a tramos, derivadas principalmente de su formulación compacta. Por una parte, es la que requiere un número mínimo de parámetros, y por otra, no aparecen restricciones entre los parámetros, lo que facilita la determinación de los mismos. Por estas razones, esta representación es especialmente adecuada para su implementación en ordenadores, ya que presenta unos requerimientos computacionales bastante reducidos si se compara con otras alternativas de naturaleza similar. De hecho, esta representación ha sido ampliamente utilizada por estas razones en el campo del modelado de dispositivos (diodos y transistores principalmente) [Chua86].

En cuanto a la determinación de los parámetros del modelo, existen varios algoritmos para su cálculo [Chua83, Chua85, Chua86, Julián98]; pero, tal vez, el más eficiente sea el algoritmo iterativo propuesto en [Chua86]. En este caso, se diferencian dos tipos de parámetros que en cada iteración se actualizan de forma diferente. Por un lado, \mathbf{a} , \mathbf{B} y \mathbf{c}_i , con respecto a los cuales el modelo presenta una dependencia lineal, con lo que la solución de mínimo error cuadrático se puede calcular de forma sencilla. El resto de parámetros, que determinan la partición del espacio de entrada, y respecto a los cuales la dependencia es no lineal, se actualizan en cada iteración mediante un método de gradiente de segundo orden, en el que se estiman el gradiente y el Hessiano de la función de coste con respecto a dichos parámetros, definiendo la dirección de la variación de los mismos en función de estas estimas. Un análisis detallado de dicho algoritmo para espacios de entrada de varias dimensiones aparece en [Chua86].

4.2.2 Redes canónicas lineales a tramos

Un aspecto interesante del modelo lineal a tramos canónico es que admite una representación como filtro adaptativo o red neuronal. A continuación se discuten dichas representaciones.

4.2.2.1 Filtros canónicos lineales a tramos adaptativos

Lin y Unbehauen propusieron el empleo de un MLT para implementar un filtro digital adaptativo no lineal [Lin90]. Dicho filtro, entrenado con un criterio de error cuadrático mínimo, supone una alternativa a los métodos basados en las series Wiener-Volterra, que han sido ampliamente utilizadas en filtrado no lineal adaptativo. La idea de este filtro consiste en utilizar como espacio de entrada un vector de retardos de la señal

$$\mathbf{x} = [x[n], x[n-1], \dots, x[n-M+1]]^T, \quad (4.2)$$

y entonces emplear la expresión (4.1) para llevar a cabo la aproximación de la no-linealidad. Los parámetros se adaptan mediante un proceso de descenso de gradiente, tipo LMS, lo que permite una sencilla implementación del filtro. En la Figura 4.2 se muestra la estructura del filtro en el caso en el que no se considera el término constante \mathbf{a}

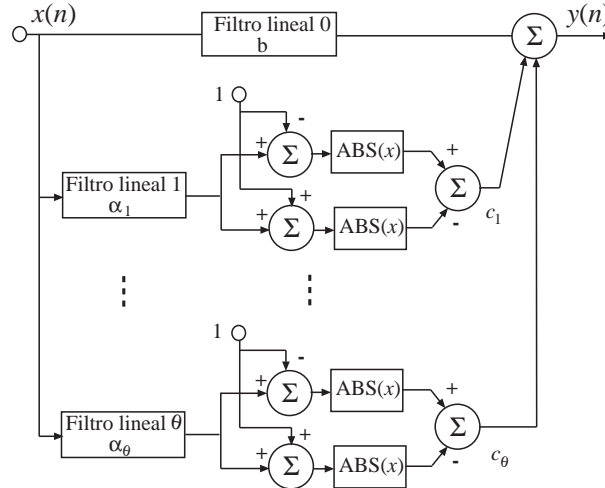


Figura 4.2: Filtro canónico lineal a tramos

En general, este método proporciona mejores resultados que otras alternativas de filtrado no lineal, como por ejemplo las basadas en series de Volterra, teniendo además la gran ventaja de tener un número de parámetros y una carga computacional sensiblemente menores. Como principal inconveniente aparecen los mínimos locales en la superficie de error: una inicialización adecuada de los parámetros mitiga este problema.

4.2.2.2 Red neuronal canónica lineal a tramos

El modelo MLT (4.1) también admite una representación en forma de red neuronal. En particular, Batruni [Batruni91] propuso una red neuronal multicapa en cascada en la que cada sección tiene una estructura canónica lineal a tramos. La Figura 4.3 muestra el esquema de una de estas secciones.

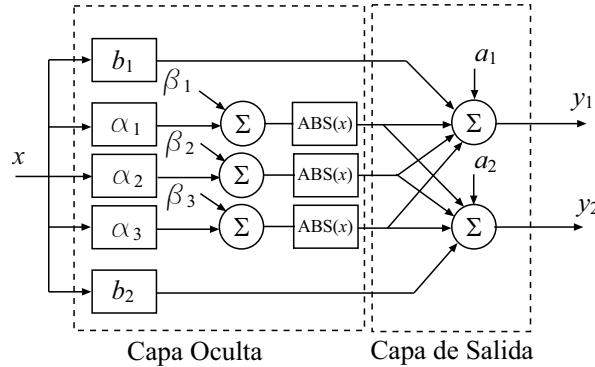


Figura 4.3: Sección de una red canónica lineal a tramos multicapa

Cada una de estas secciones, que implementa la ecuación (4.1), puede ser vista como una red neuronal con una capa oculta, en la que la función de activación es la función valor absoluto. También puede entenderse que las funciones base de la red tienen la forma de hiperplanos bisagra como los mostrados en la Figura 4.1. Batruni ha desarrollado un método de entrenamiento mediante retropropagación del error para redes multicapa formadas a partir de estas secciones. Aunque en principio Batruni empleó este tipo de redes sin justificación teórica de sus capacidades, esta justificación se ha desarrollado con posterioridad [Lin93, Lin95].

4.2.2.3 El MLT como aproximador universal

En [Lin93, Lin95], se demuestra que una red del tipo MLP de dos capas, que utiliza como función de activación la función umbral lógico, es decir

$$\sigma(x) = \begin{cases} 0, & x < 0 \\ x, & 0 \leq x < 1 \\ 1, & x \geq 1 \end{cases} \quad (4.3)$$

es una función canónica lineal a tramos que admite la representación (4.1). A partir de este resultado, y de la probada capacidad de aproximación de este tipo de redes [Hornik89], se demuestra que una red neuronal canónica lineal a tramos básica, es decir, que admite la representación (4.1), es capaz de aproximar cualquier función continua definida en un conjunto compacto de R^N con una precisión arbitraria. Así pues, se tiene que este tipo de redes es un aproximador universal de funciones continuas, aunque no sea un *representante universal* de todas las funciones lineales a tramos (una función

lineal a tramos sólo admite una representación canónica si la partición del espacio de entrada es llevada a cabo mediante hiperplanos, ver [Chua88] para más detalles).

Estas demostraciones justifican a nivel teórico la capacidad de las redes neuronales canónicas lineales a tramos. Asimismo, se han propuesto redes canónicas lineales a tramos generalizadas, que se diferenciarían por sus funciones de activación (todas ellas lineales a tramos). De entre todas las posibilidades, en [Lin93] se llega a la conclusión de que la más ventajosa es la función valor absoluto. Esto se debe a que es una función lineal a tramos con un único punto de ruptura. Desde un punto de vista geométrico, un punto de ruptura en la función de activación supone una frontera lineal en el espacio de entrada, mientras que un número mayor de puntos de ruptura (como por ejemplo la función umbral lógico, que presenta dos) supone introducir fronteras paralelas en la partición del espacio de entrada y, desde el punto de vista de aproximación funcional, la partición más eficiente del dominio de entrada no es necesariamente la llevada a cabo por grupos de fronteras paralelas. Teniendo en cuenta que esta situación es un caso particular de partición, que se puede incluir en las obtenidas con fronteras individuales, la red neuronal lineal a tramos con una función de activación valor absoluto, juega el papel más general y flexible.

Otro aspecto interesante es que, para las redes neuronales más habituales (MLP's con una función de activación sigmoideal), teniendo en cuenta que la función sigmoideal puede verse como una versión suavizada de la función umbral lógico (4.3), su comportamiento en aproximación funcional puede verse como una aproximación canónica lineal a tramos suavizada, que sería suave en las fronteras que particionan el dominio. Esta es una de las razones que invitan a la construcción de redes neuronales lineales a tramos suavizadas. Además, este tipo de redes tendrían en principio algunas ventajas sobre los tradicionales MLP's: como Lin y Unbehauen apuntan en [Lin93], el aprendizaje de las redes canónicas lineales a tramos es, en general, más rápido que el de los MLP's. Esto se debe a la naturaleza de sus relativas funciones de activación: la parte plana de la función sigmoideal hace que normalmente el aprendizaje sea lento. Por otro lado, también hay que tener en cuenta que la naturaleza no monótona de la función valor absoluto puede llevar a superficies de error más complejas, con lo que aumentaría en este caso la probabilidad de caer en mínimos locales para las redes canónicas lineales a tramos.

4.3 Suavizado de la función valor absoluto

Como ya se ha mencionado en la introducción, una de las ventajas de los modelos lineales a tramos es su sencillez, y el reducido número de parámetros que generalmente requieren para su representación, en particular para la representación canónica. Por otro lado, tal vez el principal inconveniente que presentan sea la falta de derivabilidad, que en ciertas aplicaciones, como por ejemplo el modelado de dispositivos, es un requisito importante. El modelo que se propone en este capítulo intenta solucionar esta falta de derivabilidad sin incrementar el número de parámetros requeridos para la represen-

tación del modelo. Esto se consigue sustituyendo, en la expresión del modelo canónico lineal a tramos (4.1), la función valor absoluto, responsable de la naturaleza abrupta y no derivable de las transiciones entre regiones lineales, por una función derivable, de tal forma que se suavizan las transiciones entre las diferentes regiones definidas por la partición del espacio de entrada, haciendo el modelo global suave y derivable.

Existen varias posibilidades para llevar a cabo el suavizado de transiciones entre regiones lineales. Para ilustrar este punto, vamos a considerar una transición genérica, es decir, una función $f : R \rightarrow R$ compuesta por dos secciones lineales separadas en un punto de ruptura α , que admite la siguiente representación

$$f(x) = \theta_0 + \theta_1(x - \alpha) + \theta_2|x - \alpha|. \quad (4.4)$$

De forma equivalente, (4.4) se puede reescribir de la forma siguiente utilizando la función signo (sgn)

$$f(x) = \theta_0 + \theta_1(x - \alpha) + \theta_2(x - \alpha)\text{sgn}(x - \alpha). \quad (4.5)$$

Entre las diferentes alternativas para llevar a cabo el “suavizado” de esta transición, parece claramente deseable poder disponer de una familia de funciones que, a la vez que proporcionan una transición suave, permitan un control paramétrico de la “suavidad” de la transición. Existen varias de estas funciones en la bibliografía; por ejemplo, en [Bacon71] se propone reemplazar la función $\text{sgn}(x)$ por una familia de funciones, que podemos denotar como $\text{trn}(x)$, a las que se exige el cumplimiento de las siguientes restricciones

$$\begin{aligned} (i) \quad & \lim_{x \rightarrow \pm\infty} (x \text{trn}(x) - |x|) = 0 \\ (ii) \quad & \lim_{\gamma \rightarrow \infty} \text{trn}(\gamma x) = \text{sgn}(x) \\ (iii) \quad & \text{trn}(0) = \text{sgn}(0) = 0 \end{aligned} \quad (4.6)$$

La primera condición asegura que $x \text{trn}(x)$ aproxima $|x| = x \text{sgn}(x)$ con una precisión arbitraria cuando $x \rightarrow \infty$. La segunda asegura que el modelo lineal a tramos (4.5) es una forma límite del modelo suavizado; es decir, que cuando $\gamma \rightarrow \infty$, esta familia de funciones tiende a un modelo lineal a tramos. Finalmente, la tercera de las condiciones asegura que el modelo suavizado pasa a través del punto (α, θ_0) : la intersección entre las dos líneas (que son las asíntotas de la curva).

Estas familias de funciones incluyen un parámetro, γ , que controla la suavidad de la transición: $\text{trn}(\gamma x)$. Cambiando este parámetro se controla la suavidad alrededor del punto de ruptura α . Entre las diferentes familias que cumplen las tres condiciones (4.6), se pueden encontrar las siguientes [Seber89]

$$\text{ftan}(x, \gamma) = x \tanh(\gamma x) \quad (4.7)$$

y

$$\text{fpol}(x, \gamma) = \begin{cases} \frac{1}{8\gamma}[15(\gamma x)^2 - 10(\gamma x)^4 + 3(\gamma x)^6] & |x| \leq \frac{1}{\gamma}, \\ |x|, & |x| > \frac{1}{\gamma} \end{cases} \quad (4.8)$$

Un parámetro geométrico que puede dar una idea acerca del comportamiento de estas funciones es el radio de curvatura, que está dado por la siguiente expresión

$$K = \frac{f''(x)}{(1 + (f'(x))^2)^{3/2}} \quad (4.9)$$

Si se calcula el valor del radio de curvatura para estas dos funciones se obtienen los valores que se muestran en la Tabla 4.1. Es sencillo comprobar que, para valores altos del radio de curvatura, K , las funciones tienden de forma más rápida a la función valor absoluto.

Un aspecto a tener en cuenta es que para cumplir las condiciones (4.6), todas estas funciones presentan derivadas primeras que alcanzan valores mayores en módulo que la unidad y derivadas segundas que alcanzan algunos valores negativos. Por lo tanto, estas derivadas presentan oscilaciones que dificultan la tarea de obtener derivadas suaves. La Figura 4.4 muestra las derivadas de las funciones $\text{ftan}(x, \gamma)$ y $\text{fpol}(x, \gamma)$ para un radio de curvatura de valor unidad, y en ella se puede apreciar este comportamiento.

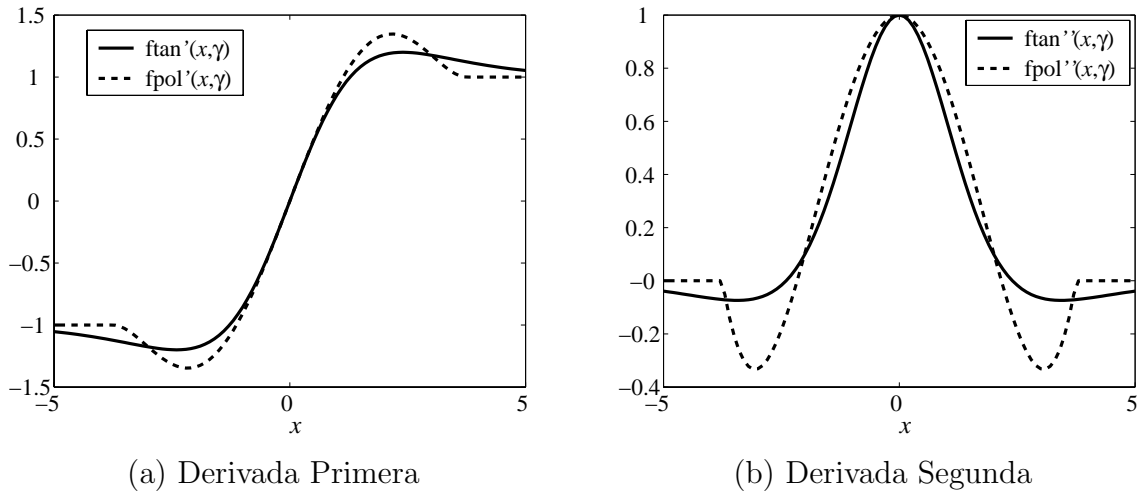


Figura 4.4: Derivadas de las funciones $\text{ftan}(x, \gamma)$ y $\text{fpol}(x, \gamma)$

Para evitar este problema es preciso emplear otro tipo de funciones: por ejemplo, Griffiths y Miller [Griffiths73], decidieron relajar la tercera condición, (4.6-iii), y propusieron emplear la función

$$\text{hyp}(x, \gamma) = \sqrt{x^2 + \gamma} \quad (4.10)$$

La función que se propone en esta Tesis para reemplazar a la función $|x|$ es la siguiente:

$$\text{lch}(x, \gamma) = \frac{1}{\gamma} \ln(\cosh(\gamma x)) \quad (4.11)$$

cuyo radio de curvatura, junto con el de las demás funciones de suavizado presentadas, aparece en la Tabla 4.1.

Función	Radio de Curvatura (K)
$\text{ftan}(x, \gamma)$	2γ
$\text{fpol}(x, \gamma)$	$\frac{15}{4}\gamma$
$\text{hyp}(x, \gamma)$	$\frac{1}{\sqrt{\gamma}}$
$\text{lch}(x, \gamma)$	γ

Tabla 4.1: Radios de curvatura de las funciones de suavizado

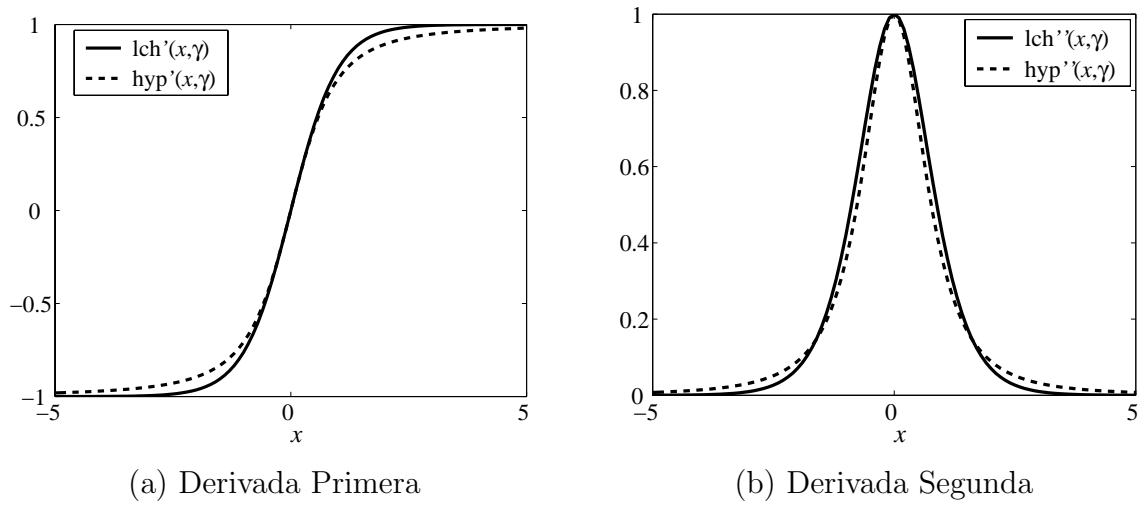
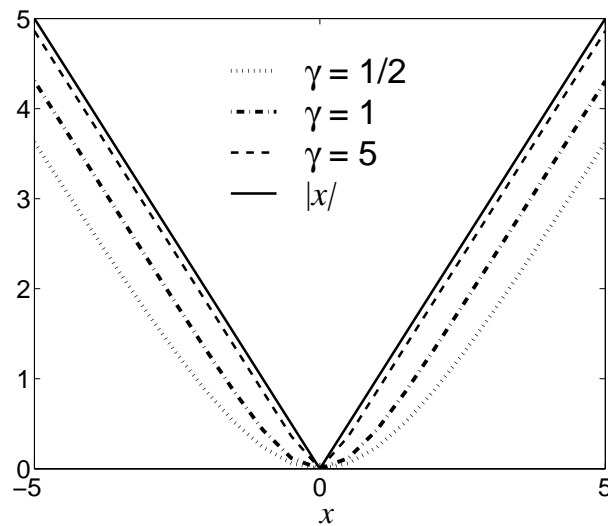
En este caso, la función propuesta, presentada en [Pantaleón94] para suavizar este tipo de transiciones, cumple la tercera de las condiciones, pero relaja la primera. Para esta función se cumple

$$\lim_{x \rightarrow \pm\infty} \left(\frac{1}{\gamma} \ln(\cosh(\gamma x) - |x|) \right) = -\frac{1}{\gamma} \ln(2) \quad (4.12)$$

La Figura 4.5 muestra las derivadas las funciones $\text{hyp}(x, \gamma)$ y $\text{lch}(x, \gamma)$. Se puede observar que las derivadas de estas funciones ya no presentan oscilaciones como las funciones ftan y fpol . Además, es sencillo comprobar que, para un mismo radio de curvatura, estas dos funciones son más suaves, es decir, la derivada segunda al cuadrado tiene menor área, que las funciones que cumplen las tres condiciones (4.6).

Asimismo, se ha comprobado que, para los problemas de modelado considerados en esta Tesis, estas funciones obtienen los mejores resultados. Finalmente, se ha tomado la decisión de emplear la función $\text{lch}(x, \gamma)$ por dos razones. Por un lado, su derivada es la función $\text{lch}'(x, \gamma) = \tanh(\gamma x)$, que es la función de activación de un aproximador universal como es el perceptrón multicapa, lo que resulta conveniente cuando se desea realizar un ajuste de las derivadas. La otra razón es la conveniencia notacional: resulta mucho más sencillo trabajar con las expresiones asociadas a esta función y a sus derivadas que con las obtenidas para la función $\text{hyp}(x, \gamma)$.

Así pues, esta va a ser la función empleada en el modelo lineal a tramos suavizado (MLTS) para sustituir a la función $|x|$. La Figura 4.6 muestra la forma de esta función para diferentes valores del parámetro de suavidad γ . A medida que aumenta el valor de γ , la función $\text{lch}(x, \gamma)$ tiende a la función valor absoluto, mientras que para valores más pequeños la curvatura en el origen es menor. Por otra parte, cuando $x \rightarrow \pm\infty$, la separación con respecto a la función valor absoluto es mayor, aunque la pendiente es la misma y, por lo tanto, su derivada coincide.

Figura 4.5: Derivadas de las funciones $lch(x, \gamma)$ y $hyp(x, \gamma)$ Figura 4.6: Función $lch(x, \gamma)$

4.4 El Modelo Lineal a Tramos Suavizado (MLTS)

El Modelo Lineal a Tramos Suavizado (MLTS) es una extensión del MLT presentado en la Sección 4.2, construido sustituyendo la función $|x|$ por la función de suavizado $\text{lch}(x, \gamma)$. De esta forma, el modelo MLTS aproxima una función $f : R^M \rightarrow R^N$ de la forma siguiente

$$f(\mathbf{x}) = \mathbf{a} + \mathbf{B}\mathbf{x} + \sum_{i=1}^{\theta} \mathbf{c}_i \text{lch}(\langle \boldsymbol{\alpha}_i, \mathbf{x} \rangle - \beta_i, \gamma_i) \quad (4.13)$$

donde \mathbf{x} y $\boldsymbol{\alpha}_i$ son vectores de la misma dimensión, M , que el espacio de entrada, \mathbf{a} y \mathbf{c}_i son vectores de dimensión N , \mathbf{B} es una matriz $N \times M$, β_i es un escalar y \langle, \rangle representa el producto escalar. Ahora, la transición en la frontera i -ésima, que divide el espacio de entrada en las distintas regiones, está gobernada por la función $\text{lch}(\langle \boldsymbol{\alpha}_i, \mathbf{x} \rangle - \beta_i, \gamma_i)$, que es una función suave y derivable, de tal forma que el modelo global está dotado también con estas propiedades.

Cada uno de los términos que aparecen ahora en el sumatorio tienen la forma de un hiperplano bisagra cuya transición se ha suavizado mediante la función $\text{lch}(x, \gamma)$. En la Figura 4.7 se muestra una de estas funciones en un espacio de entrada bidimensional, donde las fronteras dividiendo el dominio son rectas.

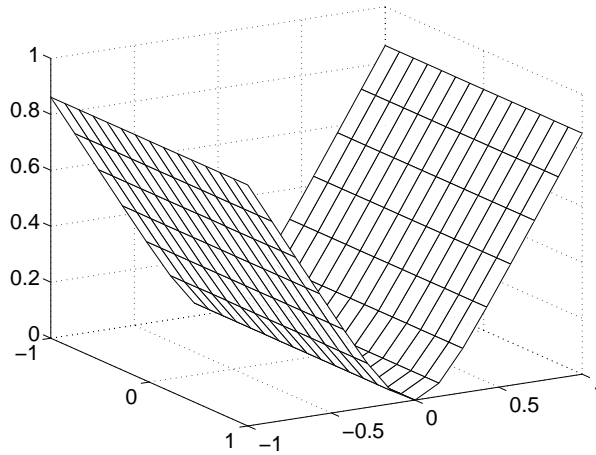


Figura 4.7: Función base para un espacio de entrada bidimensional

Si este modelo se interpreta como una red neuronal, la función de activación sería la función $\text{lch}(x, \gamma)$, y los hiperplanos bisagra suavizados deben ser interpretados como las funciones base del modelo neuronal.

4.4.1 Análisis de suavidad: relación con la teoría de la regularización

En esta sección se estudia la suavidad del modelo global demostrándose que tanto para problemas unidimensionales como bidimensionales, es una función monótona con el parámetro γ . Se ha considerado un modelo simplificado en el que se utiliza el mismo valor de γ para todos los componentes del modelo. Así, el parámetro γ actúa de forma similar a un parámetro de regularización, permitiendo establecer un compromiso entre la fidelidad o exactitud del modelo con respecto a las medidas de la función, y la suavidad del modelo (relacionada con la información de las derivadas).

Con estas premisas, se puede apuntar un enlace entre el modelo MLTS y la teoría de la regularización, cuyos principios generales se han descrito en la Sección 2.4.1. Concretamente, se ha demostrado que la relación entre la suavidad, evaluada por medio de integrales de las derivadas de segundo orden, y el parámetro de suavidad, γ , es monótona y que, por lo tanto, es posible controlar la suavidad del modelo mediante una selección adecuada de este parámetro. De esta forma, una adecuada selección de γ y la posterior minimización del error cuadrático mediante el algoritmo de entrenamiento del modelo MLTS, tiene cierta similitud con la minimización de un funcional regularizado como (2.92).

4.4.1.1 Análisis para un espacio de entrada unidimensional

En un espacio de entrada unidimensional, la expresión de los estabilizadores dados por la ecuación (2.93) es, simplemente, la derivada segunda al cuadrado. Por otro lado, la derivada segunda de la función de suavizado, $\text{lch}(x, \gamma)$, tiene la forma

$$\text{lch}''(x, \gamma) = \gamma \text{sech}^2(\gamma x) \quad (4.14)$$

Intuitivamente se ve que, en el caso de una única componente, la suavidad es monótona con γ , lo que se confirma a partir de la medida del radio de curvatura de la función en el origen: $K = \gamma$. Vamos a comprobar esta intuición midiendo la suavidad mediante el estabilizador, que en este caso toma el valor

$$\|Pf\|^2 = \int_R (\gamma \text{sech}^2(\gamma x))^2 dx = \gamma \left[\frac{\text{sech}(\gamma x)}{3 \cosh^3(\gamma x)} + \frac{2 \text{sech}(\gamma x)}{3 \cosh(\gamma x)} \right]_{-\infty}^{\infty} = \frac{4}{3} \gamma, \quad (4.15)$$

lo que confirma que para una única componente la suavidad es monótona con γ .

Estudiamos a continuación el caso en el que hay varias componentes. La derivada segunda de cada componente, (4.14), es una función localizada. Teniendo esto en cuenta, para simplificar el estudio podemos asumir que cada componente solamente se solapa con el componente más cercano. De esta forma, y aprovechando la propiedad de invariancia respecto a la traslación del estabilizador, sin perder generalidad podemos asumir que el modelo está compuesto por dos componentes, separados una distancia b

$$f(x) = c_1 \text{lch}(x, \gamma) + c_2 \text{lch}(x - b, \gamma), \quad (4.16)$$

con lo que el término de regularización vendrá dado por

$$\begin{aligned} \|Pf\|^2 &= \int_{\mathbb{R}} (c_1 \gamma \text{sech}^2(\gamma x))^2 dx_1 + \int_{\mathbb{R}} (c_2 \gamma \text{sech}^2(\gamma(x - b)))^2 dx_2 \\ &+ 2 \int_{\mathbb{R}} (c_1 c_2 \gamma^2 \text{sech}^2(\gamma x))^2 \text{sech}^2(\gamma(x - b)) dx_1 dx_2. \end{aligned} \quad (4.17)$$

Los dos primeros términos de la derecha corresponden a la contribución de cada componente por separado, mientras que el tercero corresponde a la intersección de ambas componentes. Resolviendo las integrales se llega a la siguiente expresión

$$\|Pf\|^2 = \frac{4}{3}(c_1^2 + c_2^2)\gamma + 8c_1 c_2 \gamma \text{cosech}^2(\gamma b) \left[\frac{\gamma b}{\tanh(\gamma b)} - 1 \right] \quad (4.18)$$

En la Figura 4.8 se muestra el valor de este estabilizador en función del parámetro γ y de la distancia b , para unos valores $c_1 = c_2 = 1$, observándose su comportamiento monótono con γ . El mismo comportamiento se puede observar para cualquier valor de los coeficientes de ponderación c_1 y c_2 .

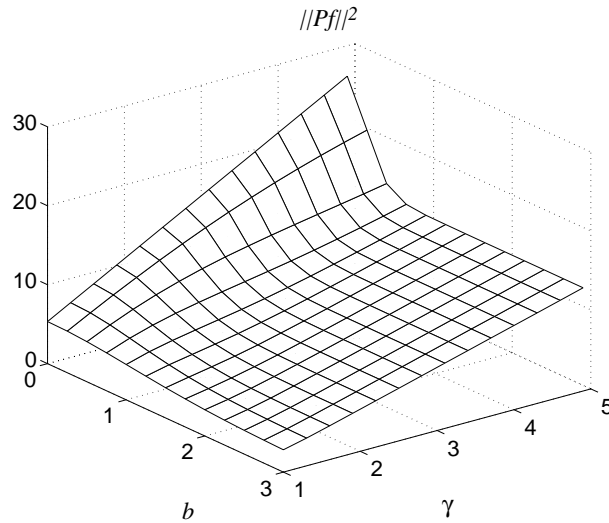


Figura 4.8: Estabilizador

4.4.1.2 Análisis para un espacio de entrada bidimensional

Para extender el análisis a un espacio de entrada bidimensional, surge el problema de que la derivada segunda de las funciones base no está localizada en la dirección del espacio que definen las rectas que dividen el dominio, con lo que no tiene un soporte finito en esa dirección. Sin embargo, este problema se puede resolver realizando la

integral sobre una región finita adecuada. Hay que tener en cuenta que en un problema real de modelado, se trabaja habitualmente sobre una región finita del espacio de entrada, que es donde interesa conocer la suavidad del modelo. Para simplificar en lo posible la evaluación de las integrales, la región de integración se delimita usando rectas perpendiculares a las rectas que definen las fronteras: la Figura 4.9 muestra un ejemplo de la forma de esta región de integración.

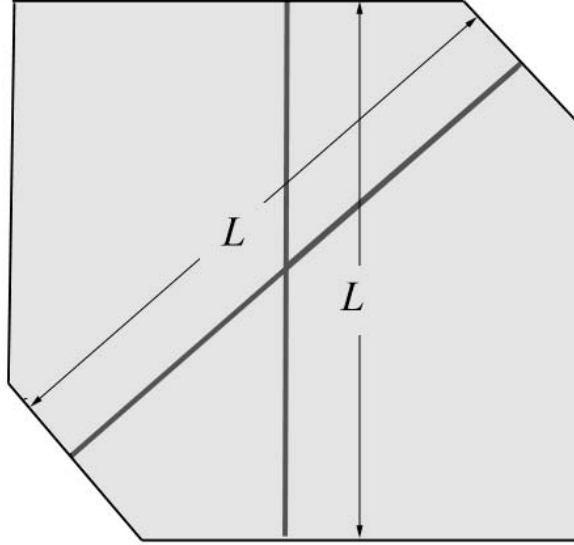


Figura 4.9: Ejemplo de región de integración en dominios bidimensionales

Teniendo esto en cuenta, si denotamos mediante R' a la región de integración así definida, el estabilizador para un espacio de entrada bidimensional tiene la forma siguiente

$$\|P^2 f\|^2 = \int_{R'} \left[\left(\frac{\partial^2 f}{\partial x_1^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x_1 \partial x_2} \right)^2 + \left(\frac{\partial^2 f}{\partial x_2^2} \right)^2 \right] dx_1 dx_2. \quad (4.19)$$

Para un dominio bidimensional, el modelo MLTS toma la expresión

$$f(x) = \sum_{i=1}^{\theta} c_i \text{lch}(\alpha_i x_1 - x_2 + \beta_i, \gamma) \quad (4.20)$$

donde, considerando la expresión (4.13), se toma $\alpha_i = [\alpha_i, -1]$, para minimizar el número de parámetros del modelo. De esta forma, se emplea la expresión más sencilla de una recta para particionar el espacio de entrada.

Si, al igual que en el caso de un espacio unidimensional, se consideran sólo dos componentes solapadas, cuyos parámetros son $\alpha_1 = \beta_1 = 0$ y $\alpha_2 = \alpha$ y $\beta_2 = \beta$ respectivamente, entonces, el término de regularización se calcula como

$$\begin{aligned} \|P^2 f\|^2 &= \int_{R'} c_1^2 \gamma^2 \operatorname{sech}^4(-\gamma x_2) dx_1 + \int_{R'} c_2^2 (\alpha^2 + 1)^2 \gamma^2 \operatorname{sech}^4(\gamma(\alpha x_1 - x_2 + \beta)) dx_2 \\ &\quad + 2 \int_R (c_1 c_2 \gamma^2 \operatorname{sech}^2(-\gamma x_1))^2 \operatorname{sech}^2(\gamma(\alpha x_1 - x_2 + \beta)) dx_1 dx_2 \end{aligned} \quad (4.21)$$

De nuevo, los dos primeros términos de (4.21), corresponden a la contribución de cada componente por separado, mientras que el tercer término corresponde a la intersección de ambas componentes. En cuanto al segundo término, éste puede ser integrado sobre todo el espacio, ya que la intersección de dos $\operatorname{sech}^2(x)$ es finita. Teniendo esto en cuenta se obtiene

$$\|P^2 f\|^2 = \frac{4}{3} \gamma (c_1^2 + c_2^2 (\alpha^2 + 1)^2) L + \frac{4}{\alpha} c_1 c_2 \quad (4.22)$$

donde L es la longitud a lo largo de la rectas que definen las fronteras del dominio sobre las que se realiza la integral (ver Figura 4.9). Nuevamente, se demuestra que la dependencia del término de regularización con γ es monótona. Incrementando γ la transición entre las regiones lineales se vuelve más abrupta, y la suavidad del modelo se reduce.

La extensión de este resultado a espacios de entrada de mayor dimensión es algo compleja y no se ha perseguido en este trabajo. No obstante, parece razonable asumir el mismo comportamiento con respecto al parámetro de suavidad, γ , que se ha observado en espacios de una y de dos dimensiones.

Como conclusión de esta sección, se ha mostrado que mediante el parámetro γ se puede controlar la suavidad del modelo MLTS, lo que hace que γ se pueda ver como un parámetro de regularización. De este modo, minimizar el error cuadrático por un lado, junto con una adecuada selección del parámetro γ , de acuerdo con algunas restricciones de suavidad, por ejemplo restricciones en las derivadas, sería equivalente, en cierto modo, a minimizar un funcional regularizado de la forma (2.92).

4.4.2 Entrenamiento del modelo

En esta sección se describe el algoritmo de entrenamiento del modelo MLTS, que es una generalización del utilizado por el modelo canónico lineal al tramos [Chua86]. Hay que tener en cuenta que el modelo MLTS tiene tres tipos claramente diferenciados de parámetros:

- Los parámetros que definen la fronteras que dividen el espacio de entrada del sistema, que en (4.13) son α_i y β_i .
- Los parámetros que definen la combinación lineal de los componentes del modelo: \mathbf{a} , \mathbf{B} , y \mathbf{c}_i .
- El parámetro de suavidad γ

El algoritmo de entrenamiento del modelo MLTS es un algoritmo iterativo basado en la sucesiva adaptación de la partición del espacio de entrada, es decir, de las fronteras, y la estima de los coeficientes óptimos que definen la combinación lineal de los componentes del modelo para esa partición. El proceso de adaptación de los parámetros que definen las fronteras del dominio se lleva a cabo mediante un método de gradiente de segundo orden, esto es, se calculan el gradiente y el Hessiano de la función error con respecto a los parámetros, y, a partir de ellos, se determina la dirección en la que han de modificarse dichos parámetros. Una vez que las fronteras del dominio han sido fijadas, el error de la aproximación es una función cuadrática de los parámetros que definen la combinación lineal de los coeficientes, y el mínimo de la misma se puede encontrar fácilmente resolviendo un problema de mínimos cuadrados. Entonces, las fronteras del dominio son adaptadas de nuevo y el mismo proceso se repite de forma iterativa hasta que el procedimiento converge. Esta es, básicamente, la idea empleada por Chua para optimizar los parámetros del modelo MLT [Chua86]. En cuanto a la adaptación del parámetro de suavidad, γ , existen varias posibilidades que se describen en detalle con posterioridad. A continuación se detallan las expresiones de este algoritmo de entrenamiento.

4.4.2.1 Estima de la fronteras del dominio y de los coeficientes de la combinación lineal

Para simplificar en lo posible la notación, se considera la aproximación de una función $g : R^M \rightarrow R$, utilizando un conjunto de N muestras entrada-salida, $(\mathbf{x}_j, \mathbf{y}_j)$, $j = 1, \dots, N$, con $\mathbf{x}_j = (x_{1,j}, x_{2,j}, \dots, x_{M,j})^T$. No obstante, la extensión a un espacio de salida de mayor dimensión es inmediata.

Asumiendo que $\alpha_{i,M} \neq 0$, es posible eliminar uno de los coeficientes de cada frontera reescribiendo $\langle \boldsymbol{\alpha}_i, \mathbf{x} \rangle - \beta_i$ como

$$d_i(\mathbf{x}) = \alpha_{i,1}x_1 + \alpha_{i,2}x_2 + \dots + \alpha_{i,M-1}x_{M-1} - x_M + \beta_i, \quad (4.23)$$

donde $d_i(\mathbf{x})$ es la i -ésima frontera evaluada en \mathbf{x} . Finalmente, teniendo en cuenta que \mathbf{B} y \mathbf{a} en (4.13) son ahora un vector $\mathbf{b} = (b_1, \dots, b_M)^T$ y un escalar a , respectivamente, el modelo MLTS genérico, con θ fronteras, se puede escribir como

$$f(\mathbf{x}) = a + \mathbf{b}^T \mathbf{x} + \sum_{i=1}^{\theta} c_i \text{lch}(d_i(\mathbf{x}), \gamma). \quad (4.24)$$

Los parámetros del modelo se pueden agrupar en dos vectores: \mathbf{z}_p , que incluye los coeficientes asociados a la combinación lineal de componentes del modelo,

$$\mathbf{z}_p = (a, b_1, \dots, b_M, c_1, \dots, c_\theta)^T, \quad (4.25)$$

y \mathbf{z}_r , que agrupa los parámetros que definen las fronteras del dominio

$$\mathbf{z}_r = (\alpha_{1,1}, \dots, \alpha_{1,M-1}, \dots, \alpha_{\theta,1}, \dots, \alpha_{\theta,M-1}, \beta_1, \dots, \beta_\theta)^T. \quad (4.26)$$

La función de coste a minimizar es el error cuadrático, dado por

$$E(\mathbf{z}_p, \mathbf{z}_r) = \sum_{j=1}^N \left(y_j - \left(a + \mathbf{b}^T \mathbf{x}_j + \sum_{i=1}^{\theta} c_i \text{lch}(d_i(\mathbf{x}_j), \gamma) \right) \right)^2 \quad (4.27)$$

Con estas premisas, el algoritmo de adaptación comienza fijando una distribución inicial de las fronteras, es decir, el vector \mathbf{z}_r . Generalmente, esta partición inicial se hace de forma aleatoria, aunque es usual imponer la restricción de que las fronteras pertenezcan al dominio en el que aparecen los datos. Una vez fijado \mathbf{z}_r , la función error de la aproximación, $E(\mathbf{z}_p, \mathbf{z}_r)$, es una función cuadrática en \mathbf{z}_p , y su mínimo se puede encontrar mediante la solución de mínimos cuadrados

$$\mathbf{z}_p = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y} \quad (4.28)$$

donde $\mathbf{y} = (y_1, \dots, y_N)^T$, y \mathbf{A} es la siguiente matriz de tamaño $(M + \theta + 1) \times N$

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_{1,1} & x_{1,2} & \cdots & x_{1,N} \\ \vdots & \vdots & \vdots & \vdots \\ x_{M,1} & x_{M,2} & \cdots & x_{M,N} \\ u_{1,1} & u_{1,2} & \cdots & u_{1,N} \\ \vdots & \vdots & \vdots & \vdots \\ u_{\theta,1} & u_{\theta,2} & \cdots & u_{\theta,N} \end{bmatrix}, \quad (4.29)$$

siendo $u_{i,j} = \text{lch}(d_i(\mathbf{x}_j), \gamma)$.

Una vez que los parámetros \mathbf{z}_p óptimos (para la partición \mathbf{z}_r inicial dada) han sido calculados, el algoritmo estima una nueva partición \mathbf{z}_r . Para calcular la nueva partición se calculan el gradiente, \mathbf{g} , y el Hessiano, \mathbf{Y} , a partir de los cuales se determina la dirección óptima para modificar \mathbf{z}_r , que es

$$\mathbf{s} = -\mathbf{Y}^{-1} \mathbf{g}. \quad (4.30)$$

El gradiente y el Hessiano se pueden calcular de la forma siguiente

$$\mathbf{g} = 2\mathbf{K}\mathbf{G}\mathbf{e}, \quad (4.31)$$

y

$$\mathbf{Y} = 2\mathbf{K}\mathbf{G}\mathbf{G}^T \mathbf{K} + 2\mathbf{K} \frac{\partial \mathbf{G}}{\partial \mathbf{z}_r} \mathbf{e}, \quad (4.32)$$

donde $\mathbf{e} = (e_1, \dots, e_N)^T$ es el vector de errores,

$$\mathbf{K} = \text{diag}(\underbrace{c_1, \dots, c_1}_{M-1 \text{ veces}}, \underbrace{c_2, \dots, c_2}_{M-1 \text{ veces}}, \dots, \underbrace{c_\theta, \dots, c_\theta}_{M-1 \text{ veces}}, c_1, c_2, \dots, c_\theta), \quad (4.33)$$

y \mathbf{G} es la matriz

$$\mathbf{G} = \begin{bmatrix} x_{1,1}p_{1,1} & x_{1,2}p_{1,2} & \cdots & x_{1,N}p_{1,N} \\ \vdots & \vdots & \vdots & \vdots \\ x_{M-1,1}p_{1,1} & x_{M-1,2}p_{1,2} & \cdots & x_{M-1,N}p_{1,N} \\ \vdots & \vdots & \vdots & \vdots \\ x_{1,1}p_{\theta,1} & x_{1,2}p_{\theta,2} & \cdots & x_{1,N}p_{\theta,N} \\ \vdots & \vdots & \vdots & \vdots \\ x_{M-1,1}p_{\theta,1} & x_{M-1,2}p_{\theta,2} & \cdots & x_{M-1,N}p_{\theta,N} \\ p_{1,1} & p_{1,2} & \cdots & p_{1,N} \\ \vdots & \vdots & \vdots & \vdots \\ p_{\theta,1} & p_{\theta,2} & \cdots & p_{\theta,N} \end{bmatrix}, \quad (4.34)$$

donde $p_{i,j} = \tanh(\gamma d_i(\mathbf{x}_j))$

El segundo término del Hessiano en (4.32) incluye la derivada segunda de las funciones base del modelo MLTS, que vale $\text{sech}^2(d_i(\mathbf{x}_j))$. Esta es una función localizada sobre las fronteras del dominio: sólo los puntos cercanos a las fronteras contribuyen a este término. En la práctica se ha observado que se puede conseguir un sensible ahorro computacional, sin apreciar una degradación sensible en las prestaciones del algoritmo, eliminado este término en la evaluación del Hessiano, es decir, calculando

$$\mathbf{Y} = 2\mathbf{K}\mathbf{G}\mathbf{G}^T\mathbf{K}. \quad (4.35)$$

Una vez que se ha calculado la dirección de búsqueda (4.30), las nuevas fronteras se estiman como

$$\mathbf{z}_r = \mathbf{z}_r + \mu\mathbf{s}, \quad (4.36)$$

donde $\mu = \text{argmin}(E(\mathbf{z}_p, \mathbf{z}_r + \mu\mathbf{s}))$. Con esta nueva partición del dominio, el proceso se repite: los coeficientes óptimos \mathbf{z}_p para esas fronteras son calculados y la partición se vuelve a adaptar. El proceso se repite hasta que un determinado nivel de error, que se considere admisible, es alcanzado o cuando se sobrepasa un número máximo de iteraciones.

4.4.2.2 Estima del parámetro de suavidad

Para la estima del parámetro de suavidad, γ , existen varias estrategias posibles. Hay que distinguir dos situaciones diferentes: cuando sólo se dispone de muestras de la función, y cuando además se dispone de muestras de las derivadas. Cuando sólo se dispone de muestras de la función, la alternativa más directa sería la de seleccionar γ para minimizar (4.27). El valor óptimo se puede obtener entonces mediante un algoritmo de descenso de gradiente

$$\gamma_{k+1} = \gamma_k - \mu \frac{\partial E(\mathbf{z}_p, \mathbf{z}_r)}{\partial \gamma}. \quad (4.37)$$

Es posible utilizar una γ diferente para cada frontera. Si bien esto puede suponer alguna mejora en los resultados obtenidos, hay que tener en cuenta que, por otro lado, supone incrementar el número de parámetros totales del modelo, con lo que hay que considerar si la ganancia conseguida sobre los resultados obtenidos empleando un único valor de γ compensa el incremento en el número de parámetros del modelo.

Otra alternativa consiste en fijar un valor para el parámetro, lo que supone fijar, de algún modo, la suavidad del modelo. Esta alternativa puede entenderse como una técnica de regularización, tal y como se ha mostrado en la Sección 4.4.1. En cuanto a la selección del valor de γ , en principio se buscará una solución lo más suave posible, sin que se degrade de forma severa la aproximación de las muestras de la función. Esto permitirá una aproximación adecuada de las derivadas.

Cuando se dispone de muestras de las derivadas, teniendo en cuenta que el parámetro γ controla la suavidad del modelo y que en general la suavidad está relacionada con las derivadas del mismo, una alternativa interesante consiste en buscar el valor óptimo de γ de acuerdo a esta información. Es posible en ese caso adaptar el resto de parámetros para minimizar (4.27) y adaptar γ para minimizar el error sobre alguna de las derivadas, por ejemplo, mediante un método de gradiente. Si, por ejemplo, se dispone de medidas de la derivada primera del sistema respecto del l -ésimo parámetro de entrada, que podemos denotar como $\mathbf{y}_{(l)}' = (y'_{(l),1}, \dots, y'_{(l),N})^T$, entonces γ se puede seleccionar para minimizar el error en la estima de esa derivada

$$E'_{(l)}(\mathbf{z}_p, \mathbf{z}_r) = \sum_{j=1}^N \left(y'_{(l),j} - \left(b_l x_{l,j} + \sum_{i=1}^{\theta} c_i m_{i,l} \tanh(\gamma d_i(\mathbf{x}_j)) \right) \right)^2, \quad (4.38)$$

donde los paréntesis interiores corresponden a la derivada del modelo MLTS con respecto de la entrada l -ésima. En este caso, el parámetro γ se obtendría aplicando la siguiente expresión

$$\gamma_{k+1} = \gamma_k + \mu \sum_{j=1}^N e'_{(l),j} \left(\sum_{i=1}^{\theta} c_i m_{i,l} b_i(\mathbf{x}_j) \operatorname{sech}^2(\gamma b_i(\mathbf{x}_j)) \right), \quad (4.39)$$

donde

$$e'_{(l),j} = \left(y'_{(l),j} - \left(b_l x_{l,j} + \sum_{i=1}^{\theta} c_i m_{i,l} \tanh(\gamma d_i(\mathbf{x}_j)) \right) \right), \quad (4.40)$$

es el error de la derivada en el punto j , y μ es el parámetro de paso.

De nuevo, un valor diferente de γ se puede emplear para cada frontera desacoplando (4.39), y naturalmente se puede tratar de minimizar el error sobre la derivada con respecto a una única dirección del espacio de entrada o con respecto a varias (la extensión en este caso es obvia).

El mismo planteamiento se puede llevar a cabo cuando se dispone de muestras de la(s) derivada(s) segunda(s), incluso con una mayor justificación intuitiva, ya que la

suavidad se acostumbra a evaluar mediante funcionales que incluyen derivadas segundas. En este caso, si se dispone de las medidas de la derivada segunda con respecto al parámetro de entrada l -ésimo, que se denotaría como $\mathbf{y}_{(l)}'' = (y'_{(l),1}, \dots, y'_{(l),N})^T$, γ se puede seleccionar para minimizar el error en la estima de esa derivada

$$E''_{(l)}(\mathbf{z}_p, \mathbf{z}_r) = \sum_{j=1}^N \left(y''_{(l),j} - \left(b_l + \sum_{i=1}^{\theta} c_i m_{i,l}^2 \gamma \operatorname{sech}^2(\gamma d_i(\mathbf{x}_j)) \right) \right)^2, \quad (4.41)$$

donde ahora los paréntesis interiores corresponden a la derivada segunda del modelo MLTS con respecto de la entrada l -ésima. En este caso, el parámetro γ se obtendría mediante

$$\begin{aligned} \gamma_{k+1} = \gamma_k - \mu \sum_{j=1}^N e''_{(l),j} & \left(\sum_{i=1}^{\theta} c_i m_{i,l}^2 \operatorname{sech}^2(\gamma b_i(\mathbf{x}_j)) \right. \\ & \left. - 2 \sum_{i=1}^{\theta} c_i m_{i,l}^2 \gamma b_i(\mathbf{x}_j) \operatorname{sech}^2(\gamma b_i(\mathbf{x}_j)) \tanh(\gamma b_i(\mathbf{x}_j)) \right), \end{aligned} \quad (4.42)$$

donde

$$e''_{(l),j} = \left(y''_{(l),j} - \left(b_l + \sum_{i=1}^{\theta} c_i m_{i,l}^2 \gamma \operatorname{sech}^2(\gamma d_i(\mathbf{x}_j)) \right) \right) \quad (4.43)$$

es el error de la derivada segunda en el punto j , y μ es el parámetro de paso del algoritmo de gradiente. De nuevo, se pueden considerar parámetros de suavidad diferentes para cada frontera, o tratar de minimizar simultáneamente varias de las posibles derivadas segundas si se dispone de medidas de las mismas.

4.4.2.3 Resumen del algoritmo

Por último, el algoritmo general de entrenamiento se puede resumir de la forma siguiente

Paso 0: Elegir un conjunto inicial de fronteras $\mathbf{z}_r^{(k)}$; $k = 0$.

Paso 1: Resolver (4.28) para obtener $\mathbf{z}_p^{(k)}$

Paso 2: Encontrar la dirección de actualización de las fronteras, $\mathbf{s}^{(k)}$, aplicando (4.30).

Paso 3: Realizar la búsqueda a lo largo de $\mathbf{s}^{(k)}$ para encontrar α en (4.36), y así obtener las nuevas fronteras del dominio, $\mathbf{z}_r^{(k+1)}$.

Paso 4: Adaptar γ de acuerdo a (4.37), (4.39) o (4.42), o dejarlo fijo para fijar la suavidad de la solución.

Paso 5: Incrementar $k = k + 1$.

Paso 6: Si $|error^{(k)} - error^{(k-1)}| < \epsilon$, finalizar el entrenamiento; si no, volver al paso 1.

4.5 Análisis de prestaciones

En esta sección se presentan algunos de los resultados obtenidos mediante el modelo MLTS.

4.5.1 Resultados en espacios de entrada unidimensionales

En primer lugar, analizaremos el caso más sencillo, que es el del modelo aplicado a un espacio de entrada unidimensional. En este caso particular, el modelo MLTS se reduce a

$$\hat{f}(x) = a + bx + \sum_{i=1}^{\theta} c_i \text{lch}(x - \beta_i, \gamma_i) \quad (4.44)$$

En este caso extremo, los hiperplanos que dividen el espacio de entrada en regiones son puntos.

4.5.1.1 Entrenamiento del parámetro de suavidad γ

Se analizan aquí las diferentes alternativas propuestas en el Apartado 4.4.2.2 para llevar a cabo el entrenamiento del parámetro de suavidad. Se han comparado las siguientes alternativas:

- Entrenamiento mediante (4.37) (en función de las medidas de la función). Se ha denominado a esta alternativa MLTS0.
- Entrenamiento mediante (4.39) (en función de las medidas de la derivada primera). Se ha denominado a esta alternativa MLTS1.
- Entrenamiento mediante (4.42) (en función de las medidas de la derivada segunda). Se ha denominado a esta alternativa MLTS2.

Para llevar a cabo la comparación de estas alternativas, se ha considerado la reconstrucción de un conjunto de funciones limitadas en banda, generadas mediante una combinación lineal de sinusoides con amplitudes y fases aleatorias y con frecuencias aleatorias comprendidas entre 0 y 0.1 Hz. El periodo de observación es de 30 segundos, y las muestras se obtienen mediante muestreo regular con periodo de muestreo $T = 1s$ (es decir, los registros generados tienen 30 puntos). Como conjunto de test se han generado 1000 señales de acuerdo a estas características (en la Figura 4.10 se muestra un ejemplo). Las prestaciones de cada modelo se han evaluado en términos de la relación señal a error (SER), en dB, de la reconstrucción, tanto para la función como para las

dos primeras derivadas, empleando para esta evaluación un periodo de muestreo 10 veces más pequeño ($T = 0.1$). Los resultados se presentan en función del valor inicial de γ .

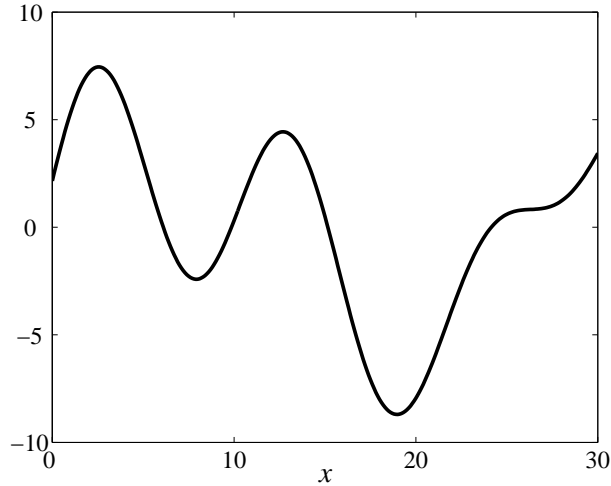


Figura 4.10: Ejemplo de función de prueba

Como ya se ha comentado, este tipo de redes tiene el problema de que la superficie de error presenta mínimos locales. Con objeto de que los resultados no reflejen este problema se ha repetido el experimento desde diferentes condiciones iniciales para los parámetros β_i del modelo MLTS en (4.44), presentando los valores obtenidos al promediar los mejores resultados (un determinado tanto por ciento de los 1000 experimentos totales realizados).

En primer lugar se ha considerado un modelo MLTS con con 5 funciones base y un único valor del parámetro γ común para todas ellas.

La Figura 4.11 presenta los resultados obtenidos en la aproximación de la función y las dos primeras derivadas con las tres alternativas de entrenamiento de γ , evaluando el 5 % de las mejores aproximaciones.

Las mejores prestaciones se obtienen con los modelos MLTS2 (línea continua), MLTS1 (línea discontinua) y MLTS0 (línea de puntos), por este orden. Esto resulta lógico si se tiene en cuenta que γ controla la suavidad del modelo, y que la derivada segunda está relacionada con la suavidad. Un aspecto interesante es que esta mejora depende fuertemente del valor inicial de γ , siendo el modelo más sensible a la inicialización el MLTS2, mientras que el menos sensible es el MLTS1.

Otra de las conclusiones que se han extraído a partir de los resultados obtenidos es que las alternativas de entrenamiento basadas en las medidas de las derivadas son más sensibles al problema de los mínimos locales. Esto se aprecia en la Figura 4.12, donde se presentan los resultados obtenidos al promediar el 10 % de los mejores resultados. En este caso los modelos MLTS2 y MLTS1 proporcionan resultados similares al modelo MLTS0. Esto indica que, si bien los mejores resultados se obtienen ajustando γ en

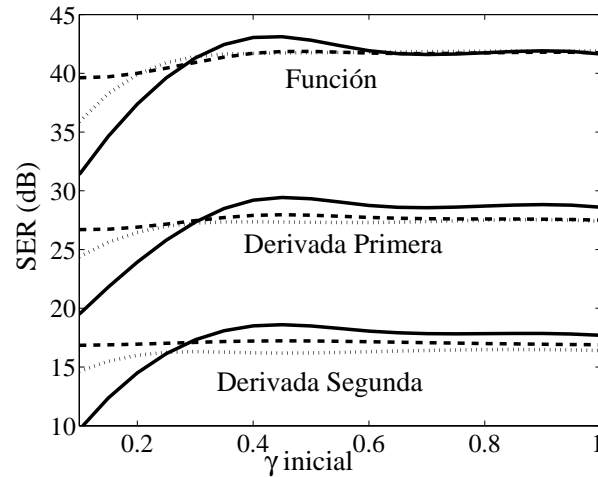


Figura 4.11: Error en la reconstrucción frente al valor inicial de γ . El parámetro de suavidad se entrena para minimizar el error cuadrático de la función (línea de puntos), de la derivada primera (línea discontinua) y de la derivada segunda (línea continua). Todas las funciones base tienen un γ común

función de las medidas de la derivada segunda, esta alternativa converge más a menudo en mínimos locales de la superficie de error. La explicación a este hecho radica en que al modificar γ para minimizar el error en las derivadas y el resto de parámetros para minimizar el error cuadrático de la función, en ocasiones se producen situaciones que conducen a un modelo peor.

Con objeto de evaluar la influencia local de la suavidad, a continuación se presentan los resultados que se obtienen cuando se emplea un parámetro de suavidad γ diferente para cada función base o frontera. La Figura 4.13 muestra los resultados obtenidos al modelar la función y las dos primeras derivadas con las tres estrategias de entrenamiento evaluando el 5 % de los mejores resultados. Comparando estos resultados con los presentados en la Figura 4.11, se comprueba que con esta alternativa se obtienen resultados sensiblemente mejores que con un valor de γ único, y que, de nuevo, los mejores resultados se obtienen con el modelo MLTS2. Además, en este caso se observa una menor sensibilidad al problema de los mínimos locales que en el caso de un γ único. Este comportamiento se ilustra en la Figura 4.14, que presenta los resultados promediando el 50 % de los mejores resultados. Los mejores resultados aún se consiguen con el modelo MLTS2, especialmente para las derivadas, aunque es cierto que la diferencia con las otras alternativas no es tan apreciable.

Hay varios aspectos, a la vista de estos resultados, que merece la pena resaltar. Por un lado se observa una fuerte dependencia de las prestaciones con respecto al valor inicial del parámetro de suavidad. Esto supone un inconveniente desde el punto de vista de que requiere un análisis previo del problema en cuestión. Una inicialización aleatoria supone la introducción de mínimos locales adicionales en la superficie de error.

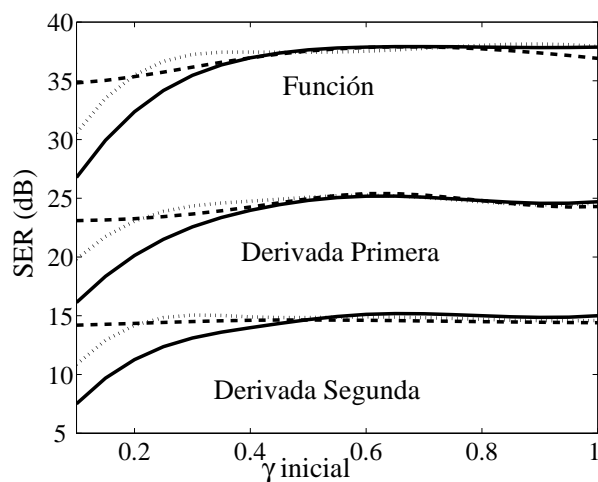


Figura 4.12: Resultados para un γ único sobre el 10 % de los mejores resultados. El parámetro de suavidad se entrena para minimizar el error cuadrático de la función (línea de puntos), de la derivada primera (línea discontinua) y de la derivada segunda (línea continua)

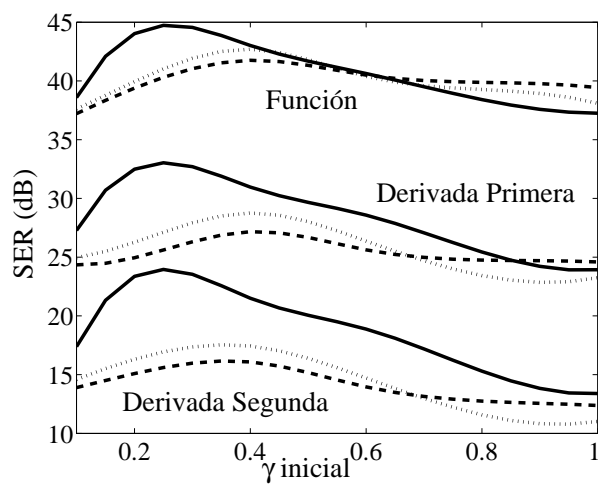


Figura 4.13: Resultados empleando un γ_i diferente para cada función base, promediando el 5 % mejor de las simulaciones. El parámetro de suavidad se entrena para minimizar el error cuadrático de la función (línea de puntos), de la derivada primera (línea discontinua) y de la derivada segunda (línea continua)

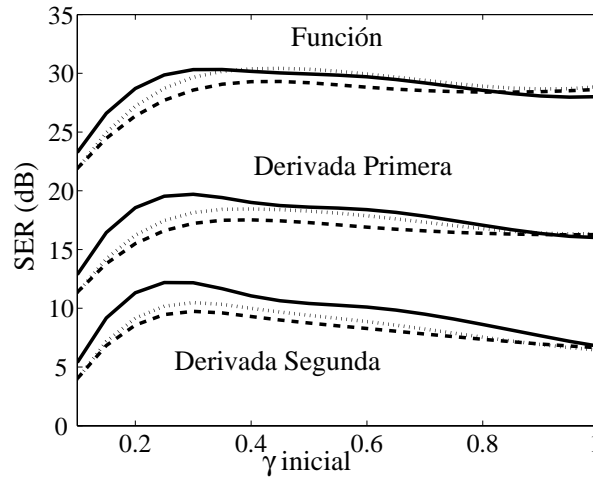


Figura 4.14: Resultados empleando un γ_i diferente para cada función base, promediando el 50% mejor de las simulaciones. El parámetro de suavidad se entrena para minimizar el error cuadrático de la función (línea de puntos), de la derivada primera (línea discontinua) y de la derivada segunda (línea continua)

Por otro lado, el hecho de emplear un parámetro de suavidad γ diferente por cada función base puede ayudar a obtener unos mejores resultados con el modelo MLTS. Esto, que parece razonable desde un punto de vista intuitivo, tiene una explicación sencilla en el hecho de que la suavidad es un concepto, en general, local, y el hecho de permitir diferentes grados de suavidad en diferentes regiones del espacio puede resultar, obviamente, beneficioso. Sin embargo, desde el punto de vista de modelado, en el que el número de parámetros del modelo resulta importante, hay que tener en cuenta que esta mejora supone un incremento, que puede ser importante, del número de parámetros. Finalmente, hay que destacar que la mejora de las prestaciones en el caso de adaptar γ mediante (4.37), (4.39) o (4.42), sobre la alternativa de emplear un γ fijo no es demasiado significativa cuando se utiliza un γ único, mientras que el incremento de carga computacional del entrenamiento de la red es bastante apreciable. La Tabla 4.2 compara esta carga para las diferentes alternativas medida en forma de flops y de tiempos promedios para el entrenamiento de cada modelo (utilizando MATLAB en un Pentium III a 450 MHz).

4.5.1.2 Comparación con la red MLP

En este apartado se presenta una comparación de los resultados obtenidos con el MLTS frente a los que proporciona el *Perceptrón Multicapa* (MLP). En este primer ejemplo, se ha considerado el mismo conjunto de funciones limitadas en banda utilizadas en el apartado anterior, comparandose los resultados obtenidos con redes parsimoniosas, un MLP con 4 neuronas o funciones base y una red MLTS con 5, ambas con el mismo número, 13, de parámetros. La red MLTS ha sido entrenada utilizando un γ fijo,

Modelo	MegaFlops	Tiempo(seg.)
γ con (4.37)(Función)	15.5	30
γ con (4.39)(Derivada 1 ^a)	12.75	23
γ con (4.42)(Derivada 2 ^a)	15	28
γ fijo	5	7.5

Tabla 4.2: Carga Computacional de las diferentes alternativas de entrenamiento de la red MLTS

cuyo valor ha sido seleccionado de forma de forma aleatoria y uniforme entre 0.1 y 1. Se han realizado 1000 experimentos con ambas redes. La Figura 4.15 muestra los resultados de SER(dB) en la reconstrucción de la función (línea continua), de la derivada primera (línea discontinua) y de la derivada segunda (línea de puntos). Para evaluar el efecto de los mínimos locales, los resultados se presentan en función del porcentaje de experimentos considerados

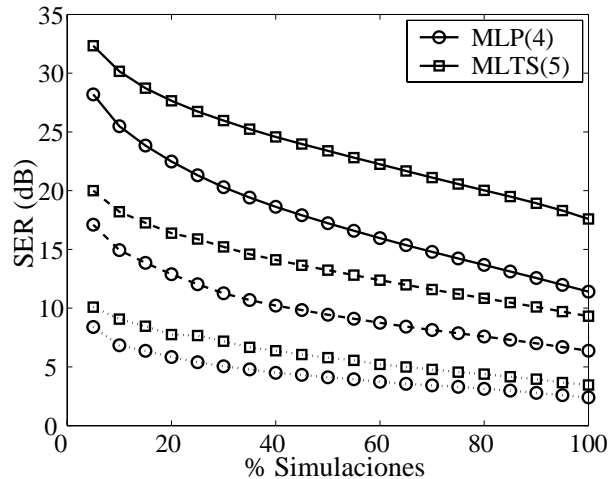


Figura 4.15: Comparación de resultados entre una red MLP de 4 neuronas y una red MLTS de 5, ambas con 13 parámetros. Se presenta la reconstrucción de la función (línea continua) y de la derivadas primera (línea discontinua) y segunda (línea de puntos).

El modelo MLTS proporciona mejores resultados que la red MLP tanto en la reconstrucción de la función como de la derivada. En cuanto al efecto de los mínimos locales, se observa un comportamiento bastante similar para ambas redes, ya que la degradación de resultados al tener en cuenta un mayor porcentaje de experimentos sigue una tendencia bastante pareja en ambos casos.

En un segundo ejemplo, se han comparado las prestaciones de ambos modelos con un mayor número de parámetros. En la Figura 4.16 se muestran los resultados obtenidos en la aproximación de la función (línea continua) y de la derivada segunda (línea de

puntos) con una red MLP de 10 neuronas y una red MLTS de 15, que presentan un número similar de parámetros (31 y 33 respectivamente). De nuevo se comprueba que el MLTS obtiene unos resultados sensiblemente superiores a los del MLP.

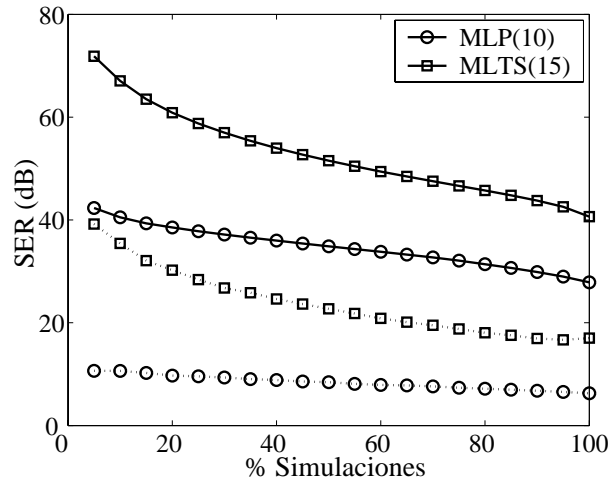


Figura 4.16: Comparación de resultados entre una red MLP de 10 neuronas y una red MLTS de 15 (31 y 33 parámetros respectivamente). Se presentan los resultados de reconstrucción de la función (línea continua) y de la derivada segunda (línea de puntos).

Otro aspecto a tener en cuenta es la carga computacional de ambas soluciones: en la Tabla 4.3 se presenta una comparación de la misma estimada mediante el número de operaciones en punto flotante (flops) y el tiempo medio de entrenamiento de estas redes. Nuevamente, estos resultados han sido obtenidos utilizando MATLAB en un Pentium III a 450 MHz.

Modelo	Parámetros	MegaFlops	Tiempo(seg.)
MLP(4)	13	128	76
MLP(10)	31	350	120
MLTS(5)	13	5	7.5
MLTS(15)	33	35	40

Tabla 4.3: Comparación de los requerimientos computacionales para el entrenamiento de las redes MLP y MLTS en un espacio de entrada unidimensional

Se aprecia la considerable diferencia existente entre ambos modelos: el MLTS es un modelo mucho más económico computacionalmente que el MLP.

4.5.2 Resultados en espacios de entrada bidimensionales

A continuación se presentan los resultados obtenidos con la red MLTS en espacios de entrada bidimensionales. En este caso, el modelo MLTS tiene la forma

$$\hat{f}(x_1, x_2) = a + b_1x_1 + b_2x_2 + \sum_{i=1}^{\theta} c_i \text{lch}(\alpha_i x_1 - x_2 + \beta_i, \gamma_i) \quad (4.45)$$

Los hiperplanos que dividen el espacio de entrada en regiones son, en este caso, rectas.

Para evaluar los resultados, se utiliza un conjunto de prueba formado por 8 funciones previamente utilizadas en [Cherkassky96] para llevar a cabo una comparación entre diferentes alternativas de modelado adaptativo, y que ya se utilizó como conjunto de prueba para espacios bidimensionales en el Capítulo 3. Las expresiones y la representación de estas funciones se muestran en el Apéndice F.

4.5.2.1 Capacidad de aproximación de las derivadas

En primer lugar se ha evaluado la capacidad del modelo MLTS para aproximar una función y sus derivadas a partir de un conjunto de medidas únicamente de la función. Para ello se han comparado los resultados obtenidos con el MLTS, el Perceptrón Multicapa (MLP) y la red de Funciones de Base Radial en su versión generalizada (GRBF). Para el entrenamiento del MLP se ha utilizado el algoritmo de retropropagación con paso adaptativo, inicializando la red con el método propuesto por Nguyen y Widrow [Nguyen90]. La GRBF se inicializa mediante el algoritmo OLS [Chen91]. Las varianzas se seleccionan de forma aleatoria dentro de unos rangos adecuados para cubrir el espacio de entrada de forma razonable. Una vez inicializada la red, la posición de los centroides y de las varianzas se actualiza mediante un algoritmo de gradiente [Santamaría99]. Para la red MLTS se ha empleado un modelo con un único parámetro de suavidad, γ , ya que se pretende utilizar redes con el menor número posible de parámetros. El valor del mismo se selecciona inicialmente de forma aleatoria dentro de un rango de valores adecuados para el espacio de entrada, manteniéndose constante a lo largo del entrenamiento.

El número de muestras de la función a partir de las que se realiza el entrenamiento es 400, distribuidas en una rejilla regular sobre el espacio de entrada. Se han realizado un total de 1000 experimentos con cada función y cada modelo. Se han considerado redes con un número reducido de parámetros, un MLP con 8 neuronas, una GRBF con 7 neuronas, y un MLTS con 10 neuronas, que tienen 33, 35 y 34 parámetros, respectivamente. Para tener en cuenta el problema de los mínimos locales, se presentan los resultados en función del porcentaje de mejores resultados considerado.

En primer lugar, la Tabla 4.4 presenta los resultados obtenidos promediando el 5% de los mejores resultados. Se ha elegido este pequeño porcentaje para mostrar la capacidad de la red para aproximar cada función, de forma que el efecto de los mínimos locales no sea importante. Se presentan los resultados de cada modelo en la aproximación de las distintas funciones y sus dos primeras derivadas (promediando los

resultados de las derivadas en las dos direcciones del espacio), empleando como figura de mérito la relación señal a error de la reconstrucción, en dB. Con el símbolo ♣ se marcan los mejores modelos en cada caso.

Aproximación de la función								
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8
MLTS	21.6 ♣	35.6	45.3 ♣	19.4 ♣	16.5 ♣	25.8	18.1	10.2
MLP	20.2	26.2	40.9	19.5 ♣	13.7	36.0 ♣	17.8	13.4
GRBF	14.6	36.4 ♣	35.7	7.3	15.0	28.5	18.9 ♣	14.5 ♣
Aproximación de la derivada primera								
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8
MLTS	12.6 ♣	18.0	32.0 ♣	15.3 ♣	10.9 ♣	13.7	9.9	5.4
MLP	11.9	16.9	28.5	15.4 ♣	8.1	20.7 ♣	10.0	8.6
GRBF	7.8	26.0 ♣	22.3	5.3	9.6	19.3	10.9 ♣	10.0 ♣
Aproximación de la derivada segunda								
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8
MLTS	5.6 ♣	4.6 ♣	21.0 ♣	10.8 ♣	4.4 ♣	1.3	5.2	1.2
MLP	5.2	1.5	19.2	10.8 ♣	1.7	8.8	5.4 ♣	4.0
GRBF	2.2	3.8	13.3	4.5	4.1	9.8 ♣	5.5 ♣	6.1 ♣

Tabla 4.4: Resultados obtenidos promediando el 5% de los mejores resultados

En primer lugar, se comprueba que no existe una solución que sea globalmente mejor que las demás: cada función tiene unas características que se ajustan mejor a un determinado modelo. Sin embargo, dentro de este comportamiento, hay ciertos aspectos que merece la pena destacar. Por un lado, se comprueba que el modelo MLTS, en cuatro de las 8 funciones, funciones 1,3,4 y 5, obtiene los mejores resultados o, al menos, resultados tan buenos como la mejor de las soluciones. En otras dos, funciones 2 y 7, está menos de 1 dB por debajo de la mejor de las soluciones. Sólo en las funciones 6 y 8 parece estar claramente por debajo de la mejor de las alternativas. En el caso de la última, por ejemplo, esto es fácilmente comprensible a la vista de la forma de la función; parece complicado poder sintetizar dicha función a partir de un número reducido de las funciones base del modelo MLTS. Es una función que, a simple vista, parece más adecuada para una función base como la de la red GRBF.

Por otro lado, es también destacable que, en términos generales, el MLTS ofrece unos resultados superiores en la aproximación de las derivadas. Por ejemplo, en la segunda derivada, en cinco de las 8 funciones, funciones 1, 2, 3, 4 y 5, el modelo MLTS obtiene los mejores resultados y en la función 7 está prácticamente al mismo nivel que las otras soluciones.

Para ver cómo afecta a cada modelo el problema de los mínimos locales se presentan también los resultados promedio considerando el 25% y el 50% de los mejores resultados, que se pueden ver en las Tablas 4.5 y 4.6, respectivamente.

A la vista de estos resultados, se puede concluir que el MLTS es más sensible a los mínimos locales que el MLP o la GRBF, especialmente en la aproximación de las

Aproximación de la función								
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8
MLTS	20.0 ♣	31.8	40.7 ♣	15.6	14.3 ♣	23.4	15.4	9.3
MLP	18.8	24.4	39.3	18.2 ♣	14.1	31.6 ♣	15.9	12.8 ♣
GRBF	13.6	34.9 ♣	35.4	6.1	12.2	24.4	17.9 ♣	11.3

Aproximación de la derivada primera								
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8
MLTS	11.0 ♣	14.5	28.2 ♣	11.8	9.1 ♣	12.6	7.7	4.8
MLP	10.9	15.1	26.8	14.5 ♣	7.14	18.6 ♣	8.4	8.1 ♣
GRBF	7.5	24.5 ♣	22.2	4.0	8.5	15.9	10.2 ♣	7.5

Aproximación de la derivada segunda								
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8
MLTS	3.2	0.3	18.0 ♣	6.6	2.8 ♣	0.9	3.2	0.3
MLP	4.1 ♣	0.4	17.6	10.0 ♣	1.2	7.6 ♣	4.0	3.5
GRBF	1.1	2.7 ♣	13.0	3.0	2.7 ♣	7.7 ♣	4.4 ♣	4.5 ♣

Tabla 4.5: Resultados obtenidos promediando el 25% de los mejores resultados

Aproximación de la función								
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8
MLTS	18.6 ♣	30.2	38.0 ♣	12.8	12.1 ♣	21.2	14.3	8.9
MLP	17.6	23.8	37.8	17.8 ♣	10.8	28.4 ♣	14.7	11.6
GRBF	13.0	33.6 ♣	33.8	5.9	11.0	22.4	16.4 ♣	10.2 ♣

Aproximación de la derivada primera								
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8
MLTS	10.0 ♣	12.9	25.6 ♣	9.4	7.3 ♣	11.2	6.8	4.3
MLP	10.0 ♣	14.7	25.5 ♣	14.1 ♣	6.0	16.9 ♣	7.5	7.3 ♣
GRBF	6.5	23.3 ♣	21.0	3.7	6.3	14.6	9.2 ♣	6.6

Aproximación de la derivada segunda								
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8
MLTS	2.0	-0.7	15.7	5.0	1.4 ♣	0.2	2.7	-0.1
MLP	3.6 ♣	0.4	16.6 ♣	9.7 ♣	0.5	6.8	3.5	2.9
GRBF	1.1	2.7 ♣	12.0	2.5	1.3 ♣	7.3 ♣	4.2 ♣	3.8 ♣

Tabla 4.6: Resultados obtenidos promediando el 50% de los mejores resultados

derivadas. A medida que se tiene en cuenta un número mayor de experimentos en la estima de los resultados, las prestaciones del MLTS se degradan más que las de las otras alternativas. Aunque en la aproximación de la función el efecto es menor y sólo en el caso de la función 4 las prestaciones se degradan por debajo de las prestaciones del MLP, en las derivadas esta degradación es más acusada. Se puede ver que ya para el 25% de resultados en la derivada segunda el MLTS sólo proporciona los mejores resultados en dos funciones, mientras que cuando se considera el 50% de los resultados sólo en la función 5 sigue teniendo los mejores resultados.

Para ilustrar este efecto con más claridad, la Figura 4.17 presenta los resultados obtenidos con el MLTS y el MLP para la función 3, en función del porcentaje de experimentos promediados. Se observa con claridad la degradación de las prestaciones del MLTS debida al efecto de los mínimos locales.

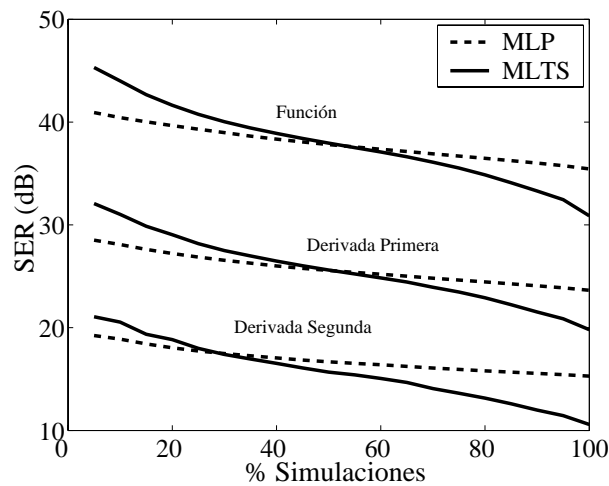


Figura 4.17: Sensibilidad a los mínimos locales de diferentes métodos

Para explicar esta mayor sensibilidad a los mínimos locales, hay que considerar que tanto el MLP como la GRBF emplean algoritmos de inicialización que permiten partir de una solución inicial aceptable, especialmente en el caso de la GRBF. Para el MLTS, por el contrario, la inicialización de las fronteras del espacio de entrada se hace de forma totalmente aleatoria. Esta estrategia, en términos generales, conduce a inicializaciones bastante pobres. Empleando una estrategia de inicialización más conveniente, esta sensibilidad se reduciría de forma considerable. Por otro lado, la selección aleatoria de γ aumenta el número de mínimos locales. Adaptando este parámetro, se reduce esta sensibilidad, especialmente si se utilizan diferentes valores del mismo para cada función base, aunque esto supone un aumento de la carga computacional y del número de parámetros.

Un último aspecto importante a tener en cuenta es la carga computacional de los algoritmos de entrenamiento de los distintos modelos. En la Tabla 4.7 se presenta la carga computacional media, en megaflops, para entrenar cada uno de los modelos. Se

muestran los valores promedio obtenidos sobre las 8 funciones de prueba, imponiendo similares criterios de parada en los algoritmos de entrenamiento de los distintos modelos.

Modelo	Iteraciones	Mflops
MLP	40000	4000
GRBF	4000	2300
MLTS	300	300

Tabla 4.7: Carga computacional de las diferentes alternativas en espacios de entrada bidimensionales

Por un lado, la carga del MLTS es mucho menor que la de los otros dos modelos, prácticamente 8 veces menor que la de la GRBF, y, por otro lado, la carga computacional de la GRBF es casi la mitad que la del MLP. Esto hace que la posible penalización por la sensibilidad a los mínimos locales se reduzca en la práctica de forma considerable, ya que el tiempo requerido para llevar a cabo cada uno de los experimentos es considerablemente menor que con las otras alternativas, especialmente cuando se compara con el MLP.

4.6 Discusión

A través de las simulaciones y resultados presentados en este capítulo, se ha mostrado que el modelo MLTS propuesto presenta una notable capacidad para el modelado o aproximación de una función, y para la generalización de las derivadas de la misma. El modelo resulta competitivo frente a algunas de las técnicas de modelado no lineal más extendidas, como son las redes neuronales MLP y GRBF.

En particular, el MLTS resulta especialmente interesante en aplicaciones en las que existe una limitación en el número de parámetros de la solución, donde se precisan modelos parsimoniosos. En este caso, la compacta formulación del modelo le permite ser especialmente competitivo.

Las características de suavidad del modelo facilitan la generalización de las derivadas de la función que se modela. En este sentido, el modelo presenta unas prestaciones en general superiores a las de las redes MLP y GRBF.

Otra de las ventajas del MLTS es la reducida carga computacional para el entrenamiento del modelo. En este aspecto, el MLTS es sensiblemente más económico que el MLP o la GRBF.

Tal vez el inconveniente más importante del modelo sea su elevada sensibilidad a los mínimos locales. Este hecho, que en parte tiene su origen en la naturaleza de la función de activación [Lin93], se ve agravado por la falta de un algoritmo de inicialización eficiente. En este sentido, una interesante línea de trabajo es la búsqueda de algoritmos de inicialización eficientes. No obstante, la penalización del modelo por esta sensibilidad

a los mínimos locales se ve reducida por la economía computacional del mismo, ya que la repetición de experimentos hasta alcanzar una solución aceptable no es muy costosa.

El parámetro γ permite el control de la suavidad del modelo y, por lo tanto, de la capacidad para modelar las derivadas. Cuando no se tiene información a priori de la suavidad del modelo este parámetro puede introducir más mínimos locales. Si se dispone de información de las derivadas es posible introducir dicha información en la adaptación de γ

Capítulo 5

Alternativas neuronales para el modelado de una función y sus derivadas

Como se vió en el Capítulo 2, las redes neuronales presentan una reconocida capacidad teórica para la aproximación simultánea de una función y sus derivadas [Hornik90, Gallant92, Ito93]; sin embargo, no existen alternativas constructivas que permitan introducir la información de las derivadas en la topología o entrenamiento de la red. En general, las aplicaciones existentes en este contexto se limitan a la obtención de un modelo adecuado de las derivadas de la función a través de la obtención de una solución para la función, aprovechando la derivabilidad de las funciones de activación habitualmente empleadas [Jordan89, Li96, Nguyen99].

En este capítulo se presentan varias alternativas constructivas que permiten introducir la información de las derivadas en la arquitectura y entrenamiento de la red. Por un lado, se ha atacado el problema a través de la modificación de la función de coste empleada en el entrenamiento de la red. La función de coste más habitual es el error cuadrático evaluado únicamente sobre las muestras de la función. Una alternativa sencilla para incorporar nuestra información a priori consiste en añadir un término que tenga en cuenta el error cuadrático sobre las muestras de las derivadas. Esta modificación se ha estudiado en los dos tipos de redes neuronales más conocidas, el MLP y la GRBF, y en el modelo MLTS presentado en el Capítulo 4. Se ha comprobado que, como era razonable esperar, la introducción de este término en el entrenamiento de la red supone una importante mejora en la aproximación de las derivadas.

Por otro lado, se ha atacado el problema a través del diseño de la arquitectura de la red. En este sentido, se ha propuesto una arquitectura modular en la que un módulo se encarga de capturar la información de la función y otro módulo la información de las derivadas. A partir de este modelo es posible obtener una reconstrucción localizada de la función y sus derivadas. Este es un modelo principalmente enfocado a la aplicación de modelado de transistores de microondas. En la arquitectura específica propuesta para esta aplicación, para el módulo relacionado con la función se ha empleado el modelo

MLTS presentado en el Capítulo 4, mientras que para el módulo encargado de capturar la información de las derivadas se ha utilizado una red GRBF. Para el entrenamiento de este módulo, una última contribución de este capítulo es el desarrollo de un nuevo algoritmo de entrenamiento para redes RBF, basado en el algoritmo EM (*“Expectation - Maximization”*), que permite acelerar el entrenamiento de la red y reduce la sensibilidad a los mínimos locales.

5.1 Introducción de la información de las derivadas en la función de coste

En problemas de modelado y aproximación funcional, el entrenamiento de las redes neuronales, en general, se traduce en la minimización de una determinada función de coste mediante un algoritmo de entrenamiento específico para cada red. La función de coste más habitual es el error cuadrático de la aproximación

$$J = \sum_n \left(f(x[n]) - \hat{f}(x[n]) \right)^2, \quad (5.1)$$

donde $f(x[n])$ son las muestras de la función a aproximar, y $\hat{f}(x[n])$ son las estimas que proporciona la red neuronal en dichos puntos.

Cuando, además de muestras de la función, se dispone de muestras de sus derivadas, una alternativa sencilla para introducir esta información en la construcción de la red consiste en modificar la función de coste para tener en cuenta el error en la estima de las derivadas. De esta forma, es posible definir una función de coste alternativa

$$J_d = \lambda_0 \sum_n \left(f(x[n]) - \hat{f}(x[n]) \right)^2 + \lambda_1 \sum_n \left(f^1(x[n]) - \hat{f}^1(x[n]) \right)^2, \quad (5.2)$$

donde $f^1(x[n])$ y $\hat{f}^1(x[n])$ son, respectivamente, las muestras de la derivada de la función a modelar y las estimas de la misma. Los parámetros λ_0 y λ_1 se encargan de ponderar la importancia relativa del error en las muestras de la señal y de la derivada. Su valor dependerá de aspectos específicos de cada aplicación, como la fidelidad (nivel de ruido) en las muestras de función y derivada, o del interés en la aproximación de las derivadas de la función.

En esta sección se presentan los resultados obtenidos al minimizar una función de coste modificada como (5.2) en el entrenamiento del Perceptrón Multicapa (MLP) y la Red de Funciones de Base Radial Generalizada (GRBF), así como en el modelo MLTS presentado en el Capítulo 4. En principio se ha limitado el estudio a la introducción de información de la primera derivada, y a espacios de entrada unidimensionales y bidimensionales. La extensión de la expresión (5.2) a espacios bidimensionales es obvia teniendo en cuenta que hay dos derivadas primeras. Utilizando la notación

$$\mathbf{x}_n = [x_1[n], x_2[n]], \quad (5.3)$$

la función de coste para espacios 2D se puede escribir como

$$J_d = \sum_n \lambda_0 \left(f(\mathbf{x}_n) - \hat{f}(\mathbf{x}_n) \right)^2 + \lambda_1 \left(f_{x_1}^1(\mathbf{x}_n) - \hat{f}_{x_1}^1(\mathbf{x}_n) \right)^2 + \lambda_2 \left(f_{x_2}^1(\mathbf{x}_n) - \hat{f}_{x_2}^1(\mathbf{x}_n) \right)^2, \quad (5.4)$$

donde $f_{x_1}^1(\mathbf{x}_n)$ y $f_{x_2}^1(\mathbf{x}_n)$ son las derivadas parciales con respecto a x_1 y x_2 .

5.1.1 Perceptrón Multicapa (MLP)

Se ha considerado un MLP con una única capa oculta, con la función de activación tangente hiperbólica, y con una función de activación lineal en las neuronas de la capa de salida. Esta es una estructura habitual en problemas de aproximación funcional. En este caso, la expresión del modelo es

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^N w_i^2 \tanh(\langle \mathbf{w}_i^1, \mathbf{x} \rangle + b_i^1) + b_i^2, \quad (5.5)$$

donde w_i^c y b_i^c son, respectivamente, los pesos y términos de sesgo asociados a la neurona i -ésima en la capa c .

La red se inicializa con el algoritmo propuesto por Nguyen y Widrow [Nguyen90]. Los parámetros de la red, w_i^c y b_i^c , se adaptan mediante un algoritmo de descenso de gradiente aplicado a la función de coste J_d (5.2) o (5.4)

$$w_i^c = w_i^c - \eta \frac{\partial J_d}{\partial w_i^c}, \quad (5.6)$$

$$b_i^c = b_i^c - \eta \frac{\partial J_d}{\partial b_i^c}, \quad (5.7)$$

donde, por simplicidad de notación, el índice indicando la iteración se ha omitido.

5.1.2 Red de Funciones de Base Radial Generalizada (GRBF)

Se ha utilizado una GRBF, permitiendo varianzas diferentes en cada una de las direcciones del espacio de entrada. La expresión de una GRBF con N neuronas viene dada por

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^N c_i o_i(\mathbf{x}), \quad (5.8)$$

donde $o_i(\mathbf{x})$ es la función de activación de la i -ésima neurona

$$o_i(\mathbf{x}) = \prod_{j=1}^{\dim(\mathbf{x})} \exp -\frac{(x_j - \mu_{i,j})^2}{2\sigma_{i,j}^2}, \quad (5.9)$$

indexando j la dirección del espacio de entrada.

La red se inicializa mediante el algoritmo OLS [Chen91]. Los centroides y las varianzas se adaptan mediante un algoritmo de descenso de gradiente

$$\mu_{i,j} = \mu_{i,j} - \eta \frac{\partial J_d}{\partial \mu_{i,j}}, \quad (5.10)$$

$$\sigma_{i,j}^2 = \sigma_{i,j}^2 - \eta \frac{\partial J_d}{\partial \sigma_{i,j}^2}, \quad (5.11)$$

y el ajuste de los pesos de la capa de salida, c_i , se realiza en cada iteración mediante un ajuste de mínimos cuadrados del funcional J_d (5.2) o (5.4).

5.1.3 Modelo Lineal a Tramos Suavizado (MLTS)

La expresión general de un modelo MLTS de N neuronas tiene la forma

$$\hat{f}(\mathbf{x}) = a + \mathbf{b}^T \mathbf{x} + \sum_{i=1}^{\theta} c_i \text{lch}(\langle \boldsymbol{\alpha}_i, \mathbf{x} \rangle - \beta_i, \gamma_i). \quad (5.12)$$

donde $\text{lch}(x, \gamma)$ es la función de suavizado

$$\text{lch}(x, \gamma) = \frac{1}{\gamma} \ln(\cosh(\gamma x)) \quad (5.13)$$

Para reducir el número de parámetros, la expresión $\langle \boldsymbol{\alpha}_i, \mathbf{x} \rangle - \beta_i$ se ha sustituido por

$$x - \beta_i, \quad (5.14)$$

en espacios de entrada 1D, y por

$$\alpha_i x_1 - x_2 + \beta_i, \quad (5.15)$$

en espacios de entrada 2D.

El algoritmo de entrenamiento de la red es similar al descrito en el Capítulo 4, sustituyendo la función de coste error cuadrático por el funcional (5.2). Los parámetros que definen la partición del espacio de entrada, α_i y β_i , se adaptan mediante un método de segundo orden, calculando el gradiente y el Hessiano de la función de coste, que definen la dirección de variación de los parámetros. Una vez obtenidos los valores óptimos de los mismos en cada iteración, los parámetros que definen la combinación lineal de los componentes del modelo, a , \mathbf{b} y c_i , se obtienen mediante un ajuste de mínimos cuadrados de la función de coste. Este procedimiento, que comienza mediante una selección aleatoria de los parámetros α_i y β_i , se itera hasta obtener una solución adecuada.

5.1.4 Resultados

A continuación se comparan los resultados obtenidos al utilizar las funciones de coste J y J_d en las redes MLP, GRBF y MLTS, en espacios de entrada bidimensionales. Nuevamente, se ha seleccionado como conjunto de prueba el formado por las 8 funciones utilizadas por Cherkassky, Gehring y Mulier [Cherkassky96] para llevar a cabo una comparación entre diferentes alternativas de modelado adaptativo. Las expresiones y la representación de estas funciones se muestran en el Apéndice F. Como factor de mérito de la reconstrucción se emplea la SER (“Signal to Error Ratio”) en la reconstrucción de la función y de la derivada, evaluada esta última mediante el promedio de la derivada con respecto a las dos direcciones del espacio de entrada.

Se ha estudiado la evolución de la SER frente al número de puntos del conjunto de entrenamiento, considerando una rejilla uniforme de $N \times N$ instantes de muestreo y variando N . Los parámetros de ponderación, λ_0 , λ_1 y λ_2 , se han obtenido mediante la relación de potencias de las secuencias de muestras de función y derivadas para que la contribución de los términos del funcional J_d sea del mismo orden. Se ha considerado una SNR de 20 dB en las muestras tanto de la función como de las derivadas.

En primer lugar se muestran los resultados obtenidos con la red MLTS. En la Figura 5.1 se representan los resultados obtenidos con una red de 32 neuronas (100 parámetros) en la reconstrucción de la función y de la derivada utilizando los funcionales J y J_d . Se presentan los resultados obtenidos para la Función 1 en función del número de instantes de muestreo considerados. Hay que tener en cuenta que el número de datos totales utilizando el funcional J_d es, en este caso, el triple que con el funcional J .

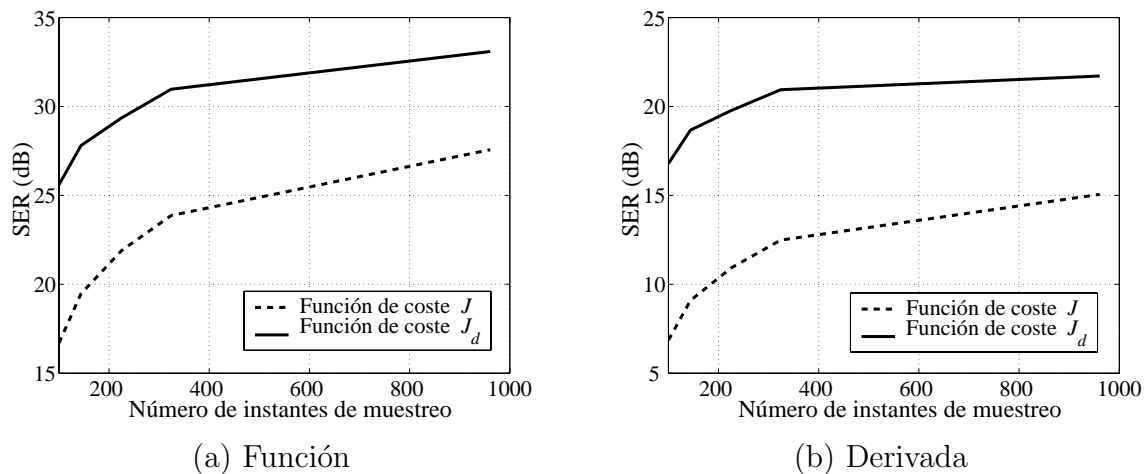


Figura 5.1: Resultados en la reconstrucción de la Función 1 con una red MLTS de 32 neuronas.

Se observa que los resultados obtenidos con la función de coste J_d son claramente superiores a los obtenidos con la función J , tanto en la función como en la derivada. Esta mejora se debe, obviamente, a la información “adicional” que introducen las muestras de la derivada primera.

Más interesante resulta resaltar que los resultados obtenidos minimizando J_d son también mejores cuando fijamos el número total de muestras, es decir, cuando el número de instantes de medida para esta alternativa es un tercio de los utilizados con el funcional J . En la derivada la mejora es mucho más notable: por ejemplo, utilizando 100 instantes de muestreo para la función y cada una de las derivadas primeras (300 muestras totales), se obtienen mejores resultados en la reconstrucción de la derivada que minimizando J utilizando 961 medidas (más del triple de puntos de medida).

El mismo comportamiento se ha observado sobre todas las funciones de prueba. La Tabla 5.1 compara los resultados obtenidos minimizando el funcional J con 961 instantes de muestreo (una rejilla de 31×31 instantes equiespaciados), con los obtenidos empleando el funcional J_d con 324 instantes de muestreo para la función y cada una de las 2 derivadas (una rejilla de 18×18 instantes equiespaciados), lo que supone un número total de 972 muestras: en todos los casos se obtiene una apreciable mejora al utilizar la función de coste J_d . La mejora es especialmente importante en la reconstrucción de la derivada.

Aproximación de la función								
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8
Func. Coste J	27.6	29.0	31.6	24.9	20.9	25.5	24.0	18.8
Func. Coste J_d	31.0	33.1	37.4	27.3	24.1	27.5	27.1	19.5

Aproximación de la derivada primera								
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8
Func. Coste J	15.1	17.3	19.4	18.5	12.3	14.1	12.9	9.9
Func. Coste J_d	21.0	25.8	29.1	23.6	18.1	20.8	19.6	13.2

Tabla 5.1: Resultados obtenidos con una red MLTS de 32 neuronas (100 parámetros) utilizando las funciones de coste J y J_d

El mismo comportamiento observado para el MLTS aparece tanto en el MLP como en la GRBF, lo que parece indicar que la mejora obtenida se debe al entrenamiento (función de coste) y no a la arquitectura de la red. Las Tablas 5.2 y 5.3 presentan los resultados obtenidos respectivamente con un MLP de 25 neuronas (101 parámetros) y una GRBF de 20 neuronas (100 parámetros) en la reconstrucción de la función y de la derivada empleando los funcionales J y J_d . De nuevo se comparan los resultados obtenidos minimizando el funcional J con 961 instantes de muestreo (una rejilla de 31×31 instantes equiespaciados) con los obtenidos empleando el funcional J_d con 324 instantes de muestreo para la función y cada una de las 2 derivadas (una rejilla de 18×18 instantes equiespaciados), con un número total de 972 muestras.

Todos estos resultados se han obtenido mediante redes con un número relativamente elevado de parámetros (en torno a 100). Resulta por ello interesante comprobar qué ocurre cuando se utilizan redes de menor tamaño. La Tabla 5.4 presenta los resultados obtenidos con un MLTS de 12 neuronas (40 parámetros) en las mismas condiciones de entrenamiento para ambas funciones de coste que en los casos anteriores. Ahora,

Aproximación de la función								
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8
Func. Coste J	22.4	20.6	29.4	16.0	16.8	20.4	18.0	15.9
Func. Coste J_d	27.2	25.9	35.5	19.4	19.5	25.8	21.5	16.6

Aproximación de la derivada primera								
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8
Func. Coste J	12.5	10.4	17.5	11.3	9.2	12.2	9.1	9.9
Func. Coste J_d	18.1	17.2	25.5	15.1	13.6	18.7	13.7	11.1

Tabla 5.2: Resultados obtenidos con un MLP de 25 neuronas (101 parámetros) utilizando las funciones de coste J y J_d

Aproximación de la función								
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8
Func. Coste J	30.2	29.6	32.5	24.3	27.8	29.1	28.0	22.9
Func. Coste J_d	34.0	34.5	37.8	25.5	29.9	31.1	29.2	22.8

Aproximación de la derivada primera								
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8
Func. Coste J	19.2	18.4	20.6	17.0	20.0	18.3	17.4	14.2
Func. Coste J_d	25.9	27.8	28.2	20.0	27.1	25.0	22.0	16.1

Tabla 5.3: Resultados obtenidos con una red GRBF de 20 neuronas (100 parámetros) utilizando las funciones de coste J y J_d

en la mayoría de las funciones, la aproximación utilizando el funcional J proporciona resultados ligeramente superiores en la aproximación de la función, pero notablemente inferiores en la aproximación de las derivadas. Podemos por tanto concluir que, en términos generales, utilizando el funcional J_d se consigue una reconstrucción más fidedigna de la función. Este mismo comportamiento se ha podido observar también al utilizar las redes MLP y GRBF.

Aproximación de la función								
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8
Func. Coste J	21.4	24.4	33.0	13.2	13.0	22.9	16.3	10.5
Func. Coste J_d	20.3	24.3	35.5	14.3	11.8	21.8	14.7	10.1

Aproximación de la derivada primera								
	Func 1	Func 2	Func 3	Func 4	Func 5	Func 6	Func 7	Func 8
Func. Coste J	11.6	14.5	22.1	9.8	7.0	12.1	8.0	5.3
Func. Coste J_d	13.3	17.6	28.7	12.0	7.9	16.6	9.4	6.1

Tabla 5.4: Resultados obtenidos con un MLTS de 12 neuronas (40 parámetros) utilizando las funciones de coste J y J_d

5.2 Una arquitectura modular para la aproximación de una función y sus derivadas

En la sección anterior se ha presentado una alternativa para introducir la información de las derivadas en una red neuronal a través de la función de coste del algoritmo de entrenamiento. En esta sección se considera cómo introducir la información de las derivadas a través de la arquitectura de la red. Para ello se propone una arquitectura modular en la que módulos diferentes se encargan de procesar la información de la función y las derivadas.

Las redes modulares se basan en el principio de dividir problemas complejos en varios subproblemas más simples, y combinar las soluciones individuales de cada subproblema. Habitualmente, los diferentes módulos se especializan en el aprendizaje en distintas regiones del espacio de entrada [Haykin94, Jacobs91]. Sin embargo, también es posible utilizar este tipo de arquitecturas cuando en el problema a resolver aparecen varios tipos de información con diferentes características. Este es el caso del problema abordado en esta Tesis, ya que la información sobre las derivadas es, en esencia, local, mientras que la información de la función es de carácter global.

Esta sección está organizada de la forma siguiente: se describe en primer lugar la arquitectura general propuesta para, a continuación, presentar la arquitectura específica para el problema de modelado de transistores de microondas. Dentro de esta arquitectura se propone un nuevo algoritmo de entrenamiento para el modelo local (realizado en esta aplicación específica mediante una GRBF) que forma parte de la misma. Este

algoritmo, una variante del algoritmo EM, permite acelerar la convergencia del entrenamiento de la red y disminuye su sensibilidad a los mínimos locales. Para finalizar, se presentan los resultados obtenidos con el modelo propuesto en el modelado de un transistor de microondas a partir de los datos obtenidos de un modelo analítico del mismo.

5.2.1 Arquitectura propuesta

En general, la información contenida en las muestras de una función y de sus derivadas tiene un carácter diferente. Las muestras de la función contienen la información del comportamiento global de la misma, mientras que las muestras de las derivadas reflejan el comportamiento local de la función. Debido a esta diferente naturaleza, parece razonable utilizar redes distintas para aproximarlas. De este modo, es posible introducir la información de las derivadas en una red neuronal a través de la arquitectura de la misma mediante una red modular: módulos diferentes se encargan de modelar, respectivamente, la función y sus derivadas. La Figura 5.2 representa esta arquitectura general.

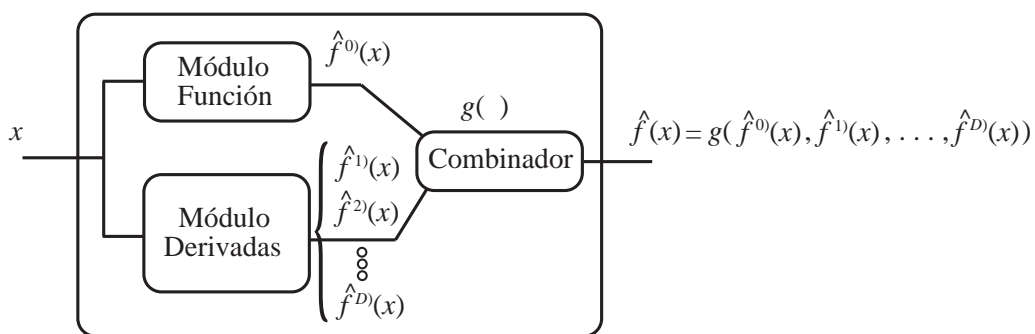


Figura 5.2: Arquitectura modular para la reconstrucción de una función utilizando la información de las derivadas

Los módulos de la función y de las derivadas se pueden obtener, de forma independiente, mediante una red neuronal que ajusta el correspondiente conjunto de muestras. El principal problema consiste en encontrar un combinador adecuado que sea capaz de obtener una aproximación razonable de la función y sus derivadas simultáneamente. Encontrar un combinador que utilice de forma adecuada la información de cada módulo, y proporcione una solución continua y derivable en todo el espacio de entrada, es un problema complejo.

5.2.1.1 Reconstrucción local de una función y sus derivadas

Una posibilidad que permite la reconstrucción localizada de una función y sus derivadas es utilizar como combinador el desarrollo en serie de Taylor de la función, $f(x)$, en torno

a un punto x_0

$$\hat{f}(x) = \sum_{d=0}^D \frac{(x - x_0)^d}{d!} \left. \frac{d^d f(x)}{dx^d} \right|_{x=x_0} \quad (5.16)$$

Empleando una estructura modular similar a la de la Figura 5.2, un módulo proporciona la estima de la función en el punto x_0 , $\hat{f}(x_0)$, y el otro módulo proporciona las estimas de las derivadas, $\hat{f}^1(x_0), \hat{f}^2(x_0), \dots, \hat{f}^D(x_0)$. A partir de estas estimas, utilizando un combinador que implementa el desarrollo (5.16), se obtiene una solución que, en torno al punto x_0 , aproxima simultáneamente la función y sus derivadas de forma razonable. Este tipo de modelo se puede extender a espacios de entrada bidimensionales, teniendo en cuenta que en el desarrollo de Taylor correspondiente intervienen también los términos de las derivadas cruzadas.

Este modelo modular no resulta adecuado cuando se desea una aproximación global de una función. Sin embargo, en aplicaciones en las que el interés se centra en obtener una aproximación local de la función y, simultáneamente, aproximar sus derivadas, esta alternativa proporciona resultados satisfactorios.

5.2.1.2 Reconstrucción global de la función con reproducción local de las derivadas

Empleando el modelo presentado en la sección anterior, la aproximación obtenida sólo es adecuada en un entorno de x_0 . Cuando se desea tener un modelo capaz de proporcionar una aproximación adecuada de la función en todo el espacio de entrada, y, además, aproximar las derivadas en un entorno local de determinados puntos del espacio de entrada, es posible emplear una solución como la que se muestra en la Figura 5.3.

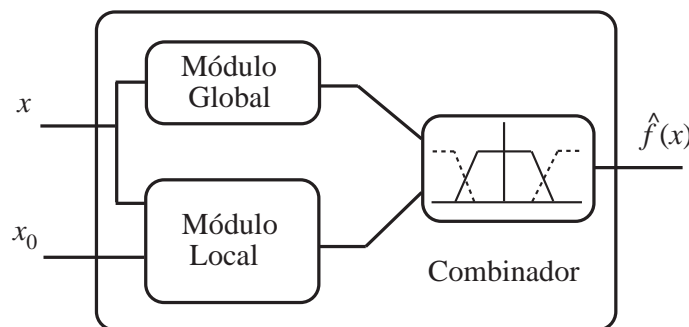


Figura 5.3: Arquitectura modular para la reconstrucción global de la función y local de las derivadas

El módulo local es el presentado en la sección anterior, mientras que para el módulo global se utiliza una red neuronal entrenada para ajustar la función $f(x)$. El combinador proporciona la aproximación final en función de la salida de ambos módulos. Para

este combinador se han utilizado funciones relacionadas con la teoría de la lógica borrosa (*"Fuzzy Logic"*), donde se plantean reglas de decisión en función del conocimiento específico del problema [Mendel95].

La idea fundamental de este modelo es que, cuando se trabaja en un entorno cercano a x_0 , la salida corresponde al módulo local (que aproxima las derivadas), mientras que en puntos lejanos a x_0 , la salida se obtiene a partir del módulo global. Para la transición entre ambas situaciones, existen varias posibilidades, dependiendo de las restricciones de continuidad que se exigen al modelo global. De esta forma, para el combinador se definen dos funciones de pertenencia complementarias, una para cada módulo, que deben sumar la unidad en todo el espacio de entrada. Como ejemplos de diferentes tipos de funciones de pertenencia para el módulo local se tienen las siguientes

- Función rectangular: en este caso se realiza una decisión "dura", cambiando de un módulo a otro en un determinado punto de transición. Con esta alternativa se obtiene un modelo global discontinuo en el punto de transición.
- Funciones con transición lineal (función triangular, función trapezoidal, etc). Proporcionan una solución global continua pero con derivadas discontinuas en los puntos de transición de la función de pertenencia.
- Funciones derivables (exponencial, trapezoidal suavizada, etc). Con este tipo de funciones se obtiene una solución continua tanto en la función como en las derivadas. En las derivadas, aparecen oscilaciones en la zona de transición.

Obviamente, la función de pertenencia correspondiente al módulo global es, en cada caso, la función complementaria.

Con el modelo propuesto se obtiene una solución que, localmente, en el entorno de x_0 , ajusta de forma adecuada tanto la función como las derivadas, mientras que lejos de x_0 la función se ajusta por un modelo neuronal convencional, con lo que la estima de las derivadas en esa zona corresponderá a las de dicha red. Por esta razón, es recomendable que esta red sea capaz de ajustar lo mejor posible dichas derivadas.

5.2.2 Arquitectura específica para el problema de modelado de transistores

Una aplicación en la que la arquitectura propuesta en la Sección 5.2.1 encuentra utilidad es el modelado no lineal de transistores de microondas. Como ya se mencionó en la introducción de la Tesis, el elemento no lineal predominante en los transistores MESFET y HEMT, que son los más comúnmente utilizados en aplicaciones de microondas, es la corriente drenador-fuente, I_{ds} , que depende de las tensiones drenador-fuente, V_{ds} , y puerta-fuente, V_{gs} . Esta dependencia se suele denominar característica corriente/tensión (I/V) del transistor. Cuando únicamente se desea reproducir el comportamiento gran señal de los transistores, en general, basta con modelar de forma adecuada la característica no lineal del dispositivo, es decir, la característica I/V [Fernández96].

Pero cuando además se desea modelar el comportamiento pequeña señal, siendo capaces de reproducir los efectos de la intermodulación, no basta con tener una buena aproximación de la característica I/V del dispositivo, sino que además es necesario caracterizar de forma adecuada las derivadas de la misma hasta el orden de los productos de intermodulación que se quiera tener en cuenta [Crosmun89, Maas91]. Para modelar de forma adecuada los efectos de la intermodulación de hasta tercer orden, que es lo más habitual para mezcladores y amplificadores en aplicaciones de comunicaciones [Maas90], es necesario aproximar hasta la tercera derivada de la característica I/V del dispositivo.

En el modo de operación normal de los transistores MESFET y HEMT, cuando se aplica una pequeña señal de RF sobre un punto de polarización, la corriente I_{ds} depende de las tensiones de polarización, (V_{ds0}, V_{gs0}) , y de las tensiones instantáneas de pequeña señal, (v_{ds}, v_{gs}) , aplicadas sobre ese punto. En un modelo de pequeña señal, es necesario modelar, en el entorno del punto de polarización, hasta la derivada de tercer orden de la característica I/V , mientras que, en un entorno de gran señal, la aproximación de dicha característica resulta suficiente.

Así pues, el problema del modelado de transistores se ajusta a las características del modelo propuesto en la Sección 5.2.1. En este caso, el modelo global es un modelo gran señal de los transistores, mientras que el modelo local corresponde a un modelo pequeña señal con capacidad de reproducir las derivadas y, por tanto, de predecir los efectos de intermodulación. En esta sección se presentan, por un lado el módulo propuesto para el modelado gran señal de los transistores, por otro lado, el módulo propuesto para el modelado pequeña señal, y, finalmente, el modelo global basado en los dos módulos anteriores.

5.2.2.1 Modelo gran señal de transistores de microondas: el MLTS

Como ya se ha mencionado, el comportamiento en gran señal de un transistor de microondas está gobernado por la característica dinámica I/V , que depende el punto de polarización [Fernández96]. Así pues, el modelo gran señal de un transistor debe modelar la corriente I_{ds} del transistor en función de las tensiones de polarización y las tensiones de pulsada aplicadas en ese punto, es decir

$$\hat{I}_{ds} = f(V_{ds0}, V_{gs0}, v_{ds}, v_{gs}). \quad (5.17)$$

Este modelo se implementa mediante una red neuronal con cuatro entradas, que proporciona como salida la estima de la corriente I_{ds} . Como estructura neuronal, se ha seleccionado el modelo MLTS presentado en el Capítulo 4. Existen varias razones para esta selección: por un lado, este modelo se ha mostrado especialmente competitivo en aplicaciones que requieren modelos parsimoniosos y es capaz de proporcionar una aproximación suave y derivable. Por otro lado, como se verá con posterioridad en el Capítulo 6, en esta aplicación específica el MLTS proporciona resultados superiores a los obtenidos por otras redes neuronales, como el MLP o la GRBF, tanto en la aproximación de la función, como de sus derivadas. Este último aspecto, aunque no resulta

determinante en un modelo gran señal, proporciona un modelo más realista del transistor, resultando beneficioso si se pretende utilizar el modelo como parte de un modelo global, válido tanto para pequeña como para gran señal, del transistor.

5.2.2.2 Modelo pequeña señal de transistores de microondas: la GRBF

Una alternativa para llevar a cabo una reconstrucción con las características ya mencionadas de un modelo pequeña señal de transistores de microondas es utilizar el desarrollo en serie de Taylor

$$\begin{aligned}
 I_{ds} = & I_{ds0} + G_m v_{gs} + G_{ds} v_{ds} + G_{m2} v_{gs}^2 + \\
 & G_{md} v_{gs} v_{ds} + G_{d2} v_{ds}^2 + G_{m3} v_{gs}^3 + \\
 & G_{m2d} v_{gs}^2 v_{ds} + G_{md2} v_{gs} v_{ds}^2 + G_{d3} v_{ds}^3,
 \end{aligned} \tag{5.18}$$

donde I_{ds0} es la corriente de polarización, v_{ds} y v_{gs} son los voltajes incrementales de drenador-fuente y puerta-fuente, respectivamente, y los parámetros (G_m, \dots, G_{d3}) son los coeficientes relacionados con las derivadas de la característica I/V evaluadas en el punto de polarización. Por ejemplo, G_{md2} vendría dado por

$$G_{md2} = \frac{1}{2} \frac{\partial^3 I_{ds}}{\partial v_{gs} \partial v_{ds}^2} \bigg|_{(V_{gs0}, V_{ds0})}, \text{ en } v_{ds} = v_{gs} = 0 \tag{5.19}$$

Las derivadas de la característica I/V , evaluadas en un determinado punto de polarización, pueden ser obtenidas experimentalmente a partir de la medida de potencias de intermodulación para una señal de excitación compuesta por dos tonos [Pedro94]. De esta forma, es posible utilizar una red neuronal para obtener los coeficientes del desarrollo (5.18) a partir de las medidas de las derivadas, y reconstruir la función mediante (5.18). En resumen, el problema que se plantea es obtener un modelo que aproxime una función $f : R^2 \rightarrow R^{10}$

$$(\hat{I}_{ds0}, \hat{G}_m, \hat{G}_{ds}, \hat{G}_{m2}, \hat{G}_{d2}, \hat{G}_{md}, \hat{G}_{m3}, \hat{G}_{d3}, \hat{G}_{m2d}, \hat{G}_{md2}) = f(V_{ds0}, V_{gs0}), \tag{5.20}$$

donde las variables de entrada son las tensiones de polarización y las variables de salida son los 10 coeficientes del desarrollo de Taylor (5.18).

Como arquitectura neuronal, para este problema particular se ha seleccionado la red GRBF. La principal motivación es que los coeficientes del desarrollo (5.18) tienen una forma que se ajusta a las características de esta función. Por ejemplo, la Figura 5.4 muestra la forma del coeficiente G_{md} .

5.2.2.3 Arquitectura modular global

A partir de los módulos de pequeña y de gran señal, se propone un modelo global del transistor combinando ambos modelos tal y como muestra la Figura 5.5.

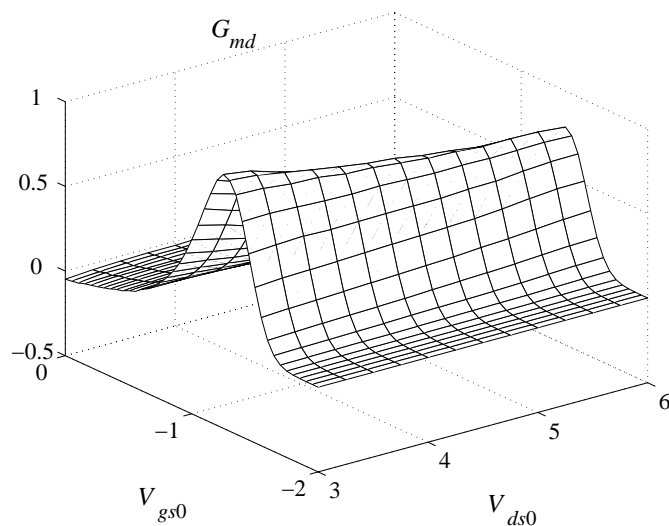


Figura 5.4: Medidas normalizadas de G_{md} en función del punto de polarización (V_{gs0}, V_{ds0}) para un transistor de microondas NE72084 MESFET.

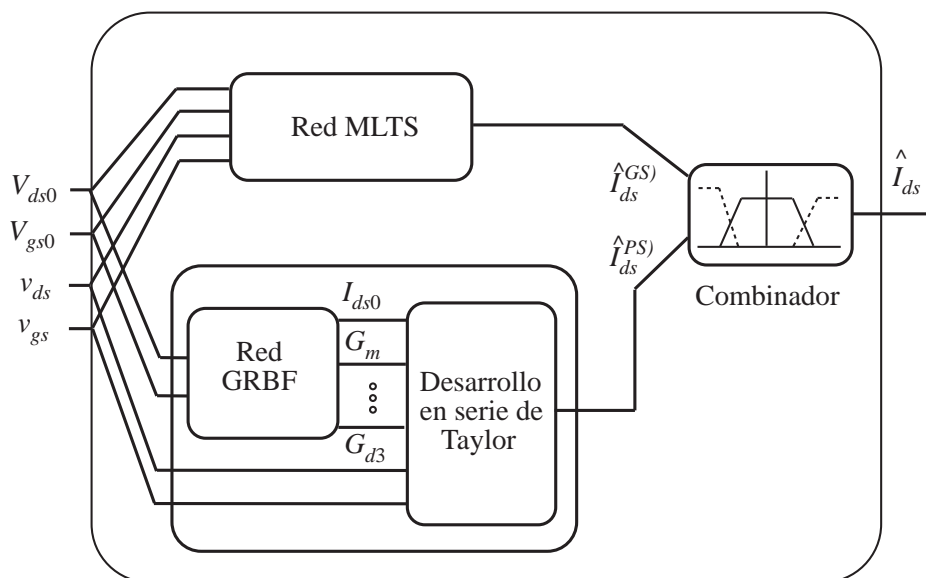


Figura 5.5: Modelo Global para el modelado de transistores de microondas

En este caso, el combinador se implementa utilizando funciones de pertenencia complementarias para ponderar la contribución de los modelos de pequeña y de gran señal. En función de los requerimientos de continuidad en la función y sus derivadas, es posible utilizar el mismo tipo de funciones definidas en la Sección 5.2.1.2. En esta aplicación, la continuidad de la función es importante. En las derivadas, dado que su importancia en el régimen gran señal no es tan determinante, esta restricción no es tan importante. Además, forzar la continuidad de las derivadas supone la aparición de oscilaciones de las mismas en la zona de transición, por lo que, en general, se prefiere utilizar funciones de pertenencia con una transición lineal, como, por ejemplo, una función trapezoidal. La forma concreta de estas funciones se discute posteriormente en la Sección 5.2.4.

5.2.3 Entrenamiento de la GRBF mediante un algoritmo EM

Como se ha visto en la sección anterior, en la arquitectura modular propuesta como modelo pequeña señal se emplea una red GRBF. Para este tipo de redes se ha propuesto un nuevo algoritmo de entrenamiento, basado en el algoritmo EM, que permite incrementar la velocidad de convergencia y disminuir la sensibilidad a los mínimos locales de este tipo de redes. En este apartado describimos este nuevo algoritmo.

5.2.3.1 Introducción al empleo del algoritmo EM

El algoritmo EM (*“Expectation-Maximization”*), formulado originalmente por Dempster, Laird y Rubin [Dempster77], es un método general para resolver problemas de estima de máxima verosimilitud de parámetros dado un conjunto incompleto de datos. El adjetivo incompleto hace referencia a que, mediante esta formulación, resulta conveniente asociar dos conjuntos de variables aleatorias con el problema, Y y Z , de los cuales sólo uno, Y , es directamente observable. Sin embargo, el modelo subyacente, se expresa en términos de Y y Z . Habitualmente, Y se denomina conjunto incompleto de datos, Z son las variables ocultas y el conjunto combinado de Y y Z se denomina conjunto completo de datos. El objetivo del algoritmo es encontrar el conjunto de parámetros, que denominaremos θ , que maximiza la verosimilitud de los valores observados de Y . El proceso de maximización de dicha verosimilitud se realiza a través de la distribución conjunta de los valores de Y y Z . Si el conjunto de datos incompletos Y tiene una función densidad de probabilidad $f_Y(y; \theta)$, entonces la función densidad de probabilidad de Z se puede expresar como

$$f_Z(z; \theta) = f_{Z|Y=y}(z; \theta) f_Y(y; \theta), \quad (5.21)$$

donde $f_{Z|Y=y}(z; \theta)$ es la función densidad de probabilidad condicional de Z dado $Y = y$. Tomando logaritmos

$$\log f_Y(y; \theta) = \log f_Z(z; \theta) - \log f_{Z|Y=y}(z; \theta). \quad (5.22)$$

Calculando ahora la esperanza condicional dado $Y = y$ para un valor de los parámetros $\theta = \theta'$

$$\log f_Y(y; \theta) = E\{\log f_Z(z; \theta) | Y = y; \theta'\} - E\{\log f_{Z|Y=y}(z; \theta) | Y = y; \theta'\}. \quad (5.23)$$

El resultado teórico más importante asociado con el algoritmo EM es que incrementar $\log f_Y(y; \theta)$ se reduce a incrementar el primer término del segundo miembro en (5.23). Esto permite maximizar la verosimilitud de los datos observados mediante el siguiente algoritmo iterativo [Dempster77]

Paso E: Estimar

$$E\{\log f_Z(z; \theta) | Y = y; \theta^{(n)}\} \quad (5.24)$$

Paso M: Calcular

$$\theta^{(n+1)} = \max_{\theta} E\{\log f_Z(z; \theta) | Y = y; \theta^{(n)}\} \quad (5.25)$$

donde $\theta^{(n)}$ es el conjunto de parámetros en la n -ésima iteración. Si (5.24) es una función continua tanto en θ como en $\theta^{(n)}$, el algoritmo converge a un punto estacionario de la función logaritmo de la verosimilitud [Wu83]. La maximización en (5.25) asegura que en cada iteración la verosimilitud se incrementa, convergiendo hacia un máximo. Este máximo puede no ser el máximo global, con lo que, en general, es preciso repetir el entrenamiento desde diferentes condiciones iniciales.

Un punto importante de este algoritmo es la selección de las variables ocultas. La elección de las mismas ha de conseguir simplificar el problema de optimización global de la verosimilitud. Dicha elección es claramente dependiente del problema.

5.2.3.2 Entrenamiento de redes RBF

Como ya se ha comentado, las redes de Funciones de Base Radial (RBF) se han convertido en una de las redes neuronales más populares, con aplicaciones en múltiples campos, como la regresión, clasificación y aproximación funcional [Haykin94, Bishop97]. La RBF aproxima una determinada función no lineal mediante una combinación lineal de funciones base gaussianas. El algoritmo de entrenamiento de una RBF debe estimar los centroides y las varianzas de cada función base y los pesos de la capa de salida. Habitualmente, el proceso de entrenamiento se separa en dos fases:

1. Se seleccionan los centroides y varianzas de las funciones base.
2. Los pesos de la capa de salida se estiman mediante un proceso de optimización lineal.

Las alternativas más simples para la selección de centroides y varianzas consisten en fijar inicialmente las varianzas y seleccionar los centroides de forma aleatoria [Broomhead88], o aplicando algoritmos de agrupamiento o cuantificación vectorial (“*clustering*”) [Moody89]. Otras alternativas tratan de solucionar este problema de

optimización utilizando procedimientos de entrenando supervisado, basados en gradiente, para estimar los parámetros de la red que minimicen una función de coste, generalmente el error cuadrático de la aproximación [Lowe89, Karayanis97, Santamaría99]. El principal inconveniente de estos métodos es que las técnicas de descenso de gradiente son, en general, computacionalmente costosas y, además, sufren el problema de los mínimos locales.

Como alternativa a las técnicas de optimización global, es posible utilizar el algoritmo EM (*“Expectation-Maximization”*) descrito anteriormente para obtener la estima de máxima verosimilitud de los parámetros de la red.

En la literatura de redes neuronales, el algoritmo EM ha sido utilizado en un gran número de problemas: aprendizaje supervisado/no supervisado, clasificación, aproximación funcional, etc. Teniendo en cuenta las características de nuestro problema, únicamente consideraremos su aplicación al aprendizaje o entrenamiento supervisado en problemas de aproximación funcional. En este contexto, Jordan y Jacobs han propuesto utilizar el algoritmo EM para entrenar una arquitectura de mezcla de expertos (*“Expert Mixtures”*) en problemas de regresión [Jordan94].

El algoritmo EM se ha aplicado también a la estima de la función densidad de probabilidad (FDP) conjunta modelada mediante una mezcla de Gaussianas (*“Gaussian Mixtures”*), para posteriormente estimar el regresor como la FDP condicional [Ghahramani94]. En ambos casos, las variables ocultas seleccionan el miembro más probable de la combinación dadas las observaciones, y, entonces, cada miembro se adapta de forma independiente.

Más recientemente, el algoritmo EM ha sido aplicado al entrenamiento de redes neuronales hacia delante y recurrentes [Ma97, Ma98]. El trabajo presentado en [Ma97] conecta con el trabajo previo de Feder y Weinstein sobre la estima de señales superpuestas en ruido [Feder88]. En ambos métodos, el conjunto de variables ocultas es el conjunto de las salidas individuales de cada neurona de la capa oculta (componentes de la señal en [Feder88]), y el paso E se reduce a descomponer en cada iteración el residuo total en N componentes, uno por neurona. En [Feder88] se muestra que las variables utilizadas para descomponer el residuo pueden ser arbitrarias, con la única restricción de que sumen 1, para garantizar la convergencia del algoritmo. Existen entonces diferentes alternativas: por ejemplo, en [Feder88], el residuo se descompone en N componentes iguales, mientras que en [Ma97] se descompone de forma proporcional a los pesos de la capa de salida. Ambas alternativas funcionan bien en redes hacia delante con funciones de activación de tipo global, tales como el MLP, pero tienden a ser lentas para redes con funciones de activación local, como la RBF, ya que cada función base se ve forzada a compensar el error en regiones alejadas de su dominio de activación.

Para evitar este inconveniente, en esta Tesis se propone un nuevo algoritmo EM de entrenamiento para redes RBF que permite incrementar la velocidad de convergencia [Lázaro2001b]. El algoritmo lleva a cabo una descomposición *suave* del residuo utilizando una secuencia de probabilidades a posteriori. De este modo, se tiene en cuenta el carácter local de las funciones de activación. Además, como se demostrará poste-

riormente, cuando se compara con alternativas de entrenamiento basadas en gradiente, el algoritmo propuesto presenta una menor sensibilidad a los mínimos locales.

5.2.3.3 Entrenamiento basado en el algoritmo EM de redes hacia delante

En esta sección se describen los métodos existentes para el entrenamiento de redes neuronales basados en el algoritmo EM. Sin pérdida de generalidad, se considera una red RBF con N neuronas, que aproxima una función unidimensional, $f(x) : \mathcal{R} \rightarrow \mathcal{R}$, de la forma siguiente

$$\hat{f}(x) = \sum_{i=1}^N \lambda_i o_i(x) \quad (5.26)$$

donde i indexa las neuronas de la red, λ_i es la amplitud, y $o_i(x)$ es la función de activación de cada neurona, que viene dada por

$$o_i(x) = \exp - \left(\frac{(x - \mu_i)^2}{2\sigma_i^2} \right). \quad (5.27)$$

El planteamiento del problema es el siguiente: dado un conjunto de entradas y sus correspondientes salidas, en general ruidosas, $\{x[n], y[n]\}$, estimar las amplitudes, λ_i , centroides, μ_i , y varianzas, σ_i^2 , de un modelo RBF. Las observaciones ruidosas se pueden caracterizar empleando en siguiente modelo

$$y[n] = \sum_i g_i(x[n]) + e[n] \quad (5.28)$$

donde $g_i(x[n]) = \lambda_i o_i(x[n])$ y, como es habitual en estos casos, se asume que $e[n]$ es un ruido blanco gaussiano de varianza σ^2 . Entonces, el logaritmo de la función de verosimilitud de los parámetros viene dado como

$$L(\mathbf{G}; \mathbf{x}, \mathbf{y}) = K - \frac{1}{2\sigma^2} \sum_n \left(y[n] - \sum_i g_i(x[n]) \right)^2 \quad (5.29)$$

donde K es una constante que puede obviarse en el proceso de optimización y

$$\mathbf{G} = (G_1, \dots, G_N), \text{ con } G_i = (\lambda_i, \mu_i, \sigma_i). \quad (5.30)$$

es el conjunto de parámetros del modelo.

De (5.29) se desprende que, asumiendo un ruido gaussiano, obtener la estima de máxima verosimilitud se reduce a un procedimiento convencional de minimización del error cuadrático medio de la aproximación. Este procedimiento de optimización no lineal multiparámetro puede resolverse mediante técnicas de descenso de gradiente, [Lowe89], [Karayanis97], [Santamaría99], cuyos principales inconvenientes ya se han mencionado.

Un procedimiento alternativo computacionalmente más eficiente para obtener las estimas de máxima verosimilitud es el algoritmo EM. En cada paso E, el algoritmo

calcula la esperanza de unas variables ocultas que no pueden ser observadas de forma directa a partir de la estima actual de los parámetros. En el paso M, se obtiene una nueva estima de los parámetros, y el proceso se itera. Cada iteración EM incrementa la verosimilitud, de forma que el algoritmo converge a un máximo (no necesariamente el máximo global) de la función de verosimilitud [Dempster77].

Un punto esencial del algoritmo EM es realizar una adecuada elección de las variables ocultas que pueden ayudar a simplificar el paso de maximización. Una elección particularmente útil de tales variables ocultas en nuestro problema es la propuesta en [Feder88]: cada observación se descompone en N variables ocultas de la forma

$$y_i[n] = g_i(x[n]) + e_i[n] \quad i = 1, \dots, N \quad (5.31)$$

donde los residuos $e_i[n]$ se obtienen también descomponiendo el residuo total

$$e[n] = y[n] - \sum_i g_i(x[n]), \quad (5.32)$$

en N componentes

$$e_i[n] = t_i[n]e[n] \quad i = 1, \dots, N. \quad (5.33)$$

En [Feder88], se muestra que las variables de desacoplo, $t_i[n]$ pueden ser escogidas de forma arbitraria, con la única restricción de que sumen 1. Por lo tanto, varias alternativas son posibles: en [Feder88] el residuo se descompone de forma equitativa entre todas las neuronas de la red

$$t_i[n] = \frac{1}{N}, \quad i = 1, \dots, N; \quad (5.34)$$

mientras que en [Ma97] se descomponen proporcionalmente a los pesos de la capa de salida

$$t_i[n] = \frac{\lambda_i^2}{\sum_j \lambda_j^2}, \quad i = 1, \dots, N. \quad (5.35)$$

Finalmente, utilizando (5.34) o (5.35) para descomponer el residuo, el algoritmo EM para el entrenamiento de redes neuronales hacia delante se puede resumir de la forma siguiente:

Paso - E: para $i = 1, \dots, N$ calcular

$$y_i[n] = g_i(x[n]) + t_i[n] \left(y[n] - \sum_j g_j(x[n]) \right) \quad (5.36)$$

Paso - M: para $i = 1, \dots, N$ estimar

$$(\lambda_i, \mu_i, \sigma_i^2) = \min_{G_i} \sum_n (y_i[n] - g_i(x[n]))^2 \quad (5.37)$$

donde el índice para denotar la iteración se ha omitido para simplificar la notación. Hay que destacar que el problema de entrenar una red con N neuronas de forma global se ha descompuesto en N problemas, más simples, de entrenar una única neurona.

5.2.3.4 Un algoritmo específico para el entrenamiento de RBF's mediante el EM

Las variables de desacoplo (5.34) o (5.35), constantes sobre todo el espacio de entrada, han mostrado ser muy efectivas en redes neuronales con funciones de activación no locales, tales como las funciones sigmoidales del MLP. Sin embargo, no resultan adecuadas para redes con funciones de activación locales, como la RBF. Para este tipo de redes la convergencia es lenta debido a que, con tales variables de desacoplo, en el paso M cada neurona debe ajustar un residuo que se extiende sobre todo el espacio de entrada, incluso fuera de la región de activación de la propia neurona. Intuitivamente, una localización del residuo asociado a cada gaussiana parece una alternativa más razonable que puede ayudar a acelerar la convergencia de la red. Esta es la idea que aprovecha el método que se propone para entrenar redes RBF. En particular, se propone una modificación que consiste en utilizar como variables de desacoplo las siguientes probabilidades a posteriori

$$t_i[n] = P(G_i|x[n], y[n]), \quad (5.38)$$

es decir, la probabilidad de la neurona i -ésima dado el n -ésimo patrón entrada-salida. Utilizando (5.38), el algoritmo lleva a cabo una descomposición suave del residuo teniendo en cuenta la naturaleza local de las funciones de activación. Se ha denotado esta modificación como EM-local como alternativa a las versiones convencionales del EM basadas en variables de desacoplo constantes en todo el espacio de entrada [Ma97, Feder88].

Ahora se considera la estima de (5.38): aplicando Bayes, las probabilidades a posteriori pueden ser estimadas mediante

$$t_i[n] = \frac{P(x[n], y[n]|G_i)}{\sum_j P(x[n], y[n]|G_j)}, \quad (5.39)$$

y las probabilidades $P(x[n], y[n]|G_j)$ se pueden obtener como

$$P(x[n], y[n]|G_j) = P(y[n]|G_j, x[n])P(x[n]|G_j). \quad (5.40)$$

Es interesante resaltar que las probabilidades $P(x[n]|G_j)$ son las variables clave responsables de introducir el carácter local de las neuronas de la red, ya que pueden ser estimadas como

$$P(x[n]|G_s) = \frac{o_s(x[n])}{\sum_j o_j(x[n])}. \quad (5.41)$$

Para estimar la salida dada la entrada y una determinada neurona, $P(y[n]|G_j, x[n])$, existen varias posibilidades en función del modelo que se asuma para los datos. Conviene aquí recordar que $y[n]$ puede descomponerse en un conjunto de variables ocultas de la forma

$$y[n] = \sum_i y_i[n], \quad (5.42)$$

y entonces considerar que

$$P(y[n]|G_j, x[n]) \propto \sum_i P(y_i[n]|G_j, x[n]). \quad (5.43)$$

Finalmente, por simplicidad, se asume para $P(y_i[n]|G_j, x[n])$ el siguiente modelo Gaussiano

$$P(y_i[n]|G_j, x[n]) \propto \frac{1}{\sqrt{2\pi\sigma_{ei}}} \exp -\frac{(y_i[n] - g_j(x[n]))^2}{2\sigma_{ei}^2}, \quad (5.44)$$

donde la varianza σ_{ei}^2 puede estimarse en cada iteración como

$$\sigma_{ei}^2 = \frac{\sum_n t_i[n]e^2[n]}{\sum_n t_i[n]}. \quad (5.45)$$

Si bien es cierto que, en este caso, considerar Gaussiana la FDP de los residuos asociados a cada neurona es una aproximación bastante grosera, en la práctica se ha podido comprobar que funciona muy bien.

Para resumir, el algoritmo de entrenamiento propuesto se puede describir mediante los siguientes pasos:

Paso 1: Inicialización: una alternativa sencilla consiste en fijar una varianza inicial igual para todas las neuronas y seleccionar la posición de los centroides mediante el algoritmo OLS [Chen91]. Las amplitudes se obtienen mediante un ajuste de mínimos cuadrados.

Paso 2: Utilizando (5.44), (5.43) y (5.41) se estiman las variables de desacoplo, $t_i[n]$, mediante (5.39).

Paso 3: A partir de estas variables se lleva a cabo el paso E del algoritmo EM, en el que se divide el residuo en N componentes y se obtienen los distintos $y_i[n]$.

Paso 4: Se aplica el paso M (maximización) obteniéndose los nuevos parámetros de la red minimizando (5.37) para cada neurona. La amplitud se calcula resolviendo un problema de mínimos cuadrados, mientras que el centroide y la varianza se pueden adaptar mediante técnicas de descenso de gradiente [Santamaría99].

$$\frac{\partial L}{\partial \mu_i} = - \sum_n e_i[n] g_i(x[n]) \frac{(x[n] - \mu_i)}{\sigma_i^2}, \quad (5.46)$$

$$\frac{\partial L}{\partial \sigma_i} = - \sum_n e_i[n] g_i(x[n]) \frac{(x[n] - \mu_i)^2}{\sigma_i^3}. \quad (5.47)$$

Paso 5: Volver al *Paso 2*: el algoritmo se detiene cuando se ha alcanzado una solución aceptable o cuando se supera un número máximo de iteraciones.

Es interesante recalcar que ahora cada neurona se adapta de forma separada, lo que simplifica el problema de optimización global y permite una sencilla paralelización. Por otro lado, la extensión a espacios multidimensionales es inmediata, tanto para redes RBF convencionales, como para redes GRBF, con varianzas diferentes en cada una de las direcciones del espacio de entrada.

A continuación, para mostrar las ventajas del método propuesto, se presentan algunos resultados obtenidos en diferentes espacios de entrada.

Ejemplo de simulación en 1D En este ejemplo se considera un conjunto de funciones unidimensionales (1D). Estas señales se generan mediante una combinación lineal de sinusoides con amplitudes fases y frecuencias aleatorias. Cada señal generada tiene 100 muestras. Se compara el funcionamiento del algoritmo EM-local propuesto con el de las alternativas convencionales basadas en el algoritmo EM [Feder88] y [Ma97], que se denotan como EM-1 y EM-2 respectivamente. En todos los experimentos se ha empleado una red RBF con 5 neuronas (15 parámetros), que se inicializa empleando el algoritmo OLS [Chen91], con diferentes valores iniciales de la varianza σ^2 , que es la misma para todas las neuronas. La Figura 5.6 presenta la evolución del error cuadrático medio, MSE, con el número de iteraciones, evaluada sobre el conjunto de señales limitadas en banda que se han empleado como funciones de test.

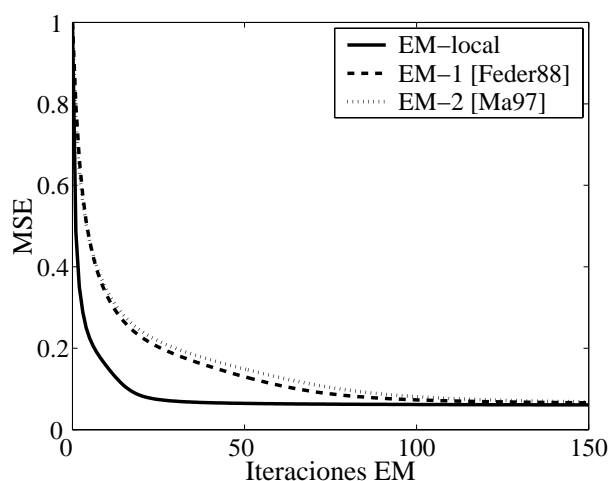


Figura 5.6: Evolución del error cuadrático medio MSE en un ejemplo 1D

Puede verse con claridad cómo la alternativa EM-local propuesta proporciona una convergencia más rápida que las alternativas EM convencionales no localizadas. El mismo comportamiento se ha observado para otras señales.

Ejemplo de simulación en 2D En este experimento se considera de nuevo como conjunto de prueba el conjunto de 8 funciones utilizado en [Cherkassky96] y que se presentan en el Apéndice F.

En este caso se utiliza una Red de Funciones de Base Radial Generalizada (GRBF), que permite diferentes varianzas para cada dirección del espacio de entrada. Se han utilizado redes con 10 neuronas (50 parámetros) inicializadas mediante el algoritmo OLS empleando diferentes valores iniciales para las varianzas σ^2 . Cuando se comparan las prestaciones del método propuesto con las alternativas EM convencionales, de nuevo se observa una importante mejora en la velocidad de convergencia, incluso más acusada que en el caso 1D. Este aspecto se ilustra en la Figura 5.7.

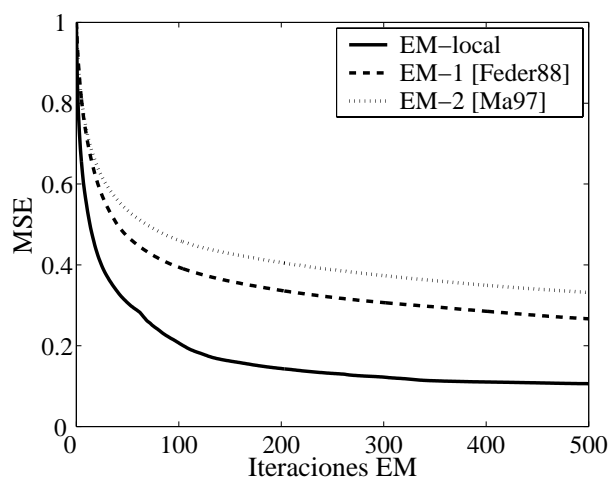


Figura 5.7: Evolución del error cuadrático medio MSE en un ejemplo 2D

Además de este incremento en la velocidad de convergencia, también se ha observado que el método EM-local presenta una menor sensibilidad al problema de los mínimos locales que las alternativas basadas en gradiente [Santamaría99]. Para ilustrar este punto se ha utilizado una red GRBF con 10 neuronas (50 parámetros) sobre la que se han evaluado estas dos alternativas sobre el mismo conjunto de prueba. Se han aplicado 500 iteraciones en ambos métodos, tanto en el de gradiente como en el EM-local. Para reducir la carga computacional de la alternativa EM-local, se aplica una única iteración de gradiente en cada paso M. De este modo, ambas alternativas presentan una carga computacional similar.

La Tabla 5.5 muestra el promedio de la relación señal a error (SER) en dB, obtenida para cada función con los dos algoritmos de entrenamiento. Se comprueba que, para la mayoría de las funciones, la alternativa EM-local ofrece unos resultados superiores a la alternativa basada en gradiente. La principal razón es que una minimización global mediante gradiente resulta atrapada con una mayor frecuencia en mínimos locales de la superficie de error, mientras que el método EM-local es capaz de evitar estos mínimos locales y, por tanto, proporcionar una mejor aproximación.

	Fun 1	Fun 2	Fun 3	Fun 4	Fun 5	Fun 6	Fun 7	Fun 8
Gradiente	18.0	23.5	26.7	5.6	11.1	19.7	12.8	10.4
EM-local	19.2	23.9	32.3	7.3	14.9	22.5	17.8	10.3

Tabla 5.5: SER promedio (en dB) para las diferentes funciones de prueba

5.2.4 Análisis de prestaciones del modelo global

Para analizar las prestaciones del modelo global propuesto para transistores de microondas, se ha aplicado el modelo a un transistor MESFET [Lázaro2000]. Los datos utilizados para el entrenamiento y validación del modelo se han obtenido empleando un modelo analítico [Navarro98] desarrollado a partir de un profundo estudio del comportamiento particular de un MESFET NEC NE72084.

Para el módulo gran señal se ha utilizado un MLTS con 12 funciones base, lo que supone 66 parámetros. El módulo de pequeña señal ha empleado una GRBF con 8 neuronas, con 114 parámetros. El entrenamiento de esta red se ha llevado a cabo utilizando el algoritmo EM propuesto en la Sección 5.2.3.

Para implementar el combinador se han definido dos funciones de pertenencia, una para cada módulo, en función de la distancia de las tensiones instantáneas, (v_{ds}, v_{gs}) , con respecto al punto de polarización, (V_{ds0}, V_{gs0}) . Definiendo la distancia

$$d = \sqrt{v_{ds}^2 + v_{gs}^2} \quad (5.48)$$

la función de pertenencia asociada al módulo de pequeña señal viene dada por

$$\mu_{PS}(d) = \begin{cases} 1, & |d| \leq d_1 \\ \frac{d_2 - d}{d_2 - d_1}, & d_1 < d < d_2 \\ 0, & |d| \geq d_2 \end{cases} \quad (5.49)$$

que es una función trapezoidal donde los parámetros d_1 y d_2 determinan la zona de transición entre módulos. La función de pertenencia para el módulo de gran señal es la función complementaria

$$\mu_{GS}(d) = 1 - \mu_{PS}(d). \quad (5.50)$$

La Figura 5.8 representa las funciones de pertenencia. En este caso se han seleccionado los valores $d_1 = 0.25$ y $d_2 = 0.3$ V.

La Figura 5.9 muestra la característica I/V para un punto de polarización, $V_{ds} = 3.5V$ y $V_{gs} = -1V$, y la aproximación que proporciona el modelo global propuesto. Se comprueba que la aproximación obtenida ajusta de forma adecuada la función I/V , con lo que el modelo representa correctamente el comportamiento gran señal del transistor. Numéricamente, la relación señal a error (SER) de la aproximación global, en todo el espacio de entrada considerado, es de 32.5 dB.

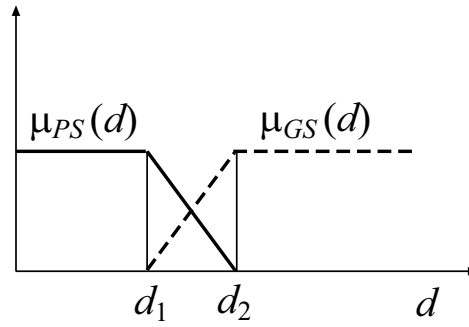


Figura 5.8: Funciones de pertenencia de los módulos gran señal y pequeña señal

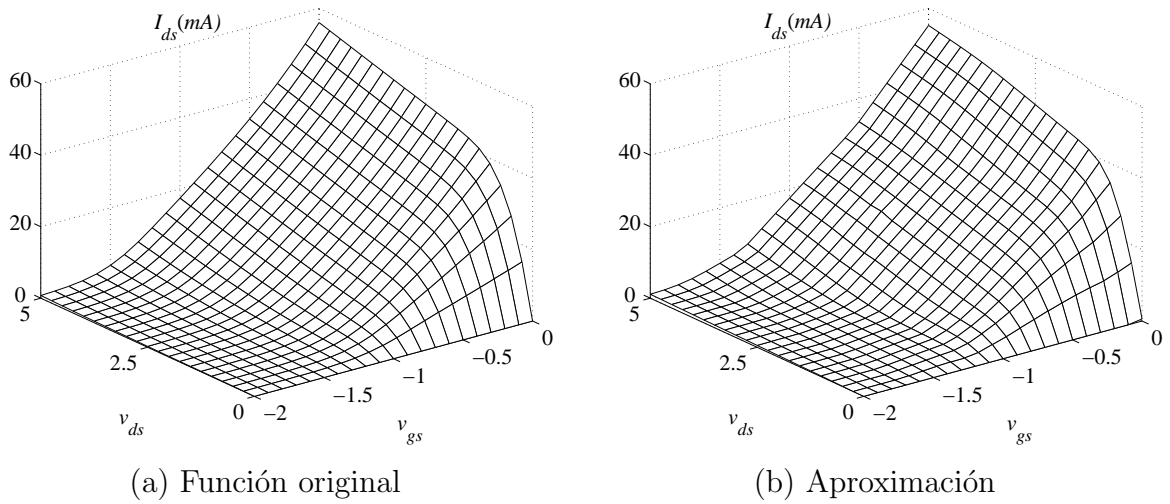


Figura 5.9: Característica I/V para un punto de polarización $V_{ds0} = 3.5V$ y $V_{gs0} = -1V$

Para mostrar el comportamiento en pequeña señal del modelo propuesto, la Figura 5.10 representa el parámetro G_m , definido como

$$G_m = \frac{1}{2} \frac{\partial I_{ds}}{\partial v_{gs}} \bigg|_{(V_{gs0}, V_{ds0})}, \text{ en } v_{ds} = v_{gs} = 0, \quad (5.51)$$

es decir, la derivada con respecto a v_{gs} en el punto de polarización. En la figura se muestran la función original y las aproximaciones que proporcionan el modelo gran señal y el modelo modular global. Se aprecia que la aproximación obtenida con el modelo gran señal es sensiblemente peor que la que proporciona el modelo global, mostrando así las ventajas de nuestro modelo modular. La degradación es más evidente a medida que aumenta el orden de las derivadas.

Resultados similares obtenidos utilizando otras arquitecturas neuronales, como el MLP, sugieren que una única red no es capaz de capturar la información necesaria para modelar de forma fiel tanto el comportamiento gran señal como el comportamiento pequeña señal. Sin embargo, utilizando una red modular como la propuesta, es posible capturar ambos comportamientos.

Para dar una idea numérica de la fidelidad de la aproximación obtenida, la Tabla 5.6 presenta la SER en dB de la aproximación de los coeficientes del modelo pequeña señal mediante la red GRBF.

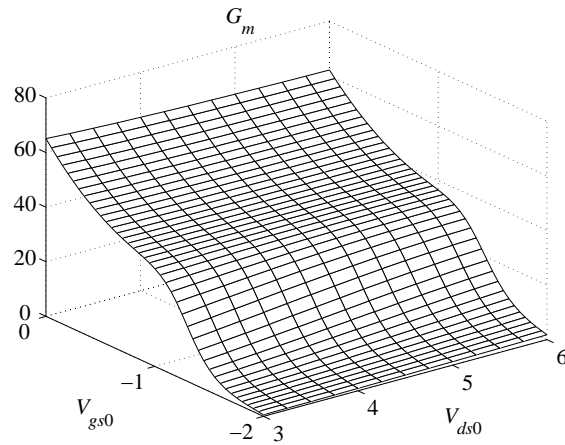
I_{ds0}	G_m	G_{ds}	G_{m2}	G_{md}	G_{d2}	G_{m3}	G_{m2d}	G_{md2}	G_{d3}
31.7	30.3	30.0	27.8	27.3	26.0	21.5	24.7	25.5	25.5

Tabla 5.6: Resultados obtenidos en la aproximación de los coeficientes de pequeña señal (SER en dB)

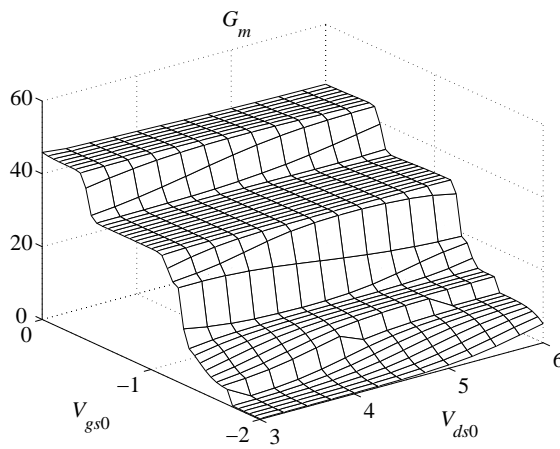
5.3 Discusión

En este capítulo hemos discutido dos alternativas para introducir información sobre las derivadas en el diseño de una red neuronal: modificando la función de coste y mediante una arquitectura modular específica para el problema.

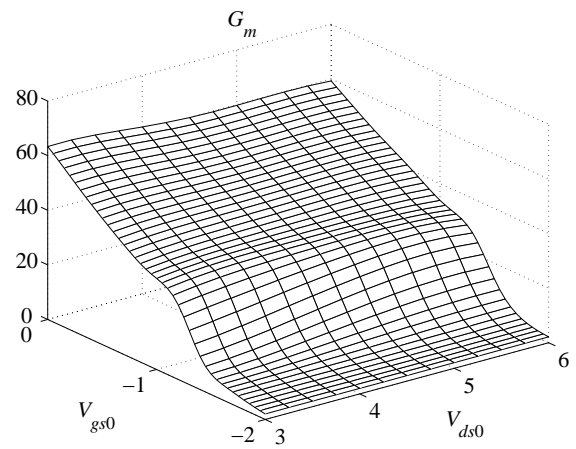
En cuanto a la primera alternativa, se ha comprobado que la introducción de la información de las derivadas en la función de coste del entrenamiento de redes neuronales proporciona varias ventajas. Por un lado, mejora apreciablemente la aproximación de las derivadas. Por otro lado, en situaciones de ruido, permite reducir el número total de muestras necesarias para obtener un determinado nivel de fidelidad en la reconstrucción de la función. Esto parece indicar que añadir información de las derivadas puede ser más importante, en el proceso de reconstrucción, que utilizar únicamente la información de la propia función. La explicación puede encontrarse en la diferente



(a) Función original



(b) Modelo gran señal



(c) Modelo global

Figura 5.10: Parámetro G_m original y las aproximaciones obtenidas con el modelo gran señal y con el modelo modular global

naturaleza de la información contenida en función y derivadas. Estas últimas proporcionan información sobre el comportamiento local de la función que puede ser muy beneficiosa en el proceso de reconstrucción.

No obstante, esta alternativa también presenta algunas limitaciones: a medida que aumenta la dimensión del espacio de entrada, la función de coste y el proceso de minimización se complican al tener que añadir nuevas derivadas, lo que limita su aplicación práctica a espacios de dimensión reducida.

También se ha demostrado que la información localizada de las derivadas se puede incluir en un modelo de redes neuronales mediante una arquitectura modular. El modelo propuesto para el modelado de transistores permite obtener un modelo global capaz de representar de forma adecuada tanto el comportamiento gran señal como el comportamiento pequeña señal de transistores. Los resultados mostrados en este capítulo para el modelo analítico de un transistor se extienden en el Capítulo 6, donde se aplica el modelo a datos reales de transistores MESFET y HEMT.

Finalmente, para el entrenamiento del modelo pequeña señal (implementado mediante una GRBF) se ha propuesto un nuevo algoritmo de entrenamiento, basado en el algoritmo EM, que permite acelerar la convergencia del entrenamiento de estas redes. Asimismo, reduce la sensibilidad a los mínimos locales del algoritmo de entrenamiento.

Capítulo 6

Aplicación al modelado de transistores de microondas

En este capítulo se presentan algunas aplicaciones de los métodos desarrollados en la Tesis al problema de modelado de transistores de microondas. Se han desarrollado varios tipos de modelos: modelos de gran señal, de pequeña señal y globales (válidos para entornos de gran señal y de pequeña señal). Asimismo, se han considerado tanto transistores MESFET como transistores HEMT.

6.1 Modelo gran señal de transistores HEMT bajo iluminación óptica utilizando un MLTS

En esta sección se presentan los resultados obtenidos en el modelado gran señal de un transistor HEMT bajo iluminación óptica [Lázaro2001]. Se comienza haciendo una breve introducción del problema que se plantea. A continuación, se muestran los resultados obtenidos utilizando el modelo MLTS para modelar un transistor real a partir de un conjunto de medidas experimentales del mismo, comparándose sus prestaciones con las obtenidas por otros modelos.

6.1.1 Introducción

Como ya se ha comentado anteriormente, el comportamiento en gran señal de un transistor viene determinado por la característica dinámica corriente-tensión (I/V) en pulsada que, además de las tensiones dinámicas instantáneas, v_{ds} , y v_{gs} , depende de las tensiones de polarización, V_{ds0} y V_{gs0} [Fernández96]. Para reproducir el comportamiento gran señal de un transistor es necesario modelar de forma adecuada esta característica, con lo que el modelo no lineal ha de tener la forma

$$\hat{I}_{ds} = f(V_{ds0}, V_{gs0}, v_{ds}, v_{gs}). \quad (6.1)$$

Por otro lado, las distintas ventajas de los sistemas de transmisión óptica y el creciente uso de las frecuencias de microondas en los sistemas de comunicaciones, junto con la capacidad para integrar componentes ópticos y de microondas en una única oblea de GaAs, han estimulado considerablemente el interés en el desarrollo de sistemas optoelectrónicos de microondas. Los circuitos ópticos tienen la ventaja de que pueden ser integrados en los circuitos de microondas sin interferir con ellos. Además, presentan bajas pérdidas, unas dimensiones reducidas, un tiempo de respuesta corto y un gran ancho de banda.

La iluminación directa de circuitos monolíticos de microondas y de ondas milimétricas resulta atractiva debido a la versatilidad de las aplicaciones relacionadas con los sistemas de comunicaciones de fibra óptica y los sistemas de control. Los transistores FET (*Field Effect Transistor*), que constituyen el elemento básico de los MMIC's, pueden ser utilizados como fotodetectores insertados en el propio chip monolítico y funcionar, de esta manera, como un puerto óptico. Por tanto, es interesante examinar el comportamiento de un FET presente en un circuito monolítico cuando se considera el efecto de la iluminación óptica.

Es conocido que cuando se ilumina un dispositivo de GaAs, aparece un interesante efecto de absorción en el espacio puerta-drenador y puerta-fuente [Madjar92, Calandra94]. De hecho, estos dispositivos exhiben efectos tanto fotoconductivos como fotovoltaicos que pueden ser convenientemente amplificados mediante circuitos resistivos externos. Esto significa que las curvas estáticas de DC, así como los parámetros del circuito equivalente en pequeña señal del transistor, varían cuando se aplica energía óptica al dispositivo.

Como ya se ha dicho, el comportamiento en gran señal de un transistor viene determinado por la característica dinámica corriente tensión (I/V) en pulsada, que depende del punto de polarización. Hasta nuestro conocimiento, el único modelo que considera los efectos de la iluminación óptica en este comportamiento dinámico, dependiente del punto de polarización, y responsable de la potencia de salida y de las características de eficiencia de estos dispositivos, es un método basado en funciones analíticas [Navarro98]. Este método se basa en una investigación exhaustiva del comportamiento dinámico en gran señal del dispositivo. Se estudian las diferentes dependencias que presenta dicho comportamiento y estas dependencias se ajustan por medio de un conjunto apropiado de funciones analíticas. Por ejemplo, en el caso particular de un transistor HEMT, se ha identificado una dependencia logarítmica con la potencia óptica y una dependencia en forma de tangente hiperbólica con la tensión en la puerta. Aunque este método obtiene resultados aceptables, tiene el inconveniente de su especificidad: para cada nuevo dispositivo hay que llevar a cabo un nuevo estudio completo y determinar las funciones analíticas óptimas para ajustar el conjunto específico de datos. Este método requiere un laborioso trabajo de análisis del dispositivo específico, por lo que claramente es deseable un método automatizado independiente del dispositivo. Para solucionar este problema se propone utilizar el modelo MLTS, presentado en el Capítulo 4, para modelar el comportamiento gran señal de estos dispositivos.

6.1.2 Resultados de modelado con el MLTS

Para modelar el comportamiento gran señal de un transistor HEMT bajo iluminación óptica, es preciso obtener un modelo de la forma

$$\hat{I}_{ds} = f(V_{ds0}, V_{gs0}, v_{ds}, v_{gs}, P_o) \quad (6.2)$$

donde V_{ds0} y V_{gs0} son las tensiones de polarización, v_{ds} y v_{gs} son las tensiones dinámicas de pulsada, y P_o es la potencia óptica aplicada sobre el dispositivo.

Se ha utilizado un modelo MLTS para modelar un transistor HEMT Philips D02AH ($4 \times 30 \mu\text{m}$), con una longitud de puerta de $0.2 \mu\text{m}$. Para el entrenamiento del modelo se dispone de un conjunto de medidas experimentales del transistor. Estas medidas han sido obtenidas por el Grupo de Microondas del Departamento de Ingeniería de Comunicaciones de la Universidad de Cantabria. El comportamiento del transistor se ha medido en una rejilla en la que las variables de entrada toman los siguientes valores:

- V_{ds0} : 0, 2 y 4 V.
- V_{gs0} : -0.75 , 0 y 0.75 V.
- v_{ds} : de 0 a 4 V en pasos de 0.25 V.
- v_{gs} : de -1 a 0.75 V en pasos de 0.25 V.
- P_o : 0, 0.25, 0.5, 0.75, 1, 2, 5 y 10 mW.

Con esta rejilla se dispone de un conjunto de 9792 muestras entrada/salida que se han utilizado para entrenar y evaluar el modelo MLTS. Los resultados obtenidos con este modelo se comparan con los que proporcionan un MLP, una GRBF, el modelo canónico lineal a tramos (MLT), y un modelo basado en funciones analíticas [Navarro98].

Para evaluar las prestaciones de cada modelo se ha utilizado como figura de mérito la relación señal a error (SER) de la aproximación expresada en dB.

En el entrenamiento de los modelos se han utilizado conjuntos de entrenamiento de 2000 muestras seleccionadas de forma aleatoria de entre los 9792 patrones totales. Se ha comprobado que utilizar un mayor número de muestras no proporciona mejores resultados, mientras que incrementa la carga computacional de forma considerable. Como conjunto de test, para evitar el sobreentrenamiento, se han utilizado las 7792 muestras restantes. En todos los casos la SER se evalúa sobre el conjunto completo de datos. El MLP se inicializa mediante el algoritmo de Nguyen y Widrow [Nguyen90] y se entrena mediante el algoritmo de retropropagación de errores con parámetro de paso adaptativo. La GRBF se inicializa mediante el algoritmo OLS [Chen91] y se entrena mediante técnicas de descenso de gradiente [Santamaría99]. El MLTS se inicializa y se entrena de acuerdo al algoritmo propuesto en el Capítulo 4, y el MLT mediante un algoritmo similar [Chua86].

La Tabla 6.1 compara los resultados obtenidos con los distintos modelos. El número entre paréntesis al lado del nombre del modelo indica el número de neuronas del mismo.

Se muestra el número de parámetros de cada modelo y la SER en la aproximación de I_{ds} y de sus primeras derivadas con respecto a las tensiones instantáneas (v_{ds} y v_{gs}). Para el transistor utilizado en este ejemplo, no se han medido estas derivadas. Se ha obtenido una estima de las mismas ajustando un spline cúbico en todo el conjunto de medidas y estimando las derivadas de dicho modelo. Se observa que, para un número similar de parámetros, el modelo MLTS proporciona los mejores resultados tanto para la función como para la derivada. Cuando se compara con el modelo analítico [Navarro98], incluso con un número menor de parámetros se obtienen resultados superiores, especialmente en el caso de la derivada con respecto a v_{ds} , donde el modelo analítico realiza una aproximación bastante pobre.

Modelo	Parámetros	I_{ds}	$\frac{\partial I_{ds}}{\partial v_{ds}}$	$\frac{\partial I_{ds}}{\partial v_{gs}}$
MLP(5)	36	25.50	12.00	14.40
MLT(5)	36	30.62	9.52	13.04
MLTS(5)	37	32.13	16.63	16.40
MLP(9)	64	27.17	12.62	14.75
MLT(10)	66	32.44	10.44	12.96
MLTS(10)	67	34.35	18.40	17.15
MLP(11)	78	28.09	13.71	14.66
MLT(12)	78	33.04	11.51	14.34
MLTS(12)	79	35.16	18.88	17.16
GRBF(8)	88	28.84	10.06	13.79
M. Analítico	98	32.92	5.15	16.53

Tabla 6.1: Comparación de resultados obtenidos en el modelado de un transistor HEMT Philips D02AH (SER en dB)

La Figura 6.1 muestra la función original y sus derivadas frente a la aproximación proporcionada por el modelo MLTS(12). Los resultados corresponden a un punto de polarización $V_{gs0} = 0V$ y $V_{ds0} = 2V$, con una potencia óptica incidente $P_o = 1mW$ y se presentan en función de las tensiones instantáneas v_{ds} y v_{gs} .

Otro aspecto importante a tener en cuenta es la carga computacional necesaria para llevar a cabo el entrenamiento del modelo. Una estima aproximada, dada por el tiempo necesario para llevar a cabo el entrenamiento de los modelos, se presenta en la Tabla 6.2. Los tiempos corresponden a la implementación de los modelos en MATLAB, utilizando un Pentium MMX 200 MHz y con criterios de parada similares para todos los modelos. Se comprueba que los modelos MLT y MLTS tienen una carga computacional sensiblemente inferior al MLP y la GRBF. En cuanto al método [Navarro98], hay que tener en cuenta que implica un exhaustivo análisis del dispositivo, lo que supone una enorme cantidad de tiempo. Además tiene el inconveniente de no ser un trabajo automatizado, con lo que hay que repetir el proceso para cada nuevo dispositivo a modelar.

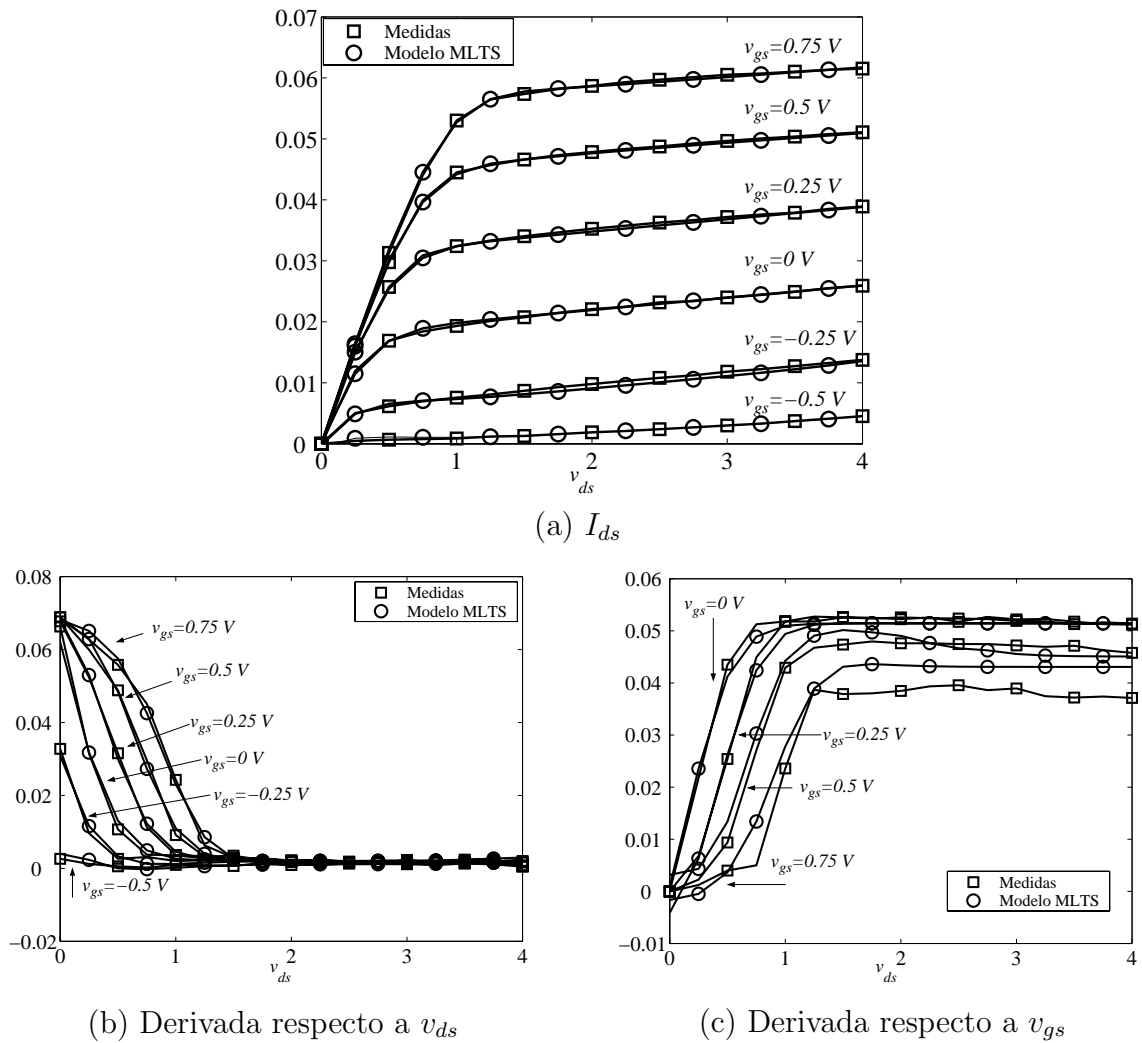


Figura 6.1: Medidas experimentales y resultados obtenidos con el modelo MLTS en un punto de polarización $V_{gs0} = 0V$ y $V_{ds0} = 2V$, con una potencia óptica incidente $P_o = 1mW$

Modelo	Tiempo de Entrenamiento (minutos)
MLP(5)	240
MLT(5)	8
MLTS(5)	12
MLP(9)	540
MLT(10)	14
MLTS(10)	22
MLP(11)	660
MLT(12)	17
MLTS(12)	26
GRBF(8)	180

Tabla 6.2: Carga computacional asociada a los distintos modelos expresada mediante el tiempo de entrenamiento

6.2 Modelo pequeña señal de transistores MESFET y HEMT para análisis de intermodulación empleando una GRBF

En este apartado se presentan los resultados del modelado pequeña señal de transistores MESFET y HEMT, donde es importante modelar el comportamiento de intermodulación. Se comienza con una introducción del problema general planteado y a continuación se presentan los resultados obtenidos al modelar transistores reales a partir de medidas experimentales con una GRBF.

6.2.1 Introducción

La predicción de los fenómenos de distorsión no lineal es muy importante en el ámbito de las microondas en la actualidad. Los nuevos sistemas de comunicaciones multiportadora han generado grandes esfuerzos en el desarrollo de técnicas de análisis y modelado en los niveles de dispositivo, circuito y sistema.

Centrándonos en el nivel de dispositivo, los transistores MESFET y HEMT son, sin duda, los más utilizados en aplicaciones de microondas. En sistemas multiportadora, el comportamiento no lineal de estos transistores produce distorsión de intermodulación (IMD). El modelado de los efectos de intermodulación de un dispositivo en pequeña señal es un problema importante y complicado. Los amplificadores funcionando por debajo del punto de 1 dB de compresión de ganancia, y los mezcladores excitados por señales de RF pequeñas, comparadas con el oscilador local, son ejemplos típicos en los que aparece este problema. En este caso, para modelar de forma adecuada los efectos de intermodulación de hasta tercer orden, que es lo que habitualmente se

requiere para estos dispositivos [Maas90], es preciso ajustar de forma adecuada no sólo la característica no lineal corriente-tensión (I/V), sino también sus derivadas hasta el mismo orden [Crosmun89, Maas91].

Las técnicas convencionales de modelado de este tipo de transistores, en general fallan a la hora de ajustar simultáneamente la función y las derivadas. Las técnicas lineales a tramos [Chua86], o basadas en funciones analíticas [Curtice85, McCamant90] proporcionan estimas adecuadas de la función, pero no de las derivadas de hasta tercer orden en la zona de operación de los dispositivos. La opción de utilizar tablas de datos [Meijer90, Root91] plantea la dificultad de su generación y gestión cuando se busca un adecuado compromiso entre la ganancia y la relación portadora-interferencia (C/I) del dispositivo.

Recientemente se han propuesto algunas soluciones utilizando redes neuronales para el modelado de dispositivos y circuitos no lineales [Rousset96, Shirakawa97, Zaabab95]. Sin embargo, prácticamente todas las soluciones propuestas se basan en el perceptrón multicapa (MLP), y la mayoría no tiene la capacidad de predecir los efectos de la intermodulación.

En esta Tesis se ha propuesto utilizar la red de funciones de base radial generalizada (GRBF) para el modelado pequeña señal de transistores MESFET y HEMT.

6.2.2 Caracterización no lineal de MESFET/HEMTs considerando el efecto de la intermodulación

Existen diferentes herramientas para el análisis no lineal de circuitos de microondas. Para un régimen de pequeña señal, la herramienta por excelencia ha sido el análisis mediante series de Volterra, y en particular la técnica de corriente no lineal. La descripción de una función de transferencia no lineal mediante series de Volterra se basa en una expansión mediante series de Taylor de las no linealidades del dispositivo alrededor de un punto de polarización en aplicaciones de amplificadores, y alrededor de una gran señal variante en el tiempo en el caso de mezcladores.

La Figura 6.2 muestra el circuito equivalente no lineal más extendido de un FET en la región de saturación (previamente presentado en la Figura 1.1). El elemento no lineal predominante es la fuente de corriente drenador-fuente (I_{ds}), que depende de las tensiones de puerta-fuente (V_{gs}) y drenador-fuente (V_{ds}). Como ya se ha comentado, cuando se aplica una pequeña señal de RF alrededor de un punto de polarización (V_{ds0}, V_{gs0}), la corriente I_{ds} puede expresarse mediante el desarrollo en serie de Taylor bidimensional

$$\begin{aligned} I_{ds}(V_{ds}, V_{gs}) = & I_{ds0} + G_m v_{gs} + G_{ds} v_{ds} + G_{m2} v_{gs}^2 + \\ & G_{md} v_{gs} v_{ds} + G_{d2} v_{ds}^2 + G_{m3} v_{gs}^3 + \\ & G_{m2d} v_{gs}^2 v_{ds} + G_{md2} v_{gs} v_{ds}^2 + G_{d3} v_{ds}^3, \end{aligned} \quad (6.3)$$

donde I_{ds0} es la corriente de polarización, v_{ds} y v_{gs} son los voltages incrementales de

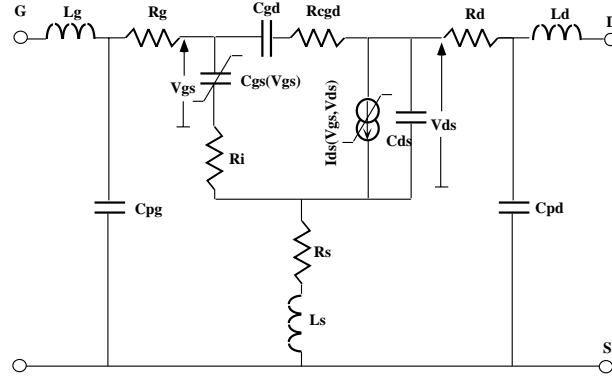


Figura 6.2: Circuito equivalente de un transistor FET

drenador-fuente y puerta-fuente, respectivamente, y los parámetros (G_m, \dots, G_{d3}) son los coeficientes relacionados con las derivadas de la característica I/V evaluadas en el punto de polarización, como (5.19). En la Figura 6.3 se representan funciones típicas de G_m y G_{m3} para transistores MESFET y HEMT en la región de saturación en función de las tensiones de polarización (V_{ds0}, V_{gs0}). El parámetro de transconductancia G_m fundamentalmente determina el comportamiento de la potencia de salida en función de V_{gs0} , mientras que el parámetro G_{m3} es el elemento predominante en la distorsión por los productos de intermodulación dentro de la banda ($2f_1 - f_2$ y $2f_2 - f_1$ en el caso de intermodulación de dos tonos de frecuencias f_1 y f_2).

En resumen, en pequeña señal el modelado de I_{ds}

$$\hat{I}_{ds} = f(V_{ds0}, V_{gs0}, v_{ds}, v_{gs}), \quad (6.4)$$

se puede plantear como un problema de aproximación de una función $f : R^2 \rightarrow R^{10}$, donde las variables de entrada son las tensiones de polarización, y las variables de salida son los 10 coeficientes del desarrollo en serie de Taylor (6.4). A partir de este mapeo, I_{ds} se reconstruye mediante dicho desarrollo.

En esta Tesis se ha propuesto utilizar una red GRBF para implementar la función $f : R^2 \rightarrow R^{10}$ antes mencionada, en el modelo pequeña señal de transistores. A la vista de la forma de los coeficientes G_i (Figura 6.3), esta red parece especialmente adecuada para el problema en cuestión. Esta intuición se refrenda a la vista de los resultados obtenidos que se presentan a continuación. Asimismo, los resultados se comparan con los obtenidos por las redes MLP y RBF.

6.2.3 Resultados de modelado de MESFETs y HEMTs

El modelo pequeña señal propuesto, basado en una GRBF, se ha utilizado para modelar un MESFET NE72084 y un HEMT D02AH en sus regiones de saturación [Santamaría99, García99b]. Las salidas del modelo

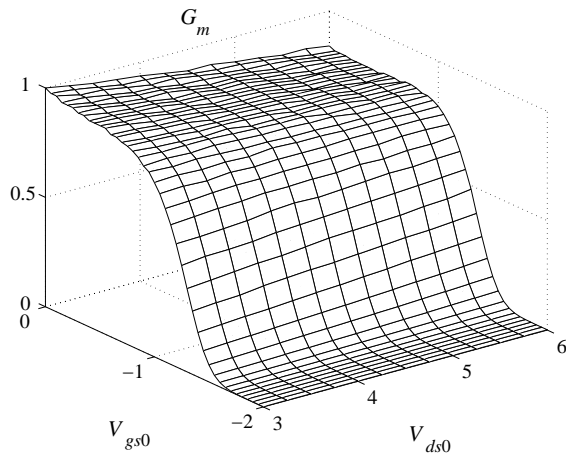
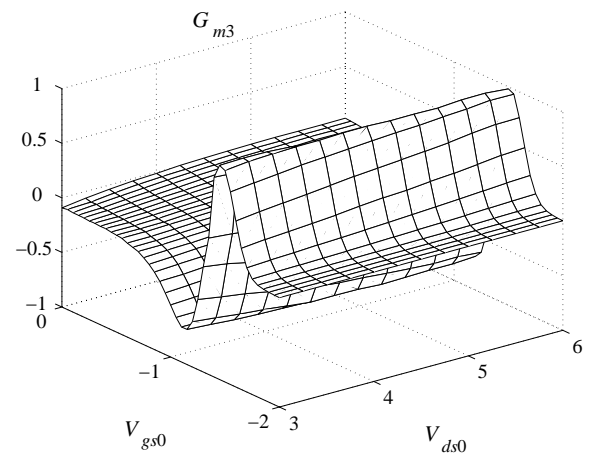
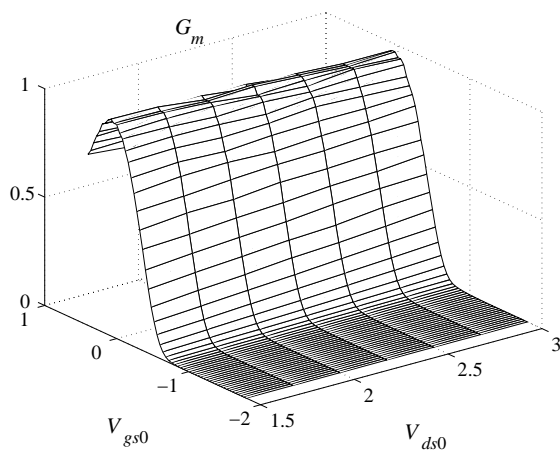
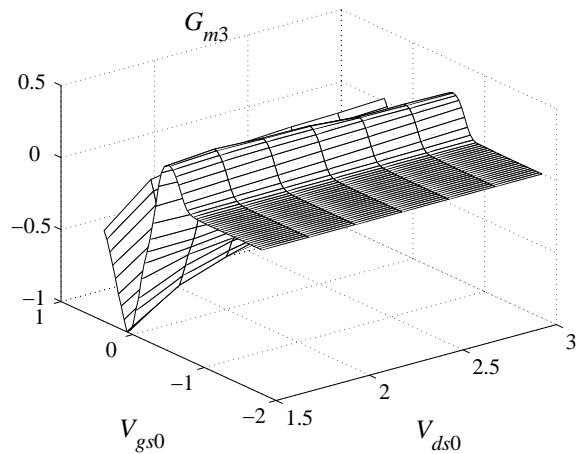
(a) G_m para un MESFET(b) G_{m3} para un MESFET(c) G_m para un HEMT(d) G_{m3} para un HEMT

Figura 6.3: Parámetros G_m y G_{m3} (normalizados) típicos para transistores MESFET y HEMT

$$(I_{ds0}, G_m, G_{ds}, G_{m2}, G_{d2}, G_{md}, G_{m3}, G_{d3}, G_{m2d}, G_{md2}),$$

se han medido para distintos puntos de polarización. En concreto, para el MESFET se han medido en una rejilla

- V_{ds0} : de 3 a 6 V en pasos de 0.25 V
- V_{gs0} : de -2 a 0 V en pasos de 0.05 V

Se mide en más puntos en V_{gs0} debido a la fuerte dependencia no lineal de todos los parámetros de salida con esta tensión de polarización. En total resulta un conjunto de 533 patrones entrada-salida.

Para el HEMT, las salidas del modelo se han medido en una rejilla

- V_{ds0} : de 1.5 a 3 V en pasos de 0.25 V
- V_{gs0} : de -1.8 a 0.4 V en pasos de 0.04 V

lo que supone un conjunto de 392 patrones de entrenamiento.

Los valores experimentales de los parámetros de primer orden, G_m y G_{ds} , se han obtenido a partir de los parámetros de scattering, S , utilizando la técnica de Dambrine [Dambrine88]. Los parámetros de orden mayor se han obtenido a partir de las medidas de potencias de intermodulación para una señal de excitación compuesta por dos tonos [Pedro94].

El conjunto de medidas, en este caso, no se ha dividido en un conjunto de entrenamiento y otro de test. Se trata de un conjunto de medidas relativamente pequeño, y, en general, la mayoría de las medidas tienen un bajo nivel de ruido. Además se han utilizado modelos con un reducido número de funciones base suaves. Por tanto, parece razonable pensar que utilizar todas las medidas en el entrenamiento de los modelos no debe afectar a la capacidad de generalización de los mismos.

Se comparan los resultados obtenidos con tres redes neuronales diferentes: la red RBF, el MLP y la GRBF propuesta. Se han comparado modelos con un número similar y reducido de parámetros. La figura de mérito utilizada para comparar las prestaciones de cada modelo es la relación señal a error (SER), expresada en dB, de la estima. En concreto, se han utilizado los siguientes modelos

MLP(N): MLP con N neuronas en la capa oculta. La red se inicializa con el algoritmo de inicialización propuesto en [Nguyen90]. Se ha entrenado con el algoritmo de retropropagación de errores utilizando un paso adaptativo (con un valor inicial de $\mu = 0.0001$). El número de iteraciones de entrenamiento se ha limitado a 5000, suficiente para lograr la convergencia.

RBF(N): RBF con N neuronas. Los centroides se seleccionan utilizando el algoritmo OLS [Chen91]. Se ha utilizado una varianza fija $\sigma^2 = 0.125$.

GRBF(N): GRBF con N neuronas. Se inicializa la red con el algoritmo OLS, con una varianza inicial aleatoria de valores acordes con el espacio de entrada: $\sigma^2 = (0.025, 0.1)$ para el MESFET y $\sigma^2 = (0.05, 0.2)$ para el HEMT. Los centroides y varianzas se actualizan mediante un algoritmo de gradiente [Santamaría99]. El número de iteraciones se ha limitado a 500, número suficiente para que el algoritmo converja.

Las Tablas 6.3 y 6.4 comparan los valores de SER en dB obtenidos con los tres modelos para un MESFET y un HEMT, respectivamente. Se observa que, para modelos con un reducido número de parámetros, la red GRBF propuesta presenta unos resultados superiores al resto de alternativas. Además, un MLP necesita al menos el doble de parámetros para obtener resultados similares en el modelado del MESFET. En el caso del HEMT, la diferencia de prestaciones es menor, pero la red GRBF proporciona, en cualquier caso, mejores resultados.

	Param	I_{ds0}	G_m	G_{ds}	G_{m2}	G_{md}	G_{d2}	G_{m3}	G_{m2d}	G_{md2}	G_{d3}
GRBF(8)	112	22.8	27.2	25.4	17.0	17.9	19.1	18.5	18.0	16.1	15.7
RBF(10)	130	12.0	12.0	12.3	6.4	6.3	10.7	2.3	3.5	3.6	9.5
MLP(8)	112	18.7	30.0	24.6	13.5	13.6	13.5	8.9	9.9	10.7	14.0
MLP(16)	224	20.0	31.7	29.1	16.3	16.3	14.4	14.3	15.4	14.2	16.7

Tabla 6.3: Resultados para un MESFET NE 72084

	Param	I_{ds0}	G_m	G_{ds}	G_{m2}	G_{md}	G_{d2}	G_{m3}	G_{m2d}	G_{md2}	G_{d3}
GRBF(8)	112	24.3	29.8	28.7	24.1	26.2	21.5	17.5	16.0	16.7	10.8
RBF(10)	130	18.4	19.8	20.7	14.6	13.9	19.8	14.5	13.9	14.8	15.0
MLP(8)	112	24.7	30.5	27.4	19.8	24.6	21.1	16.2	14.1	13.0	13.1

Tabla 6.4: Resultados para un HEMT D02AH

Finalmente, para verificar la validez del modelo propuesto para modelar el comportamiento de distorsión no lineal debido a la fuente de corriente I_{ds} , se ha calculado la potencia de salida y la relación portadora-interferencia C/I utilizando los datos experimentales y las derivadas extraídas del modelo propuesto. Como se muestra en [Pedro94, Webster96], la mayoría de los modelos existentes fallan al reproducir el comportamiento C/I al variar la carga. Por tanto, no se pueden utilizar para obtener valores de carga óptimos para diseños de baja distorsión. Este inconveniente se debe a la incapacidad de ajustar no sólo la derivada tercera predominante en los efectos de intermodulación, G_{m3} , sino también de los términos cruzados de tercer orden.

Se ha realizado un típico análisis mediante dos tonos, con frecuencias de 10 y 10.01 GHz y niveles de potencia por debajo del punto de compresión de 1 dB, con lo que se trabaja en régimen de pequeña señal, donde el comportamiento de distorsión no lineal es, en general, modelado de forma pobre. Se ha seleccionado un punto de

polarización comúnmente utilizado en las aplicaciones de amplificadores de la Clase A, $V_{gs} = -0.2V$ y $V_{ds} = 3V$. La Figura 6.4 compara los contornos de la potencia de salida para el MESFET NE72084 en función de la carga, utilizando los parámetros medidos y el modelo GRBF(8) propuesto. El casi perfecto ajuste muestra el correcto modelado de los coeficientes de primer orden, G_m y G_{ds} .

En la Figura 6.5 se representan los contornos para la relación C/I , que expresa la diferencia de potencias entre una de las señales de referencia (f_1 o f_2) y los productos de intermodulación adyacentes ($2f_1 - f_2$ o $f_1 - 2f_2$). La predicción que proporciona el modelo propuesto es significativamente fiel, lo que supone una reproducción fiable de los coeficientes de segundo y tercer orden, que no se obtiene con los métodos tradicionales de modelado. Esta predicción valida este método para un control preciso de la IMD en función del punto de polarización y de la carga en amplificadores de la clase A.

6.3 Modelo global de un MESFET

En esta sección se presentan algunos resultados obtenidos con el modelo global (pequeña señal + gran señal) propuesto en la Sección 5.2.2.3, aplicado a un transistor MESFET NE72084.

6.3.1 Módulo gran señal

Como ya se ha visto, el módulo gran señal del modelo global propuesto en esta Tesis aproxima, utilizando un MLTS, la función

$$\hat{I}_{ds}^{GS} = f(V_{ds0}, V_{gs0}, v_{ds}, v_{gs}), \quad (6.5)$$

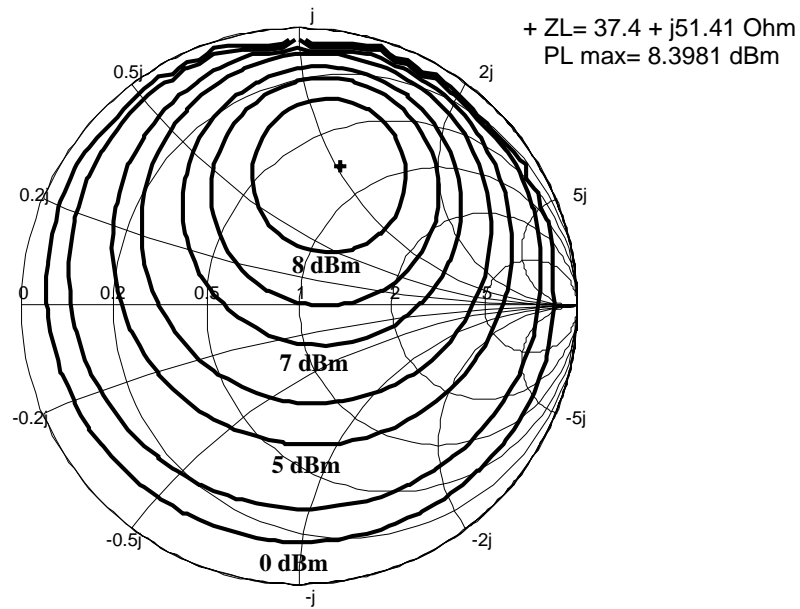
donde V_{ds0} y V_{gs0} son las tensiones de polarización, y v_{ds} y v_{gs} son las tensiones dinámicas de pulsada.

Para el transistor NE72084, se dispone de un conjunto de medidas obtenidas en una rejilla en la que las variables de entrada toman los siguientes valores.

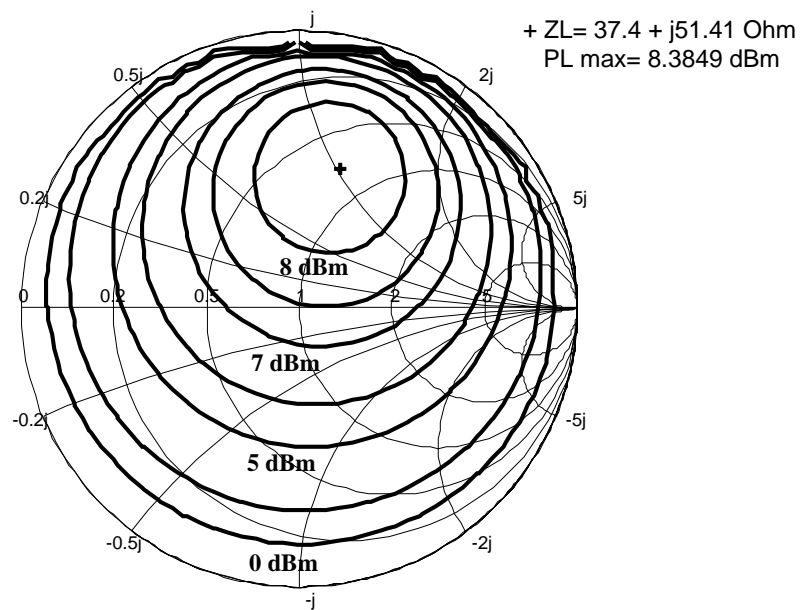
- V_{ds0} : de 0 a 1 V en pasos de 0.25 V y de 2 a 5 V en pasos de 1 V.
- V_{gs0} : de -2 a 0 V en pasos de 0.5 V.
- v_{ds} : de 0 a 1 V en pasos de 0.25 V, 1.5 V, y de 2 a 5 V en pasos de 1 V.
- v_{gs} : de -2 a 0 V en pasos de 0.25 V.

Con esta rejilla se dispone de un conjunto de 4050 muestras entrada/salida que se han utilizado para entrenar y evaluar el modelo MLTS. Se han seleccionado conjuntos de entrenamiento de 750 muestras seleccionadas de forma aleatoria.

En este caso, se ha considerado un modelo MLTS con 12 funciones base, lo que supone un total de 79 parámetros. La SER obtenida con este modelo en la aproximación del conjunto completo de datos es de 32.7 dB.

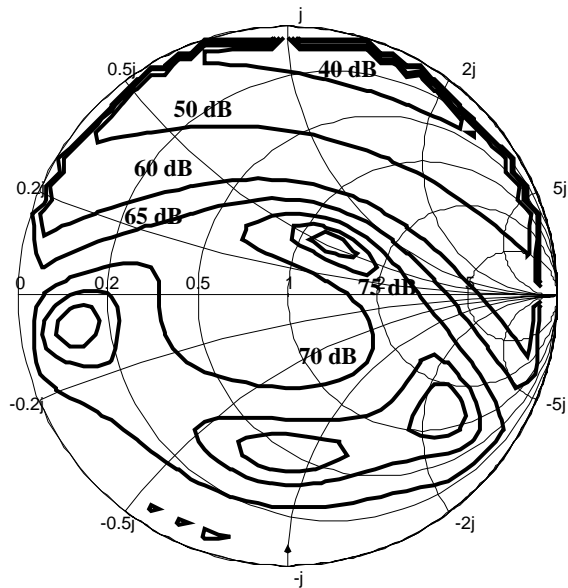


(a) Utilizando los parámetros medidos

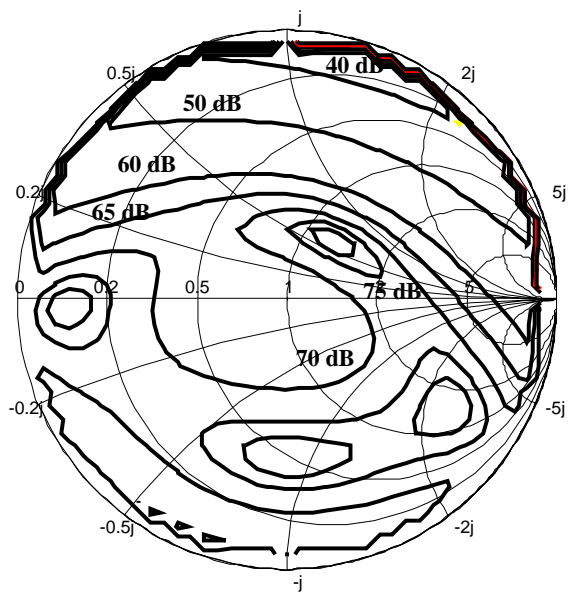


(b) Utilizando el modelo GRBF(8)

Figura 6.4: Potencia de salida en función de la carga para un MESFET NE72084



(a) Utilizando los parámetros medidos



(b) Utilizando el modelo GRBF(8)

Figura 6.5: Contornos de la relación C/I en función de la carga para un MESFET NE72084

6.3.2 Módulo pequeña señal

Para el módulo pequeña señal del transistor, en este caso, se ha utilizado una red GRBF entrenada con el algoritmo EM presentado en la Sección 5.2.3.

El conjunto de datos disponibles es el mismo que en el Apartado 6.2. Igualmente, se ha utilizado todo el conjunto de medidas como conjunto de entrenamiento, y se ha considerado una red con 8 neuronas (112 parámetros).

Los resultados obtenidos con el módulo pequeña señal considerado, expresados mediante la SER de la aproximación de los distintos coeficientes, se presentan en la Tabla 6.5.

I_{ds0}	G_m	G_{ds}	G_{m2}	G_{md}	G_{d2}	G_{m3}	G_{m2d}	G_{md2}	G_{d3}
24.7	26.5	26.4	24.1	23.7	18.2	21.3	21.0	18.9	16.0

Tabla 6.5: Resultados del módulo pequeña señal, entrenado con el algoritmo EM, para un MESFET NE 72084

6.3.3 Modelo global

A partir de los módulos gran señal y pequeña señal mostrados con anterioridad, se obtiene el modelo global del transistor. Para ello se utiliza un combinador con funciones de pertenencia trapezoidales

$$\mu_{PS}(d) = \begin{cases} 1, & |d| \leq d_1 \\ \frac{d_2 - d}{d_2 - d_1}, & d_1 < d < d_2 \\ 0, & |d| \geq d_2 \end{cases}, \quad (6.6)$$

donde los parámetros d_1 y d_2 determinan la zona de transición entre módulos, y

$$\mu_{GS}(d) = 1 - \mu_{PS}(d). \quad (6.7)$$

Estas funciones de pertenencia se representaron en la Figura 5.8. En este caso, se han seleccionado los valores $d_1 = 0.25V$ y $d_2 = 0.3V$.

Este modelo global proporciona una adecuada caracterización del comportamiento pequeña señal del transistor, gracias al correcto ajuste de los coeficientes de las derivadas de la característica I/V . Los parámetros d_1 y d_2 seleccionados son lo suficientemente grandes para que, en el régimen habitual de pequeña señal en este tipo de transistores, la salida del modelo global la proporcione el módulo pequeña señal.

En cuanto al comportamiento gran señal, la Figura 6.6 compara la característica gran señal medida y la aproximación obtenida con el modelo global, para un punto de polarización $V_{ds0} = 4V$ y $V_{gs0} = -1V$.

Finalmente, para mostrar que el modelo global desarrollado es válido para un análisis de intermodulación, la Figura 6.7 representa al espectro de la señal I_{ds} obtenido con el modelo cuando a la entrada del mismo se introducen 2 tonos de frecuencias

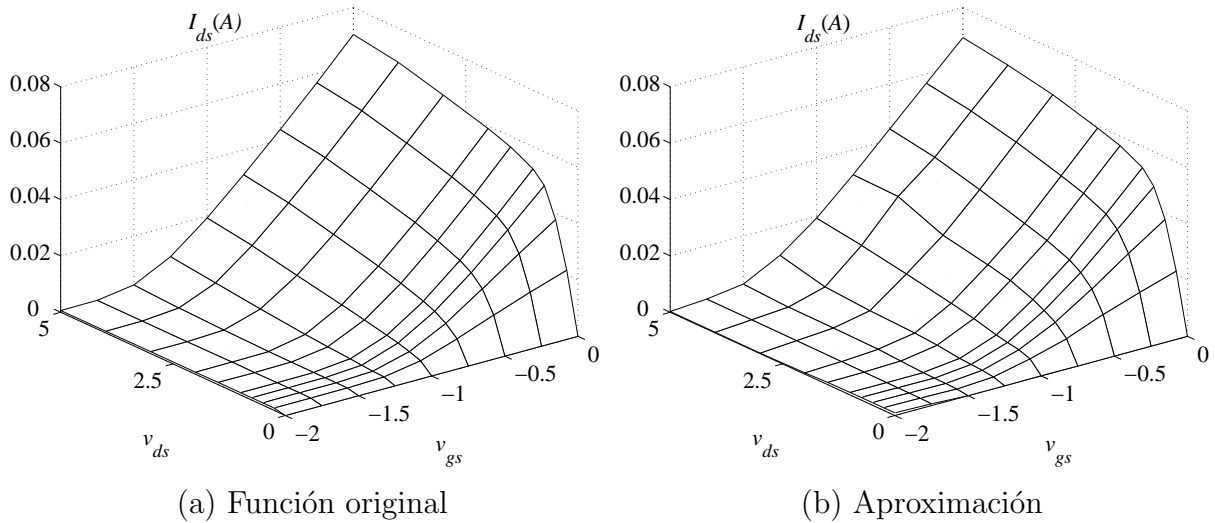


Figura 6.6: Característica I/V para un punto de polarización $V_{ds0} = 4V$ y $V_{gs0} = -1V$

$f_1 = 10$ GHz y $f_2 = 10.5$ GHz, con una potencia de -20 dBm, que está por debajo del punto de compresión de 1 dB, lo que asegura que se trabaja en un entorno de pequeña señal. Aunque para las medidas de intermodulación se suele utilizar una distancia menor en la frecuencia de los dos tonos, típicamente 1 MHz, en este caso se han considerado 5 MHz por una mera cuestión de representación, para apreciar mejor en la figura cada componente. Se observa que en la salida aparecen, además de las dos frecuencias de entrada, para las que se toma el nivel de referencia de 0 dB, las componentes de los productos de intermodulación de segundo y de tercer orden. Se puede comprobar cómo los productos $2f_1 - f_2$ y $2f_2 - f_1$ caen en la banda de la señal, con lo que no pueden ser eliminados mediante filtrado. Hay que tener en cuenta que la amplitud de los productos de intermodulación dependen de los coeficientes G_i del modelo pequeña señal: el hecho de que estos coeficientes se determinen de forma precisa a partir de medidas de productos de intermodulación permite obtener un modelo que representa de forma adecuada este fenómeno.

6.4 Modelo gran señal de un transistor con el MLR

En esta sección se presentan algunos de los resultados obtenidos al utilizar el MLR, propuesto en el Capítulo 3 para el modelado gran señal de transistores. Se propone utilizar el modelo MLR en espacios bidimensionales para realizar el modelo gran señal de un transistor, en un determinado punto de polarización (V_{ds0}, V_{gs0}) , a partir de las medidas de I_{ds} y de sus derivadas para distintos valores de las tensiones dinámicas instantáneas v_{ds} y v_{gs} .

En este caso, se ha utilizado el modelo de grado 2 para modelar un MESFET NE72084. Los valores de I_{ds} y de sus derivadas con respecto a v_{ds} , v_{gs} , y la cruzada

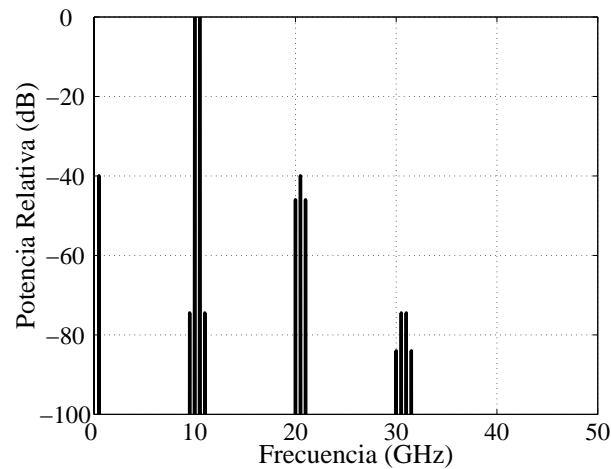
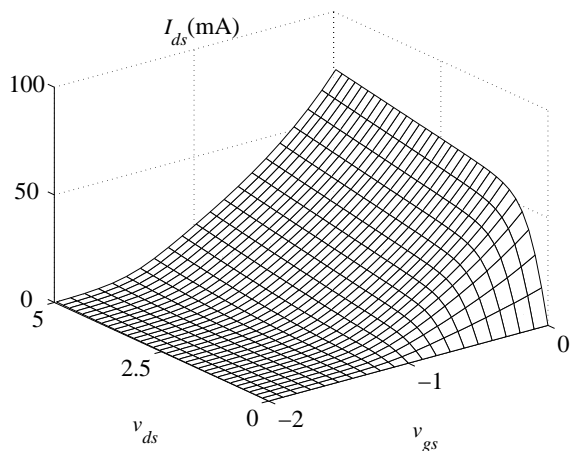


Figura 6.7: Productos de intermodulación en la característica I/V del modelo global

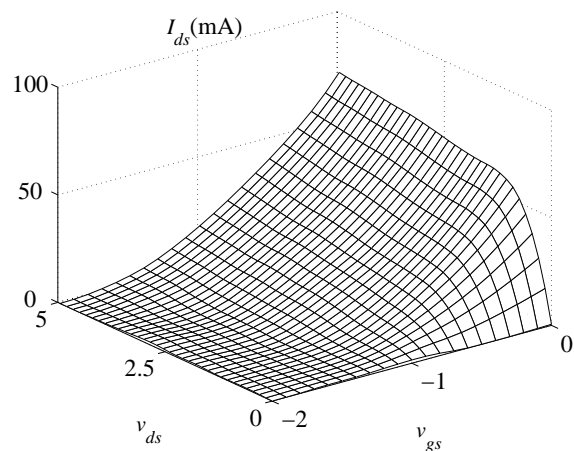
con respecto a ambas tensiones, se han obtenido mediante un modelo analítico del NE72084 [Navarro98].

La Figura 6.8 compara la aproximación obtenida con el modelo MLR con la función y derivadas originales, para un punto de polarización $V_{ds0} = 4V$ y $V_{gs0} = -1V$, y seleccionando 5 instantes de muestreo equiespaciados en cada una de las direcciones del espacio de entrada. Esto supone un total de 100 muestras para este punto de polarización. La SNR en las muestras de función y derivadas es de 30 dB. Se puede observar que con el modelo MLR se obtiene un ajuste adecuado. Numéricamente, en este caso se obtiene una SER de aproximadamente 30 dB en la reconstrucción de la función, mientras que en la derivada con respecto a v_{ds} se tienen 21.3 dB, 23.5 dB en la derivada con respecto a v_{gs} , y 14.9 dB en la derivada cruzada.

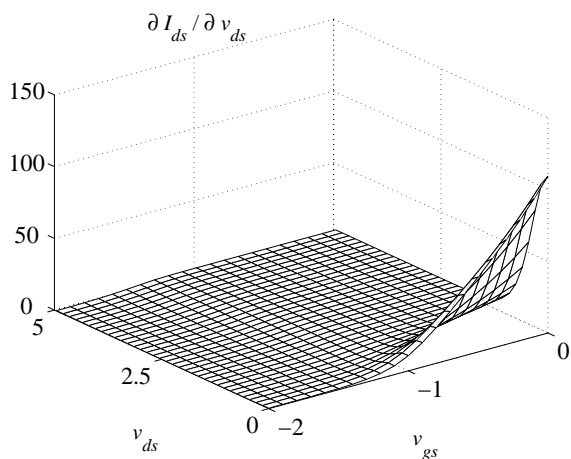
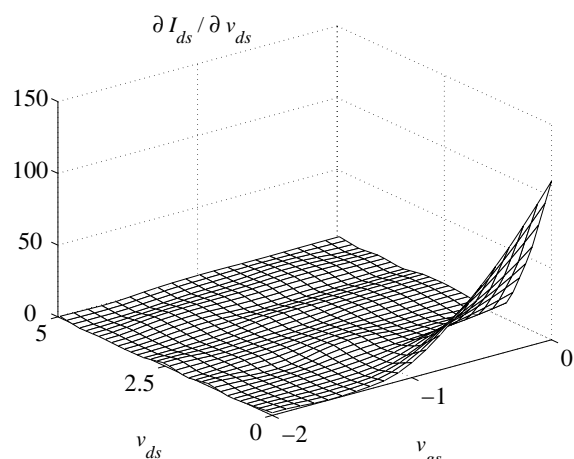
Con esta alternativa se obtiene un modelo gran señal del transistor, para un punto de polarización específico, que es capaz de proporcionar resultados adecuados. El modelo no es parsimonioso, sino que requiere un número relativamente elevado de parámetros. Por otro lado, en la actualidad no es común medir las derivadas de la característica I/V de un transistor en pulsada, sino que este tipo de medidas se restringe al modelo pequeña señal y en el punto de polarización. En gran parte, esto se debe a la falta de referencias acerca del significado físico de las medidas de las derivadas de la característica pulsada. No obstante, desde un punto de vista puramente de modelado, si se dispone de estas derivadas, es posible utilizar esta información en la reconstrucción de la característica I/V del transistor, como se ha podido comprobar.



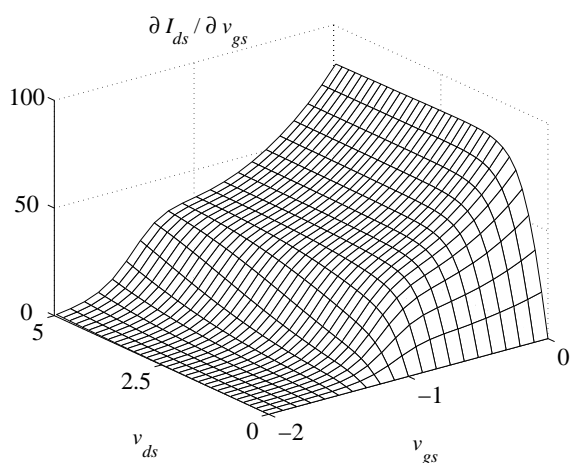
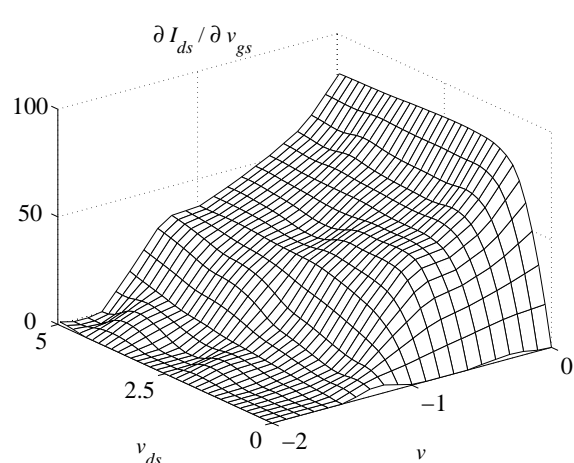
(a) Función original



(b) Aproximación

(c) $\frac{\partial I_{ds}}{\partial v_{ds}}$ original

(d) Aproximación

(e) $\frac{\partial I_{ds}}{\partial v_{gs}}$ original

(f) Aproximación

Figura 6.8: Característica I/V , y sus derivadas, para un punto de polarización $V_{ds0} = 4V$ y $V_{gs0} = -1V$, modelada mediante el modelo MLR

Capítulo 7

Conclusiones y líneas futuras de trabajo

En esta Tesis se han presentado distintas técnicas para la reconstrucción, aproximación y modelado de una función sometida a una serie de restricciones en sus derivadas. Los distintos modelos propuestos se han aplicado al modelado de transistores de microondas, en concreto transistores MESFET y HEMT.

A continuación se resumen las principales aportaciones realizadas en esta Tesis y las conclusiones que se han extraído sobre las mismas. Para finalizar, se presentan algunas líneas futuras de investigación que se consideran de interés.

7.1 Aportaciones

En el Capítulo 3 se ha presentado una nueva técnica para la interpolación simultánea de un conjunto de muestras de una función y de sus derivadas. La solución obtenida pertenece al subespacio de funciones splines. Las principales aportaciones presentadas en este capítulo son las siguientes:

- La formulación de un método de interpolación de las muestras de una función y sus derivadas en un espacio de entrada 1D. Esta formulación admite una implementación mediante un banco de filtros MIMO en el que todos sus componentes son filtros FIR, de modo que la reconstrucción de la función se realiza de forma local.
- La regularización del conjunto de muestras de la función y sus derivadas, que permite reducir de forma apreciable la sensibilidad al ruido del método de interpolación propuesto.
- La extensión del método de interpolación a espacios de entrada bidimensionales. Se han propuesto alternativas diferentes dependiendo de si se dispone o no de la información de las derivadas cruzadas. Cuando se dispone de la información de las derivadas cruzadas, todo el proceso de reconstrucción se basa en el modelo

unidimensional, con lo que es posible implementarlo mediante filtrado FIR y obtener una reconstrucción local de la función.

- La regularización de las muestras en espacios de entrada bidimensionales. Se han presentado distintas alternativas: unas basadas en el modelo unidimensional, y otra, cuando se dispone de la información de las derivadas cruzadas, empleando un funcional de regularización 2D.
- La obtención de expresiones explícitas para los parámetros de regularización en función de la potencia de ruido en las muestras de la función y de las derivadas.

Como conclusiones más interesantes obtenidas sobre los modelos desarrollados cabe destacar las siguientes:

- La información de las derivadas proporciona algunas ventajas: permite una reconstrucción local de la función (utilizando sencillos filtros FIR) y puede contribuir a mejorar el ajuste de la función y, especialmente, de las derivadas.
- La sensibilidad al ruido en la reconstrucción de las derivadas crece al aumentar la frecuencia de muestreo y es más notable a medida que crece el orden de la derivada.
- La regularización de las muestras reduce la sensibilidad al ruido de los métodos de interpolación propuestos. La mejora obtenida es más significativa a medida que aumenta la frecuencia de muestreo de las señales, cuando la sensibilidad al ruido es mayor.
- Los métodos propuestos tienen como limitación, para su extensión a espacios de entrada de dimensión elevada, el gran número de derivadas necesarias.

Pasando del ámbito de la interpolación/regularización, que requiere tantos parámetros como número de medidas, al ámbito del modelado, en el Capítulo 4 se ha presentado el MLTS, un modelo parsimonioso que, con un número reducido de parámetros, proporciona una aproximación suave y derivable de una función. Como principales aportaciones de este capítulo pueden reseñarse:

- La obtención de un modelo suave y derivable, a partir del Modelo Lineal a Tramos Canónico, utilizando la función logaritmo de coseno hiperbólico, $\text{lch}(x, \gamma)$, para sustituir la función $|x|$, responsable de las transiciones entre tramos lineales.
- El análisis de suavidad del modelo MLTS: se ha determinado que la suavidad del mismo se controla mediante un único parámetro, el parámetro de suavidad, γ . En este sentido, γ se puede interpretar como un parámetro de regularización.
- El estudio de diferentes alternativas para el entrenamiento del parámetro de suavidad del modelo. Dado que la suavidad está relacionada con las derivadas de la función, se han comparado las posibilidades de adaptar γ en función de la información de la función o de sus derivadas.

Como conclusiones, se pueden destacar:

- El MLTS se ha mostrado muy eficaz en la aproximación de una función y sus derivadas en problemas con restricciones en el número de parámetros del modelo. En estas circunstancias, en general, el MLTS proporciona resultados superiores a los obtenidos con otras redes neuronales, como el MLP o la GRBF.
- El algoritmo de entrenamiento propuesto presenta una reducida carga computacional cuando se compara con el de redes como el MLP o la GRBF.
- El principal inconveniente del modelo es su sensibilidad a los mínimos locales. En gran parte, esto es debido a que no se dispone de un algoritmo de inicialización eficiente.
- Entre las alternativas para el entrenamiento de γ , el entrenamiento a partir de la información de la derivada segunda se ha mostrado como la más efectiva, especialmente cuando se toma un γ distinto para cada función base del modelo, aunque esto significa aumentar el número total de parámetros del mismo.

En el Capítulo 5 se han presentado varias alternativas para introducir la información de las derivadas en la construcción de una red neuronal. Como principales aportaciones en este capítulo cabe destacar:

- La utilización de una nueva función de coste, que incluye la información de las derivadas, en el algoritmo de entrenamiento de distintas redes neuronales.
- La propuesta de una arquitectura modular para la reconstrucción global de una función con reproducción local de las derivadas. En esta línea, se ha propuesto una arquitectura específica para el modelado global de transistores de microondas, utilizando un MLTS para el modelo gran señal y una GRBF para el modelo pequeña señal de los mismos. El modelo pequeña señal propuesto permite el modelado de los efectos de la intermodulación.
- Un nuevo algoritmo de entrenamiento, basado en el algoritmo EM, para redes RBF dentro del módulo pequeña señal.

Algunas de las conclusiones obtenidas son las siguientes:

- Con la función de coste propuesta se consigue una mejor aproximación de las derivadas de la función. Además, en ciertos casos, considerar la información de las derivadas resulta más interesante que considerar la información de la propia función. Por ejemplo, se ha podido comprobar que, cuando el número de parámetros de la red es suficientemente elevado, la información de las derivadas permite obtener los mismos resultados que con una función de coste convencional, reduciendo el tamaño del conjunto de entrenamiento.

- La arquitectura modular propuesta para el modelado de transistores es capaz representar correctamente tanto el comportamiento pequeña señal, donde se modela el efecto de la intermodulación, como el comportamiento gran señal.
- El algoritmo EM propuesto para redes RBF permite acelerar la convergencia del entrenamiento, a la vez que reduce la sensibilidad a los mínimos locales, aprovechando el carácter local de las funciones de activación de este tipo de redes.

Los modelos propuestos se han aplicado, en el Capítulo 6, al modelado de transistores de microondas. En este caso hay que destacar las siguientes aportaciones:

- Un modelo gran señal para un transistor HEMT bajo iluminación óptica utilizando el modelo MLTS.
- Un modelo pequeña señal para transistores MESFET y HEMT, con capacidad para predecir la distorsión de intermodulación, utilizando una red GRBF.
- Un modelo global (pequeña señal + gran señal) para un transistor MESFET. El módulo gran señal se implementa con un MLTS, y el pequeña señal con una GRBF. La salida de los dos módulos se combina utilizando unas funciones de pertenencia de lógica borrosa.

A partir de los resultados obtenidos en este capítulo, se han extraído las siguientes conclusiones:

- Con el modelo propuesto para el modelado del efecto de la iluminación en la característica gran señal de transistores de microondas se han obtenido, con un número reducido de parámetros, resultados superiores a los obtenidos con otras alternativas. Hay que destacar que prácticamente no hay modelos que tengan en cuenta el efecto de la iluminación (hasta nuestro conocimiento sólo un modelo basado en funciones analíticas).
- Los modelos pequeña señal propuestos permiten una correcta caracterización del efecto de la intermodulación. Con el modelo propuesto es posible analizar aspectos como la potencia o la distorsión de intermodulación en función de la carga, lo que permite un diseño con este tipo de requerimientos.

Finalmente, hay que reseñar también que la realización de esta Tesis ha dado lugar a una serie de publicaciones. Una lista con la referencia de las mismas se muestra en el Apéndice G.

7.2 Líneas futuras

A continuación se presentan algunas líneas futuras de investigación, a partir de los resultados obtenidos en la Tesis, que se consideran interesantes.

7.2.1 SVM's en problemas de regresión con información de las derivadas

Una línea futura de investigación que consideramos de especial interés es la aplicación de la arquitectura de aprendizaje conocida como Máquina de Vectores Soporte o "Support Vector Machine" (SVM), a los problemas de aproximación y modelado abordados en esta Tesis. Empleando criterios derivados de la Teoría Estadística del Aprendizaje ("Statistical Learning Theory") [Vapnik95], la SVM fue inicialmente propuesta para resolver problemas de clasificación binaria [Vapnik95, Cortes95], aunque posteriormente ha sido extendida para su uso en problemas de regresión/aproximación funcional [Smola98, Vapnik97, Muller98].

La idea fundamental del método consiste en realizar una transformación no lineal del espacio de entrada en otro espacio de "características", de dimensión mayor (posiblemente infinita), en el que es posible resolver el problema de clasificación/regresión mediante un hiperplano óptimo. Un aspecto crucial para la aplicación de las SVM's es que habitualmente no es necesario conocer la transformación no lineal entre el espacio de entrada y el espacio de características; basta con conocer el producto escalar entre los vectores transformados que, bajo ciertas condiciones [Aizerman64], puede obtenerse de forma muy sencilla evaluando una función núcleo ("kernel") simétrica. Por ejemplo, la función de activación Gaussiana, habitualmente empleada en la Red de Funciones de Base Radial, es un kernel de estas características. Otro aspecto interesante, del que deriva su nombre esta arquitectura de aprendizaje, es que el vector de pesos que define el hiperplano óptimo en el espacio transformado admite una expansión en términos de una serie de patrones representativos, denominados vectores soporte, de entre el conjunto de patrones de entrenamiento. Así, comparando por ejemplo con una RBF convencional, una SVM proporciona directamente los centroides y amplitudes de cada gaussiana. Además de esta ventaja, las SVM's han demostrado unas prestaciones superiores a las Redes Neuronales en distintos problemas de clasificación [Schölkopf97] y predicción [Muller98].

Estas ventajas nos sugieren que, dentro del problema considerado en esta Tesis, sería interesante estudiar cómo incorporar la información de las derivadas en una SVM para regresión. De esta manera podríamos aprovechar la potencia del marco teórico en el que se desarrollan las SVM's, así como su capacidad de generalización.

Conviene en este punto formalizar con un poco más de detalle el empleo de SVM's en regresión, con objeto de indicar cómo es posible modificar la formulación para incluir valores de las derivadas de la función, así como los aspectos que requieren un estudio más profundo dentro de esta nueva línea de investigación.

Dado un conjunto de N muestras de la función a aproximar: (\mathbf{x}_i, y_i) , $i = 1, \dots, N$; una SVM construye un regresor lineal en un espacio transformado

$$\hat{f}(x) = \mathbf{w}^T \Phi(\mathbf{x}) + b, \quad (7.1)$$

donde los parámetros del hiperplano \mathbf{w} y b son la solución del siguiente problema de minimización con restricciones

$$\text{Minimizar} \quad \frac{1}{2} \|\mathbf{w}\|^2 \quad (7.2)$$

sujeto a las restricciones

$$(\mathbf{w}^T \Phi(\mathbf{x}_i) + b) - y_i \leq \epsilon \quad (7.3)$$

$$y_i - (\mathbf{w}^T \Phi(\mathbf{x}_i) + b) \leq \epsilon. \quad (7.4)$$

Supongamos ahora que adicionalmente a las muestras de la función se conoce también la primera derivada de la función en dichos puntos: (\mathbf{x}_i, y_i^1) , $i = 1, \dots, N$. Esta información adicional puede incorporarse en el proceso de minimización añadiendo las restricciones

$$(\mathbf{w}^T \Phi^1(\mathbf{x}_i) + b) - y_i^1 \leq \epsilon_1 \quad (7.5)$$

$$y_i^1 - (\mathbf{w}^T \Phi^1(\mathbf{x}_i) + b) \leq \epsilon_1 \quad (7.6)$$

donde $\Phi^1(\mathbf{x}_i)$ denota la derivada de la transformación no lineal al espacio de características. Por ejemplo, si consideramos una aproximación en un espacio 1D mediante polinomios de tercer grado

$$\begin{aligned} \Phi: \mathcal{R} &\rightarrow \mathcal{R}^3 \\ x &\rightarrow (x, x^2, x^3) \end{aligned} \quad (7.7)$$

entonces

$$\begin{aligned} \Phi^1: \mathcal{R} &\rightarrow \mathcal{R}^3 \\ x &\rightarrow (1, 2x, 3x^2). \end{aligned} \quad (7.8)$$

Introduciendo los multiplicadores de Lagrange $\alpha_i, \alpha_i^*, \alpha_{1i}, \alpha_{1i}^*$, la minimización de (7.2) sujeta a las restricciones (7.3), (7.4), (7.5) y (7.6), es equivalente a minimizar la siguiente función

$$\begin{aligned} L(\alpha_i, \alpha_i^*, \alpha_{1i}, \alpha_{1i}^*) &= \frac{1}{2} \sum_{i,j} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle \\ &+ \frac{1}{2} \sum_{i,j} (\alpha_{1i} - \alpha_{1i}^*)(\alpha_{1j} - \alpha_{1j}^*) \langle \Phi^1(\mathbf{x}_i), \Phi^1(\mathbf{x}_j) \rangle \\ &+ \sum_{i,j} (\alpha_{1i} - \alpha_{1i}^*)(\alpha_j - \alpha_j^*) \langle \Phi^1(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle \\ &- \sum_i ((\alpha_i^* - \alpha_i)y_i + (\alpha_{1i}^* - \alpha_{1i})y_i^1) \\ &+ \sum_i ((\alpha_i^* + \alpha_i)\epsilon + (\alpha_{1i}^* + \alpha_{1i})\epsilon_1) \end{aligned} \quad (7.9)$$

sujeta a las restricciones

$$0 \leq \alpha_i, \alpha_i^*, \alpha_{1i}, \alpha_{1i}^*, \quad (7.10)$$

y

$$\sum_i ((\alpha_i - \alpha_i^*) + (\alpha_{1i} - \alpha_{1i}^*)) = 0. \quad (7.11)$$

Como en otros problemas de regresión mediante SVM's, se trata de un problema de optimización cuadrático que puede escribirse de forma más compacta como

$$L(\mathbf{\Lambda}) = \frac{1}{2} \mathbf{\Lambda}^T \mathbf{D} \mathbf{\Lambda} - \mathbf{\Lambda}^T \mathbf{y} + |\mathbf{\Lambda}|^T \epsilon \quad (7.12)$$

donde $\mathbf{\Lambda}$ es un vector columna de dimensiones $2N \times 1$ cuyos elementos son $\alpha_i - \alpha_i^*$, y $\alpha_{1i} - \alpha_{1i}^*$. Por su parte \mathbf{D} es una matriz de dimensiones $2N \times 2N$ que tiene como elementos los productos escalares entre los vectores transformados: $\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$, $\langle \Phi^1(\mathbf{x}_i), \Phi^1(\mathbf{x}_j) \rangle$ y $\langle \Phi^1(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$.

Una vez obtenidos los valores de los multiplicadores de Lagrange, la constante b se puede obtener a partir de cualquiera de las restricciones (7.3), (7.4), (7.5) o (7.6), para un valor i tal que $\alpha_i - \alpha_i^* \neq 0$, o $\alpha_{1i} - \alpha_{1i}^* \neq 0$.

Finalmente, la expresión de la aproximación no lineal proporcionada por la SVM es

$$\hat{f}(\mathbf{x}) = \sum_i (\alpha_i - \alpha_i^*) \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}_i) \rangle + \sum_i (\alpha_{1i} - \alpha_{1i}^*) \langle \Phi(\mathbf{x}), \Phi^1(\mathbf{x}_i) \rangle + b. \quad (7.13)$$

Este resultado preliminar muestra que es posible extender la idea de SVM's para regresión de modo que sea posible incluir información adicional respecto a las derivadas de la función. Obviamente ciertos aspectos requieren un análisis más detallado, siendo también necesario un estudio de sus prestaciones y limitaciones en comparación con otras técnicas como las propuestas en esta Tesis. En concreto, algunos aspectos a tratar dentro de esta futura línea de investigación son los siguientes:

- Tal como ha sido planteado, la aplicación del método requiere una transformación no lineal $\Phi(\cdot)$ conocida (obviamente a un espacio de dimensión finita), cuya derivada $\Phi^1(\cdot)$ sea también conocida. De esta manera es posible evaluar directamente todos los productos escalares involucrados en el problema de optimización cuadrática y en la expansión de la SVM en términos de sus vectores soporte. Sería posible de esta manera obtener soluciones que pertenezcan al subespacio de los splines: establecer una comparación entre esta alternativa y los métodos propuestos en el Capítulo 3 nos parece una idea interesante.
- La extensión del método a espacios de dimensión infinita requiere que los productos escalares puedan evaluarse a través de un kernel. Es en este punto donde

creemos pueden existir algunas dificultades teóricas ya que, a partir de un kernel tal que $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$, no conocemos ningún resultado que nos asegure que existen o nos indique cómo obtener los kernels $K^1(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi^1(\mathbf{x}_i), \Phi^1(\mathbf{x}_j) \rangle$ y $K^2(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi^1(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$.

- Caso de existir, la solución proporcionada por la SVM se encuentra como máximo a una distancia ϵ de las muestras de la función y ϵ_1 de las muestras de la derivada. Pudiendo estos valores controlar de forma independiente la precisión en la aproximación de la función y su derivada. Habitualmente es necesario relajar las restricciones del problema mediante la introducción de variables que permitan obtener una solución aun cuando no todas las restricciones puedan cumplirse de forma exacta: esto conduce a una formulación alternativa y más potente del método. Asimismo, es posible considerar otras funciones de coste alternativas a la función de pérdidas de Vapnik empleada en nuestro desarrollo. En esta línea cabe mencionar el empleo de una función de coste cuadrática, la función de Huber [Smola98], o el empleo de una una función tangente hiperbólica [Pérez2000]. Cuando las muestras están contaminadas por ruido el empleo de estas funciones de coste puede ser especialmente interesante.
- Solucionar de forma eficiente el problema de optimización cuadrática es otro aspecto que merece estudio. En problemas de clasificación existen algoritmos eficientes para la obtención de la SVM: algunos conocidos desde hace algunos años como el Adatron [Anlauf89, Frieb98], otros más recientes [Pérez99, Platt98]. Una vía sencilla de abordar el problema consiste por lo tanto en extender estos métodos para su aplicación a nuestro problema particular de regresión.
- Finalmente, es posible generalizar estas ideas considerando la incorporación de otro tipo de información (no necesariamente las derivadas de la función) en el problema de aproximación funcional: suavidad, ancho de banda, etc.

7.2.2 Otras líneas

Otras posibles líneas de continuación del trabajo realizado en esta Tesis son:

- El desarrollo de modelos de interpolación/regularización en espacios multiresolución, utilizando funciones de escala. En esta Tesis nos hemos centrado en el espacio de funciones splines debido a su sencillez. Sin embargo, el método de interpolación en espacios unidimensionales propuesto por Djokovic y Vaidyanathan [Djokovic97] es válido para cualquier función de escala que cumpla las condiciones de multiresolución. Una alternativa interesante es estudiar este tipo de funciones y la posibilidad de extender el método a espacios de entrada bidimensionales, así como la búsqueda de extensiones regularizadas.
- El desarrollo de algoritmos de inicialización para el modelo MLTS. En las simulaciones realizadas en esta Tesis este modelo se inicializa de forma aleatoria. Una

inicialización más inteligente del mismo puede contribuir a mejorar sus prestaciones, al reducir su sensibilidad a los mínimos locales.

- El desarrollo de nuevas arquitecturas neuronales modulares para la reconstrucción de una función y sus derivadas. Aquí se ha considerado el caso particular en el que la información de las derivadas es solamente de interés en un entorno de un punto del espacio de entrada. Resulta interesante la posibilidad de estudiar distintos tipos de combinadores que permitan integrar la información de la función y de sus derivadas, procedentes en principio de distintos módulos, para llevar a cabo una reconstrucción global de la función en todo el espacio de entrada.

Apéndice A

Notación y Abreviaturas

En este apéndice se describe la notación utilizada en esta Tesis, así como una lista de las abreviaturas más frecuentes.

A.1 Notación

En esta sección se presentan, por un lado, las reglas básicas para la representación de los diferentes tipos de variables, y por otro lado, algunos aspectos específicos relacionados con el tema que aborda la Tesis.

A.1.1 Reglas generales

La notación empleada se rige básicamente por las siguientes reglas:

- **Escalares:** se emplean minúsculas o mayúsculas en itálica, x , N , sin distinción entre naturales, enteros o complejos.
- **Vectores:** se emplean minúsculas en negrita, \boldsymbol{x} . La operación de transposición se indica con el superíndice T, \boldsymbol{x}^T .
- **Matrices:** se utilizan mayúsculas en negrita, \mathbf{A} .
- **Parámetros y conjuntos de Parámetros:** se emplean letras griegas minúsculas para los parámetros, λ , y mayúsculas para los conjuntos de parámetros, Θ .

A.1.2 Aspectos específicos de la Tesis

A continuación se presentan algunos aspectos de notación específicos de la temática de esta Tesis: la interpolación, modelado y aproximación funcional de una función y sus derivadas.

- **Función y derivadas:** la función que se pretende modelar se denota como $f(x)$, en espacios de entrada unidimensionales, y como $f(\mathbf{x})$, en espacios de entrada n -dimensionales. En este caso, el vector de variables de entrada, \mathbf{x} , para un espacio de entrada de dimensión N , tiene la forma

$$\mathbf{x} = (x_1, x_2, \dots, x_N)$$

La derivada de orden d se denota como $f^{(d)}(x)$ para espacios de entrada unidimensionales. En espacios 2D, si sólo se deriva respecto a una variable, x_n , se denota

$$f_{x_n}^{(d)}(x_1, x_2).$$

Para las derivadas cruzadas se emplea la notación

$$f_{x_1, x_2}^{(d_1, d_2)}(x_1, x_2)$$

- **Muestras de la función y derivadas:** las secuencias de muestras de la función y las derivadas se denotan mediante $y^{(d)}[n]$, con d indicando el orden de la derivada, y n la posición dentro de la secuencia. En general, el superíndice $d = 0$, para las muestras de la función, se obviará por cuestiones de simplicidad notacional. El instante de muestreo asociado a cada muestra, en espacios de entrada unidimensionales, se indica mediante la secuencia $x[n]$, es decir

$$y^{(d)}[n] = f^{(d)}(x[n]).$$

En problemas de muestreo regular, con periodo de muestreo T , se asume que $x[n] = nT$.

En espacios de entrada 2D, para derivadas con respecto a una única variable de entrada, x_n , se emplea la notación

$$y_{x_n}^{(d)}[n_1, n_2] = f_{x_n}^{(d)}(x_1[n_1], x_2[n_2]),$$

y para las derivadas cruzadas

$$y_{x_1, x_2}^{(d_1, d_2)}[n_1, n_2] = f_{x_1, x_2}^{(d_1, d_2)}(x_1[n_1], x_2[n_2]).$$

En general se considera muestreo regular, donde $x_1[n_1] = n_1T_1$ y $x_2[n_2] = n_2T_2$.

A.2 Abreviaturas

EM	<i>Expectation-Maximization</i>
FET	<i>Field Effect Transistor</i>
FIR	<i>Finite Impulse Response</i>
GRBF	Red de Funciones de Base Radial Generalizada (<i>Generalized Radial Basis Function network</i>)
HEMT	<i>High Electron Mobility Transistor</i>
I/V	Característica Corriente/Tensión
IIR	<i>Infinite Impulse Response</i>
MESFET	<i>Metal Semiconductor Field Effect Transistor</i>
ML I	Modelo Local de Interpolación
MLI1D	Modelo Local de Interpolación en espacios 1D
MLP	Perceptrón Multicapa (<i>Multilayer Perceptron</i>)
MLR	Modelo Local Regularizado
MLRs	Modelo Local Regularizado Simple
MLR1D	Modelo Local Regularizado en espacios 1D
MLT	Modelo Lineal a Tramos Canónico
MLTS	Modelo Lineal a Tramos Suavizado
MRA	Análisis Multiresolución (<i>Multiresolution Analysis</i>)
MSI	Modelo Semilocal de Interpolación
MSR	Modelo Semilocal Regularizado
OLS	<i>Orthogonal Least Squares</i>
RBF	Red de Funciones de Base Radial (<i>Radial Basis Function network</i>)
RF	<i>Radio Frequency</i>
RKHS	Espacio de Hilbert con Kernel de Reproducción (<i>Reproducing Kernel Hilbert Space</i>)
SER	Relación Señal a Error (<i>Signal to Error Ratio</i>)
SNR	Relación Señal a Ruido (<i>Signal to Noise Ratio</i>)
SVM	Máquina de Vectores Soporte (<i>Support Vector Machine</i>)

Apéndice B

Descomposiciones de funciones trigonométricas en fracciones parciales

En este apéndice se muestran algunas relaciones trigonométricas que se han aplicado en diferentes apartados de la Tesis. Algunas de estas relaciones se han obtenido de [Abramowitz64].

$$\sum_{k=-\infty}^{\infty} \frac{1}{(x - k\pi)} = \cot(x) \quad (\text{B.1})$$

$$\sum_{k=-\infty}^{\infty} \frac{(-1)^k}{(x - k\pi)} = \csc(x) \quad (\text{B.2})$$

$$\sum_{k=-\infty}^{\infty} \frac{1}{(x - k\pi)^2} = \csc^2(x) \quad (\text{B.3})$$

$$\sum_{k=-\infty}^{\infty} \frac{(-1)^k}{(x - k\pi)^2} = \frac{\csc^2(x/2)}{2} - \csc^2(x) \quad (\text{B.4})$$

$$\sum_{k=-\infty}^{\infty} \frac{1}{(x - k\pi)^3} = \cot^3(x) + \cot(x) \quad (\text{B.5})$$

$$\sum_{k=-\infty}^{\infty} \frac{(-1)^k}{(x - k\pi)^3} = \csc^3(x) - \frac{\csc(x)}{2} \quad (\text{B.6})$$

Apéndice C

Modelos MLI de grados 2 y 3

C.1 Modelo de grado 3

En este apartado se estudia el problema particular de la interpolación sobre el espacio de los splines cúbicos, mediante el modelo MLI, de las muestras de una función y de sus derivadas primera y segunda ($D = 2$).

C.1.1 Expresiones de interpolación y reconstrucción

Se parte de nuevo de un conjunto de N muestras de la función y de las derivadas, es decir

$$y^d[n] = y^d(nT), \quad n = \{0, 1, \dots, N - 1\} \quad d = \{0, 1, 2\} \quad (\text{C.1})$$

De nuevo se supone un muestreo uniforme con periodo de muestreo T . Si se desea un modelo en el que la derivada segunda sea lineal a tramos, para conseguir cumplir las restricciones impuestas por las condiciones de interpolación y de continuidad que definen las muestras, en este caso se van a insertar dos instantes de ruptura entre cada dos instantes de muestreo consecutivos, o lo que es lo mismo, se realiza una interpolación de las secuencias de muestras por un factor 3, dando lugar a las secuencias interpoladas

$$y_i^d[n] \quad n = \{0, 1, \dots, 3N - 3\} \quad d = \{1, 2, 3\} \quad (\text{C.2})$$

Planteando un sistema en el que la derivada segunda es lineal a tramos sobre su secuencia interpolada correspondiente, con las restricciones impuestas por las condiciones de interpolación y de continuidad definidas por las muestras de la función y de las derivadas, se obtienen los valores de la función y las derivadas en los instantes de ruptura insertados, llegando a las siguientes expresiones:

$$y_i^d[3n] = y^d[n], \quad n = 0, \dots, N - 1 \quad (\text{C.3})$$

$$y_i^2[3n + 1] = \frac{y[n + 1] - y[n]}{T_i^2} - \frac{2y^1[n] + y^1[n + 1]}{T_i} + \frac{2y^2[n + 1] - 5y^2[n]}{6} \quad (\text{C.4})$$

$$y_i^{(2)}[3n+2] = \frac{y[n] - y[n+1]}{T_i^2} + \frac{2y^{(1)}[n+1] + y^{(1)}[n]}{T_i} + \frac{2y^{(2)}[n] - 5y^{(2)}[n+1]}{6} \quad (\text{C.5})$$

$$y_i^{(1)}[3n+1] = \frac{y[n+1] - y[n]}{2T_i} - \frac{y^{(1)}[n+1]}{2} + \frac{y^{(2)}[n] + 2y^{(2)}[n+1]}{12}T_i \quad (\text{C.6})$$

$$y_i^{(1)}[3n+2] = \frac{y[n+1] - y[n]}{2T_i} - \frac{y^{(1)}[n]}{2} - \frac{2y^{(2)}[n] + y^{(2)}[n+1]}{12}T_i \quad (\text{C.7})$$

$$y_i[3n+1] = \frac{5y[n] + y[n+1]}{6} + \frac{4y^{(1)}[n] - y^{(1)}[n+1]}{6}T_i + \frac{2y^{(2)}[n+1] + 7y^{(2)}[n]}{36}T_i^2 \quad (\text{C.8})$$

$$y_i[3n+2] = \frac{y[n] + 5y[n+1]}{6} + \frac{y^{(1)}[n] - 4y^{(1)}[n+1]}{6}T_i + \frac{2y^{(2)}[n] + 7y^{(2)}[n+1]}{36}T_i^2 \quad (\text{C.9})$$

donde T_i es el periodo de muestreo asociado a las secuencias interpoladas, que en este caso vale $T_i = T/3$ y $n = 0, \dots, N-2$ para las ecuaciones (C.4)-(C.9).

A partir de estas secuencias interpoladas, se plantea un modelo con la segunda derivada lineal a tramos entre las muestras de su secuencia interpolada asociada, pudiendo reconstruirse la función mediante la integración sucesiva de la misma. De esta forma, para un instante x situado entre los instantes de tiempo asociados a las muestras n y $n+1$ de la secuencia interpolada, en el caso de muestreo regular, entre nT_i y $(n+1)T_i$, la reconstrucción de la función vendrá dada por la siguiente expresión

$$f(x) = y_i[n] + y_i^{(1)}[n]\Delta x + y_i^{(2)}[n]\frac{(\Delta x)^2}{2} + \frac{y_i^{(2)}[n+1] - y_i^{(2)}[n]}{6T_i}(\Delta x)^3 \quad (\text{C.10})$$

donde Δx es la diferencia entre x y el instante de tiempo asociado a la muestra n de la secuencia interpolada, que en el caso de muestreo regular valdría $nT/3$.

De nuevo, al igual que en el caso del modelo de grado 2, el proceso de reconstrucción es totalmente local. El cálculo de los valores de función y derivadas en los instantes de ruptura depende tan sólo de las muestras en los dos instantes entre los que se encuentran, y de nuevo la reconstrucción a partir de las secuencias interpoladas sólo depende de los valores de las mismas en los instantes entre los que se está reconstruyendo en cada momento. De esta forma se sigue manteniendo un bajo coste computacional, y se facilita la aplicación directa al caso de muestreo irregular.

C.1.2 Representación mediante un banco de filtros

Al igual que se vió para el caso del modelo de grado 2, este modelo también admite una representación sencilla utilizando un banco de filtros digitales MIMO seguido de un banco de filtros analógicos de reconstrucción. La figura C.1 muestra esta estructura de bancos de filtros para el modelo MLI de grado 3.

La interpolación de las secuencias de muestras se lleva a cabo mediante los filtros polifase que componen el banco de filtros digitales MIMO. Cada una de las secuencias interpoladas, $y_i[n]$, $y_i^{(1)}[n]$ y $y_i^{(2)}[n]$ pasa a través de un filtro reconstructor analógico y

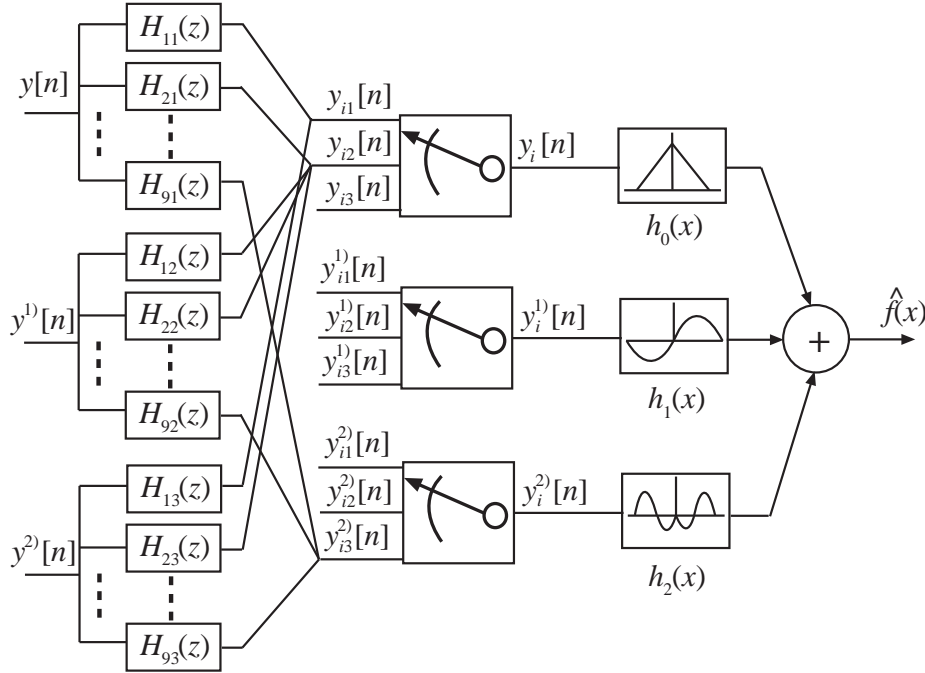


Figura C.1: Representación mediante filtros MIMO del modelo MLI de grado 3

finalmente la señal reconstruida, $\hat{f}(x)$, se obtiene mediante la suma de las salidas de cada uno de estos tres filtros reconstructores.

La respuesta de los filtros MIMO se puede obtener fácilmente a partir de las expresiones de las señales interpoladas descritas por las ecuaciones (C.3) - (C.9). Así se puede comprobar que los filtros tienen las siguientes respuestas

$$H_{11}(z) = H_{42}(z) = H_{73} = 1;$$

$$H_{12}(z) = H_{13}(z) = H_{41}(z) = H_{43}(z) = H_{71}(z) = H_{72}(z) = 0;$$

$$H_{21}(z) = \frac{5+z}{6}; \quad H_{31}(z) = \frac{1+5z}{6}; \quad H_{51}(z) = \frac{z-1}{2T_i};$$

$$H_{61}(z) = \frac{z-1}{2T_i}; \quad H_{81}(z) = \frac{z-1}{T_i^2}; \quad H_{91}(z) = \frac{1-z}{T_i^2};$$

$$H_{22}(z) = \frac{4-z}{6}T_i; \quad H_{32}(z) = \frac{1-4z}{6}T_i; \quad H_{52}(z) = -\frac{z}{2}; \quad (C.11)$$

$$H_{62}(z) = -\frac{1}{2}; \quad H_{82}(z) = -\frac{2+z}{T_i}; \quad H_{92}(z) = \frac{1+2z}{T_i};$$

$$H_{23}(z) = \frac{2z + 7}{36}T_i^2; \quad H_{33}(z) = \frac{2 + 7z}{36}T_i^2; \quad H_{53}(z) = \frac{1 + 2z}{12}T_i;$$

$$H_{63}(z) = -\frac{2 + z}{12}T_i; \quad H_{83}(z) = \frac{2z - 5}{6}; \quad H_{93}(z) = \frac{2 - 5z}{6};$$

En cuanto a los filtros reconstructores, $h_0(x)$ y $h_1(x)$, coinciden con los filtros reconstructores del modelo de grado 2 dados por las ecuaciones (3.14) y (3.15), mientras que el filtro $h_2(x)$ tiene la siguientes respuestas al impulso y frecuencial

$$h_2(x) = \frac{x^2}{4} - \frac{|x|^3}{6T_i} - \frac{|x|}{12}T_i \quad \text{para } |x| \leq T_i. \quad (\text{C.12})$$

$$H_2(\Omega) = \left(\frac{2 - \Omega^2 T_i^2 / 6}{\Omega^4 T_i} \cos(\Omega T_i) + \frac{1}{\Omega^3} \sin(\Omega T_i) - \frac{2 - \Omega^2 T_i^2 / 6}{\Omega^4 T_i} \right) \quad (\text{C.13})$$

Estas respuestas se representa en la figura C.2 para un valor de $T_i = 1$.

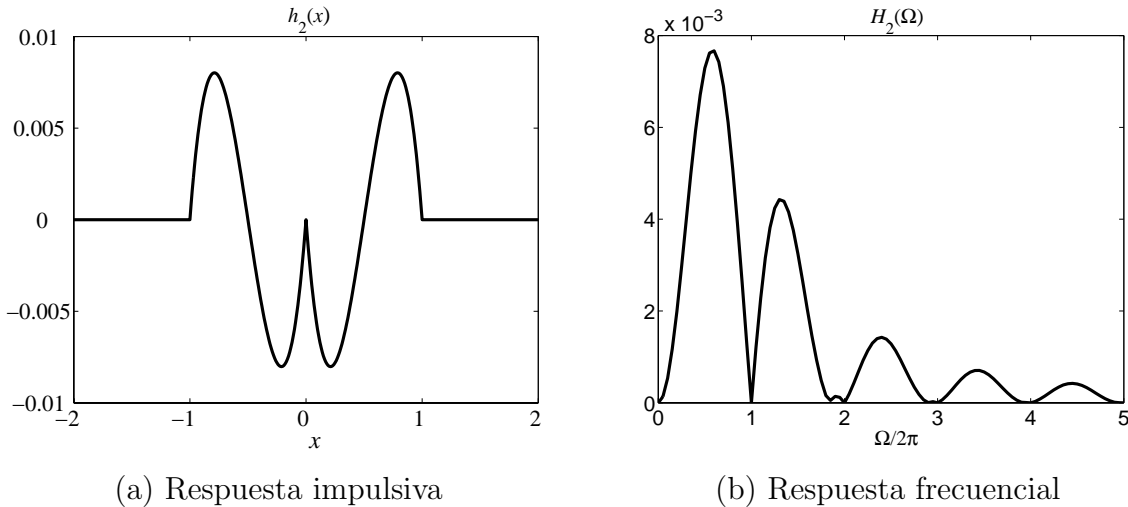


Figura C.2: Respuestas impulsiva y frecuencial del filtro reconstructor $h_2(x)$.

C.2 Modelo de grado 4

En este apartado se estudia el problema particular de la interpolación sobre el espacio de los splines de grado 4, mediante el modelo MLI, de las muestras de una función y de sus derivadas primera, segunda y tercera ($D = 3$).

C.2.1 Expresiones de interpolación y de reconstrucción

Se parte de nuevo de un conjunto de N muestras de la función y de las derivadas, es decir

$$y^d[n] = y^d(nT), \quad n = \{0, 1, \dots, N-1\}, \quad d = \{0, 1, 2, 3\} \quad (\text{C.14})$$

De nuevo se supone un muestreo uniforme con periodo de muestreo T . Ahora, el modelo presenta una derivada tercera lineal a tramos y se insertan tres instantes de ruptura entre cada dos instantes de muestreo consecutivos, es decir, se realiza una interpolación de las secuencias de muestras por un factor 4, dando lugar a las secuencias interpoladas

$$y_i^d[n] \quad n = \{0, 1, \dots, 4N-4\}, \quad d = \{0, 1, 2, 3\}. \quad (\text{C.15})$$

A partir de las condiciones de interpolación y de continuidad definidas por las muestras de la función y de la derivada, se obtienen los valores de la función y las derivadas en los instantes de ruptura insertados, llegándose a las siguientes expresiones:

$$y_i^d[4n] = y^d[n], \quad n = 0, \dots, N-1 \quad (\text{C.16})$$

$$\begin{aligned} y_i^3[4n+1] &= \frac{y[n+1] - y[n]}{T_i^3} - \frac{5y^1[n] + 3y^1[n+1]}{2T_i^2} \\ &+ \frac{11y^2[n+1] - 35y^2[n]}{12T_i} - \frac{13y^3[n] + 3y^3[n+1]}{12} \end{aligned} \quad (\text{C.17})$$

$$\begin{aligned} y_i^3[4n+2] &= \frac{2(y[n] - y[n+1])}{T_i^3} + \frac{4(y^1[n] + y^1[n+1])}{T_i^2} \\ &+ \frac{17(y^2[n] + y^2[n+1])}{6T_i} + \frac{5(y^3[n] + y^3[n+1])}{6} \end{aligned} \quad (\text{C.18})$$

$$\begin{aligned} y_i^3[4n+3] &= \frac{y[n+1] - y[n]}{T_i^3} - \frac{3y^1[n] + 5y^1[n+1]}{2T_i^2} \\ &- \frac{11y^2[n] - 35y^2[n+1]}{12T_i} - \frac{3y^3[n] + 13y^3[n+1]}{12} \end{aligned} \quad (\text{C.19})$$

$$\begin{aligned} y_i^2[4n+1] &= \frac{y[n+1] - y[n]}{2T_i^2} - \frac{5y^1[n] + 3y^1[n+1]}{4T_i} \\ &+ \frac{11(y^2[n+1] - y^2[n])}{24} - \frac{T_i(y^3[n] + 3y^3[n+1])}{24} \end{aligned} \quad (\text{C.20})$$

$$y_i^2[4n+2] = \frac{y^1[n+1] - y^1[n]}{2T_i} - \frac{y^2[n] + y^2[n+1]}{2} + \frac{T_i(y^3[n+1] - 3y^3[n])}{6} \quad (\text{C.21})$$

$$y_i^2[4n+3] = \frac{y[n] - y[n+1]}{2T_i^2} + \frac{3y^1[n] + 5y^1[n+1]}{4T_i} + \frac{11(y^2[n] - y^2[n+1])}{24} + \frac{T_i(3y^3[n] + y^3[n+1])}{24} \quad (\text{C.22})$$

$$y_i^1[4n+1] = \frac{y[n+1] - y[n]}{6T_i} + \frac{7y^1[n] - 3y^1[n+1]}{12} + \frac{37y^2[n] + 11y^2[n+1]}{72}T_i + \frac{11y^3[n] - 3y^3[n+1]}{72}T_i^2 \quad (\text{C.23})$$

$$y_i^1[4n+2] = \frac{2(y[n+1] - y[n])}{3T_i} - \frac{5(y^1[n] + y^1[n+1])}{6} + \frac{4(y^2[n+1] - y^2[n])}{9}T_i - \frac{y^3[n] + y^3[n+1]}{9}T_i^2 \quad (\text{C.24})$$

$$y_i^1[4n+3] = \frac{y[n+1] - y[n]}{6T_i} + \frac{7y^1[n+1] - 3y^1[n]}{12} - \frac{11y^2[n] + 37y^2[n+1]}{72}T_i + \frac{11y^3[n+1] - 3y^3[n]}{72}T_i^2 \quad (\text{C.25})$$

$$y_i[4n+1] = \frac{23y[n] + y[n+1]}{24} + \frac{43y^1[n] - 3y^1[n+1]}{48}T_i + \frac{109y^2[n] + 11y^2[n+1]}{288}T_i^2 + \frac{23y^3[n] - 3y^3[n+1]}{288}T_i^3 \quad (\text{C.26})$$

$$y_i[4n+2] = \frac{y[n] + y[n+1]}{2} + \frac{17(y^1[n] - y^1[n+1])}{24}T_i + \frac{5(y^2[n] + y^2[n+1])}{12}T_i^2 + \frac{y^3[n] - y^3[n+1]}{9}T_i^3 \quad (\text{C.27})$$

$$\begin{aligned}
y_i[4n+3] = & \frac{y[n] + 23y[n+1]}{24} + \frac{3y^1[n] - 43y^1[n+1]}{48} T_i \\
& + \frac{11y^2[n] + 109y^2[n+1]}{288} T_i^2 + \frac{3y^3[n] - 23y^3[n+1]}{288} T_i^3 \quad (C.28)
\end{aligned}$$

donde T_i es el periodo de muestreo asociado a las secuencias interpoladas, que en este caso vale $T_i = T/4$ y $n = 0, \dots, N-2$ para las ecuaciones (C.17)-(C.28).

A partir de estas secuencias interpoladas, se plantea un modelo con la derivada tercera lineal a tramos entre las muestras de su secuencia interpolada asociada, pudiendo reconstruirse la función mediante la integración sucesiva de la misma. De esta forma, para un instante de tiempo x situado entre los instantes de tiempo asociados a las muestras n y $n+1$ de la secuencia interpolada, en el caso de muestreo regular, entre nT_i y $(n+1)T_i$, la reconstrucción de la función vendrá dada por la siguiente expresión

$$y(t) = y_i[n] + y_i^1[n]\Delta x + y_i^2[n]\frac{(\Delta x)^2}{2} + y_i^3[n]\frac{(\Delta x)^3}{6} + \frac{y_i^3[n+1] - y_i^3[n]}{24T_i}(\Delta x)^4 \quad (C.29)$$

donde Δx es la diferencia entre x y el instante de tiempo asociado a la muestra n de la secuencia interpolada, que en el caso de muestreo regular vale $nT/4$.

C.2.2 Representación mediante un banco de filtros

Este modelo también admite una representación sencilla utilizando un banco de filtros digitales MIMO seguido de un banco de filtros analógicos de reconstrucción. Las expresiones de los filtros polifase se obtienen de forma similar a los casos de grado 2 y 3 a partir de las expresiones (C.17)-(C.28).

En cuanto a los filtros reconstructores analógicos, los asociados a las secuencias interpoladas de función y derivadas primera y segunda son equivalentes a los vistos en los modelos de grado 2 y 3. En cuanto al filtro asociado a la derivada tercera se puede comprobar que su respuesta al impulso y su respuesta en frecuencia tienen la forma siguiente

$$h_3(x) = \frac{x^3}{12} - \frac{|x|x^3}{24T_i} - \frac{|x|x}{24}T_i \quad (C.30)$$

$$H_3(\Omega) = j \left\{ \frac{12 - T_i^2\Omega^2}{3\Omega^5 T_i} \sin^2\left(\Omega \frac{T_i}{2}\right) - \frac{2}{\Omega^4} \cos\left(\Omega \frac{T_i}{2}\right) \sin\left(\Omega \frac{T_i}{2}\right) \right\} \quad (C.31)$$

Estas respuestas se representa en la figura C.3 para un valor de $T_i = 1$.

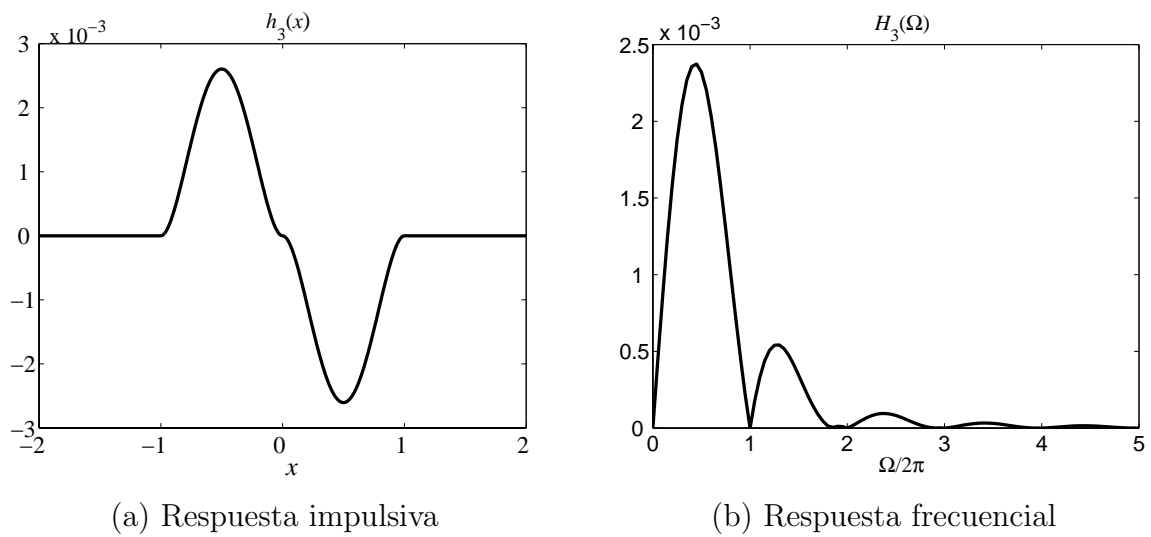


Figura C.3: Respuestas impulsiva y frecuencial del filtro reconstructor $h_3(x)$.

Apéndice D

Modelos MLR de grados 3 y 4

D.1 Modelo de grado 3

En este apartado se presentan las ecuaciones para el caso particular $D = 2$, es decir, cuando se dispone de muestras de la función y de las dos primeras derivadas.

Las ecuaciones que definen el sistema para encontrar el mínimo de la función de coste J a partir de las secuencias de muestras regularizadas son

$$\frac{\partial J}{\partial y_r[n]} = 2\lambda_0(y_r[n] - y[n]) + \lambda_r \frac{\partial J_r}{\partial y_r[n]}, \quad (\text{D.1})$$

$$\frac{\partial J}{\partial y_r^1[n]} = 2\lambda_1(y_r^1[n] - y^1[n]) + \lambda_r \frac{\partial J_r}{\partial y_r^1[n]}. \quad (\text{D.2})$$

$$\frac{\partial J}{\partial y_r^2[n]} = 2\lambda_2(y_r^2[n] - y^2[n]) + \lambda_r \frac{\partial J_r}{\partial y_r^2[n]}. \quad (\text{D.3})$$

Las derivadas del término de regularización J_r respecto de las distintas muestras vienen dadas por las expresiones

$$\left\{ \begin{array}{l} \frac{\partial J_r}{\partial y_r[n]} = \frac{2}{T^3}(y_r[n] - y_r[n+1]) + \frac{3}{T^2}(y_r^1[n] + y_r^1[n+1]) \\ \quad + \frac{5}{6T}(y_r^2[n] - y_r^2[n+1]) \\ \\ \frac{\partial J_r}{\partial y_r^1[n]} = \frac{3}{T^2}(y_r[n] - y_r[n+1]) + \frac{1}{3T}(16y_r^1[n] + 11y_r^1[n+1]) \\ \quad + \frac{3}{2}y_r^2[n] - y_r^2[n+1] \\ \\ \frac{\partial J_r}{\partial y_r^2[n]} = \frac{5}{6T}(y_r[n] - y_r[n+1]) + \frac{3}{2}y_r^1[n] + y_r^1[n+1] \\ \quad + T(y_r^2[n] - \frac{1}{4}y_r^2[n+1]) \end{array} \right. , \quad (\text{D.4})$$

para $n = 0$,

$$\left\{ \begin{array}{l} \frac{\partial J_r}{\partial y_r[n]} = \frac{2}{T^3}(y_r[n] - y_r[n-1] - y_r[n+1]) + \frac{3}{T^2}(y_r^1[n+1] - y_r^1[n-1]) \\ \quad + \frac{5}{6T}(2y_r^2[n] - y_r^2[n-1] - y_r^2[n+1]) \\ \\ \frac{\partial J_r}{\partial y_r^1[n]} = \frac{3}{T^2}(y_r[n-1] - y_r[n+1]) + \frac{1}{3T}(32y_r^1[n] + 11y_r^1[n-1] + 11y_r^1[n+1]) \\ \quad + y_r^2[n-1] - y_r^2[n+1] \\ \\ \frac{\partial J_r}{\partial y_r^2[n]} = \frac{5}{6T}(2y_r[n] - y_r[n-1] - y_r[n+1]) + y_r^1[n+1] - y_r^1[n-1] \\ \quad + T \left(2y_r^2[n] - \frac{1}{4}(y_r^2[n-1] + y_r^2[n+1]) \right) \end{array} \right. , \quad (\text{D.5})$$

para $0 < n < N - 1$, y finalmente

$$\left\{ \begin{array}{l} \frac{\partial J_r}{\partial y_r[n]} = \frac{2}{T^3}(y_r[n] - y_r[n-1]) - \frac{3}{T^2}(y_r^1[n-1] + y_r^1[n]) \\ \quad + \frac{5}{6T}(y_r^2[n] - y_r^2[n-1]) \\ \frac{\partial J_r}{\partial y_r^1[n]} = \frac{3}{T^2}(y_r[n-1] - y_r[n]) + \frac{1}{3T}(11y_r^1[n-1] + 16y_r^1[n]) \\ \quad + y_r^2[n-1] - \frac{3}{2}y_r^2[n] \\ \frac{\partial J_r}{\partial y_r^2[n]} = \frac{5}{6T}(y_r[n] - y_r[n-1]) - y_r^1[n-1] - \frac{3}{2}y_r^1[n] \\ \quad + T(y_r^2[n] - \frac{1}{4}y_r^2[n-1]) \end{array} \right. , \quad (\text{D.6})$$

para $n = N - 1$.

D.2 Modelo de grado 4

En este apartado se presentan las ecuaciones para el caso particular $D = 3$, cuando se dispone de muestras de la función y de las tres primeras derivadas.

Las ecuaciones que definen el sistema para encontrar el mínimo de la función de coste J a partir de las secuencias de muestras regularizadas son

$$\frac{\partial J}{\partial y_r[n]} = 2\lambda_0(y_r[n] - y[n]) + \lambda_r \frac{\partial J_r}{\partial y_r[n]}, \quad (\text{D.7})$$

$$\frac{\partial J}{\partial y_r^1[n]} = 2\lambda_1(y_r^1[n] - y^1[n]) + \lambda_r \frac{\partial J_r}{\partial y_r^1[n]}. \quad (\text{D.8})$$

$$\frac{\partial J}{\partial y_r^2[n]} = 2\lambda_2(y_r^2[n] - y^2[n]) + \lambda_r \frac{\partial J_r}{\partial y_r^2[n]}. \quad (\text{D.9})$$

$$\frac{\partial J}{\partial y_r^3[n]} = 2\lambda_3(y_r^3[n] - y^3[n]) + \lambda_r \frac{\partial J_r}{\partial y_r^3[n]}. \quad (\text{D.10})$$

Las derivadas del término de regularización J_r respecto de las distintas muestras vienen dadas por las expresiones

$$\left\{ \begin{array}{l} \frac{\partial J_r}{\partial y_r[n]} = \frac{4}{3T^3}(y_r[n] - y_r[n+1]) + \frac{8}{3T^2}(y_r^1[n] + y_r^1[n+1]) \\ \quad + \frac{131}{90T}(y_r^2[n] - y_r^2[n+1]) + \frac{16}{45}(y_r^3[n] + y_r^3[n+1]) \\ \frac{\partial J_r}{\partial y_r^1[n]} = \frac{8}{3T^2}(y_r[n] - y_r[n+1]) + \frac{1}{30T}(183y_r^1[n] + 137y_r^1[n+1]) \\ \quad + \frac{1}{45}(155y_r^2[n] - 107y_r^2[n+1]) + \frac{T}{360}(309y_r^3[n] + 203y_r^3[n+1]) \\ \frac{\partial J_r}{\partial y_r^2[n]} = \frac{131}{90T}(y_r[n] - y_r[n+1]) + \frac{1}{45}(155y_r^1[n] + 107y_r^1[n+1]) \\ \quad + \frac{T}{135}(443y_r^2[n] - 155y_r^2[n+1]) + \frac{T^2}{540}(455y_r^3[n] + 137y_r^3[n+1]) \\ \frac{\partial J_r}{\partial y_r^3[n]} = \frac{16}{45}(y_r[n] - y_r[n+1]) + \frac{T}{360}(309y_r^1[n] + 203y_r^1[n+1]) \\ \quad + \frac{T^2}{540}(455y_r^2[n] - 137y_r^2[n+1]) + \frac{T^3}{135}(33y_r^3[n] + 7y_r^3[n+1]) \end{array} \right. , \quad (\text{D.11})$$

para $n = 0$,

$$\left\{ \begin{array}{l} \frac{\partial J_r}{\partial y_r[n]} = \frac{4}{3T^3}(2y_r[n] - y_r[n-1] - y_r[n+1]) + \frac{8}{3T^2}(y_r^1[n+1] - y_r^1[n-1]) \\ \quad + \frac{131}{90T}(2y_r^2[n] - y_r^2[n-1] - y_r^2[n+1]) + \frac{16}{45}(y_r^3[n+1] + y_r^3[n-1]) \\ \frac{\partial J_r}{\partial y_r^1[n]} = \frac{8}{3T^2}(y_r[n-1] - y_r[n+1]) + \frac{1}{30T}(366y_r^1[n] + 137(y_r^1[n-1] + y_r^1[n+1])) \\ \quad + \frac{107}{45}(y_r^2[n-1] - y_r^2[n+1]) + \frac{T}{360}(618y_r^3[n] + 203(y_r^3[n-1] + y_r^3[n+1])) \\ \frac{\partial J_r}{\partial y_r^2[n]} = \frac{131}{90T}(2y_r[n] - y_r[n-1] - y_r[n+1]) + \frac{107}{45}(y_r^1[n+1] - y_r^1[n-1]) \\ \quad + \frac{T}{135}(886y_r^2[n] - 155(y_r^2[n-1] + y_r^2[n+1])) + \frac{137T^2}{540}(y_r^3[n+1] - y_r^3[n-1]) \\ \frac{\partial J_r}{\partial y_r^3[n]} = \frac{16}{45}(y_r[n-1] - y_r[n+1]) + \frac{T}{360}(618y_r^1[n] + 203(y_r^1[n-1] + y_r^1[n+1])) \\ \quad + \frac{137T^2}{540}(y_r^2[n-1] - y_r^2[n+1]) + \frac{T^3}{135}(66y_r^3[n] + 7(y_r^3[n-1] + y_r^3[n+1])) \end{array} \right. , \quad (\text{D.12})$$

para $0 < n < N - 1$, y finalmente

$$\left\{ \begin{array}{l}
\frac{\partial J_r}{\partial y_r[n]} = \frac{4}{3T^3}(y_r[n] - y_r[n-1]) - \frac{8}{3T^2}(y_r^1[n] + y_r^1[n-1]) \\
\quad + \frac{131}{90T}(y_r^2[n] - y_r^2[n-1]) - \frac{16}{45}(y_r^3[n] + y_r^3[n-1]) \\
\frac{\partial J_r}{\partial y_r^1[n]} = \frac{8}{3T^2}(y_r[n-1] - y_r[n]) + \frac{1}{30T}(183y_r^1[n] + 137y_r^1[n-1]) \\
\quad + \frac{1}{45}(107y_r^2[n-1] - 155y_r^2[n]) + \frac{T}{360}(203y_r^3[n-1] + 309y_r^3[n]) \\
\frac{\partial J_r}{\partial y_r^2[n]} = \frac{131}{90T}(y_r[n] - y_r[n-1]) - \frac{1}{45}(107y_r^1[n-1] + 155y_r^1[n]) \\
\quad + \frac{T}{135}(443y_r^2[n] - 155y_r^2[n-1]) - \frac{T^2}{540}(455y_r^3[n] + 137y_r^3[n-1]) \\
\frac{\partial J_r}{\partial y_r^3[n]} = \frac{16}{45}(y_r[n-1] - y_r[n]) + \frac{T}{360}(309y_r^1[n] + 203y_r^1[n-1]) \\
\quad + \frac{T^2}{540}(137y_r^2[n-1] - 455y_r^2[n]) + \frac{T^3}{135}(33y_r^3[n] + 7y_r^3[n-1])
\end{array} \right. , \quad (\text{D.13})$$

para $n = N - 1$.

Apéndice E

Demostración de la continuidad del modelo MSI

En este apéndice se muestran las expresiones de reconstrucción en el interior de cada cuadrícula definida por dos instantes de muestreo consecutivos en cada dirección del espacio, tal y como muestra la Figura E.1.

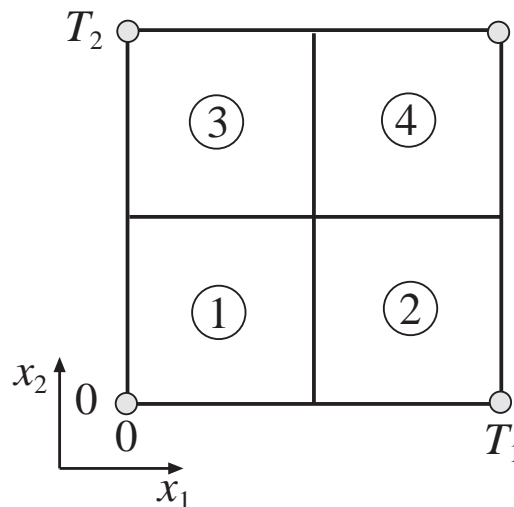


Figura E.1: Cuadrícula básica del espacio de entrada

Con el método de reconstrucción utilizado en el modelo MSI, la reconstrucción de la función divide cada rejilla en 4 regiones en las que se tiene una expresión de reconstrucción distinta. Para simplificar en lo posible las expresiones se considera el caso particular $D = 1$ (modelo de grado 2) y $T_1 = T_2 = T$. Seleccionando x_1 como dirección de reconstrucción, es preciso conocer $\hat{f}(x_1, x_2)$ y $\hat{f}_{x_1}(x_1, x_2)$ particularizadas para $x_1 = 0$ y $x_1 = T$, para aplicar el modelo MLI1D en la dirección x_1 . Vamos a comenzar evaluando los valores de estas dos funciones para $0 \leq x_2 \leq T/2$, con lo que

se puede reconstruir la función en las regiones marcadas en la Figura E.1 como 1 y 2. En este caso, utilizando el MLI1D en la dirección x_2 se tiene

$$\hat{f}(0, x_2) = \hat{f}(0, 0) + \hat{f}_{x_2}(0, 0)x_2 + \frac{\hat{f}_{x_2}(0, T/2) - \hat{f}_{x_2}(0, 0)}{T}x_2^2, \quad (\text{E.1})$$

donde $\hat{f}(0, 0)$ y $\hat{f}_{x_2}(0, 0)$ vienen dados por las muestras disponibles, y

$$\hat{f}_{x_2}(0, T/2) = 2\frac{\hat{f}(0, T) - \hat{f}(0, 0)}{T} - \frac{\hat{f}_{x_2}(0, T) - \hat{f}_{x_2}(0, 0)}{2}. \quad (\text{E.2})$$

Igualmente,

$$\hat{f}(T, x_2) = \hat{f}(T, 0) + \hat{f}_{x_2}(T, 0)x_2 + \frac{\hat{f}_{x_2}(T, T/2) - \hat{f}_{x_2}(T, 0)}{T}x_2^2, \quad (\text{E.3})$$

donde $\hat{f}(T, 0)$ y $\hat{f}_{x_2}(T, 0)$ vienen dados por las muestras disponibles, y

$$\hat{f}_{x_2}(T, T/2) = 2\frac{\hat{f}(T, T) - \hat{f}(T, 0)}{T} - \frac{\hat{f}_{x_2}(T, T) - \hat{f}_{x_2}(T, 0)}{2}. \quad (\text{E.4})$$

Para modelar la derivada con respecto a x_1 se toma un spline cuadrático que se puede escribir, de forma genérica

$$\hat{f}_{x_1}(0, x_2) = ax_2^2 + bx_2 + c, \quad (\text{E.5})$$

y

$$\hat{f}_{x_1}(T, x_2) = dx_2^2 + ex_2 + g. \quad (\text{E.6})$$

De este modo, en la región 1 la función se reconstruye como

$$\hat{f}^{R1}(x_1, x_2) = \hat{f}(0, x_2) + \hat{f}_{x_1}(0, x_2)x_1 + \frac{\hat{f}_{x_1}(T/2, x_2) - \hat{f}_{x_1}(0, x_2)}{T}x_1^2, \quad (\text{E.7})$$

donde

$$\hat{f}_{x_1}(T/2, x_2) = \frac{\hat{f}(0, x_2) + \hat{f}(T, x_2)}{2} + T\frac{\hat{f}_{x_1}(0, x_2) - \hat{f}_{x_1}(T, x_2)}{8}. \quad (\text{E.8})$$

En la región 2

$$\begin{aligned} \hat{f}^{R2}(x_1, x_2) = & \hat{f}(T/2, x_2) + \hat{f}_{x_1}(T/2, x_2)(x_1 - T/2) \\ & + \frac{\hat{f}_{x_1}(T, x_2) - \hat{f}_{x_1}(T/2, x_2)}{T}(x_1 - T/2)^2, \end{aligned} \quad (\text{E.9})$$

donde

$$\hat{f}(T/2, x_2) = 2\frac{\hat{f}(0, x_2) - \hat{f}(T, x_2)}{T} - \frac{\hat{f}_{x_1}(0, x_2) + \hat{f}_{x_1}(T, x_2)}{2}. \quad (\text{E.10})$$

Sustituyendo todas las expresiones, se llega en ambos casos a expresiones que dependen únicamente de los valores de la función y las derivadas en los instantes de muestreo (0

y T). Si se evalúa la continuidad en $x_1 = T/2$, la función y la derivada con respecto de x_1 son continuas por proceder de un MLI1D. En cuanto a la derivada con respecto a x_2 , si se derivan las expresiones (E.7) y (E.9) y, para simplificar, se evalúa la diferencia entre las expresiones resultantes, que denotamos como $Df_{x_2}^{R1-R2}(x_1, x_2)$, se llega a la expresión

$$\begin{aligned}
Df_{x_2}^{R1-R2}(x_1, x_2) = & -\frac{(2x_1 - T)^2}{2T^4} \left(8x_2(\hat{f}(0, T) + \hat{f}(T, 0) - \hat{f}(0, 0) - \hat{f}(T, T)) \right. \\
& + 2x_2T(3\hat{f}_{x_2}(T, 0) + \hat{f}_{x_2}(T, T) - 3\hat{f}_{x_2}(0, 0) - \hat{f}_{x_2}(0, T)) \quad (\text{E.11}) \\
& \left. + 2T^2(\hat{f}_{x_2}(0, 0) - \hat{f}_{x_2}(T, 0)) + T^3(b + e + 2x_2(a + d)) \right).
\end{aligned}$$

Se puede ver que la expresión se anula para $x_1 = T/2$, con lo que se asegura la continuidad de la derivada con respecto a x_2 .

Siguiendo la misma idea, se pueden calcular de forma sencilla las expresiones de reconstrucción en las regiones 3 y 4. Evaluando la continuidad, tanto de la función como de las dos derivadas, en las fronteras que dividen las regiones, se llega siempre a la conclusión de continuidad de las mismas. Por ejemplo, si se evalúa la diferencia entre las derivadas con respecto a x_1 en las regiones 1 y 3, que denotaremos como $Df_{x_1}^{R1-R3}(x_1, x_2)$, se obtiene la expresión

$$\begin{aligned}
Df_{x_1}^{R1-R3}(x_1, x_2) = & -2x_1 \frac{(2x_2 - T)^2}{T^4} \left(\right. \\
& T(\hat{f}_{x_2}(T, 0) + \hat{f}_{x_2}(T, T) - \hat{f}_{x_2}(0, 0) - \hat{f}_{x_2}(0, T)) \quad (\text{E.12}) \\
& \left. + 2(\hat{f}_{x_2}(T, 0) + \hat{f}_{x_2}(0, T) - \hat{f}_{x_2}(0, 0) - \hat{f}_{x_2}(T, T)) \right),
\end{aligned}$$

que se anula para $x_2 = T/2$, lo que indica la continuidad de la función. Es sencillo comprobar que el mismo resultado se obtiene para la función y las derivadas con respecto a x_1 y x_2 en cada frontera entre dos regiones. Por tanto, se demuestra que la solución obtenida con la alternativa MSI es continua dentro de cada rectángulo.

Apéndice F

Funciones de prueba en espacios bidimensionales

En este apéndice se presentan las funciones de prueba que se han utilizado en esta Tesis para evaluar las prestaciones de las diferentes técnicas propuestas en espacios de entrada bidimensionales. Estas funciones son las mismas que utilizan Cherkassky, Gehring y Mulier [Cherkassky96] para llevar a cabo una comparación entre diferentes alternativas de modelos de tipo adaptativo. Se trata de un conjunto de funciones obtenidas de diferentes referencias bibliográficas, cada una de diferentes características, por lo que forman un conjunto de pruebas suficientemente representativo. La Tabla F.1 presenta estas funciones, que se pueden ver en las Figuras F.1 y F.2.

Función 1. Obtenida de [Breiman91].

$$y = \sin(x_1 x_2)$$

con x_1 y x_2 uniformemente distribuidos entre $[-2, 2]$.

Función 2. Obtenida de [Breiman91].

$$y = \exp(x_1 \sin(\pi x_2))$$

con x_1 y x_2 uniformemente distribuidos entre $[-1, 1]$.

Función 3. Función GBCW.

$$a = 40 * \exp(8((x_1 - 0.5)^2 + (x_2 - 0.5)^2))$$

$$b = \exp(8((x_1 - 0.2)^2 + (x_2 - 0.7)^2))$$

$$c = \exp(8((x_1 - 0.7)^2 + (x_2 - 0.2)^2))$$

$$y = a/(b + c)$$

con x_1 y x_2 uniformemente distribuidos entre $[0, 1]$.

Función 4.

$$y = (1 + \sin(2x_1 + 3x_2))/(3.5 + \sin(x_1 - x_2))$$

con x_1 y x_2 uniformemente distribuidos entre $[-2, 2]$.

Función 5. Obtenida de [Maechler90].

$$y = 42.659(0.1 + x_1(0.05 + x_1^4 - 10x_1^2x_2^2 + 5x_2^4))$$

con x_1 y x_2 uniformemente distribuidos en $[-0.5, 0.5]$.

Función 6. Obtenida de [Maechler90].

$$y = 1.3356[1.5(1 - x_1) + \exp(2x_1 - 1)\sin(3\pi(x_1 - 0.6)^2) + \exp(3(x_2 - 0.5))\sin(4\pi(x_2 - 0.9)^2)]$$

con x_1 y x_2 uniformemente distribuidos en $[0, 1]$.

Función 7. Obtenida de [Maechler90].

$$y = 1.9[1.35 + \exp(x_1)\sin(13(x_1 - 0.6)^2) + \exp(3(x_2 - 0.5))\sin(4\pi(x_2 - 0.9)^2)]$$

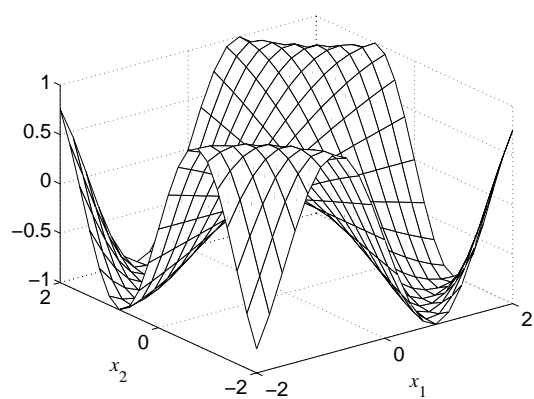
con x_1 y x_2 uniformemente distribuidos en $[0, 1]$.

Función 8. Obtenida de [Cherkassky91].

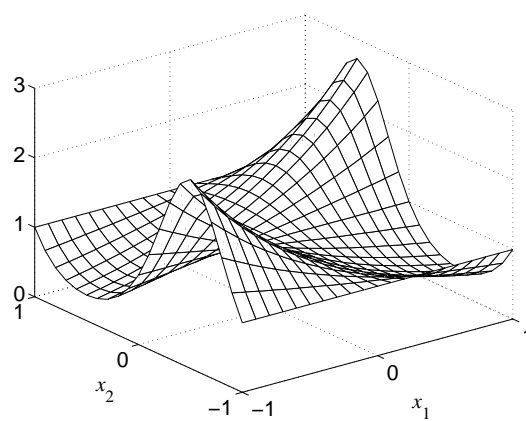
$$y = \sin(2\pi\sqrt{x_1^2 + x_2^2})$$

con x_1 y x_2 uniformemente distribuidos entre $[-1, 1]$.

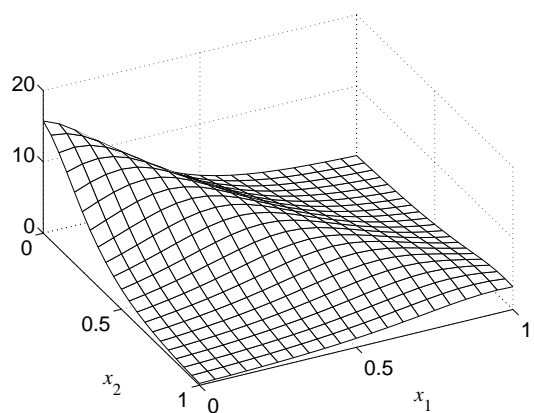
Tabla F.1: Funciones empleadas para la comparación de resultados



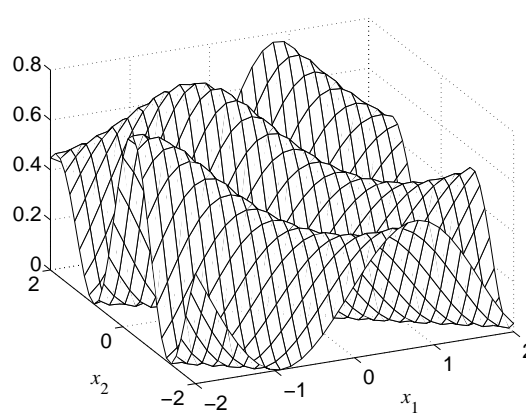
Función 1



Función 2



Función 3



Función 4

Figura F.1: Funciones de dos variables utilizadas en la comparación de resultados

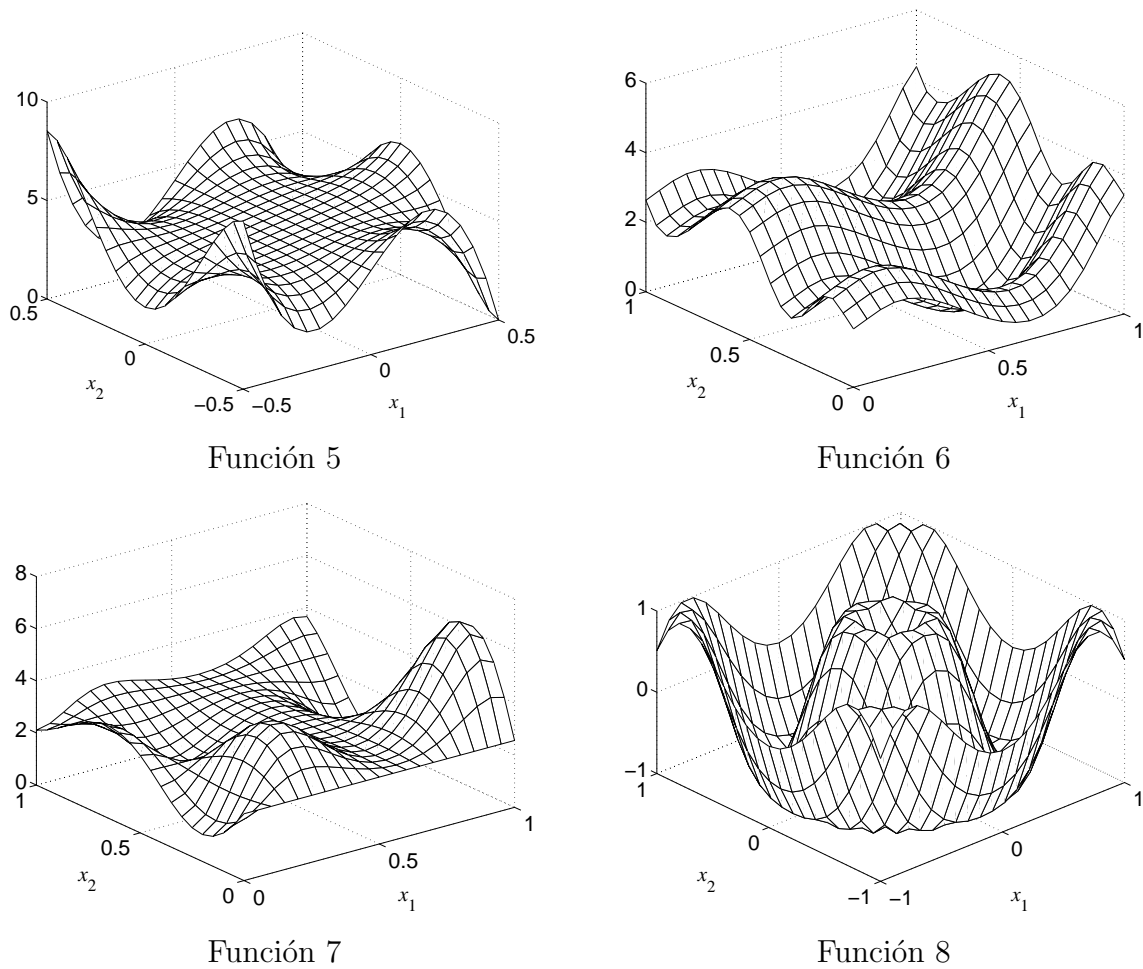


Figura F.2: Funciones de dos variables utilizadas en la comparación de resultados

Apéndice G

Publicaciones a las que ha dado lugar la realización de la Tesis

La realización de esta Tesis ha dado lugar al siguiente conjunto de publicaciones

Revistas:

- I. Santamaría, M. Lázaro, C. J. Pantaleón, J. A. García, A. Tazón, y A. Mediavilla. “A nonlinear MESFET model for intermodulation analysis using a generalized radial basis function network”. *Neurocomputing*, vol. 25, pp. 1–18, 1999.
- J. A. García, A. Tazón, A. Mediavilla, I. Santamaría, M. Lázaro, C. Pantaleón, y J. C. Pedro. “Modeling MESFETs and HEMTs intermodulation distortion behavior using a generalized radial basis function network”. *International Journal of RF and Microwave Computer Aided Engineering*, , no. 9, pp. 261–276, 1999.
- M. Lázaro, I. Santamaría, C. Pantaleón, A. Mediavilla, A. Tazón, y T. Fernández. “Smoothing the canonical piecewise-linear model: An efficient and derivable large-signal model for MESFET/HEMT transistors”. *IEEE Transactions on Circuits and Systems I*, vol. 48, no. 2, pp. 184–192, Febrero 2001.
- M. Lázaro, I. Santamaría, C. Pantaleón, “Neural Networks for Large and Small-Signal Modeling of MESFET/HEMT Transistors”. Aceptado para su publicación en *IEEE Transactions on Instrumentation and Measurement*
- A. Mediavilla, A. Tazón, J.A. Pereda, M. Lázaro y C. Pantaleón, I.Santamaría. “High Speed Analysis and Optimization of Waveguide Band-Pass Filter Structures Using Simple Neural Architectures”. Aceptado para publicación en *Microwave Journal*.

- M. Lázaro, I. Santamaría, y C. Pantaleón. “A new technique to avoid local minima and improve the convergence of EM-based training algorithms for RBF networks”. Enviado para su publicación en *Neural Networks*, Pergamon Press. (En período de revisión)

Congresos Internacionales:

- M. Lázaro, I. Santamaría, C. Pantaleón, “Neural Networks for Large and Small-Signal Modeling of MESFET/HEMT Transistors: A Comparative Study”. En *Proceedings of the 17 IEEE Instrumentation and Measurement Technology Conference (IMTC)*, vol 3, pp 1272-1277, Baltimore, EEUU, Mayo 2000.
- M. Lázaro, I. Santamaría, C. Pantaleón, C. Navarro, A. Tazón, y T. Fernández. “A modular neural network for global modeling of microwave transistors”. En *Proceedings of the International Joint Conference in Neural Networks (IJCNN)*, Como, Italia, Julio 2000.
- A. Mediavilla, A. Tazón, J. A. Pereda, M. Lázaro, I. Santamaría, C. Pantaleón. “Neuronal architecture for waveguide inductive iris bandpass filter optimization”. En *Proceedings of the International Joint Conference in Neural Networks (IJCNN)*, Como, Italia, Julio 2000.
- M. Lázaro, I. Santamaría, C. Pantaleón, “A Smooth and Derivable Large-Signal Model for Microwave HEMT Transistors”. En *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, Mayo 2000.
- M. Lázaro, I. Santamaría y C. Pantaleón. “Accelerating the convergence of EM-based training algorithms for RBF networks”. En *Proceedings of the International Work-conference on Artificial and Natural Neural Networks (IWANN)*, Granada, Junio 2001.
- M. Lázaro, I. Santamaría y C. Pantaleón. “A Regularized Digital Filtering Technique for the Simultaneous Reconstruction of a Function and its Derivatives”. En *Proceedings of the 8th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, Malta, Septiembre 2001.

Congresos Nacionales:

- M. Lázaro, J. A. García, C. Pantaleón, I. Santamaría, y J. Ibáñez. “Modelado no lineal de dispositivos mediante funciones lineales a tramos suavizadas”. En *Actas del XII Simposium Nacional de la Unión Científica Internacional de Radio (URSI)*, Bilbao, Septiembre de 1997.

- M. Lázaro, J. A. García, C. Pantaleón, y I. Santamaría. “Modelado no lineal de un MESFET para análisis de intermodulación mediante funciones lineales a tramos suavizadas”. En *Actas del XIII Simposium Nacional de la Unión Científica Internacional de Radio (URSI)*, Pamplona, Septiembre de 1998.
- M. Lázaro, I. Santamaría, C. Pantaleón, y J. Ibáñez. “Una nueva técnica para la interpolación simultánea de una función y sus derivadas”. En *Actas del XV Simposium Nacional de la Unión Científica Internacional de Radio (URSI)*, Zaragoza, Septiembre del 2000.
- M. Lázaro, I. Santamaría, y C. Pantaleón. “Introducción de la información de las derivadas de una función en el entrenamiento de redes neuronales”. En *Actas del XVI Simposium Nacional de la Unión Científica Internacional de Radio (URSI)*, Madrid, Septiembre del 2000.

Bibliografía

- [Abramowitz64] M. Abramowitz y I. A. Stegun. *Handbook of mathematical functions*. Dover, 1964.
- [Ahlberg67] J. H. Ahlberg, E. N. Nilson, y J.L. Walsh. *The theory of splines and their applications*. Academic Press, New York, 1967.
- [Aizerman64] M. Aizerman, E. Braverman y L. Rozonoer, “Theoretical foundations of the potential function method in pattern recognition learning”, *Automation and Remote Control*, vol. 25, pp. 821-837, 1964.
- [Aldroubi92] A. Aldroubi, M. Unser, y M. Eden. “Cardinal spline filters: stability and convergence to the ideal sinc interpolator”. *Signal Processing*, vol. 28, no. 2, pp. 127–138, 1992.
- [Anholt95] R. Anholt. *Electrical and thermal characterization of MESFET's, HEMT's and HBT's*. Artech House, Norwood, MA, 1995.
- [Anlauf89] J. K. Anlauf y M. Biehl, “The AdaTron: an adaptive perceptron algorithm”, *Europhys. Lett.*, vol. 10, pp. 687-692, 1989.
- [Bacon71] D. W. Bacon y D.W. Watts. “Estimating the transition between two intersecting straight lines”. *Biometrika*, vol. 58, pp. 525–534, 1971.
- [Barron93] A. R. Barron. “Universal approximation bounds for superpositions of a sigmoidal function”. *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 930–945, Mayo 1993.
- [Bartels87] R. H. Bartels, J.C Beaty, y B.A. Barsky. *Splines for use in computer graphics*. Morgan Kaufmann, Los Altos, CA, 1987.
- [Batruni91] R. Batruni. “A multilayer neural network with piecewise-linear structure and back-propagation learning”. *IEEE Transactions on Neural Networks*, vol. 2, no. 3, pp. 395–403, Mayo 1991.
- [Benedetto95] J. J. Benedetto y M.W. Frazier. *Wavelets: mathematics and applications*. CRC, Boca Raton, USA, 1995.

- [Bishop97] C. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1997.
- [Breiman91] L. Breiman. “The PI method for estimating multivariate functions from noisy data”. *Technometrics*, vol. 3, no. 2, pp. 125–160, 1991.
- [Breiman93] L. Breiman. “Hinging hiperplanes for regression, classification, and function approximation”. *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 999–1013, Mayo 1993.
- [Broomhead88] D. S. Broomhead y D. Lowe. “Multivariable functional interpolation and adaptive networks”. *Complex Systems*, vol. 2, pp. 321–355, 1988.
- [Burrus98] C. S. Burrus, R. A. Gopinath, y H. Guo. *Introduction to wavelets and wavelet transforms. A primer*. Prentice Hall, Upper Saddle River, New Jersey, 1998.
- [Cairon2000] J. M. Cairon, A. Tazón, T. Fernández, A. Mediavilla, C. Navarro, y J. Rodríguez. “A dynamic drain current model for the MES-FET/HEMT valid under varyng static bias and temperature conditions”. *Microwave Journal*, vol. 43, no. 11, pp. 122–134, Noviembre 2000.
- [Calandra94] E. F. Calandra y G. Sirna. “CAD-oriented modeling of the optically-controlled GaAs MESFET”. En *Proceedings of the GAAS'94 Symposium*, pp. 401–404, Torino, 1994.
- [Candocia98] F. Candocia y J. C. Principe. “Comments on sinc interpolation of discrete periodic signals”. *IEEE Transactions on Signal Processing*, vol. 46, no. 7, pp. 2044–2047, Junio 1998.
- [Cardaliaguet92] P. Cardaliaguet y G. Euvrard. “Approximation of a function and its derivative with a neural network”. *Neural Networks*, vol. 5, pp. 207–220, 1992.
- [Chen91] S. Chen, C. F. N. Cowan, y P. M. Grant. “Orthogonal least squares learning algorithm for radial basis functions”. *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302–309, Marzo 1991.
- [Cherkassky91] V. Cherkassky, Y. Lee, y H. Lari-Najafi. “Self-organizing network for regression: Efficient implementation and comparative evaluation”. En *Proceedings of the International Joint Conference on Neural Networks*, vol. 1, pp. 79–84, 1991.
- [Cherkassky96] V. Cherkassky, D. Gehring, y F. Mulier. “Comparison of adaptive methods for function estimation from samples”. *IEEE Transactions on Neural Networks*, vol. 7, no. 4, pp. 969–984, Julio 1996.

- [Chua85] L. O. Chua y A.C. Deng. “Canonical piecewise-linear analysis: generalized breakpoint hopping algorithm”. Technical Report UCB/ERL M85/36, University of California, Berkeley, Abril 1985.
- [Chua86] L. O. Chua y A.C. Deng. “Canonical piecewise-linear modeling”. *IEEE Transactions on Circuits and Systems*, vol. 33, no. 5, pp. 511–525, 1986.
- [Chua88] L. O. Chua y A. C. Deng. “Canonical piecewise-linear representation”. *IEEE Transactions on Circuits and Systems*, vol. 35, no. 1, pp. 101–111, 1988.
- [Chua77] L. O. Chua y S. M. Kang. “Section-wise, piecewise-linear functions: Canonical representation, properties and applications”. En *Proceedings of the IEEE*, vol. 65, pp. 915–929, 1977.
- [Chua83] L. O. Chua y R. Ying. “Canonical piecewise-linear analysis”. *IEEE Transactions on Circuits and Systems*, vol. 30, pp. 125–140, 1983.
- [Chui92] C. K. Chui. *An introduction to wavelets*. Academic Press, New York, 1992.
- [Cortes95] C. Cortes, V. Vapnik, “Support-Vector networks”, *Machine Learning*, vol. 20, pp- 273-297, 1995.
- [Crosmun89] A. M. Crosmun y S. A. Maas. “Minimization of intermodulation distortion in GaAs MESFET small-signal amplifiers”. *IEEE Transactions Microwave Theory and Techniques*, vol. 37, no. 9, pp. 1411–1417, 1989.
- [Curtice85] A. M. Curtice y M. Ettenberg. “A nonlinear GaAs FET model for use in the design of output circuits for power amplifiers”. *IEEE Transactions Microwave Theory and Techniques*, vol. 33, no. 12, pp. 1383–1394, 1985.
- [Cybenko89] G. Cybenko. “Approximation by superposition of a sigmoidal function”. *Math. Control, Signals and Systems*, vol. 2, pp. 303–314, 1989.
- [Dambrine88] G. Dambrine, A. Cappy, F. Heliodore, y E. Playez. “A new method for determining the FET small-signal equivalent circuit”. *IEEE Transactions Microwave Theory and Techniques*, , no. 36, pp. 1151–1159, 1988.
- [Daubechies92] I. Daubechies. *Ten lectures on wavelets*. SIAM, Philadelphia, PA, 1992.
- [DeBoor72] C. de Boor. “On calculating with B-splines”. *Journal on Approximation Theory*, vol. 6, pp. 50–62, 1972.

- [DeBoor78] C. de Boor. *A practical guide to splines*. Springer-Verlag, New York, 1978.
- [Dempster77] A. P. Dempster, N. M. Laird, y D. B. Rubin. “Maximum likelihood from incomplete data via the EM algorithm”. *Journals of Royal Statistics Society B*, vol. 39, pp. 1–38, 1977.
- [Diercks95] P. Diercks. *Curve and surface fitting with splines*. Oxford University Press, Walton Street, Oxford, 1995.
- [Djokovic97] I. Djokovic y P. P. Vaidyanathan. “Generalized sampling theorems in multiresolution subspaces”. *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 583–599, Marzo 1997.
- [Dooley2000] S. R. Dooley y A. K. Nandi. “Notes on the interpolation of discrete periodic signals using sinc function related approaches”. *IEEE Transactions on Signal Processing*, vol. 48, no. 4, Abril 2000.
- [Duchon77] J. Duchon. “Spline minimizing rotation-invariant semi-norms in Sobolev spaces. Constructive theory of functions of several variables”. *Lecture Notes in Mathematics*, vol. 571, 1979.
- [Elbadawi83] I. Elbadawi, A.R. Gallant, y G. Souza. “An elasticity can be estimated consistently without a priori knowledge of functional form”. *Econometrica*, vol. 51, pp. 1731–1752, 1983.
- [Feder88] M. Feder y E. Weinstein. “Parameter estimation of superimposed signals using the EM algorithm”. *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, pp. 477–489, 1988.
- [Feichtinger95] H. G. Feichtinger y K. Gröchenig. *Wavelets: mathematics and applications*, de J. J. Benedetto y M. W. Frazier, Capítulo: “Theory and practice of irregular sampling”. CRC, Boca Raton, USA, 1995.
- [Fernández96] T. Fernández, Y. Newport, J. M. Zamanillo, A. Tazón, y A. Mediavilla. “Extracting a bias-dependent large signal MESFET model from I/V measurements”. *IEEE Transactions Microwave Theory and Techniques*, vol. 44, no. 3, pp. 372–378, Marzo 1996.
- [Fogel55] L. J. Fogel. “A note on the sampling theorem”. *IRE Trans. Inform. Theory*, vol. IT-1, pp. 47–48, 1955.
- [Frieb98] T. Frieb, N. Cristianini y C. Campbell, “The kernel-Adatron algorithm: a fast and simple learning procedure for support vector machines”, in *Machine Learning: Proc. of the 15th Int. Conf.*, Shavlik, J. Ed., San Francisco: Morgan Kaufman Publishers, 1998.

- [Funahashi89] K. Funahashi. “On the approximate realization of continuous mappings by neural networks”. *Neural Networks*, vol. 2, pp. 183–192, 1989.
- [Gallant92] A. R. Gallant y H. White. “On learning the derivatives of an unknown mapping with multilayer feedforward networks”. *Neural Networks*, vol. 5, pp. 129–138, 1992.
- [García99] J. A. García, J. C. Pedro, M. L. De la Fuente, N. Carvalho, A. Mediavilla, y A. Tazón. “Resistive FET mixer conversion loss and IMD optimization by selective drain bias”. *IEEE Transactions Microwave Theory and Techniques*, vol. 47, no. 12, pp. 2382–2392, Diciembre 1999.
- [García99b] J. A. García, A. Tazón, A. Mediavilla, I. Santamaría, M. Lázaro, C. Pantaleón, y J. C. Pedro. “Modeling MESFETs and HEMTs intermodulation distortion behavior using a generalized radial basis function network”. *International Journal of RF and Microwave Computer Aided Engineering*, , no. 9, pp. 261–276, 1999.
- [Ghahramani94] Z. Ghahramani y M. I. Jordan. “Supervised learning from incomplete data via an EM approach”. En J. D. Cowan, G. Tesauro, y J. Alspector, editors, *Advances in NIPS VI*, San Mateo, CA, 1994. Morgan Kaufmann, CA.
- [Gilstrap89] L. Gilstrap y R. Dominy. “A general explanation and interrogation system for neural networks”. En *Proceedings of the International Joint Conference on Neural Networks*, Washington D.C., 1989.
- [Golio91] J. M. Golio. *Microwave MESFET's and HEMT's*. Artech House, Norwood, MA, 1991.
- [Goshtasby90] A. Goshtasby, F. Cheng, y B. A. Barsky. “B-spline curves and surfaces viewed as digital filters”. *Computer Vision, Graphics, and Image Processing*, vol. 52, pp. 264–275, 1990.
- [Griffits73] D.A. Griffits y A. Miller. “Hyperbolic regresion - a model based on two phase piecewise linear regression with a smooth transition between regimes”. *Commun. Stat.*, vol. 2, pp. 561–569, 1973.
- [Haar10] A. Haar. “Zur theorie der orthogonalen funktionensysteme”. *Mathematische Annalen*, vol. 69, pp. 331–371, 1910.
- [Haykin94] S. Haykin. *Neural networks: A comprehensive foundation*. Macmillan Publishing Company, Englewoods Cliffs, NJ, USA, 1994.

- [Hecht89] R. Hecht-Nielsen. “Theory of the backpropagation neural network”. En *Proceedings of the International Joint Conference on Neural Networks*, vol. I, pp. 593–606, Washington D.C., 1989.
- [Hecht90] R. Hecht-Nielsen. *Neurocomputing*. Addison-Wesley, Reading, MA, USA, 1990.
- [Higgins96] J. R. Higgins. *Sampling Theory in Fourier and Signal Analysis*. Oxford University Press, Walton Street, Oxford, NY, 1996.
- [Hornik89] K. Hornik, M. Stinchcombe, y H. White. “Multilayer feedforward networks are universal approximators”. *Neural Networks*, vol. 2, pp. 183–192, 1989.
- [Hornik90] K. Hornik, M. Stinchcombe, y H. White. “Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks”. *Neural Networks*, vol. 3, pp. 551–560, 1990.
- [Hoskins94] R. F. Hoskins y J. Sousa Pinto. *Distributions, ultradistributions and other generalized functions*. Ellis Horwood, Chichester, 1994.
- [Hush92] D. R. Hush y B. Horne. “An overview of neural networks. part I: Static networks”. *Informática y Automática*, vol. 25, no. 1, pp. 19–36, 1992.
- [Hush92a] D. R. Hush y B. Horne. “An overview of neural networks. part II: Dynamic networks”. *Informática y Automática*, vol. 25, no. 2, pp. 17–32, 1992.
- [Ito93] Y. Ito. “Extension of approximation capability of three layered neural networks to derivatives”. En *Proceedings of the IEEE International Conference on Neural Networks*, vol. 3, pp. 366–370, New York, NY, USA, 1993.
- [Jacobs91] R. A. Jacobs, M.I Jordan, S. J. Nowlan, y G. E. Hinton. “Adaptive mixtures of local experts”. *Neural Computation*, , no. 3, pp. 79–87, 1991.
- [Janssen93] A. J. E. M. Janssen. “The Zak transform and sampling theorems for wavelet subspaces”. *IEEE Transactions on Signal Processing*, vol. 41, pp. 3360–3365, Diciembre 1993.
- [Jerri77] A. J. Jerri. “The Shannon sampling theorem - its various extension and applications: A tutorial review”. *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1565–1596, 1977.
- [Jordan89] M. Jordan. “Generic constraints on under specified target trajectories”. En *Proceedings of the International Joint Conference on Neural Networks*, vol. 1, pp. 593–606, 1989.

- [Jordan94] M. I. Jordan y R. A. Jacobs. “Hierarchical mixtures of experts and the EM algorithm”. *Neural Computation*, vol. 6, pp. 181–214, 1994.
- [Julián98] P. Julián, M. Jordán, y A. Desages. “Canonical piecewise-linear approximation of smooth functions”. *IEEE Transactions on Circuits and Systems I*, vol. 45, no. 5, pp. –, Mayo 1998.
- [Kang78] S. M. Kang y L. O. Chua. “A global representation of multidimensional piecewise-linear functions with linear partitions”. *IEEE Transactions on Circuits and Systems*, vol. 25, no. 11, pp. 938–949, 1978.
- [Karayanis97] N. B. Karayanis. “Gradient descent learning of radial basis neural networks”. En *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1825–1830, Houston, TX, 1997.
- [Katok95] A. Katok y B. Hasselblatt. *Introduction to the modern theory of dynamical systems*. Cambridge University Press, Cambridge, 1995.
- [Kotelnikov33] V. A. Kotelnikov. “On the transmission capacity of ‘ether’ and wire in electrocommunications”. *Izd. Red. Upr. Svyazzi RKKA (Moscow)*, 1933.
- [Lancaster86] P. Lancaster y K. Salkauskas. *Curve and surface fitting: An introduction*. Academic Press, London, 1986.
- [Lázaro2000] M. Lázaro, I. Santamaría, C. Pantaleón, C. Navarro, A. Tazón, y T. Fernández. “A modular neural network for global modeling of microwave transistors”. En *Proceedings of the International Joint Conference in Neural Networks (IJCNN2000)*, Como, Italia, 2000.
- [Lázaro2001] M. Lázaro, I. Santamaría, C. Pantaleón, A. Mediavilla, A. Tazón, y T. Fernández. “Smoothing the canonical piecewise-linear model: An efficient and derivable large-signal model for MESFET/HEMT transistors”. *IEEE Transactions on Circuits and Systems I*, vol. 48, no. 2, pp. 184–192, Febrero 2001.
- [Lázaro2001b] M. Lázaro, I. Santamaría, y C. Pantaleón. “Accelerating the convergence of EM-based training algorithms for RBF networks”. En *Proceedings of the International Work-conference on Artificial and Natural Neural Networks (IWANN)*, Granada, Junio 2001.
- [Leshno93] M. Leshno, V. Lin, A. Pinkus, y S. Schocken. “Multilayer feedforward networks with a nonpolynomial activation function can approximate any function”. *Neural Networks*, vol. 6, pp. 861–867, 1993.

- [Li96] X. Li. “Simultaneous approximations of multivariate functions and their derivatives by neural networks with one hidden layer”. *Neurocomputing*, vol. 12, pp. 327–343, 1996.
- [Lin90] J. N. Lin y R. Unbehauen. “Adaptive nonlinear digital filter with canonical piecewise-linear structure”. *IEEE Transactions on Circuits and Systems*, vol. 37, no. 3, pp. 347–353, Marzo 1990.
- [Lin93] J. N. Lin y R. Unbehauen. “Canonical PWL network and multilayer perceptron-like networks: A unified view”. En *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 2588–2591, New York, NY, USA, Mayo 1993.
- [Lin95] J. N. Lin y R. Unbehauen. “Canonical piecewise-linear networks”. *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 43–50, Enero 1995.
- [Linden60] D. A. Linden y N. M. Abramson. “A generalization of the sampling theorem”. *Inform. Control*, vol. 3, pp. 26–31, 1960.
- [Linden59] D. A. Linden. “A discussion of sampling theorems”. En *Proceedings of the IRE*, vol. 47, pp. 1219–1266, 1959.
- [Lowe89] D. Lowe. “Adaptive radial basis functions nonlinearities and the problem of generalization”. *1st IEEE Conference on Artificial Neural Networks*, 1989.
- [Ma97] S. Ma, C. Ji, y J. Farmer. “An efficient EM-based training algorithm for feedforward neural networks”. *Neural Networks*, vol. 10, pp. 243–256, 1997.
- [Ma98] S. Ma y C. Ji. “Fast training of recurrent networks based on the EM algorithm”. *IEEE Transactions on Neural Networks*, vol. 9, pp. 11–26, 1998.
- [Maas90] S. A. Maas y D. Neilson. “Modeling MESFETs for intermodulation analysis of mixers and amplifiers”. *IEEE Transactions Microwave Theory and Techniques*, vol. 38, no. 12, pp. 1964–1971, 1990.
- [Maas91] S. A. Maas. “How to model intermodulation distortion”. *IEEE MTT-S International Microwave Symposium Digest*, pp. 149–151, 1991.
- [Madjar92] A. Madjar, P. R. Herczfeld, y A. Paoella. “Analytical model for optically generated currents in GaAs MESFETs”. *IEEE Transactions Microwave Theory and Techniques*, vol. 40, 1992.

- [Maechler90] M. Maechler, D. Martin, J. Schimert, M. Csoppensky, y J. N. Hwang. “Projection pursuit learning networks for regression”. En *Proceedings of the International Conference in Tools AI*, vol. 1, pp. 350–358, 1990.
- [Mallat89] S. G. Mallat. “Multifrequency channel decomposition of images and wavelet models”. *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, pp. 2091–2110, Diciembre 1989.
- [Marvasti93] F. Marvasti. *Advanced topics in Shannon sampling and interpolation theory*, Capítulo: “Nonuniform sampling”. Springer-Verlag, New York, 1993.
- [McCamant90] A. McCamant, G. McCormak, y D. Smith. “An improved GaAs MES-FET for SPICE”. *IEEE Transactions Microwave Theory and Techniques*, vol. 38, no. 6, pp. 822–824, 1990.
- [Meijer90] P. B. L. Meijer. “Fast and smooth highly nonlinear multidimensional table models for device modeling”. *IEEE Transactions on Circuits and Systems*, , no. 37, pp. 335–346, 1990.
- [Meinguet79] J. Meinguet. “Multivariate interpolation at arbitrary points made simple”. *Journal Applied Mathematics and Physics*, vol. 30, pp. 292–304, 1979.
- [Mendel95] J. M. Mendel. “Fuzzy logic systems for engineering: A tutorial”. *Proceedings of the IEEE*, vol. 83, no. 3, pp. 345–377, Marzo 1995.
- [Mhaskar96] H. N. Mhaskar. “Neural networks for optimal approximation of smooth and analytic functions”. *Neural Computation*, vol. 8, pp. 164–177, 1996.
- [Moody89] J. E. Moody y C. J. Darken. “Fast learning in networks of locally-tuned processing units”. *Neural Computation*, vol. 1, pp. 281–294, 1989.
- [Muller98] K. R. Muller, A. J. Smola, G. Ratsch, B. Schölkopf, J. Kohlmorgen, V. Vapnik, “Using Support Vector Machines for time series prediction”, in B. Schölkopf, C. J. Burges, A. J. Smola, *Advances in Kernel Methods: Support Vector Learning*, Cambridge, MA, MIT Press., 1998.
- [Nachbin67] L. Nachbin. *Elements of approximation theory*. Princeton, New Jersey, 1967.
- [Navarro98] C. Navarro, J. M. Zamanillo, A. Mediavilla, y J. L. García. “Large signal dynamic properties of GaAs MESFET/HEMT devices under optical illumination”. En *Proceedings of the GAAS’98 Symposium*, pp. 350–353, 1998.

- [Nguyen90] D. Nguyen y B. Widrow. “Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights”. En *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, pp. 21–26, New York, USA, 1990.
- [Nguyen99] T. Nguyen-Thien y T. Tran-Cong. “Approximation of functions and their derivatives: A neural network implementation with applications”. *Neurocomputing*, vol. 23, pp. 687–704, 1999.
- [Pantaleón94] C. J. Pantaleón. *Nuevos métodos de modelado localmente lineal*. PhD thesis, Universidad Politécnica de Madrid, Madrid, Spain, 1994.
- [Park91] J. Park y I.W. Sandberg. “Universal approximation using radial basis function networks”. *Neural Computation*, vol. 3, pp. 246–257, 1991.
- [Pedro94] J. C. Pedro y J. Pérez. “Accurate simulation of GaAs MESFETs intermodulation using a new drain-source current model”. *IEEE Transactions Microwave Theory and Techniques*, vol. 42, no. 1, pp. 25–33, 1994.
- [Pérez99] F. Pérez, A. Navia, J. L. Rojo y A. Artés, “A new training algorithm for support vector machines”, In *Proc. of the 5th Bayona Workshop on Emerging Technologies in Telecommunications*, vol. I, Bayona, Spain, 1999.
- [Pérez2000] F. Pérez, A. Navia, P. L. Alarcón, A. Artés, “Support vector classifier with hyperbolic tangent penalty function”, *Proc. of ICASSP 2000*, Istanbul, Turkey, June, 2000.
- [Perry93] K. M. Perry y S. J. Reeves. “A practical stopping rule for iterative signal restoration”. En *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. III, pp. 440–443, Minneapolis (MI), 1993.
- [Platt98] J. Platt, “Fast training of support vector machine sequential minimal optimization”, in B. Schölkopf, C. J. Burges, A. J. Smola, *Advances in Kernel Methods: Support Vector Learning*, Cambridge, MA, MIT Press., 1998.
- [Poggio90] T. Poggio y F. Girosi. “Networks for approximation and learning”. *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1481–1497, 1990.
- [Powell85] M. J. D. Powell. “Radial basis functions for multivariable interpolation: A review”. En *IMA Conference on Algorithms for the Approximation of Functions and Data*, pp. 143–167, 1985.

- [Powell92] M.J.D. Powell. “The theory of radial basis functions approximation”. *Advances in Numerical Analysis III, Wavelets, Subdivision Algorithms and Radial Basis Functions*, pp. 105–210, 1992.
- [Prenter75] P. M. Prenter. *Splines and variational methods*. Wiley, New York, 1975.
- [Rawn89] M. D. Rawn. “A stable nonuniform sampling expansion involving derivatives”. *IEEE Transactions on Information Theory*, vol. 35, no. 6, pp. 1223–1227, 1989.
- [Razafinjatovo95] H. N. Razafinjatovo. “Iterative reconstructions in irregular sampling with derivatives”. *Journ. Fourier Anal. Appl.*, vol. 1, no. 3, pp. 281–295, 1995.
- [Reeves90] S. J. Reeves y R. M. Mersereau. “Optimal estimation of the regularization parameters and stabilizing functional for regularized image restoration”. *Optical Engineering*, vol. 29, no. 5, pp. 446–454, Mayo 1990.
- [Reinsh67] C. H. Reinsh. “Smoothing by spline functions”. *Numer. Math.*, vol. 10, pp. 177–183, 1967.
- [Rodríguez2000] J. Rodríguez, T. Fernández, A. Mediavilla, y A. Tazón. “Pseudo random pulsed I/V characterisation of MESFET/HEMT devices”. *Electronic Letters*, vol. 36, no. 12, pp. 1075–1076, Junio 2000.
- [Root91] D. E. Root, S. Fan, y J. Meyer. “Technology independent large signal nonquasi-static fet models by direct construction from automatically characterized device data”. En *Proceedings of the 21th European Microwave Conference*, pp. 923–927, Stuttgart, Germany, 1991.
- [Rousset96] J. Rousset, Y. Harkouss, J. M. Collantes, y M. Campovecchio. “An accurate neural network model of fet for intermodulation and power analysis”. En *Proceedings of the 26th European Microwave Conference*, Praga, 1996.
- [Rumelhart86] D. E. Rumelhart y J. L. McClelland. *Parallel distributed procesing: Exploring in the microstructure of cognition*. MIT Press, Cambridge, MA, USA, 1986.
- [Rumelhart86b] D. E. Rumelhart, G. E. Hinton, y R. J. Willians. “Learning representations by back-propagating errors”. *Nature (London)*, vol. 323, pp. 533–536, 1986.

- [Santamaría99] I. Santamaría, M. Lázaro, C. J. Pantaleón, J. A. García, A. Tazón, y A. Mediavilla. “A nonlinear MESFET model for intermodulation analysis using a generalized radial basis function network”. *Neurocomputing*, vol. 25, pp. 1–18, 1999.
- [Sauer87] K. D. Sauer y J.P. Allebach. “Iterative reconstruction of bandlimited images from nonuniformly spaced samples”. *IEEE Transactions on Circuits and Systems*, vol. 34, no. 2, pp. 1497–1506, 1987.
- [Schanze95] T. Schanze. “Sinc interpolation of discrete periodic signals”. *IEEE Transactions on Signal Processing*, vol. 43, no. 6, pp. 1502–1503, Junio 1995.
- [Schoenberg46] I. J. Schoenberg. “Contribution to the problem of approximation of equidistant data by analytic functions”. *Quart. Appl. Math.*, vol. 4, pp. 45–99, 112–141, 1946.
- [Schoenberg64] I. J. Schoenberg. “Spline functions and the problem of graduation”. *Proc. Nat. Acad. Sci.*, vol. 52, pp. 947–950, 1964.
- [Schoenberg73] I. J. Schoenberg. *Cardinal spline interpolation*. Society of Industrial and Applied Mathematics, Philadelphia, 1973.
- [Schölkopf97] B. Schölkopf, K.K. Sung, C. J. Burges, F. Girosi, P. Niyogi, T. Poggio y V. Vapnik, “Comparing Support Vector Machines with Gaussian kernels to Radial Basis Function Classifiers”, *IEEE Trans. on Signal Processing*, vol. 45, no. 11, pp. 2758-2765, 1997.
- [Schumaker81] L. L. Schumaker. *Spline functions: Basic theory*. Wiley, New York, 1981.
- [Seber89] G. A. F. Seber y C.J. Wild. *Nonlinear Regression*. John Wiley, New York, 1989.
- [Shannon49] C. E. Shannon. “Communications in the presence of noise”. En *Proceedings of the IRE*, vol. 37, pp. 10–21, 1949.
- [Shirakawa97] K. Shirakawa, M. Shimiz, N. Okubo, y Y. Daido. “A large signal characterization of a hemt using a multilayered neural network”. *IEEE Transactions Microwave Theory and Techniques*, vol. 9, no. 45, pp. 1630–1633, 1997.
- [Smola98] A. J. Smola, *Learning with Kernels*, PhD Thesis, Technische Universität Berlin, 1998.

- [Stinchcombe89] M. Stinchcombe y H. White. “Universal approximation using feedforward networks with non-sigmoidal hidden layer activation functions”. En *Proceedings of the International Joint Conference on Neural Networks*, vol. I, pp. 613–617, Washington D. C., 1989.
- [Tikhonov77] A. N. Tikhonov y V. Y. Arsenin. *Solution to ill-posed problems*. Winston and Sons, Washington D. C., 1977.
- [Unser91] M. Unser, A. Aldroubri, y M. Eden. “Fast B-spline transforms for continuous image representation and interpolation”. *IEEE Transactions on Pattern Anal. Machine Intell.*, vol. 13, no. 3, pp. 277–285, 1991.
- [Unser93a] M. Unser, A. Aldroubri, y M. Eden. “B-spline signal processing: Part I - theory”. *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 821–833, 1993.
- [Unser93b] M. Unser, A. Aldroubri, y M. Eden. “B-spline signal processing: Part II - efficient design and applications”. *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 834–848, 1993.
- [Unser99] M. Unser. “Splines: A perfect fit for signal and image processing”. *IEEE Signal Processing Magazine*, vol. 16, no. 6, pp. 21–38, nov 1999.
- [Unser2000] M. Unser. “Sampling - 50 years after Shannon”. *Proceedings of the IEEE*, vol. 88, no. 4, pp. 569–587, Abril 2000.
- [Vaidyanathan2001] P. P. Vaidyanathan y B. Vrcelj. “On sampling theorems for non bandlimited signals”. En *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, Utah, USA, vol. 6, 2001.
- [Vapnik95] V. Vapnik, *The Nature of Statistical Learning Theory*, New York: Springer Verlag, 1995.
- [Vapnik97] V. Vapnik, S. Golowich y A. Smola, “Support vector method for function approximation, regression estimation, and signal processing”. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 281–287, Cambridge, MA, MIT Press., 1997.
- [Varian78] H. Varian. *Microeconomic analysis*. Ed Norton, New York, 1978.
- [Vinod85] H. D. Vinod y A. Ullah. “Flexible production function estimation by non-parametric kernel estimators. Technical report, Economie Department, West Ontario University, 1995.

- [Walter92] G. G. Walter. “A sampling theorem for wavelet subspaces”. *IEEE Transactions on Information Theory*, vol. 38, pp. 881–884, Marzo 1992.
- [Webster96] D. R. Webster. *Low-Power HF microelectronics*, Capítulo: Distortion in short channel FET circuits. G. Machado Ed., London, 1996.
- [Werbos74] P. J. Werbos. *Beyond regression: New tools for prediction and analysis in the behavioral sciences*. PhD thesis, Harvard University, Cambridge, MA, USA, 1974.
- [Whittaker29] J. M. Whittaker. “The Fourier theory of the cardinal functions”. En *Proceedings of the Mathematical Society of Edinburgh*, vol. 1, pp. 169–176, 1929.
- [Whittaker35] J. M. Whittaker. *Interpolatory function theory*. Cambridge University Press, 1935.
- [Wu83] C. F. Wu. “On the convergence properties of the EM algorithm”. *Annals of Statistics*, , no. 11, pp. 95–103, 1983.
- [Young80] R. M. Young. *An introduction to nonharmonic Fourier series*. Academic Press, New York, 1980.
- [Zaabab95] A. H. Zaabab, Q. J. Zhang, y M. Nakhla. “A neural network modeling approach to circuit optimization and statistical design”. *IEEE Transactions Microwave Theory and Techniques*, vol. 6, no. 43, pp. 1349–1358, 1995.

Índice de Tablas

3.1	Resultados obtenidos en la reconstrucción de las funciones de prueba, utilizando únicamente las derivadas direccionales, con 5 instantes de muestreo en cada dirección del espacio de entrada y una SNR de 20 dB en las muestras de la función y las derivadas	83
3.2	Resultados obtenidos en la reconstrucción de las funciones de prueba, utilizando únicamente las derivadas direccionales, con 15 instantes de muestreo en cada dirección del espacio de entrada y una SNR de 20 dB en las muestras de la función y las derivadas	84
3.3	Resultados obtenidos en la reconstrucción de las funciones de prueba, utilizando las derivadas direccionales y cruzadas, con 5 instantes de muestreo en cada dirección del espacio de entrada y una SNR de 20 dB en las muestras de la función y las derivadas	86
3.4	Resultados obtenidos en la reconstrucción de las funciones de prueba, utilizando las derivadas direccionales y cruzadas, con 15 instantes de muestreo en cada dirección del espacio de entrada y una SNR de 20 dB en las muestras de la función y las derivadas	86
4.1	Radios de curvatura de las funciones de suavizado	97
4.2	Carga Computacional de las diferentes alternativas de entrenamiento de la red MLTS	114
4.3	Comparación de los requerimientos computacionales para el entrenamiento de las redes MLP y MLTS en un espacio de entrada unidimensional	115
4.4	Resultados obtenidos promediando el 5% de los mejores resultados . . .	117
4.5	Resultados obtenidos promediando el 25% de los mejores resultados . .	118
4.6	Resultados obtenidos promediando el 50% de los mejores resultados . .	118
4.7	Carga computacional de las diferentes alternativas en espacios de entrada bidimensionales	120
5.1	Resultados obtenidos con una red MLTS de 32 neuronas (100 parámetros) utilizando las funciones de coste J y J_d	128
5.2	Resultados obtenidos con un MLP de 25 neuronas (101 parámetros) utilizando las funciones de coste J y J_d	129
5.3	Resultados obtenidos con una red GRBF de 20 neuronas (100 parámetros) utilizando las funciones de coste J y J_d	129

5.4	Resultados obtenidos con un MLTS de 12 neuronas (40 parámetros) utilizando las funciones de coste J y J_d	130
5.5	SER promedio (en dB) para las diferentes funciones de prueba	146
5.6	Resultados obtenidos en la aproximación de los coeficientes de pequeña señal (SER en dB)	148
6.1	Comparación de resultados obtenidos en el modelado de un transistor HEMT Philips D02AH (SER en dB)	154
6.2	Carga computacional asociada a los distintos modelos expresada mediante el tiempo de entrenamiento	156
6.3	Resultados para un MESFET NE 72084	161
6.4	Resultados para un HEMT D02AH	161
6.5	Resultados del módulo pequeña señal, entrenado con el algoritmo EM, para un MESFET NE 72084	165
F.1	Funciones empleadas para la comparación de resultados	204

Índice de Figuras

1.1	Circuito equivalente de un transistor MESFET	3
2.1	Funciones de escala de Haar y triangular	24
2.2	Interpretación mediante banco de filtros de reconstrucción perfecta del problema de reconstrucción de una función a partir de un conjunto de muestras de la misma y de sus derivadas	27
2.3	Interpretación polifase del problema de reconstrucción de una función a partir de las muestras de la misma y de sus derivadas	27
2.4	B-splines, $\beta^k(x)$, de grados 0 hasta 3	31
2.5	Respuestas al impulso y en frecuencia de un spline cúbico interpolador	35
2.6	Funciones $\varphi(x)$ y $\varphi^{(1)}(x)$ y sus muestras en puntos enteros	38
2.7	Arquitectura de red neuronal hacia delante típica del Perceptrón Multicapa	44
3.1	Posición de los puntos de ruptura insertados	55
3.2	Representación mediante un banco de filtros MIMO del modelo MLI de grado 2 ($D = 1$)	57
3.3	Respuestas al impulso de los filtros reconstructores	58
3.4	Derivadas del modelo para el cálculo de la respuesta del filtro reconstuctor $h_1(x)$	59
3.5	Respuesta en frecuencia de los filtros reconstructores	59
3.6	Esquema de reconstrucción en la dirección x_1	65
3.7	Reconstrucción de una señal limitada en banda muestreada a 2 veces la velocidad de Nyquist con el modelo MLI y con un spline cúbico. La SNR en las muestras de la función es de 20 dB, y los resultados se presentan en función de la SNR común en las muestras de las dos derivadas	74
3.8	Reconstrucción de una señal limitada en banda muestreada a 3 veces la velocidad de Nyquist con el modelo MLI y con un spline cúbico con una SNR en las muestras de la función de 20 dB	75
3.9	Reconstrucción de una función y su derivada primera con los modelos MLI y MLR de grado 3	76
3.10	Reconstrucción de una función a partir de las muestras de la misma y de las derivadas primera y segunda muestreadas cada una a 2/3 de la frecuencia de Nyquist y con una SNR en las muestras de la función de 10 dB, en función de la SNR en las muestras de las derivadas.	77

3.11	Reconstrucción de una función a partir de las muestras de la misma y de sus derivadas primera y segunda muestreadas a la frecuencia de Nyquist y con una SNR en las muestras de la función de 10 dB, en función de la SNR en las muestras de las derivadas.	78
3.12	Reconstrucción de una función de ancho de banda 1, con una SNR en las muestras de función y derivadas de 10 dB, en función del período de muestreo.	78
3.13	Reconstrucción de la Función 2 mediante los modelos MSI y MSR con una SNR de 10 dB en las muestras de función y derivadas, y con 10 instantes de muestreo equiespaciados en cada dirección del espacio de entrada	80
3.14	Resultados en la reconstrucción de la Función 1 con una SNR de 20 dB en las muestras de la función y 15 instantes de muestreo en cada dirección del espacio de entrada	81
3.15	Resultados en la reconstrucción de la Función 1 con una SNR de 20 dB en las muestras de la función y 5 instantes de muestreo en cada dirección del espacio de entrada	82
3.16	Resultados en la reconstrucción, mediante los modelos de grado 3, de la Función 1 con una SNR de 20 dB en las muestras de la función	82
3.17	Resultados en la reconstrucción, mediante los modelos de grado 3, de la Función 1 con una SNR de 20 dB en las muestras de la función y 15 instantes de muestreo en cada dirección del espacio de entrada	85
3.18	Reconstrucción de las derivadas direccionales y de las derivadas cruzadas empleando los modelos MSR y MLRs, con 15 instantes de muestreo en cada dirección del espacio de entrada	87
3.19	Reconstrucción de las derivadas direccionales y de las derivadas cruzadas empleando los modelos MSR y MLRs, con 5 instantes de muestreo en cada dirección del espacio de entrada	88
4.1	Hiperplanos bisagra del modelo MLT	91
4.2	Filtro canónico lineal a tramos	92
4.3	Sección de una red canónica lineal a tramos multicapa	93
4.4	Derivadas de las funciones $f_{\text{tan}}(x, \gamma)$ y $f_{\text{pol}}(x, \gamma)$	96
4.5	Derivadas de las funciones $l_{\text{ch}}(x, \gamma)$ y $h_{\text{yp}}(x, \gamma)$	98
4.6	Función $l_{\text{ch}}(x, \gamma)$	98
4.7	Función base para un espacio de entrada bidimensional	99
4.8	Estabilizador	101
4.9	Ejemplo de región de integración en dominios bidimensionales	102
4.10	Ejemplo de función de prueba	110

4.11	Error en la reconstrucción frente al valor inicial de γ . El parámetro de suavidad se entrena para minimizar el error cuadrático de la función (línea de puntos), de la derivada primera (línea discontinua) y de la derivada segunda (línea continua). Todas las funciones base tienen un γ común	111
4.12	Resultados para un γ único sobre el 10 % de los mejores resultados. El parámetro de suavidad se entrena para minimizar el error cuadrático de la función (línea de puntos), de la derivada primera (línea discontinua) y de la derivada segunda (línea continua)	112
4.13	Resultados empleando un γ_i diferente para cada función base, promediando el 5 % mejor de las simulaciones. El parámetro de suavidad se entrena para minimizar el error cuadrático de la función (línea de puntos), de la derivada primera (línea discontinua) y de la derivada segunda (línea continua)	112
4.14	Resultados empleando un γ_i diferente para cada función base, promediando el 50% mejor de las simulaciones. El parámetro de suavidad se entrena para minimizar el error cuadrático de la función (línea de puntos), de la derivada primera (línea discontinua) y de la derivada segunda (línea continua)	113
4.15	Comparación de resultados entre una red MLP de 4 neuronas y una red MLTS de 5, ambas con 13 parámetros. Se presenta la reconstrucción de la función (línea continua) y de la derivadas primera (línea discontinua) y segunda (línea de puntos).	114
4.16	Comparación de resultados entre una red MLP de 10 neuronas y una red MLTS de 15 (31 y 33 parámetros respectivamente. Se presentan los resultados de reconstrucción de la función (línea continua) y de la derivada segunda (línea de puntos).	115
4.17	Sensibilidad a los mínimos locales de diferentes métodos	119
5.1	Resultados en la reconstrucción de la Función 1 con una red MLTS de 32 neuronas.	127
5.2	Arquitectura modular para la reconstrucción de una función utilizando la información de las derivadas	131
5.3	Arquitectura modular para la reconstrucción global de la función y local de las derivadas	132
5.4	Medidas normalizadas de G_{md} en función del punto de polarización (V_{gs0}, V_{ds0}) para un transistor de microondas NE72084 MESFET.	136
5.5	Modelo Global para el modelado de transistores de microondas	136
5.6	Evolución del error cuadrático medio MSE en un ejemplo 1D	144
5.7	Evolución del error cuadrático medio MSE en un ejemplo 2D	145
5.8	Funciones de pertenencia de los módulos gran señal y pequeña señal	147
5.9	Característica I/V para un punto de polarización $V_{ds0} = 3.5V$ y $V_{gs0} = -1V$	147

5.10	Parámetro G_m original y las aproximaciones obtenidas con el modelo gran señal y con el modelo modular global	149
6.1	Medidas experimentales y resultados obtenidos con el modelo MLTS en un punto de polarización $V_{gs0} = 0V$ y $V_{ds0} = 2V$, con una potencia óptica incidente $P_o = 1mW$	155
6.2	Circuito equivalente de un transistor FET	158
6.3	Parámetros G_m y G_{m3} (normalizados) típicos para transistores MESFET y HEMT	159
6.4	Potencia de salida en función de la carga para un MESFET NE72084 .	163
6.5	Contornos de la relación C/I en función de la carga para un MESFET NE72084	164
6.6	Característica I/V para un punto de polarización $V_{ds0} = 4V$ y $V_{gs0} = -1V$	166
6.7	Productos de intermodulación en la característica I/V del modelo global	167
6.8	Característica I/V , y sus derivadas, para un punto de polarización $V_{ds0} = 4V$ y $V_{gs0} = -1V$, modelada mediante el modelo MLR	168
C.1	Representación mediante filtros MIMO del modelo MLI de grado 3 . .	187
C.2	Respuestas impulsiva y frecuencial del filtro reconstructor $h_2(x)$	188
C.3	Respuestas impulsiva y frecuencial del filtro reconstructor $h_3(x)$	192
E.1	Cuadrícula básica del espacio de entrada	199
F.1	Funciones de dos variables utilizadas en la comparación de resultados .	205
F.2	Funciones de dos variables utilizadas en la comparación de resultados .	206

Índice Alfabético

- A posteriori
 - probabilidades, 139, 142
- A priori, 1, 4, 6, 41, 123
- Aereo
 - control de tráfico , 5
- Amplificadores, 2, 134, 156, 157
 - de la Clase A, 162
- Análisis multiresolución, 23, 25
- Analíticas
 - funciones, 152, 157, 172
- Analítico
 - modelo, 131, 146, 150, 154, 167
- Analógicos, 57
- Aplicaciones, 9
- Aproximación, 1, 11, 169
 - capacidad de, 47
 - funcional, 2
- Aproximador
 - universal, 48
- Arquitectura, 133
 - de la red, 8, 43, 123, 128, 130
 - de mezcla de expertos, 139
 - específica, 123, 130, 171
 - general, 130
 - modular, 123, 130, 135, 148, 171
 - neuronal, 135, 148
- B-splines, 29, 30, 34
- Backpropagation, 44
- Banco
 - de filtros, 7, 25, 26, 38, 53, 55, 57
- Banda, 158
 - ancha, 2
 - ancho de, 77, 176
 - funciones limitadas en, 19, 32, 75, 109, 113
 - limitada en, 17
 - señales limitadas en, 6, 11, 21, 73
 - señales no limitadas en, 79
- Base
 - de Riezs, 6
 - de Riezs, 25
 - función, 23, 99, 111, 138
 - funciones, 31, 33, 46, 101, 113, 117, 146, 160
- Bayes, 142
- Bernouilli, 13
- Bisagra
 - hiperplanos, 90, 99
- Borrosa
 - lógica, 133, 172
- Campana, 30, 49
- Caótico
 - sistema , 5
- Capa, 44
 - de salida, 44, 46, 126, 138
 - oculta, 44, 47, 93, 125
- Capacidad de aproximación, 47
- Cardinal spline, 33
- Carga, 161
 - computacional, 7, 52, 63, 72, 84, 88, 92, 113, 115, 119, 145, 154, 171
 - valores de, 161
- Centroide, 45
- Centroides, 46, 116, 126, 138, 140, 160
- Cerebro, 43
- Circuito equivalente, 2, 157
- Clase A
 - amplificadores, 162
- Coefficientes, 25, 29, 104, 135, 158, 162, 165, 166

- B-spline, 30, 33
 - de las derivadas, 165
 - secuencia de, 22, 24, 26
- Combinador, 131, 132, 137, 146, 165
- Compacta
 - formulación, 91
- Compacto, 93
 - soporte, 25, 26, 30, 34
- Completo
 - conjunto, 137
- Computacional
 - ahorro, 106
- Comunicación
 - teoría de la, 12
- Comunicaciones, 134
 - aplicaciones de, 2
 - sistemas de, 152, 156
- Conclusiones, 9
- Condicionado, 41
- Condiciones, 96
- Conjunto
 - completo, 137
 - de entrenamiento, 153
 - de prueba, 145
 - incompleto, 137
- Consciencia, 43
- Consumo, 5
- Continua, 29, 65, 67, 70
- Continuidad, 37, 53, 137
 - condiciones de, 53, 58, 66
 - de las derivadas, 137
- Contornos, 162
 - integración de, 13
- Contribuciones, 7
- Convergencia, 9, 20, 21, 34, 131, 142, 144, 150, 172
 - uniforme, 18
 - velocidad de, 137, 139, 145
- Cosecante, 15
- Coste
 - computacional, 35, 54
 - función de, 8, 41, 45, 63, 91, 123–127, 139, 148, 171
- Cotangente, 15
- Crosmun, 2
- Cruzada
 - derivada, 68, 71, 72, 167
 - validación, 63
- Cruzadas
 - derivadas, 52, 64, 68, 71, 83, 84, 169
- Cuadrático
 - error, 42, 43, 46, 71, 91, 92, 100, 103, 111, 123, 124, 126, 139, 140
 - problema de optimización, 175
 - spline, 38, 67, 68, 70
- Cuadrado integrable, 11, 32
- Cúbico
 - spline, 29, 32, 36, 39, 42, 56, 73, 74, 154
- Curvatura, 29, 32, 97
 - radio de, 96, 97, 100
- Dambrine
 - técnica de, 160
- Derivabilidad, 94, 123
- Derivable, 89, 95, 99, 133, 134, 170
- Derivadas, 1, 3, 6, 12
 - ajuste de las, 97
 - aproximación de las, 117, 130
 - cruzadas, 52, 68, 71, 83, 84, 132, 169
 - direccionales, 51, 64, 70, 79
 - información de las, 124, 130, 148, 170, 171
- Desacoplo
 - variables de, 141, 142
- Desarrollo de Taylor, 3
- Diferenciable, 30
- Digital
 - filtrado, 32, 33
 - procesado, 22
- Digitales, 57
- Dinámica
 - característica, 134, 151, 152
- Dinámicas
 - tensiones, 151
- Dirac, 30

- Direccionales
 - derivadas, 51, 64
- Discontinua, 67
- Discreta, 22
- Drenador, 2, 133
- Economía, 5
- Elasticidades, 5
- EM, 9, 124, 131, 137, 139, 140, 142, 144, 146, 150, 165, 171
- Empresa, 5
- Entrenamiento, 43, 46, 49, 103, 108, 109, 116, 123, 126, 128, 138, 142
 - algoritmo de, 124, 130, 137
 - conjunto de, 127, 160, 162, 165, 171
 - conjuntos de, 153
 - de los modelos, 153
 - patrones de, 160
 - supervisado, 139
- Error, 21, 42, 45
 - cuadrático, 91, 100, 103, 111, 123, 124, 126, 139, 140
 - de reconstrucción, 81
 - función, 104
 - superficie de, 92, 111
- Escala
 - filtro de, 23
 - función de, 22, 25, 26, 28, 38
 - funciones de, 6, 21, 176
- Escalado
 - condición de, 24
- Estabilidad, 15
- Estabilizador, 41, 45, 61, 100, 102
- Expectation-Maximization, 9, 137, 139
- Filtrado inverso, 33
- Filtro
 - adaptativo, 92
 - digital, 92
- Filtros, 58
 - analógicos, 57
 - banco de, 25, 26, 38, 53, 55, 57
 - canónicos lineales a tramos, 92
 - de retardo fraccional, 15
 - digitales, 57
 - FIR, 40, 59, 169
 - ideales, 32
 - IIR, 40, 59
 - polifase, 57, 60
 - recursivos, 33
- FIR, 33, 40, 59, 169
- Fotodetectores, 152
- Fourier
 - transformada de, 25, 26, 47
 - transformada discreta de, 15
- Fraccional
 - filtros de retardo, 15
- Fronteras, 104, 106, 108
- Fuente, 2, 133
- Función de coste, 8
- Funciones analíticas, 3
- GaAs, 152
- García, 4, 158
- Gaussiana, 45
- Gaussianas
 - mezcla de, 139
- Gaussiano
 - modelo, 143
 - ruido, 140
 - ruido blanco, 60, 140
- Generadora, 24
 - función, 22, 24
- Gradiente, 63, 72, 91, 104–106, 125, 126, 139, 140, 143, 145, 153, 161
- Grado de libertad, 37
- Grados
 - de libertad, 29, 53, 90
 - de suavidad, 113
- Gran señal, 2, 133–135, 172
 - comportamiento, 148, 151, 153
 - comportamiento dinámico, 152
 - módulo, 146, 162
 - modelos de, 151
- GRBF, 8, 46, 116, 123–125, 130, 135, 144–146, 153, 156, 160, 165
- HEMT, 2, 9, 133, 150–152, 156, 169

- Hermite, 36, 40
 - interpolación de, 28
- Hessiano, 91, 104, 105, 126
- Hilbert, 6, 11, 19
 - espacio de, 22, 25
- Hiperplano
 - óptimo, 173
- Hiperplanos, 90, 94, 109

- IIR, 40, 59
- Iluminación, 152
 - óptica, 151
 - efecto de la, 172
- Imagen, 32
- Incompleto
 - conjunto, 137
- Impulsiva
 - respuesta, 57
- Infinitas
 - series, 19
- Instantáneas
 - tensiones, 151, 154
- Integración, 54
- Inteligencia artificial, 43
- Intermodulación, 2, 4, 134, 135, 156, 157, 160, 161, 171
 - análisis de, 165
 - distorsión de, 156
 - efecto de la, 172
 - productos de, 158, 166
- Interpolación, 1, 2, 6, 11, 20, 21, 28, 34, 51
 - condiciones de, 53, 61, 66
 - de Hermite, 28, 36
 - lineal, 58
 - local, 64
 - métodos de, 88
 - modelos de, 64, 70
 - regularizada, 41
 - semilocal, 64
- Interpolada
 - secuencia, 54, 56, 57, 61, 66, 69, 70
- Intervalo, 56

- Irregular
 - muestreo, 20, 53, 54
- Iteración, 21
- Iterativa, 20
- Iterativo, 91, 104, 138
- Ito, 49

- Jacobiano, 4

- Kernel, 14, 33, 173, 175
 - de reproducción, 25
- Knots, 29

- Lógica borrosa, 133, 172
- Líneas de investigación, 6, 9, 172
- Lagrange, 13, 18, 174, 175
- Lebesgue, 11
- Lineal
 - a tramos, 89, 93
 - combinación, 90, 103, 126
 - dependencia, 91
 - expansión, 22
 - frontera, 94
 - función de activación, 125
 - optimización, 138
 - sistema de ecuaciones, 62
 - transición, 133, 137
- Lineales
 - a tramos, 94, 157
 - particiones, 90
 - regiones, 90, 95, 103
 - secciones, 95
- Local, 7, 28, 51, 54, 56, 60, 64, 68, 70, 169
 - aproximación, 132
 - carácter, 139, 142, 172
 - comportamiento, 131, 150
 - entorno, 132
 - funciones de activación, 139
 - influencia, 111
 - módulo, 132
 - modelo, 130
 - reconstrucción, 59, 74, 170
 - reproducción, 171

- Lyapunov, 5
- Máximo, 138, 141
- Mínimo, 56, 61, 105
 - error cuadrático, 91
 - error cuadrático, 92
 - número de parámetros, 91
- Mínimos
 - cuadrados, 47, 104, 105, 126, 143
 - locales, 9, 92, 94, 110, 111, 114, 116, 117, 124, 131, 137, 139, 140, 145, 150, 171, 172
- Maldición de la dimensionalidad, 46
- MATLAB, 115
- Mente, 43
- MESFET, 2, 9, 133, 146, 150, 151, 156, 169
- Mezcladores, 2, 134, 156, 157
- microondas, 7
- MIMO, 57
- MLI, 7, 51, 52, 57, 58, 64, 68, 73, 75
- MLI1D, 65, 66, 68, 69
- MLP, 8, 44, 47, 93, 113, 116, 123–125, 130, 153, 157, 160
- MLR, 8, 51, 60, 64, 70, 74, 75, 83
- MLR1D, 70–72
- MLRs, 52, 70, 83
- MLT, 90, 153
- MLTS, 8, 89, 97, 99, 100, 103, 110, 113, 116, 123, 124, 126, 134, 146, 151, 153, 170
- Modelado, 1, 2, 11, 89, 113, 169
 - de dispositivos, 91, 94
 - de transistores, 133
 - no lineal, 133
 - problema de, 102
 - problemas de, 97, 124
- Modelo
 - global, 9, 162, 165
 - gran señal, 166, 172
 - pequeña señal, 172
- Modular, 8, 132
 - arquitectura, 123
 - red, 148
- Monótona, 94, 100, 103
- Monótono, 101
- MRA, 23
- MSI, 52, 64, 70, 79
- MSR, 52, 70, 72, 79
- Muestreo
 - frecuencia de, 87, 170
 - instantes de, 19, 21, 53, 56, 65, 67, 83, 167
 - irregular, 7, 17, 19, 20, 53, 54
 - multibanda, 25
 - no uniforme periódico, 25
 - período de, 53, 58, 65, 75, 77, 109
 - regular, 7, 55
 - teorema de, 7, 11, 13, 79
 - teoremas de, 55
 - tiempo de, 55
 - uniforme, 18, 53
 - velocidad de, 77, 81
- Multibanda
 - muestreo, 25
- Multiresolución, 21–25, 37
 - espacio, 28
 - espacios, 55, 176
- Neurona, 43–45, 47, 139
- Neuronas, 113, 116, 125–128, 140, 141, 145, 146, 160
- Norma
 - L_2 , 18, 20, 24
 - L_p , 34
 - ponderada, 46
- Nyquist, 14
 - frecuencia de, 12, 73–75
- Objetivos, 1
- Ocultas
 - variables, 137–139, 141, 142
- OLS, 46, 116, 126, 143, 145, 153, 160
- Óptica
 - energía, 152
 - fibra, 152
 - iluminación, 151–153, 172

- potencia, 152, 154
- sistemas de transmisión, 152
- Ortogonal
 - base, 24
 - proyección, 20, 21
- Oscilaciones, 29, 96, 97, 137
 - de Gibbs, 32
- Osculatoria, 28
- Parsimonioso, 6, 14, 43, 89, 113, 134, 167, 170
- Partición, 90, 91, 94, 106
 - de la unidad, 19, 22, 24
- Pedro, 3
- Pequeña señal
 - módulo, 146
- Pequeña Señal, 2
- Pequeña señal, 134, 135, 156, 172
 - ciercuito equivalente, 152
 - comportamiento, 148, 150, 165
 - módulo, 146, 165
 - modelo, 156
 - modelos de, 151
- Periódica, 15
 - secuencia discreta, 14
- Pertenencia
 - funciones de, 137, 146, 165, 172
- Perturbaciones, 19
- Planteamiento
 - y motivación del problema, 1
 - y objetivos de la Tesis, 6
- Polarización
 - corriente de, 135
 - punto de, 3, 8, 134, 135, 146, 152, 157, 165
 - tensiones de, 134, 151, 153, 158
- Polifase
 - componente, 57
 - filtros, 57, 60
 - matrices, 26
 - matriz, 38, 39
- Polinomios, 28
- Ponderación
 - parámetros de, 127
- Potencias de intermodulación, 3
- Procesado de señal, 7, 12, 32, 34, 79
- Producción
 - funciones de , 5
- Producto
 - de tensores, 35, 70
 - escalar, 99
- Proyección, 52
 - ortogonal, 20, 21
- Proyecciones, 19
- Publicaciones, 172
- Puerta, 2
- Pulsada, 152, 167
 - característica, 167
 - tensiones de, 134
 - tensiones dinámicas de, 153
- Pulso, 24, 30
- RBF, 8, 44, 45, 47, 138, 140, 142, 160
- Reconstrucción, 2, 169
 - de la función, 68
 - dirección de, 69
 - global, 132, 171
 - localizada, 123, 131
 - procedimiento de, 68
 - proceso de, 150
 - unidimensional, 66
- Red neuronal, 99, 131
 - canónica lineal a tramos, 93
- Redes
 - multicapa, 49
 - neuronales, 5–8, 42, 47, 123, 134, 140, 157, 160
- Regular
 - muestreo, 55
- Regularización, 8, 41, 51, 74, 169
 - de las secuencias, 80
 - estricta, 71
 - funcional de, 61, 63, 70, 71, 87
 - parámetro de, 41, 42, 90, 100, 103, 170
 - parámetros de, 63, 71, 72, 77, 170

- simple, 84
- simplificada, 72
- técnica de, 107
- término de, 62, 71, 102
- teoría de la, 100
- Regularizada
 - aproximación, 6
- Regularizado
 - funcional, 70, 100, 103
 - valor, 70
- Rejilla, 153, 160, 162
- Relajación
 - parametro de, 20
- Residuo, 139, 141–143
- Respuesta
 - en frecuencia, 58
 - impulsiva, 57
- Restricciones, 1, 6, 11, 20, 29, 36, 41, 47, 95
 - de suavidad, 103
 - en las derivadas, 1, 169
 - entre los parámetros, 91
 - minimización con, 173
- Retardo fraccional, 15
- Retropropagación, 44, 50, 93, 153, 160
- Revisión, 11
- Riezs, 6
 - base de, 22, 24, 25
- Robótica, 4
- Root, 3
- Rubin, 137
- Ruido, 41, 51, 53, 73, 74
 - blanco gaussiano, 60, 140
 - inmunidad al, 50
 - nivel de, 124, 160
 - potencia de, 60
 - potencia de, 63, 72
 - relación señal a, 73
 - sensibilidad al, 60, 61, 75, 77, 81, 84, 169
 - señales superpuestas en, 139
- Ruptura
 - instantes de, 56
- punto de, 94, 95
- puntos de, 29, 35, 53, 55
- Síntesis
 - funciones de, 25, 28, 39, 40
- Saturación, 158
 - región de, 2, 157, 158
- Secuencia
 - interpolada, 61
 - regularizada, 61, 63
- Semilocal, 52, 64, 68, 70
- Sensibilidad
 - al ruido, 51
- SER, 73, 75, 79, 109, 127, 145, 153
- Sigmoidal, 48, 49, 94
- Sigmoidales, 142
- Simultánea
 - aproximación, 6, 123
 - interpolación, 51, 169
 - reconstrucción, 51
- Sistemas
 - de comunicaciones, 152, 156
 - teoría de, 18
- SNR, 73, 75
- Sobrentrenamiento, 153
- Sobremuestreo, 25, 74, 77, 79
- Soporte
 - finito, 101
 - maquinas de vectores, 173
- Spline, 32
- Splines, 7, 28
 - espacio de, 53, 56, 63
 - espacio de funciones, 51, 53
- Suave, 89
- Suavidad, 8, 29, 32, 36, 41, 42, 51, 55, 90, 95, 103, 176
 - análisis de, 100, 170
 - condiciones de, 60
 - influencia local, 111
 - parámetro de, 97, 100, 103, 106, 109, 113, 116, 170
 - parámetros de, 108
- Suavizado, 94, 99

- función de, 100
- modelo, 95
- modelo lineal a tramos, 97, 99
- SVM, 173
- Taylor
 - desarrollo en serie, 3, 28, 131, 135, 157, 158
- Tensores
 - producto de, 35, 70
- Teoría
 - de la regularización, 41, 100
 - de Shannon, 32
 - de sistemas, 18
 - estadística del aprendizaje, 173
- Teorema de muestreo, 7, 11
- Thin plate splines, 42
- Tikhonov, 41
- Tonos, 3, 158, 160, 161
- Tráfico aéreo, 5
- Tramos
 - constantes a, 33
 - de polinomios, 29
 - funciones polinómicas a, 29
 - lineal a, 52–54, 58, 67, 89
 - lineales a, 157
 - modelo lineal a, 90
- Transconductancia, 158
- Transformada
 - de Fourier, 25, 26, 47
 - wavelet, 23
 - z, 26, 33
- Transistor, 134, 153
- Transistores, 9, 91, 123, 130, 133, 146, 150, 151, 169
- Triangular
 - función de escala, 24
 - respuesta, 58
- Trigonométrica, 16
- Umbral lógico, 93, 94
- Unidad
 - partición de la, 22, 24
- Uniforme, 65, 68
- convergencia, 18
- distribución, 73, 75
- muestreo, 53
- rejilla, 127
- Universal
 - aproximador, 93, 97
 - capacidad de aproximación, 47
 - representante, 93
- Valor absoluto, 90, 94, 96
- Varian, 5
- Varianza, 160
 - de ruido, 60
- Varianzas, 46, 116, 125, 138, 140, 145
- Verosimilitud, 137–141
- Volterra, 92, 157
- Wavelet, 23
- Z
 - transformada, 26, 33