

UNIVERSIDAD DE CANTABRIA



**DEPARTAMENTO DE INGENIERÍA DE
COMUNICACIONES**

TESIS DOCTORAL

**ESTIMACIÓN ÓPTIMA DE SECUENCIAS
CAÓTICAS CON APLICACIÓN EN
COMUNICACIONES**

**Autor : David Luengo García
Directores : Carlos Pantaleón Prieto
Ignacio Santamaría Caballero**

Grupo de Tratamiento Avanzado de Señal

Septiembre de 2006

Capítulo 6

Estimación de Mapas Caóticos

6.1. Introducción

En este capítulo se considera el problema complementario al de los capítulos anteriores: obtener buenos estimadores de mapas caóticos a partir de las observaciones ruidosas. En realidad, este problema genérico engloba otros dos problemas más específicos: la estimación de los parámetros del mapa que ha generado una secuencia dada cuando su forma general es conocida (*inferencia paramétrica*), y la obtención de un mapa que represente de manera adecuada a un conjunto de observaciones disponibles generadas por un sistema caótico desconocido (*modelado*).

El único problema analizado en este capítulo es el primero. En primer lugar, en la Sección 6.2 se plantea el problema desde el punto de vista de los dos marcos de estimación óptima considerados: ML y Bayesiano. A continuación, en la Sección 6.3 se presenta una implementación aproximada del estimador ML para mapas unimodales y uniparamétricos con dependencia lineal de su único parámetro basada en el mismo principio que el estimador ML de la secuencia: división del espacio de parámetros en regiones en función del itinerario, búsqueda de estimadores ML locales, y obtención del estimador ML global como el mejor de todos los locales.

La complejidad del estimador ML de los parámetros es muy superior a la del estimador ML de la secuencia, incluso para los mapas PWL más simples. Además, no existe ninguna FDP a priori natural que se pueda usar para plantear estimadores Bayesianos. En consecuencia, el resto del capítulo considera una función de coste simplificada que sólo tiene en cuenta el error cometido entre cada observación y la predicción realizada usando la observación anterior. Para minimizar dicha función de coste, en la Sección 6.4 se recurre a diversos estimadores bloque (método de los momentos, mínimos cuadrados y mínimos cuadrados totales), mientras que en la Sección 6.5 se propone un algoritmo novedoso de gradiente estocástico basado en la competición entre las distintas regiones del mapa caótico por modelar las observaciones. Finalmente, el capítulo se cierra con la presentación y análisis de los resultados más relevantes en la Sección 6.6, y la discusión de las distintas alternativas de estimación en la Sección 6.7.

6.2. Estimación de Parámetros de Mapas Caóticos

6.2.1. Modelo Matemático

El modelo matemático utilizado es el mismo que el de los capítulos anteriores. La ecuación de estado del sistema es

$$x[k] = f(x[k-1]; \boldsymbol{\theta}), \quad (6.1)$$

para un cierto valor de $x[0]$ prefijado y con $1 \leq k \leq N$. La única diferencia entre (6.1) y (3.1) es la presencia en (6.1) de $\boldsymbol{\theta} = [\theta_1, \dots, \theta_p]^T$, el vector de parámetros que caracteriza al mapa caótico, de manera explícita. En cuanto a la ecuación de medida, es exactamente la misma que (3.2):

$$y[k] = x[k] + w[k], \quad k = 0, \dots, N. \quad (6.2)$$

Este capítulo se va a centrar en el estudio de aquellas funciones que exhiben una dependencia lineal con sus parámetros:

$$f(x; \boldsymbol{\theta}) = \sum_{i=1}^p \theta_i g_i(x) + g_0(x) = [1 \ \boldsymbol{\theta}^T] \mathbf{g}(x) = \mathbf{g}^T(x) [1 \ \boldsymbol{\theta}^T]^T, \quad (6.3)$$

donde $\mathbf{g}(x) = [g_0(x), g_1(x), \dots, g_p(x)]^T$ es el conjunto de funciones que definen la forma del mapa, de las cuales al menos una debe ser no lineal y no invertible.

Nótese que, aunque la restricción de dependencia lineal con los parámetros puede parecer excesivamente severa, existen muchos mapas caóticos unidimensionales que la satisfacen: el TM, el S-TM, el mapa de Markov de la Figura 2.4, los mapas del seno y logístico, etc. Por el contrario, ejemplos importantes de mapas con dependencia no lineal en sus parámetros son el SK-TM y el BSK-TM (debido a que el parámetro afecta al límite entre sus dos regiones lineales), el mapa de Markov de la Figura 2.5(a), y el mapa exponencial.

Dentro de la clase de mapas con dependencia lineal en sus parámetros se encuentran incluidos los mapas PWL, siempre que los límites de sus intervalos no pertenezcan al conjunto de parámetros a estimar. Para un mapa PWL genérico con M regiones prefijadas su vector de parámetros tiene dimensión $2M$: $\boldsymbol{\theta} = [\mathbf{a}^T \ \mathbf{b}^T]^T$, con $\mathbf{a} = [a_1, \dots, a_M]^T$ y $\mathbf{b} = [b_1, \dots, b_M]^T$. En este caso, la estructura de las funciones que definen la dependencia del mapa con cada parámetro es muy sencilla:

$$\mathbf{g}(x) = [x \cdot \boldsymbol{\chi}^T \ \boldsymbol{\chi}^T]^T,$$

siendo $\boldsymbol{\chi}$ simplemente un vector de longitud M que indica a que región pertenece x ,

$$\boldsymbol{\chi} = [\chi_{E_1}(x), \dots, \chi_{E_M}(x)]^T,$$

donde $\chi_{E_i}(x)$ es la función característica del mapa, definida por (2.12).

6.2.2. Estimadores ML y Bayesianos

En esta sección se desarrolla el estimador ML para un mapa arbitrario con dependencia lineal en sus parámetros. En primer lugar, la FDP de las observaciones para este problema es exactamente la misma que la mostrada en la Sección 3.2.2: una Gaussiana multidimensional. En consecuencia, el estimador ML se obtiene minimizando una función de coste idéntica a (3.8),

$$J(x[n], \mathbf{s}_{0:n-1}; \boldsymbol{\theta}) = \sum_{k=0}^{n-1} \left(y[k] - f_{\mathbf{s}_{k:n-1}}^{-(n-k)}(x[n]; \boldsymbol{\theta}) \right)^2 + \sum_{k=n}^N \left(y[k] - f^{k-n}(x[n]; \boldsymbol{\theta}) \right)^2, \quad (6.4)$$

con respecto a $\boldsymbol{\theta}$ en esta ocasión. Derivando (6.4) con respecto a cada uno de los p parámetros e igualando a cero se obtiene un sistema de p ecuaciones con p incógnitas similares a (3.10):

$$\sum_{k=1}^N y[k] \frac{\partial f_{\hat{\mathbf{s}}_{\text{ML}}}^{k-n}(x[n]; \boldsymbol{\theta})}{\partial \theta_i} = \sum_{k=1}^N f_{\hat{\mathbf{s}}_{\text{ML}}}^{k-n}(x[n]; \boldsymbol{\theta}) \frac{\partial f_{\hat{\mathbf{s}}_{\text{ML}}}^{k-n}(x[n]; \boldsymbol{\theta})}{\partial \theta_i}, \quad (6.5)$$

donde $\hat{\mathbf{s}}_{\text{ML}}$ denota la estima ML de los n primeros símbolos, e $i = 1, \dots, p$.

Nótese en primer lugar que en este caso siempre va a ser necesario disponer de una muestra de referencia conocida, $x[n]$, o estimarla previamente. Además, cuando dicha muestra de referencia no es $x[0]$, resulta imprescindible estimar también las n primeras muestras de la secuencia simbólica, así como generar n muestras de la secuencia iterando hacia atrás, para las que en general el mapa inverso no va a presentar una dependencia lineal con sus parámetros.

Por lo tanto, en este capítulo se va a considerar únicamente el caso en el que la muestra de referencia es $x[0]$, de modo que (6.5) se convierte en

$$\sum_{k=1}^N y[k] \frac{\partial f^k(x[0]; \boldsymbol{\theta})}{\partial \theta_i} = \sum_{k=1}^N f^k(x[0]; \boldsymbol{\theta}) \frac{\partial f^k(x[0]; \boldsymbol{\theta})}{\partial \theta_i}, \quad (6.6)$$

en la que ha desaparecido la dependencia con la secuencia simbólica. Aunque esta expresión es muy similar a (3.10), su resolución resulta mucho más complicada, no pudiéndose obtener una fórmula cerrada ni tan siquiera para los mapas PWL más simples. Como prueba de la complejidad de (6.6), en el Apéndice F se demuestra que la derivada que aparece en la misma se puede expresar como:

$$\begin{aligned} \frac{\partial f^k(x[0]; \boldsymbol{\theta})}{\partial \theta_i} &= g_i(f^{k-1}(x[0]; \boldsymbol{\theta})) + \boldsymbol{\theta}^T \dot{\mathbf{g}}(f^{k-1}(x[0]; \boldsymbol{\theta})) \frac{\partial f^{k-1}(x[0]; \boldsymbol{\theta})}{\partial \theta_i} \\ &= g_i(x[k-1]) + \sum_{m=1}^{k-1} g_i(x[k-m-1]) \prod_{l=1}^m \boldsymbol{\theta}^T \dot{\mathbf{g}}(x[k-l]). \end{aligned} \quad (6.7)$$

Analizando (6.7) resulta evidente que se trata de un polinomio de orden $k-1$ en sus parámetros con términos cruzados en los que van a aparecer en general los p

parámetros a estimar. Este carácter polinómico de la derivada resulta más aparente cuando el mapa depende de un único parámetro, en cuyo caso

$$\frac{\partial f^k(x[0]; \theta)}{\partial \theta} = \sum_{m=0}^{k-1} c_{m,k}(\mathbf{x}) \theta^m, \quad (6.8)$$

donde

$$c_{m,k}(\mathbf{x}) = \begin{cases} g(x[k-1]), & m = 0; \\ g(x[k-m-1]) \prod_{l=1}^m \dot{g}(x[k-l]), & 1 \leq m \leq k-1. \end{cases} \quad (6.9)$$

Nótese que la dependencia polinómica de (6.7) y (6.8) es un tanto artificiosa, ya que ambas dependen de la secuencia $\mathbf{x}_{1:k-1} = [x[1], \dots, x[k-1]]$, que a su vez depende de $\boldsymbol{\theta}$ de una forma no necesariamente polinómica. No obstante, estas dos ecuaciones resultan útiles en el Apéndice F para calcular el CRLB, y permiten ilustrar la dificultad de hallar el estimador ML de forma cerrada.

De hecho, la iteración k -ésima del mapa, $f^k(x[0]; \boldsymbol{\theta})$, que aparece tanto en (6.6) como en las ecuaciones de la derivada, presenta una expresión complicada incluso para los mapas más sencillos. Por ejemplo, para el TM y el S-TM f^k es un polinomio de grado k en β , para el mapa logístico también se tiene un polinomio, aunque de grado $2^k - 1$ en este caso, y para un mapa PWL genérico, en el que se deben estimar sus a_i y b_i , f^k es igualmente un polinomio de grado k en el que aparecen términos cruzados con productos de a_i y b_j ($1 \leq i, j \leq M$). Para otras funciones, como el mapa del seno, la dependencia de f^k con su parámetro resulta mucho más complicada aún.

Obviamente, una ecuación que incluya funciones de este tipo no va a resultar sencilla de minimizar para obtener el estimador ML. Así pues, en esta ocasión no parece que exista una clase amplia de mapas útiles y sencillos de estimar, como ocurría en el caso de la estimación de la secuencia con los mapas PWL. Estas dificultades motivan la utilización de algoritmos subóptimos en general, salvo en casos muy concretos, como el TM, el S-TM, y otros mapas unimodales simples con dependencia lineal en su parámetro de bifurcación, para los que se propone un método que permite obtener el estimador ML de manera aproximada en la Sección 6.3.

Por último, en relación con los estimadores Bayesianos, a la hora de plantearlos se debe proponer una FDP a priori para los parámetros que se desean estimar que condense el conocimiento disponible acerca de los mismos antes de llevar a cabo ningún experimento. Aunque en principio el marco de estimación Bayesiana admite la utilización de cualquier FDP a priori, resulta conveniente elegir una FDP ajustada a la naturaleza de los datos por dos motivos [Box1973]:

1. La elección de una FDP a priori muy distinta de la FDP real de los datos puede oscurecer la información contenida en los mismos, dando lugar a conclusiones erróneas respecto a los resultados obtenidos en la estimación.
2. El criterio de rendimiento utilizado habitualmente, y en esta Tesis en particular, es el de minimizar el MSE de los estimadores, o equivalentemente su varianza, ya

que se buscan estimadores insesgados (o al menos asintóticamente insesgados). En este sentido, los estimadores Bayesianos únicamente van a resultar ventajosos frente al ML, proporcionando un menor MSE, cuando su FDP a priori esté bien ajustada a la FDP real de los datos.

En el caso de la estimación Bayesiana de la secuencia caótica, discutido en el Capítulo 4, existe una FDP a priori natural de los datos: la FDP invariante natural asociada al mapa caótico. Sin embargo, para el problema de la estimación de los parámetros de un mapa caótico no existe ninguna FDP a priori natural. En estos casos se pueden plantear estimadores Bayesianos utilizando FDPs a priori no informativas, como por ejemplo una FDP uniforme dentro del rango conocido de los parámetros, o muy poco informativas [Box1973]. No obstante, esta aproximación presumiblemente no va a conseguir mejorar en gran medida el rendimiento del estimador ML, mientras que su coste computacional sí que puede incrementarse sensiblemente. En consecuencia, en este capítulo no se considera el desarrollo de estimadores Bayesianos.

6.3. Estimador ML del Parámetro de Bifurcación de Mapas Unimodales

En esta sección se va a considerar la estimación ML de mapas PWL unimodales cuya forma es conocida, con una partición natural $\mathcal{P} = \{[e_0, e_1], [e_1, e_2]\}$, y caracterizados por un único parámetro, θ , con el que la dependencia es lineal. El método propuesto requiere que la dependencia de f^k con respecto a θ dado un itinerario sea polinómica para $1 \leq k \leq N$, de modo que podría extenderse en general a mapas de la forma

$$f(x) = \theta \sum_{i=1}^M f_i(x) \chi_{E_i}(x) + g_0(x), \quad (6.10)$$

con

$$f_i(x) = \sum_{k=0}^{Q_i} d_i[k] x^k,$$

donde M es el número de regiones de la partición natural del mapa (que puede ser mayor que dos), $f_i(x)$ ($1 \leq i \leq M$) es su forma en cada intervalo, un polinomio de grado Q_i , $d_i[k]$ son los coeficientes de dicho polinomio, y $g_0(x)$ es la función que engloba todos aquellos términos que no dependen de θ . Resulta evidente que (6.10) es un caso particular de (6.3) en el que $p = 1$ y $g_1(x)$ es una función PWL:

$$g_1(x) = \sum_{i=1}^M f_i(x) \chi_{E_i}(x) = \sum_{i=1}^M \chi_{E_i}(x) \sum_{k=0}^{Q_i} d_i[k] x^k.$$

Ejemplos de mapas pertenecientes a esta clase son todos los PWL con un único parámetro de bifurcación que no afecta a los límites entre regiones, como el TM o el S-TM

($M = 2, Q_1 = Q_2 = 1$), el mapa logístico ($M = 1, Q_1 = 2$), las distintas versiones propuestas de mapas cuadráticos y cúbicos mostradas en la Tabla 2.2 ($M = 1, Q_1 = 2$ ó 3), o el mapa de Singer ($M = 1, Q_1 = 4$).

En primer lugar, en la Sección 6.3.1.1 se muestra como llevar a cabo una partición del espacio de parámetros en función del itinerario de manera análoga a la partición del espacio de fases realizada en la Sección 3.4.3. Aunque el algoritmo exacto desarrollado para realizar la partición es válido para cualquier mapa con dependencia lineal de θ y polinómica con x , requiere calcular los ceros de $f_s^k(x[0]; \theta)$ en el espacio de parámetros para $k = 1, \dots, N$. En consecuencia, es necesario disponer de una expresión cerrada para $f_s^k(x[0]; \theta)$, algo que actualmente sólo es posible para mapas PWL como el TM o el S-TM. Sin embargo, efectuando la suposición (comprobada experimentalmente) de que $f_s^k(x[0]; \theta)$ presenta un sólo cruce por el umbral entre las dos regiones de la partición, e_1 , dentro de cada región de la partición obtenida a partir de $f_s^{k-1}(x[0]; \theta)$, en la Sección 6.3.1.2 se presenta un método eficiente basado en técnicas del tipo descenso de gradiente, que podría permitir hallar la partición para mapas no PWL.

A continuación, en la Sección 6.3.2 se muestra un algoritmo que permite encontrar el estimador ML de θ de manera aproximada. Su funcionamiento es similar al del estimador ML de $x[0]$: se obtiene una estima ML local para cada región de la partición del espacio de parámetros, y se escoge como estima ML global la mejor de todas las estimas ML locales. La principal dificultad en este caso radica en que no se dispone de una fórmula cerrada para las estimas ML locales, de modo que se obtienen mediante una búsqueda de rejilla o simplemente eligiendo el punto medio de la región (aprovechando el hecho de que estas regiones son cada vez más pequeñas conforme N aumenta). Por último, en esta sección también se propone una aproximación del tipo HC-ML consistente en utilizar únicamente el itinerario obtenido a partir de las medidas ruidosas para evitar el crecimiento exponencial con N del coste computacional del algoritmo ML aproximado.

6.3.1. Partición del Espacio de Parámetros en Función del Itinerario

En la Sección 3.4.3 se vio cómo, dada una muestra de referencia, $x[n]$, el espacio de fases de un mapa caótico se puede dividir en regiones en función de su itinerario hacia delante y hacia atrás. La iteración hacia delante produce regiones disjuntas cuya unión comprende todo el espacio de fases (esto es, da lugar a una partición en el sentido estricto), mientras que la iteración hacia atrás produce regiones cuya intersección en general es no nula. Si se elige como muestra de referencia $x[0]$, entonces toda la secuencia caótica se genera mediante iteración hacia delante, y la región de posibles valores de $x[0]$ asociada a un itinerario dado, $\mathbf{s} = [s[0], \dots, s[N-1]]^T$, resulta

$$\begin{aligned} R_s^N &= \{x[0] : f^k(x[0]) \in E_{s[k]}, 0 \leq k \leq N-1\} \\ &= \{x[0] : x[0] \in f_{s[0]}^{-k}, \dots, s_{[k-1]}(E_{s[k]}), 0 \leq k \leq N-1\} = [\eta_s^N, \kappa_s^N]. \end{aligned} \quad (6.11)$$

Para un valor de N finito esta región se va reduciendo conforme aumenta N , dado que $R_{\mathbf{s},s[N]}^{N+1} \subseteq R_{\mathbf{s}}^N$, y, si la división en regiones del mapa original constituye una partición generadora, en el límite cuando N tiende a infinito $R_{\mathbf{s}}^N$ contiene un único punto del espacio de fases [Beck1993]. Este hecho permite estimar cada vez mejor $x[0]$ conforme N aumenta (al menos cuando la SNR tiende a infinito), y es la base de los estimadores óptimos y subóptimos planteados en los capítulos anteriores.

Esta idea es la que se va a extender en esta sección al *espacio de parámetros*, utilizándose como base para el planteamiento de buenas aproximaciones al estimador óptimo en secciones posteriores. Es decir, del mismo modo que θ y \mathbf{s} definen una región en el espacio de fases, $R_{\mathbf{s}}^N$, $x[0]$ y \mathbf{s} permiten determinar una región en el espacio de parámetros, $Q_{\mathbf{s}}^N$, que se define de un modo similar a (6.11):

$$\begin{aligned} Q_{\mathbf{s}}^N &= \{\theta : f^k(x[0]; \theta) \in E_{s[k]}, 1 \leq k \leq N-1\} \\ &= \{\theta : x[0] \in f_{s[0], \dots, s[k-1]}^{-k}(E_{s[k]}; \theta), 1 \leq k \leq N-1\} = [\vartheta_{\mathbf{s}}^N, \varphi_{\mathbf{s}}^N]. \end{aligned} \quad (6.12)$$

Nótese que de nuevo la unión de todos los posibles conjuntos $Q_{\mathbf{s}}^N$ conforma una partición, en esta ocasión del espacio de parámetros, ya que para un cierto valor de $x[0]$ cada valor de θ da lugar a un único itinerario de longitud N . Y, una vez más, $Q_{\mathbf{s},s[N]}^{N+1} \subseteq Q_{\mathbf{s}}^N$, de modo que el tamaño de $Q_{\mathbf{s}}^N$ decrece conforme N aumenta, indicando que en teoría es posible obtener una estima arbitrariamente precisa de θ al incrementar N , al menos asintóticamente cuando la SNR tiende a infinito.

6.3.1.1. Partición Exacta del Espacio de Parámetros

Obtener una partición exacta en función del itinerario en el espacio de parámetros resulta mucho más costoso que en el espacio de fases. No obstante, la tarea de calcular los límites de dichas regiones en función de \mathbf{s} es posible gracias al siguiente lema, que es una extensión del presentado en [Panta2000c]:

Lema 6.1 *Sea una condición inicial, $x[0]$, y sea un mapa caótico $f(x; \theta)$ unimodal, continuo tanto en x como en θ , y con una partición natural $\mathcal{P} = \{[e_0, e_1], [e_1, e_2]\}$. Los elementos del conjunto de parámetros formado por todos los ceros de $f^k - e_1$, $\Xi_N = \{\theta : f^k(x[0]; \theta) - e_1 = 0, 1 \leq k \leq N-1\}$, dividen el espacio de parámetros, $\Theta = [\theta_{\min}, \theta_{\max}]$, en un número finito de intervalos (regiones) de tal modo que todas las señales generadas iterando f a partir de $x[0]$ usando un valor de θ cualquiera dentro de un mismo intervalo comparten la misma secuencia simbólica de longitud N .*

Demostración 6.1 *Este lema se demuestra por reducción al absurdo. Considérense dos parámetros distintos, θ_1 y θ_2 , pertenecientes al mismo intervalo, y supóngase que sus itinerarios de longitud N para una misma condición inicial, $x[0]$, son distintos. En este caso, tiene que haber al menos un $k < N$ tal que $f^k(x[0]; \theta_1) \in E_1$ y $f^k(x[0]; \theta_2) \in E_2$ o viceversa. Y puesto que $f(x; \theta)$ es una función continua tanto en θ como en x , entonces $f^k(x; \theta)$ debe ser también continua. En consecuencia, por continuidad tiene que existir un θ^* entre θ_1 y θ_2 tal que $f^k(x[0]; \theta^*) = e_1$, lo que contradice el hecho de que θ_1 y θ_2 pertenezcan a la misma región. \square*

El Lema 6.1 permite garantizar que la región asociada a cada itinerario está compuesta por un número finito de intervalos y proporciona un mecanismo para hallarlos. En [Panta2000c] se propuso un método que permitía encontrar dichas regiones para el S-TM, y que se puede generalizar fácilmente a cualquier mapa con dependencia lineal con θ y polinómica con x , como se muestra en el Algoritmo 6.1. Nótese que en la demostración del Lema 6.1 únicamente se han utilizado la continuidad de f y el hecho de que sea unimodal. Por consiguiente, tanto el Lema 6.1 como el Algoritmo 6.1 son válidos para cualquier mapa unimodal continuo, no sólo para mapas con dependencia lineal con θ y polinómica con x . Sin embargo, su aplicación práctica requiere encontrar los ceros de $f_s^k(x[0]; \theta)$ para $1 \leq k \leq N - 1$, algo que resulta muy complicado para funciones arbitrarias. Así pues, su uso se va a restringir a mapas con dependencia lineal con θ y polinómica con x , en cuyo caso simplemente es necesario factorizar un polinomio en θ de orden creciente con k .

1. Inicialización del algoritmo. Se establecen los límites iniciales de las regiones a partir del rango admisible de parámetros, $\Xi_0 = \Theta = [\theta_{\min}, \theta_{\max}]$, y la matriz con los itinerarios iniciales se hace igual a la región a la que pertenece $x[0]$, $\mathbf{S}_0 = s[0]$.
2. Para $1 \leq k \leq N$ se buscan las raíces de $f_{\mathbf{S}_{0:k-1}}^k(x[0]; \theta) - e_1$ y se añaden o no a Ξ_k utilizando el siguiente algoritmo:
 - 2.1. Inicializar Ξ_k como el conjunto vacío: $\Xi_k = []$.
 - 2.2. Para $1 \leq i \leq L_{k-1} - 1$, siendo L_{k-1} la longitud de Ξ_{k-1} , repetir los pasos siguientes:
 - 2.2.1. Añadir el límite de la región i -ésima a la partición k -ésima: $\Xi_k = [\Xi_k, \Xi_{k-1}(i)]$.
 - 2.2.2. Construir el polinomio en θ correspondiente a $f_{\mathbf{S}_{k-1}(i,1), \dots, \mathbf{S}_{k-1}(i,k)}^k(x[0]; \theta) - e_1$, identificando los coeficientes para cada grado de θ .
 - 2.2.3. Obtener las raíces de $f_{\mathbf{S}_{k-1}(i,1), \dots, \mathbf{S}_{k-1}(i,k)}^k(x[0]; \theta) - e_1$ usando algún método estándar para factorizar polinomios.
 - 2.2.4. Seleccionar aquellas raíces que pertenezcan al intervalo $[\Xi_{k-1}(i), \Xi_{k-1}(i+1)]$ y añadirlas a Ξ_k en orden creciente.
 - 2.3. Añadir el límite superior de Ξ_{k-1} a Ξ_k : $\Xi_k = [\Xi_k, \Xi_{k-1}(L_{k-1})]$.
 - 2.4. Construir una nueva matriz con el itinerario correspondiente a cada región, \mathbf{S}_k , usando $x[0]$ como condición inicial y $\theta^{i,k} = (\Xi_k(i) + \Xi_k(i+1))/2$ como parámetro, de modo que $\mathbf{S}_k(i, j) = f^{j-1}(x[0]; \theta^{i,k})$, con $1 \leq i \leq L_k - 1$ y $1 \leq j \leq k + 1$.
3. La partición deseada viene dada por Ξ_N , y el conjunto de itinerarios por \mathbf{S}_N .

Algoritmo 6.1: Obtención de una partición exacta del espacio de parámetros en función del itinerario para un mapa unimodal continuo en θ y x , y con dependencia lineal de θ y polinómica con x .

Aunque el Algoritmo 6.1 es capaz de encontrar la partición exacta en el espacio de parámetros, su principal dificultad se halla en el punto 2.2.2: la necesidad de conocer

los coeficientes del polinomio en θ dado por $f_{s_{0,k-1}}^k(x[0]; \theta) - e_1$ para $k = 1, \dots, N$. Desafortunadamente, en la práctica estos coeficientes sólo se conocen para dos mapas unimodales de entre todos los considerados: el TM y el S-TM. La iteración k -ésima de ambos mapas viene dada por (3.15) con $n = 0$, siendo en los dos casos

$$A_s^{m,k} = \prod_{l=m}^{k-1} a_{s[l]} = \beta^{k-m} \prod_{l=m}^{k-1} (-1)^{s[l]-1} \quad (6.13)$$

para $0 \leq m \leq k-1$, ya que $a_1 = \beta$ y $a_2 = -\beta$ tanto para el TM como para el S-TM, y $A_s^{k,k} = 1$. Para el S-TM $b_1 = b_2 = \beta - 1$, de modo que, sustituyendo (6.13) en (3.15), $f_s^k(x[0]; \beta) - e_1 = f^k(x[0]; \beta)$ ($e_1 = 0$ para el S-TM) se convierte en

$$f_s^k(x[0]; \beta) = \beta^k x[0] \prod_{l=0}^{k-1} (-1)^{s[l]-1} + (\beta - 1) \sum_{m=0}^{k-1} \beta^{k-m-1} \prod_{l=m+1}^{k-1} (-1)^{s[l]-1}. \quad (6.14)$$

A partir de (6.14) resulta evidente que la iteración k -ésima del S-TM se puede expresar como un polinomio de orden k en β ,

$$f_s^k(x[0]; \beta) = \sum_{r=0}^k c_s^{r,k} \beta^r, \quad (6.15)$$

estando los coeficientes $c_s^{r,k}$ dados por

$$c_s^{r,k} = \begin{cases} -1, & r = 0; \\ 1 + (-1)^{s[k-1]}, & r = 1; \\ (1 + (-1)^{s[k-r]}) \prod_{l=k-r+1}^{k-1} (-1)^{s[l]-1}, & 2 \leq r \leq k-1; \\ (1 + (-1)^{s[0]-1} x[0]) \prod_{l=1}^{k-1} (-1)^{s[l]-1}, & r = k. \end{cases} \quad (6.16)$$

En cuanto al TM, aparentemente existe una dificultad añadida en el hecho de que b_1 no depende de β , $b_1 = 0$, mientras que b_2 sí, $b_2 = \beta$. No obstante, este problema se puede solventar fácilmente notando que $b_{s[m]}$, el término que aparece en (3.15), se puede expresar como

$$b_{s[m]} = \beta \cdot \chi_{E_2}(x[m]). \quad (6.17)$$

Haciendo uso de (3.15), (6.13) y (6.17), $f_s^k(x[0]; \beta) - e_1 = f^k(x[0]; \beta) - 1/2$ es

$$f_s^k(x[0]; \beta) - e_1 = \beta^k x[0] \prod_{l=0}^{k-1} (-1)^{s[l]-1} + \sum_{m=0}^{k-1} \beta^{k-m} \chi_{E_2}(x[m]) \prod_{l=m+1}^{k-1} (-1)^{s[l]-1} - \frac{1}{2}. \quad (6.18)$$

Obviamente, (6.18) se puede expresar de nuevo conforme a (6.15), siendo los coeficientes del polinomio en esta ocasión

$$c_{\mathbf{s}}^{r,k} = \begin{cases} -1/2, & r = 0; \\ \chi_{E_2}(x[k-1]), & r = 1; \\ \chi_{E_2}(x[k-r]) \prod_{l=k-r+1}^{k-1} (-1)^{s[l]-1}, & 2 \leq r \leq k-1; \\ (\chi_{E_2}(x[0]) + (-1)^{s[0]-1}x[0]) \prod_{l=1}^{k-1} (-1)^{s[l]-1}, & r = k. \end{cases} \quad (6.19)$$

Para finalizar, nótese que los polinomios que se deben factorizar para ambos mapas presentan una estructura muy particular que se podría tratar de aprovechar para obtener sus raíces de un modo eficiente. Por ejemplo, en el caso del S-TM $c_{\mathbf{s}}^{0,k} = -1$ para cualquier valor de k , $c_{\mathbf{s}}^{1,k} \in \{0, 2\}$, $c_{\mathbf{s}}^{r,k} \in \{-2, 0, 2\}$ para $2 \leq r \leq k-1$, y $c_{\mathbf{s}}^{k,k}$ es el único coeficiente del polinomio que puede tomar un amplio abanico de valores gracias a la presencia en el mismo de $x[0]$: $c_{\mathbf{s}}^{k,k} \in [-\beta, \beta]$ cuando $\beta \in [1, 2]$, el rango de valores interesantes del parámetro. En el caso del TM ocurre algo similar: $c_{\mathbf{s}}^{0,k} = -1/2$, $c_{\mathbf{s}}^{1,k} \in \{0, 1\}$, $c_{\mathbf{s}}^{r,k} \in \{-1, 0, 1\}$ para $2 \leq r \leq k-1$, y $c_{\mathbf{s}}^{k,k} \in [-(1 + \beta/2), 1 + \beta/2]$ cuando $\beta \in [1, 2]$. No obstante, en las simulaciones realizadas en esta Tesis no se ha explotado esta estructura, dejándose su potencial aprovechamiento para factorizar los polinomios de un modo más eficiente como una posible línea futura.

6.3.1.2. Método Eficiente de Búsqueda de la Partición

Aunque el Algoritmo 6.1 proporciona muy buenos resultados, permitiendo encontrar la partición exacta en el espacio de parámetros en función del itinerario para el TM y el S-TM, no resulta aplicable para valores medios/altos de N a causa de su elevada complejidad, debida a dos motivos:

1. Su coste computacional depende del número de regiones a buscar (esto es, del número de itinerarios válidos, $P(N)$), que crece exponencialmente con N .
2. La factorización de $f_{\mathbf{s}_{0:k-1}}^k(x[0]; \theta) - e_1$ dentro de cada una de las regiones asociadas a $\mathbf{s}_{0:k-2}$, $Q_{\mathbf{s}_{0:k-2}}^{k-1}$, requiere la obtención de los ceros de un polinomio de orden creciente con k .

La primera dificultad es inherente al problema, y resulta inevitable si se desea encontrar la partición completa en el espacio de parámetros. Sin embargo, la segunda dificultad es evitable, pudiéndose plantear un algoritmo eficiente gracias a la siguiente proposición.

Proposición 6.1 *Sea una condición inicial, $x[0]$, un itinerario de longitud $k-1$ cualquiera, $\mathbf{s}_{0:k-2} = [s[0], \dots, s[k-2]]^T$, y $Q_{\mathbf{s}_{0:k-2}}^{k-1}$ la región en el espacio de parámetros asociada a $x[0]$ y $\mathbf{s}_{0:k-2}$. La función $f^k(x[0]; \beta)$ es monótona dentro de $Q_{\mathbf{s}_{0:k-2}}^{k-1}$, tanto para el TM como para el S-TM, de modo que únicamente puede existir como mucho un cero de $f^k(x[0]; \beta) - e_1$ dentro de $Q_{\mathbf{s}_{0:k-2}}^{k-1}$. Además, las regiones que conforman el espacio de parámetros en este caso, $Q_{\mathbf{s}_{0:k-1}}^k$, son convexas.*

Nótese que la Proposición 6.1 garantiza que la región asociada a cada itinerario en el espacio de parámetros es *conexa* (esto es, que está compuesta por un único intervalo) y *convexa* (es decir, que $J(x[0]; \beta)$ presenta como mucho un mínimo dentro de la misma). Desafortunadamente, no se ha conseguido llevar a cabo una demostración teórica de la Proposición 6.1, aunque se ha comprobado experimentalmente a lo largo de numerosas simulaciones. Como muestra, en la Figura 6.1 se presentan dos ejemplos del comportamiento de $f^k(x[0]; \beta)$ y $J(x[0]; \beta)$ dentro de las regiones de la partición para el TM y el S-TM, confirmandose su monotonía y convexidad respectivamente en ambos casos.

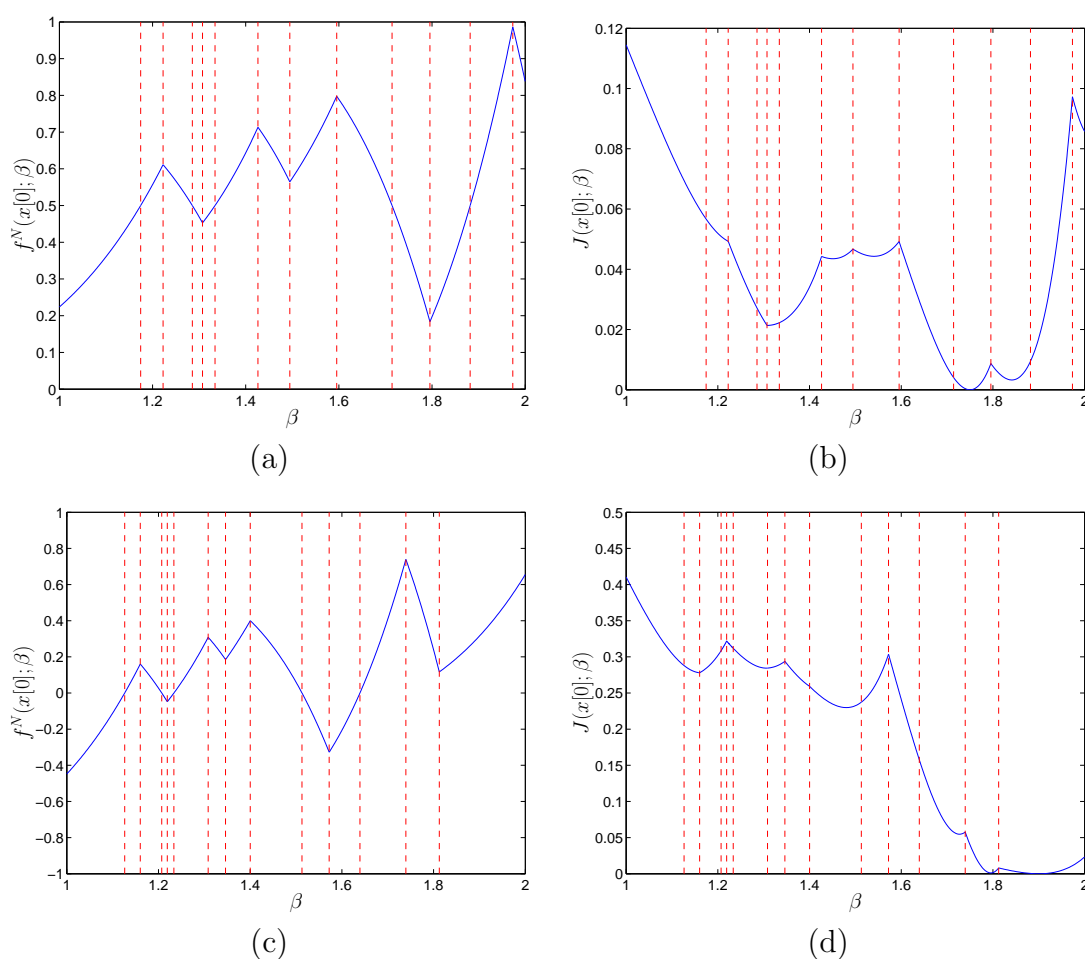


Figura 6.1: Monotonía de $f^N(x[0]; \beta)$ dentro de las regiones de la partición $(N - 1)$ -ésima y convexidad de $J(x[0]; \beta)$ para el TM y el S-TM. (a) $f^N(x[0]; \beta)$ para el TM con $x[0] = 0,2238$, $\beta = 1,75$ y $N = 5$. (b) $J(x[0]; \beta)$ para el mismo caso. (c) $f^N(x[0]; \beta)$ para el S-TM con $x[0] = -0,4483$, $\beta = 1,9$ y $N = 5$. (d) $J(x[0]; \beta)$ para el mismo caso.

Una segunda característica evidenciada por la Figura 6.1 es que el número de regiones en general es mucho menor que M^N (de las 32 regiones posibles en el ejemplo

del TM únicamente existen 13 y en el del S-TM 14), reduciéndose por lo tanto notablemente el coste computacional. Para corroborar estas dos propiedades y resaltar su cumplimiento para cualquier valor de N , β y $x[0]$, en la Figura 6.2 se muestra un tercer ejemplo para $\beta = 2$ y $N = 10$: solamente existen 137 regiones de las 1024 posibles, y en todas ellas $f^k(x[0]; \beta)$ es monótona (para $1 \leq k \leq N$) y $J(x[0]; \beta)$ es convexa.

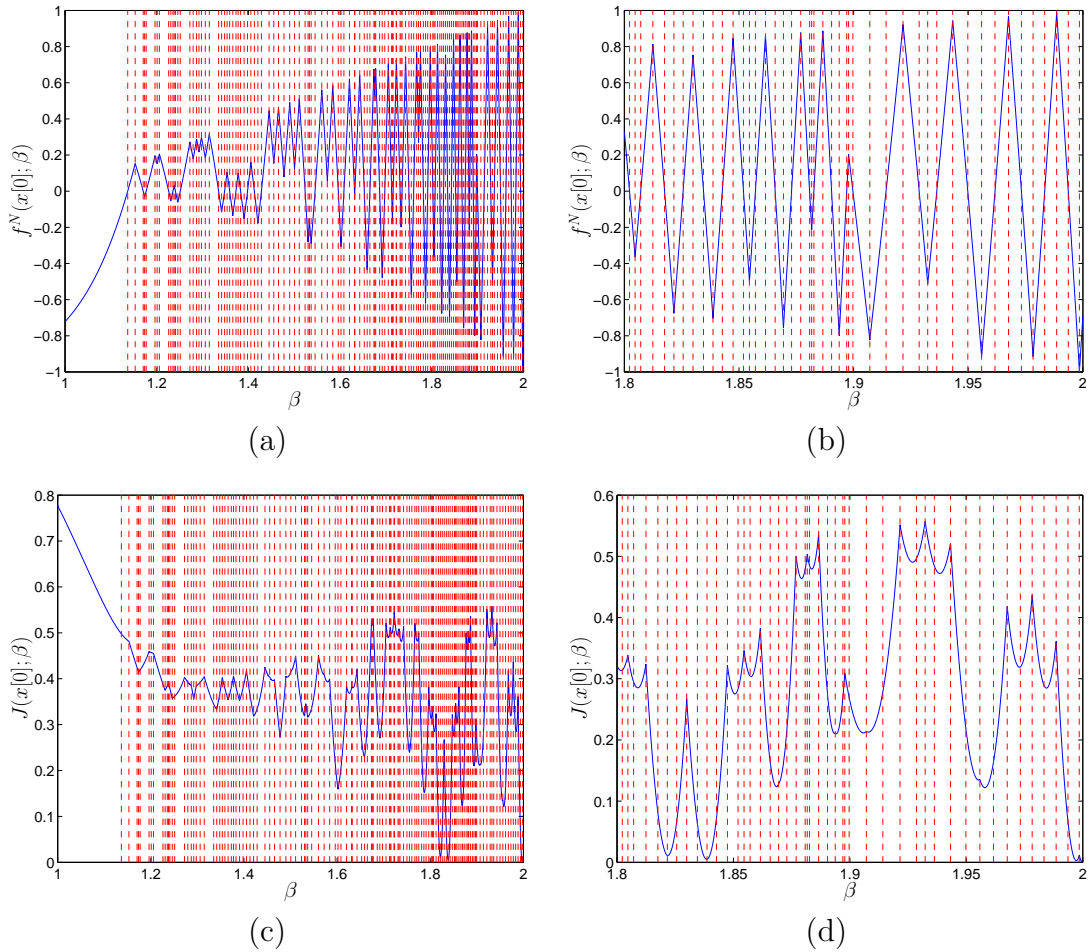


Figura 6.2: Monotonía de $f^k(x[0]; \beta)$ y convexidad de $J(x[0]; \beta)$ para el S-TM con $x[0] = -0,7223$, $\beta = 2$ y $N = 10$. (a) $f^N(x[0]; \beta)$ para $1 \leq \beta \leq 2$. (b) $f^N(x[0]; \beta)$ para $1,8 \leq \beta \leq 2$. (c) $J(x[0]; \beta)$ para $1 \leq \beta \leq 2$. (d) $J(x[0]; \beta)$ para $1,8 \leq \beta \leq 2$.

La Proposición 6.1 permite simplificar en gran medida el Algoritmo 6.1, ya que, haciendo uso de dicha hipótesis, solamente es necesario buscar una raíz de $f^k - e_1$ dentro de cada $Q_{s_{0:k-2}}^{k-1}$, como se muestra en el Algoritmo 6.2, en lugar de factorizar un polinomio de orden k y buscar cuales de sus raíces pertenecen a la región deseada. En este caso, la búsqueda del único cero posible dentro de cada región procede evaluando f^k en los límites inferior y superior de $Q_{s_{0:k-2}}^{k-1}$: si $f^k(x[0]; \vartheta_{s_{0:k-2}}^{k-1}) \in E_1$ y $f^k(x[0]; \varphi_{s_{0:k-2}}^{k-1}) \in E_2$ o viceversa, entonces existe un cruce de región que se puede obtener mediante algún

algoritmo del tipo descenso de gradiente; en caso contrario, no existe ningún cruce y uno de los dos nuevos itinerarios de longitud k , $\mathbf{s}_{0:k-1}^1 = [\mathbf{s}_{0:k-2}, 1]$ o $\mathbf{s}_{0:k-1}^2 = [\mathbf{s}_{0:k-2}, 2]$, va a ser inválido. Además, la Proposición 6.1 facilita la obtención de los estimadores ML locales, ya que, al no poder existir más de un mínimo dentro de cada región, este se puede encontrar mediante un algoritmo iterativo local cualquiera.

1. Inicialización del algoritmo. Se establecen los límites iniciales de las regiones a partir del rango admisible de parámetros, $\Xi_0 = \Theta = [\theta_{\min}, \theta_{\max}]$, y la matriz con los itinerarios iniciales se hace igual a la región a la que pertenece $x[0]$, $\mathbf{S}_0 = s[0]$.
2. Para $1 \leq k \leq N$ se busca la única raíz posible de $f_{\mathbf{s}_{0:k-1}}^k(x[0]; \theta) - e_1$ dentro de $Q_{\mathbf{s}_{0:k-2}}^{k-1}$ mediante el siguiente algoritmo:
 - 2.1. Inicializar Ξ_k y \mathbf{S}_k como el conjunto vacío: $\Xi_k = []$ y $\mathbf{S}_k = []$. Inicializar también el contador con el número de raíces de $f_{\mathbf{s}_{0:k-1}}^k(x[0]; \theta) - e_1$: $j = 1$.
 - 2.2. Para $1 \leq i \leq L_{k-1} - 1$, siendo L_{k-1} la longitud de Ξ_{k-1} , repetir los pasos siguientes:
 - 2.2.1. Añadir el límite de la región i -ésima a la partición k -ésima: $\Xi_k = [\Xi_k, \Xi_{k-1}(i)]$.
 - 2.2.2. Evaluar f^k en los límites de la región i -ésima de la partición $(k-1)$ -ésima: $x_1[k] = f^k(x[0]; \Xi_{k-1}(i))$ y $x_2[k] = f^k(x[0]; \Xi_{k-1}(i+1))$.
 - 2.2.3. Si $x_1[k] \in E_1$ y $x_2[k] \in E_2$ o viceversa, entonces existe un nuevo cero que añadir a la partición k -ésima entre $\Xi_{k-1}(i)$ y $\Xi_{k-1}(i+1)$:
 - Encontrar ese punto, θ^* , aplicando un algoritmo de descenso de gradiente o de Newton-Raphson estándar en el espacio de parámetros a $f^k(x[0]; \theta) - e_1$.
 - Añadirlo a la partición: $\Xi_k = [\Xi_k, \theta^*]$.
 - Incorporar las nuevas secuencias simbólicas a la matriz de itinerarios: $\mathbf{S}_k(j, 1 : k) = \mathbf{S}_k(j+1, 1 : k) = \mathbf{S}_{k-1}(i, 1 : k)$, $\mathbf{S}_k(j, k+1) = s_1[k]$ y $\mathbf{S}_k(j+1, k+1) = s_2[k]$, siendo $s_1[k]$ y $s_2[k]$ el intervalo al que pertenecen $x_1[k]$ y $x_2[k]$ respectivamente.
 - Actualizar el contador con el número de ceros: $j = j + 2$.
 - 2.2.4. Si $x_1[k]$ y $x_2[k]$ pertenecen al mismo intervalo, entonces no existe ningún cero que añadir a la partición:
 - Incorporar la nueva secuencia simbólica a la matriz de itinerarios: $\mathbf{S}_k(j, 1 : k) = \mathbf{S}_{k-1}(i, 1 : k)$ y $\mathbf{S}_k(j, k+1) = s_1[k] = s_2[k]$.
 - Actualizar el contador con el número de ceros: $j = j + 1$.
 - 2.3. Añadir el límite superior de Ξ_{k-1} a Ξ_k : $\Xi_k = [\Xi_k, \Xi_{k-1}(L_{k-1})]$.
3. La partición deseada viene dada por Ξ_N , y el conjunto de secuencias simbólicas asociadas a cada región por \mathbf{S}_N .

Algoritmo 6.2: Obtención eficiente de una partición del espacio de parámetros en función del itinerario para un mapa unimodal, continuo en θ y x , y que cumple la Proposición 6.1.

6.3.2. Estimadores ML y HC-ML

La obtención del estimador ML exacto del parámetro de bifurcación de un mapa caótico siguiendo la metodología propuesta en el Capítulo 3 requeriría:

1. Encontrar la partición del espacio de parámetros en función del itinerario.
2. Calcular los mínimos de la función de coste (ceros de la derivada) situados dentro de la región correspondiente a cada secuencia simbólica, y seleccionar el mejor de todos ellos, obteniéndose de este modo hasta $P(N)$ estimadores ML locales.
3. Seleccionar el mejor estimador ML local (esto es, el que proporcione el mínimo valor de la función de coste) como estimador ML global.

Este mecanismo de estimación va a resultar muy costoso, por lo que se procede de un modo alternativo calculando la partición en el espacio de parámetros, y utilizando como estima local el punto medio de cada región o un valor del parámetro obtenido mediante una búsqueda en una rejilla o un algoritmo de descenso de gradiente (en el caso del TM o el S-TM), en lugar de la verdadera estima ML local, cuya obtención precisa resulta complicada. El procedimiento detallado se muestra en el Algoritmo 6.3. Nótese que, aunque las estimas locales no son en general las de máxima verosimilitud (y por lo tanto la estima global tampoco es la ML), sin embargo sí que tienden hacia ellas conforme aumenta N (y otro tanto ocurre con la estima global), debido a la disminución progresiva del tamaño de las regiones de la partición. Así pues, se puede afirmar que la estima encontrada mediante el Algoritmo 6.3 converge asintóticamente hacia la estima ML del parámetro θ conforme N tiende a infinito.

El Algoritmo 6.3 funciona muy bien para valores pequeños de N . Sin embargo, conforme N aumenta el número de regiones de la partición crece exponencialmente, lo que imposibilita su uso para valores medios/altos de N . En estos casos se propone una estima del tipo HC-ML consistente en utilizar el itinerario obtenido a partir de la secuencia de observaciones ruidosas (posiblemente modificando aquellas más susceptibles de ser erróneas) como alternativa computacionalmente eficiente [Panta2000c]. El funcionamiento detallado de este estimador se muestra en el Algoritmo 6.4, y su rendimiento se estudia en la Sección 6.6, mostrándose su eficiencia estadística asintótica: sus estimas son asintóticamente insesgadas y su varianza alcanza el CRLB para un valor de SNR suficientemente elevado.

6.4. Estimadores Bloque

En esta sección se presentan otras alternativas de estimación, subóptimas y con menor rendimiento, pero mucho más sencillas de implementar. Todas ellas se basan en alguna función de $y[k]$ y $f(y[k-1]; \theta)$, bien obtenida a partir de una estima de sus momentos de orden r , bien como una variante del *error de predicción de primer orden* (esto es, de la diferencia entre $y[k]$ y $f(y[k-1]; \theta)$). Esta clase de funciones

1. Construir la partición del espacio de parámetros, Ξ_N , en función del itinerario usando el Algoritmo 6.1 o el Algoritmo 6.2.
2. Para $1 \leq i \leq L_N - 1$, siendo L_N la longitud de Ξ_N , obtener una aproximación a la estima ML local i -ésima de θ : bien como el punto intermedio de la región

$$\hat{\theta}_{\text{ML}}^i = \frac{1}{2} (\Xi_N(i) + \Xi_N(i + 1));$$

bien mediante una búsqueda de rejilla,

$$\hat{\theta}_{\text{ML}}^i = \arg \min_{\theta^{i,k}} J(x[0]; \theta^{i,k}),$$

siendo $\theta^{i,k} = \Xi_N(i) + (\Xi_N(i + 1) - \Xi_N(i))k/(N_r - 1)$, N_r el número de puntos de la rejilla, y $k = 0, \dots, N_r - 1$; o bien mediante un algoritmo iterativo local de descenso de gradiente (sólo adecuado para regiones convexas).

3. Seleccionar como estima ML global aproximada la mejor de todas las estimas ML locales:

$$\hat{\theta}_{\text{ML}} = \arg \min_{\hat{\theta}_{\text{ML}}^i} J(x[0]; \hat{\theta}_{\text{ML}}^i).$$

Algoritmo 6.3: Estimador ML aproximado para mapas con dependencia lineal de un único parámetro, θ , y polinómica con x .

de coste no aprovechan toda la información contenida en las observaciones, pero dan lugar a estimadores de bajo coste computacional y que son aplicables a una amplia clase de mapas caóticos: todos aquellos que presenten una dependencia lineal con sus parámetros. Es decir, en general mapas cuya forma viene dada por (6.3),

$$f(x; \boldsymbol{\theta}) - g_0(x) = \boldsymbol{\theta}^T \mathbf{g}_{1:p}(x), \quad (6.20)$$

aunque también se pueden aplicar a aquellos mapas en los que aparezca dicha dependencia lineal después de una transformación invertible, como el mapa exponencial de la Tabla 2.2:

$$\ln f(x; c) - \ln x = c(1 - x).$$

En la Tabla 6.1 se muestran los valores de $g_0(x)$ y $g_1(x)$, necesarios para plantear cualquiera de los estimadores de esta sección, para los seis mapas caóticos de la Tabla 2.2 usados en las simulaciones: el “tent-map” (TM), el “tent-map” simétrico (S-TM), el mapa logístico (Log.), el mapa de Singer (Singer), el mapa sinusoidal 1 (Sin.), y el mapa exponencial (Exp.).

A continuación se presentan los tres estimadores bloque considerados: el basado en el método de los momentos (MBE), el de mínimos cuadrados (LS), y el de mínimos cuadrados totales (TLS). Y posteriormente se discute el modo de aplicar dichos esti-

1. Estimar el itinerario a partir de la secuencia de observaciones ruidosas: $\hat{s} = [\hat{s}[0], \dots, \hat{s}[N-1]]^T$ con $\hat{s}[n] = i \Leftrightarrow y_T[n] \in E_i$, e $y_T[n]$ dada por (5.8).
2. Obtener la región de la partición asociada a \hat{s} mediante una versión modificada del Algoritmo 6.1 o del Algoritmo 6.2 en la que únicamente se explora la región del espacio de parámetros de interés. Esto es, sólo se pretende encontrar aquella región asociada al itinerario \hat{s} , y no la partición completa.
3. Encontrar la estima ML correspondiente a \hat{s} del mismo modo que en el Algoritmo 6.3: como el punto medio de la región, mediante una búsqueda de rejilla, o utilizando un algoritmo de descenso de gradiente local.
4. Si la secuencia simbólica no es válida o si se desean explorar otras secuencias, modificar \hat{s} siguiendo el mismo procedimiento que para el estimador HC-ML(k) de $x[0]$, dado por el Algoritmo 5.2, y repetir los pasos 2 y 3 para cada uno de los itinerarios alternativos.
5. Seleccionar la mejor de todas las estimas obtenidas (es decir, la que proporcione un menor valor de la función de coste) como estima HC-ML de θ .

Algoritmo 6.4: Estimador HC-ML(k) aproximado para mapas con dependencia lineal de un único parámetro, θ , y polinómica con x .

madores a mapas cuyos parámetros definen los límites de la partición natural, como el SK-TM y el BSK-TM.

Mapa	$g_0(x)$	$g_1(x)$
TM	0	$(1 - 2 x - 0,5)/2$
S-TM	-1	$1 - x $
Log.	0	$x(1 - x)$
Singer	0	$7,86x - 23,31x^2 + 28,75x^3 - 13,3x^4$
Sin.	0	$\sin(\pi x)$
Exp.	$\ln x$	$1 - x$

Tabla 6.1: Funciones necesarias para construir los distintos estimadores subóptimos de los seis mapas caóticos utilizados en las simulaciones.

6.4.1. Método de los Momentos (MBE)

Habitualmente se dice que un estimador está basado en el método de los momentos (MBE) cuando se obtiene mediante la solución de una ecuación teórica (generalmente sencilla) que involucra uno o varios de los primeros *momentos* de la FDP de las observaciones. Los estimadores obtenidos mediante el método de los momentos no cumplen ningún criterio de optimalidad. No obstante, en ocasiones puede resultar interesante su uso debido a que suelen ser fáciles de calcular e implementar, y resultan muy útiles

cuando el registro de datos es suficientemente largo, ya que usualmente son asintóticamente insesgados y consistentes [Kay1993].

El caso más general es aquel en que se dispone de un conjunto de $N+1$ observaciones que dependen de $\boldsymbol{\theta}$, $y[k; \boldsymbol{\theta}]$ con $0 \leq k \leq N$, siendo $\boldsymbol{\theta}$ de dimensión $p \times 1$. En esta ocasión es necesario plantear p ecuaciones basadas en p momentos de diferentes órdenes tales que den lugar a un conjunto de ecuaciones linealmente independientes:

$$\mu_{r_i} = E(y[k; \boldsymbol{\theta}]^{r_i}) = h_i(\boldsymbol{\theta}), \quad (6.21)$$

con $1 \leq i \leq p$ y normalmente $r_i = i$ (es decir, se usan los p primeros momentos). Agrupando las p ecuaciones en una única ecuación vectorial,

$$\boldsymbol{\mu} = \mathbf{h}(\boldsymbol{\theta}),$$

con $\boldsymbol{\mu} = [\mu_{r_1}, \dots, \mu_{r_p}]^T$ y $\mathbf{h}(\boldsymbol{\theta}) = [h_1(\boldsymbol{\theta}), \dots, h_p(\boldsymbol{\theta})]^T$, y asumiendo que \mathbf{h} es invertible, entonces resulta inmediato obtener el vector de parámetros:

$$\boldsymbol{\theta} = \mathbf{h}^{-1}(\boldsymbol{\mu}). \quad (6.22)$$

Obviamente, en la práctica no es posible calcular el valor exacto de μ_{r_i} , ya que se desconoce el valor real de $\boldsymbol{\theta}$. No obstante, se pueden reemplazar las esperanzas matemáticas que aparecen en (6.21) por su “estimador natural”, la *media muestral*:

$$\hat{\mu}_{r_i} = \frac{1}{N+1} \sum_{k=0}^N y[k]^{r_i}. \quad (6.23)$$

Finalmente, sustituyendo en (6.22) $\boldsymbol{\mu}$ por su estima, $\hat{\boldsymbol{\mu}} = [\hat{\mu}_{r_1}, \dots, \hat{\mu}_{r_p}]^T$ con $\hat{\mu}_{r_i}$ dado por (6.23), se obtiene el estimador buscado basado en el método de los momentos:

$$\hat{\boldsymbol{\theta}}_{\text{MBE}} = \mathbf{h}^{-1}(\hat{\boldsymbol{\mu}}).$$

Para un mapa genérico con p parámetros (caso vectorial) puede resultar muy complicado encontrar un sistema de ecuaciones apropiado e invertirlo para hallar la solución exacta de $\hat{\boldsymbol{\theta}}_{\text{MBE}}$. Sin embargo, para $p = 1$ (caso escalar) es muy sencillo. En esta ocasión únicamente se debe calcular su momento no centrado de orden r ,

$$\mu_r = E(y[k; \theta]^r) = h(\theta),$$

y, asumiendo que la función $h(\theta)$ es invertible, obtener θ invirtiéndola:

$$\theta = h^{-1}(\mu_r).$$

Reemplazando de nuevo la esperanza matemática por la media muestral, se obtiene el estimador basado en el método de los momentos de θ :

$$\hat{\theta}_{\text{MBE}} = h^{-1} \left(\frac{1}{N+1} \sum_{k=0}^N y[k]^r \right). \quad (6.24)$$

En el caso de un mapa caótico con dependencia lineal de su único parámetro, θ , el momento no centrado de orden r de las observaciones, $y[k]$, menos el término independiente de θ , $g_0(x)$, se puede escribir como

$$\mu_r = E((y[k] - g_0(y[k-1]))^r) = \theta^r \cdot E(g_1(y[k-1])^r),$$

y despejando el parámetro a estimar se obtiene que

$$\theta = \left(\frac{\mu_r}{E(g_1(y[k-1])^r)} \right)^{1/r} = \left(\frac{E((y[k] - g_0(y[k-1]))^r)}{E(g_1(y[k-1])^r)} \right)^{1/r}. \quad (6.25)$$

Finalmente, sustituyendo las dos esperanzas matemáticas de (6.25) por su media muestral, la estima MBE de θ resulta

$$\hat{\theta}_{\text{MBE}} = \left(\frac{\sum_{k=1}^N z[k]^r}{\sum_{k=1}^N g_1(y[k-1])^r} \right)^{1/r}, \quad (6.26)$$

con

$$z[k] = y[k] - g_0(y[k-1]). \quad (6.27)$$

En esta Tesis se van a considerar los estimadores obtenidos para $r = 1, 2, 3$ y 4 . La forma del MBE en el caso más sencillo ($r = 1$, para el que el estimador se obtiene a partir del promedio de las observaciones) para cada uno de los seis mapas de la Tabla 6.1 se presenta en la Tabla 6.2 (mostrada al final de la Sección 6.4.2) junto con la del estimador LS, de modo que se aprecien claramente sus semejanzas y diferencias.

6.4.2. Estimador de Mínimos Cuadrados (LS)

En el sentido estricto, un estimador de mínimos cuadrados (LS) es todo aquel que minimiza el error cuadrático existente entre las observaciones y el modelo de señal considerado. En consecuencia, resulta evidente que los estimadores ML vistos anteriormente, tanto de la señal como del parámetro, son estimadores de mínimos cuadrados. La diferencia fundamental entre ambos es que en el primero la dependencia con el parámetro, $x[n]$, es cuadrática para una secuencia simbólica dada (y por lo tanto existe una solución única, que además se puede obtener de manera analítica para mapas PWL), mientras que en el segundo aparece una dependencia fuertemente no lineal diferente para cada mapa.

Para conseguir una dependencia cuadrática en esta ocasión es necesario limitar la clase de mapas analizados a aquellos cuya dependencia con los parámetros es lineal (posiblemente tras una transformación invertible). Considerando una ecuación de la forma (6.20), y teniendo en cuenta que se dispone de $N + 1$ observaciones, se puede plantear un sistema lineal de N ecuaciones con p incógnitas,

$$\mathbf{z}_{1:N} + \mathbf{r} = \mathbf{G}^T \boldsymbol{\theta}, \quad (6.28)$$

en el que $\mathbf{z}_{1:N} = [y[1] - g_0(y[0]), \dots, y[N] - g_0(y[N-1])]^T = [z[1], \dots, z[N]]^T$, con $z[k]$ dado por (6.27), es el vector con las observaciones que se desean modelar, $\mathbf{G} = [\mathbf{g}_{1:p}(y[0]), \dots, \mathbf{g}_{1:p}(y[N-1])]$ es la matriz $p \times N$ correspondiente a las predicciones del modelo, y \mathbf{r} es el vector con el error de modelado debido a que en general las observaciones no se ajustan perfectamente al modelo. El estimador LS de $\boldsymbol{\theta}$ es aquel que minimiza la norma L_2 del error, $\|\mathbf{r}\|_2^2 = \|\mathbf{z}_{1:N} - \mathbf{G}^T \boldsymbol{\theta}\|_2^2$, y para un sistema indeterminado (esto es, aquel en el que el número de parámetros a ajustar, p , es menor o igual que el número de pares de observaciones, N) se obtiene mediante la *pseudoinversa* o *inversa generalizada de Moore-Penrose* [Golub1996]:

$$\hat{\boldsymbol{\theta}}_{\text{LS}} = \mathbf{G}^\dagger \mathbf{z}_{1:N} = (\mathbf{G}\mathbf{G}^T)^{-1} \mathbf{G} \mathbf{z}_{1:N}. \quad (6.29)$$

En esta sección se van a estudiar solamente los mapas con un único parámetro, θ , para los que la función de coste está compuesta por el sumatorio al cuadrado del error de predicción de $y[k]$ a partir de $y[k-1]$:

$$J_{\text{LS}}(\theta) = \sum_{k=1}^N \varepsilon_{\text{LS}}[k]^2, \quad (6.30)$$

siendo $\varepsilon_{\text{LS}}[k]$ el error de predicción para la muestra k -ésima,

$$\varepsilon_{\text{LS}}[k] = y[k] - f(y[k-1]; \theta) = z[k] - \theta \cdot g_1(y[k-1]), \quad (6.31)$$

y con $z[k]$ dado por (6.27). Derivando (6.30) con respecto a θ e igualando a cero se puede encontrar fácilmente una expresión cerrada para el estimador LS de θ :

$$\hat{\theta}_{\text{LS}} = \frac{\sum_{k=1}^N z[k] g_1(y[k-1])}{\sum_{k=1}^N g_1(y[k-1])^2}. \quad (6.32)$$

En la Tabla 6.2 se comparan las expresiones del MBE para $r = 1$ con las del estimador LS para los seis mapas de la Tabla 6.1. Nótese que se ha utilizado

$$p_s(y[k-1]) = 7,86y[k-1] - 23,31y[k-1]^2 + 28,75y[k-1]^3 - 13,3y[k-1]^4$$

para denotar el polinomio de orden cuatro que caracteriza al mapa de Singer con el fin de evitar expresiones innecesariamente largas y complejas.

6.4.3. Estimador de Mínimos Cuadrados Totales (TLS)

El estimador LS desarrollado en la Sección 6.4.2 se puede ver como la solución de un problema de regresión no lineal en el que se trata de ajustar una función, $f(x; \theta)$, de tal modo que el error cuadrático del conjunto de $N + 1$ observaciones sea mínimo. En este sentido, el estimador LS resuelve el problema suponiendo que únicamente existe

Mapa	MBE	LS
TM	$\hat{\beta}_{\text{MBE}} = \frac{2 \sum_{k=1}^N y[k]}{\sum_{k=1}^N (1-2 y[k-1]-0,5)}$	$\hat{\beta}_{\text{LS}} = \frac{2 \sum_{k=1}^N y[k](1-2 y[k-1]-0,5)}{\sum_{k=1}^N (1-2 y[k-1]-0,5)^2}$
S-TM	$\hat{\beta}_{\text{MBE}} = \frac{\sum_{k=1}^N (1+y[k])}{\sum_{k=1}^N (1- y[k-1])}$	$\hat{\beta}_{\text{LS}} = \frac{\sum_{k=1}^N (1+y[k])(1- y[k-1])}{\sum_{k=1}^N (1- y[k-1])^2}$
Log.	$\hat{\lambda}_{\text{MBE}} = \frac{\sum_{k=1}^N y[k]}{\sum_{k=1}^N y[k-1](1-y[k-1])}$	$\hat{\lambda}_{\text{LS}} = \frac{\sum_{k=1}^N y[k]y[k-1](1-y[k-1])}{\sum_{k=1}^N y[k-1]^2(1-y[k-1])^2}$
Singer	$\hat{c}_{\text{MBE}} = \frac{\sum_{k=1}^N y[k]}{\sum_{k=1}^N p_s(y[k-1])}$	$\hat{c}_{\text{LS}} = \frac{\sum_{k=1}^N y[k]p_s(y[k-1])}{\sum_{k=1}^N p_s(y[k-1])^2}$
Sin.	$\hat{c}_{\text{MBE}} = \frac{\sum_{k=1}^N y[k]}{\sum_{k=1}^N \sin(\pi y[k-1])}$	$\hat{c}_{\text{LS}} = \frac{\sum_{k=1}^N y[k] \sin(\pi y[k-1])}{\sum_{k=1}^N \sin(\pi y[k-1])^2}$
Exp.	$\hat{c}_{\text{MBE}} = \frac{\sum_{k=1}^N \ln(y[k]/y[k-1])}{\sum_{k=1}^N (1-y[k-1])}$	$\hat{c}_{\text{LS}} = \frac{\sum_{k=1}^N \ln(y[k]/y[k-1])(1-y[k-1])}{\sum_{k=1}^N (1-y[k-1])^2}$

Tabla 6.2: Forma de los estimadores MBE y LS para los seis mapas caóticos de la Tabla 6.1 utilizados en las simulaciones.

error en el eje de las abscisas (es decir, en $\mathbf{z}_{1:N}$), y minimizando la distancia entre los puntos correspondientes a dos observaciones consecutivas, $(y[k-1], y[k])$, y a los pares entrada-salida del modelo, $(y[k-1], f(y[k-1]; \theta))$. Teniendo esto en cuenta, el error de predicción del estimador LS, (6.31), se puede expresar alternativamente como

$$\begin{aligned} \varepsilon_{\text{LS}}[k] &= d([y[k-1], y[k]]^T, [y[k-1], f(y[k-1]; \theta)]^T) \\ &= \|[0, y[k] - f(y[k-1]; \theta)]\|_2, \end{aligned} \quad (6.33)$$

donde $d(\mathbf{x}, \mathbf{y})$ indica la distancia euclídea entre los vectores \mathbf{x} e \mathbf{y} .

Sin embargo, en el problema considerado resulta evidente que en realidad existe error tanto en las abscisas como en las ordenadas, de modo que es preferible un estimador que trate de hallar la solución de *mínimos cuadrados totales* (TLS) [Golub1996]. Algebraicamente esto significa que el sistema de ecuaciones que se pretende resolver en esta ocasión es

$$\mathbf{z}_{1:N} + \mathbf{r} = (\mathbf{G} + \mathbf{E})^T \boldsymbol{\theta}, \quad (6.34)$$

donde \mathbf{r} denota de nuevo el error en las observaciones, y con respecto a (6.28) se ha introducido una fuente adicional de error, \mathbf{E} , que evidencia el error en el modelo representado por la matriz \mathbf{G} . Para un mapa genérico con múltiples parámetros no es posible plantear una ecuación cerrada para el estimador TLS, y el problema se resuelve habitualmente haciendo uso de la descomposición en valores singulares (SVD) [Golub1996].

No obstante, para un mapa con un único parámetro y considerando de nuevo la estimación como un problema de regresión, existe una interpretación geométrica sencilla del estimador TLS [Golub1980], que se muestra en la Figura 6.3 para el mapa logístico.

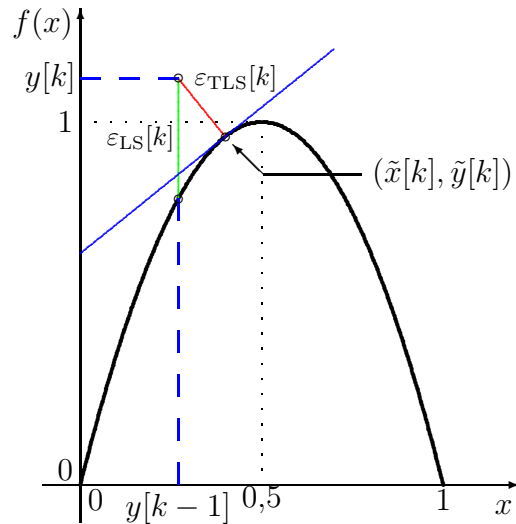


Figura 6.3: Interpretación geométrica del error de predicción para los estimadores LS y TLS con el mapa logístico.

Como se puede apreciar claramente en la figura, mientras que el LS obtiene el error de predicción para cada observación proyectando verticalmente sobre el modelo (error únicamente en la abscisa), el TLS lo obtiene proyectando de manera perpendicular al mismo (error tanto en la abscisa como en la ordenada). A la vista de esta observación, resulta evidente que, para un sólo parámetro, es posible plantear una función de coste para el TLS equivalente a (6.30),

$$J_{\text{TLS}}(\theta) = \sum_{k=1}^N \varepsilon_{\text{TLS}}[k]^2, \quad (6.35)$$

donde $\varepsilon_{\text{TLS}}[k]$ es la distancia de $(y[k-1], y[k])$ al par entrada-salida compuesto por $(\tilde{x}[k], \tilde{y}[k])$ de la Figura 6.3:

$$\begin{aligned} \varepsilon_{\text{TLS}}[k] &= d([y[k-1], y[k]]^T, [\tilde{x}[k], \tilde{y}[k]]^T) \\ &= \|[y[k-1] - \tilde{x}[k], y[k] - \tilde{y}[k]]\|_2. \end{aligned} \quad (6.36)$$

Teniendo en cuenta que la pendiente del modelo viene dada por la derivada de $f(x; \theta)$, y que la pendiente de la recta perpendicular a f que pasa por $(\tilde{x}[k], \tilde{y}[k])$ es $-1/f'(\tilde{x}[k]; \theta)$, el punto de corte entre esta recta y el modelo, $\tilde{x}[k]$, se obtiene resolviendo la ecuación

$$y[k] - \frac{x - y[k-1]}{f'(x; \theta)} = f(x; \theta),$$

que mediante una simple manipulación se puede reescribir como

$$f'(x; \theta)f(x; \theta) + x - f'(x; \theta)y[k] - y[k - 1] = 0. \quad (6.37)$$

Aunque en general resulta muy sencillo plantear esta ecuación, normalmente no va a ser fácil encontrar su solución en función del parámetro a estimar. Por ejemplo, para el mapa logístico se obtiene la siguiente ecuación cúbica,

$$x^3 - \frac{3}{2}x^2 + \frac{1 + 2\lambda y[k] + \lambda^2}{2\lambda^2}x - \frac{\lambda y[k] + y[k - 1]}{2\lambda^2} = 0,$$

cuyas soluciones presentan una expresión cerrada [Abramo1965], pero que va a ser una función complicada de λ . En consecuencia, para un mapa genérico no se puede obtener una expresión analítica del estimador TLS como las mostradas en la Tabla 6.2 para los estimadores MBE y LS, y el problema debe resolverse haciendo uso de la SVD.

No obstante, existen dos casos en los que (6.37) sí va a dar lugar a una solución cerrada sencilla: el TM y el S-TM. En el primer caso $f(x; \beta)$ viene dada por (2.17) o (2.18), y su derivada es

$$f'(x; \beta) = -\beta s(x - 0,5),$$

mientras que en el segundo caso $f(x; \beta)$ se puede escribir como (2.21) o (2.22), siendo su derivada

$$f'(x; \beta) = -\beta s(x),$$

y $s(x)$ la función signo. En ambos casos se llega a ecuaciones muy sencillas de resolver (lineales) notando que el intervalo al que pertenecen $y[k - 1]$ y $\tilde{x}[k]$ es el mismo. Es decir, que $s(\tilde{x}[k] - 0,5) = s(y[k - 1] - 0,5)$ para el TM y que $s(\tilde{x}[k]) = s(y[k - 1])$ para el S-TM. Esto resulta evidente a la vista de la Figura 6.3, ya que, al obtenerse $(\tilde{x}[k], \tilde{y}[k])$ proyectando $(y[k - 1], y[k])$ perpendicularmente sobre el modelo, el intervalo al que pertenecen $\tilde{x}[k]$ e $y[k - 1]$ tiene que ser siempre el mismo. A continuación, planteando (6.37) para el TM y despejando x se obtiene

$$\tilde{x}[k] = \frac{\beta^2(1 + s(y[k - 1] - 0,5))/2 - \beta s(y[k - 1] - 0,5)y[k] + y[k - 1]}{1 + \beta^2}, \quad (6.38)$$

y llevando a cabo el mismo proceso para el S-TM el resultado es

$$\tilde{x}[k] = \frac{\beta^2 s(y[k - 1]) - \beta(1 + y[k])s(y[k - 1]) + y[k - 1]}{1 + \beta^2}. \quad (6.39)$$

Ahora, para obtener $\tilde{y}[k]$ simplemente hay que iterar $\tilde{x}[k]$, y, a pesar de que la forma de (6.38) y (6.39) parezca implicar una expresión complicada para $\tilde{y}[k]$, en realidad se obtiene una ecuación muy sencilla,

$$\tilde{y}[k] = f(\tilde{x}[k]; \beta) = \frac{f(y[k - 1]; \beta) + \beta^2 y[k]}{1 + \beta^2}, \quad (6.40)$$

que es válida tanto para el TM como para el S-TM, aunque $f(x; \beta)$ depende del mapa en cuestión lógicamente. Sustituyendo (6.38) o (6.39) y (6.40) en (6.36) y operando, se puede demostrar que el error de predicción para el TLS viene dado por

$$\varepsilon_{\text{TLS}}[k] = \frac{\varepsilon_{\text{LS}}[k]}{\sqrt{1 + \beta^2}}, \quad (6.41)$$

y la función de coste asociada resulta

$$J_{\text{TLS}}(\beta) = \frac{J_{\text{LS}}(\beta)}{1 + \beta^2} = \frac{1}{1 + \beta^2} \sum_{k=1}^N (z[k] - \beta g_1(y[k-1]))^2. \quad (6.42)$$

Por último, ya sólo resta derivar $J_{\text{TLS}}(\beta)$ con respecto a β , igualar a cero y resolver la ecuación cuadrática resultante,

$$\beta^2 - \frac{d_1}{d_0}\beta - 1 = 0,$$

cuya solución es [Abramo1965]

$$\hat{\beta}_{\text{TLS}} = \frac{d_1 \pm \sqrt{4d_0^2 + d_1^2}}{2d_0}, \quad (6.43)$$

siendo

$$d_0 = \sum_{k=1}^N z[k]g_1(y[k-1]), \quad (6.44)$$

$$d_1 = \sum_{k=1}^N (z[k]^2 - g_1(y[k-1])^2), \quad (6.45)$$

y $z[k] = y[k] - g_0(y[k-1])$, y estando $g_0(y[k-1])$ y $g_1(y[k-1])$ definidas de acuerdo con la Tabla 6.1 para el TM y el S-TM.

Así pues, en resumen, la obtención del estimador TLS para el TM o el S-TM se puede llevar a cabo de una manera eficiente del siguiente modo:

1. Calcular d_0 y d_1 a partir de las observaciones usando (6.44) y (6.45).
2. Obtener $\hat{\beta}_{\text{TLS}}$ aplicando (6.43) y eligiendo la raíz apropiada, esto es, la que proporcione un $\beta \in [1, 2]$.

En el resto de los casos el estimador TLS se obtiene, con un mayor coste computacional, a partir de la SVD de la matriz del modelo “extendida” $N \times (p + 1)$, $\mathbf{H} = [\mathbf{G}^T, \mathbf{z}_{1:N}]$, aplicando el Algoritmo 12.3.1 de [Golub1996].

6.4.4. Estimación Bloque del SK-TM y BSK-TM: HCLS

Todos los estimadores bloque propuestos se basan en la linealidad con el parámetro de la relación existente entre $x[k]$ y $x[k - 1]$,

$$x[k] = f(x[k - 1]; \theta) = \theta g_1(x[k - 1]) + g_0(x[k - 1]),$$

siendo necesario además disponer de $s[k - 1]$ o poder estimarlo de manera sencilla y fiable a partir de las observaciones en el caso de mapas cuya forma depende de la región del espacio de fases, como los mapas PWL. Desafortunadamente para algunos mapas PWL, como el SK-TM o el BSK-TM, no se cumple ninguna de estas dos premisas:

1. En el caso del SK-TM, $g_1(x) = x$ en ambos intervalos, pero la relación con su parámetro aparece a través de $1/c$ en el primer intervalo y a través de $-1/(1 - c)$ en el segundo, mientras que en el caso del BSK-TM ocurre algo similar: $g_1(x) = x$, y la dependencia con el parámetro es $2/(1 + c)$ en E_1 , y $-2/(1 - c)$ en E_2 .
2. El parámetro c determina la frontera entre las dos regiones de su partición, de modo que si se desconoce no existe ningún modo evidente de obtener una estima de la secuencia simbólica.

Por suerte, tanto para el SK-TM como para el BSK-TM es posible solucionar ambos problemas. El primero de ellos se resuelve simplemente considerando la relación inversa,

$$x[k - 1] = f_{s[k-1]}^{-1}(x[k]; \theta) = \theta h_1(x[k]) + h_0(x[k]),$$

que para el SK-TM resulta

$$\begin{aligned} f_{s[k-1]}^{-1}(x[k]; c) &= \begin{cases} c x[k], & s[k - 1] = 1; \\ c x[k] + 1 - x[k], & s[k - 1] = 2 \end{cases} \\ &= c x[k] + (s[k - 1] - 1)(1 - x[k]), \end{aligned} \quad (6.46)$$

y para el BSK-TM también presenta una expresión similar,

$$\begin{aligned} f_{s[k-1]}^{-1}(x[k]; c) &= \begin{cases} \frac{1+x[k]}{2}c - \frac{1-x[k]}{2}, & s[k - 1] = 1; \\ \frac{1+x[k]}{2}c + \frac{1-x[k]}{2}, & s[k - 1] = 2 \end{cases} \\ &= \frac{1 + x[k]}{2}c + (2s[k - 1] - 3)\frac{1 - x[k]}{2}. \end{aligned} \quad (6.47)$$

A la vista de estas dos ecuaciones resulta evidente que la relación entre $x[k - 1]$ y $x[k]$ se va a poder expresar de un modo semejante a la de la relación directa explotada en las secciones 6.4.1 a 6.4.3, aunque con una diferencia importante: ahora la dependencia con el itinerario aparece en el término independiente de θ , $h_0(x)$.

Respecto al segundo problema, la manera obvia de solucionarlo consiste en probar las $P(N)$ secuencias simbólicas válidas, obtener la estima correspondiente para cada

una de ellas, y seleccionar finalmente la mejor de todas. El crecimiento exponencial del número de itinerarios con la longitud de la secuencia hace que esta solución resulte inviable para valores de N medios/altos. En consecuencia, en [Panta2001a, Luengo2001b] se propuso una solución computacionalmente eficiente, conocida como HCLS (“Hard Censoring Least Squares”), consistente en ordenar las observaciones de acuerdo con su amplitud y dividir las en dos conjuntos continuos de tal modo que aquellas con amplitud menor que un cierto umbral se asume que pertenecen a E_1 , y el resto a E_2 . Situando este umbral sucesivamente entre cada par de muestras ordenadas se obtienen $N + 1$ estimas que se corresponden con otros tantos itinerarios explorados (los de mayor probabilidad), seleccionándose la mejor de todas ellas como estima global.

Aunque el estimador bloque puede ser cualquiera de los anteriores, en lo sucesivo se considera únicamente el LS. En este caso, la función de coste se puede escribir como

$$J_{\hat{s}}(c) = \|\mathbf{d}_{\hat{s}} - \mathbf{h} \cdot c\|_2^2 = (\mathbf{d}_{\hat{s}} - \mathbf{h} \cdot c)^T (\mathbf{d}_{\hat{s}} - \mathbf{h} \cdot c),$$

siendo $\mathbf{d}_{\hat{s}} = [d_{\hat{s}[0]}[0], \dots, d_{\hat{s}[N-1]}[N-1]]^T$, con $d_{\hat{s}[k-1]}[k-1] = y[k-1] - h_0(y[k])$, que en el caso del SK-TM vale

$$d_{\hat{s}[k-1]}[k-1] = y[k-1] - (\hat{s}[k-1] - 1)(1 - y[k]),$$

y en el caso del BSK-TM es

$$d_{\hat{s}[k-1]}[k-1] = 2y[k-1] - (2\hat{s}[k-1] - 3)(1 - y[k]),$$

y $\mathbf{h} = [h[1], \dots, h[N]]^T$ con $h[k] = h_1(y[k])$, que se convierte simplemente en $h[k] = y[k]$ para el SK-TM, y en $h[k] = 1 + y[k]$ para el BSK-TM.

En ambos casos la solución LS para un itinerario dado viene dada por (6.29):

$$\hat{c}_{\text{LS}}^{\hat{s}} = \mathbf{h}^\dagger \mathbf{d}_{\hat{s}} = \frac{\mathbf{h}^T \mathbf{d}_{\hat{s}}}{\mathbf{h}^T \mathbf{h}}, \quad (6.48)$$

donde $\mathbf{h}^\dagger = (\mathbf{h}^T \mathbf{h})^{-1} \mathbf{h}^T = \mathbf{h}^T / (\mathbf{h}^T \mathbf{h})$ no depende del itinerario, y por lo tanto únicamente es necesario calcularlo una vez. Alternativamente, (6.48) se puede escribir como

$$\hat{c}_{\text{LS}}^{\hat{s}} = \frac{\sum_{k=1}^N h[k] d_{\hat{s}[k-1]}[k-1]}{\sum_{k=1}^N h[k]^2}. \quad (6.49)$$

A la vista de (6.49), se aprecia claramente que si se explora el conjunto de $N + 1$ secuencias simbólicas de manera ordenada de tal modo que entre el estimador r -ésimo y el $(r - 1)$ -ésimo únicamente una observación de la secuencia ordenada cambia de región a la que pertenece, entonces se puede obtener el estimador r -ésimo de manera eficiente a partir del $(r - 1)$ -ésimo simplemente sumando y restando un término al numerador y realizando la división por el denominador (que se mantiene constante), como se muestra en el Algoritmo 6.5.

1. Construir el vector $\tilde{\mathbf{y}} = [\tilde{y}[0], \dots, \tilde{y}[N-1]]^T$, que contiene las N primeras observaciones, $\mathbf{y}_{0:N-1}$, en orden creciente de acuerdo con su amplitud. Esto es, suponiendo que no existen dos observaciones que tengan exactamente la misma amplitud, $\tilde{y}[r] = y[k-1]$ ($0 \leq r \leq N-1$, $1 \leq k \leq N$) si y sólo si existen únicamente $r-1$ observaciones cuya amplitud es menor que $y[k-1]$, cumpliéndose además que $\tilde{y}[r] > \tilde{y}[r-1]$ para $r = 1, \dots, N-1$.
2. Construir el vector $\tilde{\mathbf{z}} = [\tilde{z}[0], \dots, \tilde{z}[N-1]]^T$, que contiene las observaciones siguientes a las de $\tilde{\mathbf{y}}$ del conjunto sin ordenar. Es decir, si $\tilde{y}[r] = y[k-1]$, entonces $\tilde{z}[r] = y[k]$.
3. A partir de $\tilde{\mathbf{y}}$ y $\tilde{\mathbf{z}}$, generar los vectores con la señal deseada cuando $s[k-1] = 1$, $\tilde{\mathbf{d}}_1 = [\tilde{d}_1[0], \dots, \tilde{d}_1[N-1]]^T$ con $\tilde{d}_1[r] = \tilde{y}[r] - h_0(\tilde{z}[r]; 1)$, y cuando $s[k-1] = 2$, $\tilde{\mathbf{d}}_2 = [\tilde{d}_2[0], \dots, \tilde{d}_2[N-1]]^T$ con $\tilde{d}_2[r] = \tilde{y}[r] - h_0(\tilde{z}[r]; 2)$, el vector diferencia, $\tilde{\boldsymbol{\delta}} = [\tilde{\delta}[0], \dots, \tilde{\delta}[N-1]]^T = \tilde{\mathbf{d}}_1 - \tilde{\mathbf{d}}_2$, y el vector $\tilde{\mathbf{h}} = [\tilde{h}[0], \dots, \tilde{h}[N-1]]^T$, para el que $\tilde{h}[r] = h_1(\tilde{z}[r])$.
4. Calcular el estimador LS suponiendo que todas las muestras de la señal pertenecen a E_2 , que viene dado por

$$\hat{c}_{\text{LS}}^0 = \frac{\hat{N}_{\text{LS}}^0}{\hat{D}_{\text{LS}}},$$

siendo $\hat{N}_{\text{LS}}^0 = \tilde{\mathbf{h}}^T \tilde{\mathbf{d}}_2$ y $\hat{D}_{\text{LS}} = \tilde{\mathbf{h}}^T \tilde{\mathbf{h}}$, y evaluar su error como

$$J_{\text{LS}}^0(\hat{c}_{\text{LS}}^0) = \|\tilde{\mathbf{E}} \tilde{\mathbf{d}}_2\|_2^2 = \|\tilde{\mathbf{e}}_0\|_2^2,$$

siendo $\tilde{\mathbf{E}} = \mathbf{I} - \tilde{\mathbf{H}} = \mathbf{I} - \tilde{\mathbf{h}}(\tilde{\mathbf{h}}^T \tilde{\mathbf{h}})^{-1} \tilde{\mathbf{h}}^T$ y $\tilde{\mathbf{e}}_0 = \tilde{\mathbf{E}} \tilde{\mathbf{d}}_2$.

5. Para $1 \leq r \leq N$, encontrar el estimador LS correspondiente a suponer que $\tilde{\mathbf{y}}_{0:r-1} \in E_1$ e $\tilde{\mathbf{y}}_{r:N-1} \in E_2$ (cuando $r = N$ todas las muestras pertenecen a E_1), del modo siguiente:

- 5.1. Calcular el estimador LS local r -ésimo a partir del $(r-1)$ -ésimo mediante

$$\hat{c}_{\text{LS}}^r = \frac{\hat{N}_{\text{LS}}^r}{\hat{D}_{\text{LS}}},$$

siendo

$$\hat{N}_{\text{LS}}^r = \hat{N}_{\text{LS}}^{r-1} + \tilde{h}[r-1] \tilde{\delta}[r-1] = \hat{N}_{\text{LS}}^{r-1} + \tilde{h}[r-1] (\tilde{d}_1[r-1] - \tilde{d}_2[r-1]).$$

- 5.2. Obtener su error de nuevo a partir del error de la estima anterior, $J_{\text{LS}}^r = \|\tilde{\mathbf{e}}_r\|_2^2$, con

$$\tilde{\mathbf{e}}_r = \tilde{\mathbf{e}}_{r-1} + \tilde{\mathbf{E}}(:, r) \tilde{\delta}[r-1] = \tilde{\mathbf{e}}_{r-1} + \tilde{\mathbf{E}}(:, r) (\tilde{d}_1[r-1] - \tilde{d}_2[r-1]).$$

6. El estimador HCLS global es aquel de los $N+1$ estimadores locales calculados cuyo error sea menor. Esto es, $\hat{c}_{\text{LS}} = \hat{c}_{\text{LS}}^i$, con

$$i = \arg \min_r J_{\text{LS}}^r(\hat{c}_{\text{LS}}^i), \quad \text{para } r = 0, \dots, N.$$

Algoritmo 6.5: Estimador HCLS para mapas cuyos límites de la partición dependen del parámetro que se desea estimar.

Por último, nótese que, para determinar cual de los $N + 1$ estimadores es el más adecuado, es necesario evaluar el error de cada uno de ellos, que viene dado por

$$J_{\hat{s}}(c) = \|\mathbf{E}\mathbf{d}_{\hat{s}}\|_2^2,$$

donde \mathbf{E} es una matriz de error independiente de la secuencia simbólica, $\mathbf{E} = \mathbf{I} - \mathbf{H}$, siendo \mathbf{I} la matriz identidad $N \times N$, y $\mathbf{H} = \mathbf{h}\mathbf{h}^\dagger = \mathbf{h}(\mathbf{h}^T\mathbf{h})^{-1}\mathbf{h}^T$. Esta evaluación del error del estimador r -ésimo también se puede hacer de un modo eficiente a partir del error $(r - 1)$ -ésimo, como se muestra en el Algoritmo 6.5.

6.5. Algoritmo de Estimación Competitivo

En el caso de mapas con dependencia lineal de sus parámetros, como los de la Tabla 6.1, se puede recurrir a algún algoritmo iterativo y/o adaptativo local (descenso de gradiente, Newton-Raphson, LMS, etc.) para encontrar $\boldsymbol{\theta}$ como alternativa a los métodos bloque descritos en la Sección 6.4. Estos métodos siguen la formulación estándar y consiguen el mismo rendimiento que el correspondiente estimador bloque [Panta2001b, Luengo2001a, Luengo2001b], de modo que no se consideran en lo sucesivo. Desafortunadamente, estos algoritmos convencionales no son aplicables en el caso de mapas cuyos límites entre regiones dependen de los parámetros a estimar, como el SK-TM, el BSK-TM o los mapas de Bernouilli. Sin embargo, en esta sección se desarrolla un estimador competitivo para esta clase de mapas en el que los diferentes intervalos del mapa caótico pugnan por modelar cada par de observaciones, adaptándose los parámetros de acuerdo con el ganador en cada ocasión.

6.5.1. Planteamiento del Problema: Aprendizaje Competitivo

En esta sección se va a explotar un paradigma bien conocido en computación neuronal y que se puede utilizar tanto para plantear algoritmos iterativos como adaptativos: el *aprendizaje competitivo*. La idea básica del aprendizaje competitivo es la siguiente. Se dispone de una colección de *vectores* o *patrones de entrenamiento*, $\mathbf{m}[k]$ con $1 \leq k \leq N$, y de un conjunto de *vectores* o *patrones de referencia*, inicializados aleatoriamente, $\mathbf{v}_i[0]$ con $0 \leq i \leq M - 1$, que se van ajustando de acuerdo con el siguiente procedimiento:

1. En la iteración n -ésima se selecciona un patrón de entrenamiento, $\mathbf{m}[n]$. Si se trata de un problema “on-line”, entonces $\mathbf{m}[n] = \mathbf{m}[k]$, mientras que para problemas bloque (“batch”) habitualmente se escoge $\mathbf{m}[k]$ aleatoriamente entre los N vectores disponibles.
2. Se compara $\mathbf{m}[n]$ con el valor actual de cada uno de los vectores de referencia, $\mathbf{v}_i[n]$, usando una métrica especificada *a priori*.
3. El vencedor de esta competición reduce su distancia (en la métrica de referencia) con respecto al vector de entrenamiento siguiendo un algoritmo adaptativo sencillo, como el LMS por ejemplo.

Si el método está bien diseñado y alcanza la convergencia, en el estado estacionario cada vector de referencia representa a un subconjunto (“cluster”) de los datos de entrenamiento [Haykin1999, Princi2000].

Así pues, el aprendizaje competitivo provoca que cada uno de los vectores de referencia se concentre en un determinado grupo de patrones de entrada con ciertas características comunes. Esta idea se ha extendido al área de modelado en [Panta1994, Panta1996]: varios modelos lineales compiten por el ajuste de los patrones de entrada, concentrándose cada uno de ellos en un grupo de los mismos que presentan algún tipo de semejanza. También es posible aplicar el concepto de competición al planteamiento de estimadores iterativos y/o adaptativos de parámetros de mapas PWL caóticos, como se mostró en [Luengo2001a, Luengo2004a] y se discute en secciones posteriores. En este caso se dispone de un único modelo (el mapa PWL) cuyos múltiples intervalos compiten entre sí por ajustar las observaciones disponibles, adaptándose el subconjunto de parámetros del mapa correspondiente al intervalo vencedor. Por último, nótese que la formulación general del problema presentada en la Sección 6.5.2, y el estimador competitivo desarrollado en la Sección 6.5.3 también se pueden aplicar a otros problemas en procesamiento de señal y comunicaciones, como por ejemplo los problemas ciegos inversos (deconvolución ciega, separación ciega de fuentes e igualación ciega [Luengo2004a]) en los que la señal de entrada es dispersa (“sparse”) [Luengo2005d] o aplicaciones de “clustering” en general.

6.5.2. Formulación General del Problema

La formulación genérica del problema es la siguiente [Luengo2004a]. Se dispone de un conjunto de observaciones contaminadas por AWGN de acuerdo con (3.4),

$$y[k] = x[k] + w[k],$$

con $k = 0, \dots, N$. La señal de interés, $x[k]$, se obtiene mediante una transformación lineal de un vector de parámetros, $\boldsymbol{\theta}$, que se desea estimar, más un posible término adicional que no depende de $\boldsymbol{\theta}$:

$$x[k] = \boldsymbol{\Phi}^T \mathbf{v}[k] = \boldsymbol{\theta}^T \mathbf{u}[k] + v_0[k], \quad (6.50)$$

donde $\boldsymbol{\Phi} = [1, \boldsymbol{\theta}^T]^T$, $\mathbf{v}[k] = [v_0[k], v_1[k], \dots, v_p[k]]^T$, $\mathbf{u}[k] = \mathbf{v}_{1:p}[k] = [v_1[k], \dots, v_p[k]]^T$ y $k = 0, \dots, N$. Cada vector $\mathbf{v}[k]$ se genera a su vez a partir de otro vector q -ario, $\mathbf{s}[k]$, mediante una transformación posiblemente no lineal,

$$\mathbf{v}[k] = \mathbf{T}(\mathbf{s}[k]), \quad (6.51)$$

siendo $\mathbf{s}[k] = [s_1[k], \dots, s_{L_s}[k]]^T$, con $s_i[k] \in \{l_1, \dots, l_q\}$. En consecuencia, aunque el objetivo último sea estimar $\boldsymbol{\theta}$, es preciso determinar previamente el valor de $\mathbf{v}[k]$ (o equivalentemente de $\mathbf{s}[k]$) de entre los hasta $M = q^{L_s}$ vectores posibles para cada observación (como de costumbre, pueden existir vectores $\mathbf{s}[k]$ inválidos).

Nótese que (6.50) es muy parecida a (6.3). De hecho la única diferencia entre ambas es que en (6.3) $\mathbf{g}(x)$ denota un conjunto de $p + 1$ funciones cualquiera, mientras que en (6.50) cada uno de los $M = q^{L_s}$ vectores $\mathbf{v}[k]$ representa uno de los M posibles conjuntos de $p + 1$ funciones disponibles:

$$\mathbf{v}_i[k] = [g_{0,i}(x[k-1]), g_{1,i}(x[k-1]), \dots, g_{p,i}(x[k-1])]^T,$$

con $i = 0, \dots, M - 1$.

A la vista de (6.50) y (6.51), resulta evidente que para cada observación se trata de resolver un problema de *chequeo de múltiples hipótesis compuestas* (“multiple composite hypothesis testing”) [VanTre1968, Kay1998a]: la selección de una de entre M posibles hipótesis cuando la FDP de por lo menos una de ellas no está completamente especificada. En el caso binario (sólo dos hipótesis) esta clase de problemas, también conocidos como problemas de detección y estimación conjunta [Olmo2000] ya que requieren la estimación de unas variables y la detección de otras, se pueden resolver de manera óptima mediante un *cociente de verosimilitud generalizado* (“Generalized Likelihood Ratio Test”, GLRT), que reemplaza aquellos parámetros desconocidos por sus estimas ML y aplica un cociente de verosimilitud estándar para determinar la hipótesis más probable [VanTre1968, Kay1998a]:

$$\hat{\boldsymbol{\theta}}_{\text{ML}} = \begin{cases} \hat{\boldsymbol{\theta}}_{\text{ML}}^0, & p(\mathbf{y}; \hat{\boldsymbol{\theta}}_{\text{ML}}^0, \mathcal{H}_0) \geq p(\mathbf{y}; \hat{\boldsymbol{\theta}}_{\text{ML}}^1, \mathcal{H}_1); \\ \hat{\boldsymbol{\theta}}_{\text{ML}}^1, & p(\mathbf{y}; \hat{\boldsymbol{\theta}}_{\text{ML}}^0, \mathcal{H}_0) < p(\mathbf{y}; \hat{\boldsymbol{\theta}}_{\text{ML}}^1, \mathcal{H}_1); \end{cases}$$

donde $\hat{\boldsymbol{\theta}}_{\text{ML}}^i$ es la estima ML de $\boldsymbol{\theta}$ suponiendo que la hipótesis i -ésima, \mathcal{H}_i , es la correcta, $p(\mathbf{y}; \hat{\boldsymbol{\theta}}_{\text{ML}}^i, \mathcal{H}_i)$ es la función de verosimilitud asociada a \mathcal{H}_i , e $i \in \{0, 1\}$.

Aunque es posible extender el GLRT al caso M -ario fácilmente, su rendimiento puede no ser satisfactorio en un caso general por dos motivos [Kay1998a]:

1. La verosimilitud de una hipótesis, $p(\mathbf{y}; \hat{\boldsymbol{\theta}}_{\text{ML}}^i, \mathcal{H}_i)$, es una función creciente con el número de parámetros de los que depende, lo que introduce un sesgo en el GLRT a favor de las hipótesis más complejas.
2. El GLRT no es capaz de discriminar hipótesis con espacios de parámetros anidados (esto es, hipótesis cuyo conjunto de parámetros sea un subconjunto de los de otra hipótesis), seleccionando siempre aquella hipótesis más general.

El modo habitual de solventar esta dificultad consiste en recurrir a funciones de coste basadas en criterios relacionados con la teoría de la información, que introducen un término que penaliza la complejidad de los modelos. Las dos reglas utilizadas con mayor frecuencia son el *principio de mínima longitud de la descripción* (“Minimum Description Length”, MDL) de Schwartz y Rissanen [Schwar1975, Rissan1978] y el *criterio de la información de Akaike* (“Akaike’s Information Criterion”, AIC) [Akaike1973, Akaike1974].

Sin embargo, en nuestra aplicación todas las hipótesis (intervalos del mapa caótico) presentan la misma complejidad, pudiendo incluso compartir los parámetros, como ocurre por ejemplo en el caso del SK-TM y del BSK-TM, de modo que estas dos limitaciones no son importantes. Suponiendo que los vectores $\mathbf{v}[k]$ son independientes, y dado que se dispone de $N + 1$ observaciones, existen un total de $M^{N+1} = q^{L_s(N+1)}$ combinaciones diferentes de vectores $\mathbf{s}[k]$ (hipótesis). Aplicando la filosofía del GLRT a este problema, este va a elegir la hipótesis r -ésima,

$$r = \arg \max_i p(\mathbf{y}; \hat{\boldsymbol{\theta}}_{\text{ML}}^i, \mathcal{H}_i), \quad (6.52)$$

con $i = 0, \dots, M^{N+1} - 1$. De acuerdo con el modelo de señal dado por (6.50) y (6.51), y puesto que el ruido es blanco y Gaussiano, resulta evidente que cada una de las FDPs de (6.52) es también Gaussiana,

$$p(\mathbf{y}; \boldsymbol{\theta}, \mathcal{H}_i) = \frac{1}{(2\pi\sigma^2)^{(N+1)/2}} \exp \left(-\frac{1}{2\sigma^2} \sum_{k=0}^N (y[k] - \boldsymbol{\Phi}^T \mathbf{v}_i[k])^2 \right), \quad (6.53)$$

siendo $\mathbf{v}_i[k]$ el vector de referencia para la hipótesis i -ésima en el instante k -ésimo. Obviamente, (6.52) se puede reformular de tal forma que r se obtenga alternativamente minimizando la función de coste cuadrática habitual:

$$r = \arg \min_i \{ -\ln p(\mathbf{y}; \hat{\boldsymbol{\theta}}_{\text{ML}}^i, \mathcal{H}_i) \} = \arg \min_i J(\hat{\boldsymbol{\theta}}_{\text{ML}}^i; \mathcal{H}_i), \quad (6.54)$$

con

$$J(\boldsymbol{\theta}; \mathcal{H}_i) = \sum_{k=0}^N (y[k] - v_0[k] - \boldsymbol{\theta}^T \mathbf{v}_{1:p}^i[k])^2 = \sum_{k=0}^N (y[k] - \boldsymbol{\Phi}^T \mathbf{v}_i[k])^2. \quad (6.55)$$

Utilizando esta formulación, el procedimiento para estimar $\boldsymbol{\theta}$ mediante el GLRT sería el siguiente:

1. Para cada una de las M^{N+1} hipótesis obtener $\hat{\boldsymbol{\theta}}_{\text{ML}}^i$ mediante la pseudo inversa,

$$\hat{\boldsymbol{\theta}}_{\text{ML}}^i = \mathbf{U}_i^\dagger (\mathbf{y} - \mathbf{v}_i^0) = (\mathbf{U}_i \mathbf{U}_i^T)^{-1} \mathbf{U}_i (\mathbf{y} - \mathbf{v}_i^0), \quad (6.56)$$

donde $\mathbf{v}_i^0 = [v_0^i[0], \dots, v_0^i[N]]^T$ es el vector $(N+1) \times 1$ con los términos de “offset” para cada observación, y $\mathbf{U}_i = [\mathbf{u}_i[0], \dots, \mathbf{u}_i[N]]^T$ es una matriz $p \times (N+1)$ con la parte de todos los vectores de referencia que multiplica a $\boldsymbol{\theta}$.

2. Evaluar (6.55) para cada hipótesis y seleccionar como estima ML final aquella que proporcione un menor valor.

Aunque esta metodología es perfectamente válida, pone de manifiesto una tercera limitación crucial del GLRT para el problema estudiado: el número de hipótesis crece

exponencialmente con la longitud de la secuencia. En consecuencia, una vez más no resulta factible la aplicación directa de este método (nótese que en el caso de la estimación de la secuencia también se ha estado aplicando el GLRT de un modo implícito), por lo que se recurre a un algoritmo de gradiente estocástico de baja complejidad [Luengo2001a, Luengo2004a] cuya formulación general se desarrolla en la Sección 6.5.3.

6.5.3. Formulación General del Estimador Competitivo

Para desarrollar el algoritmo competitivo de gradiente estocástico, en primer lugar se va a replantear el problema considerado. En lugar de abordarlo como un test de múltiples hipótesis compuestas con M^{N+1} hipótesis diferentes se va a analizar como un problema de minimización funcional en el que para cada instante de tiempo se desea encontrar el vector de referencia óptimo. Desde este punto de vista la función de coste que se debe minimizar es

$$J_d(\boldsymbol{\theta}) = \sum_{k=0}^N (y[k] - \boldsymbol{\Phi}^T \mathbf{v}_{c_k}[k])^2 = \sum_{k=0}^N \varepsilon_{c_k}[k]^2, \quad (6.57)$$

donde $c_k \in \{0, 1, \dots, M-1\}$ indica el mejor patrón de referencia (hipótesis) posible para la observación k -ésima, $\mathbf{v}_{c_k}[k]$, de entre los M disponibles:

$$c_k = \arg \min_i \varepsilon_i[k]^2, \quad (6.58)$$

siendo

$$\varepsilon_i[k] = y[k] - \boldsymbol{\Phi}^T \mathbf{v}_i[k] \quad (6.59)$$

el error de aproximación usando el vector de referencia i -ésimo, con $i = 0, \dots, M-1$. Desafortunadamente la mejor hipótesis para cada observación, \mathcal{H}_{c_k} , es una función discontinua del vector de parámetros, $\boldsymbol{\theta}$, y del conjunto de todos los posibles patrones de referencia para la muestra k -ésima, $\mathcal{V}_k = \{\mathbf{v}_0[k], \dots, \mathbf{v}_{M-1}[k]\}$. En consecuencia, no es posible llevar a cabo una minimización directa de (6.57) mediante métodos bloque o iterativos estándar con un bajo coste computacional. Es decir, el único modo de garantizar que la solución obtenida es la óptima consiste en probar las M^{N+1} hipótesis posibles y seleccionar la mejor de todas ellas.

No obstante, extendiendo la idea de [Kohon1991], es posible resolver este problema mediante un algoritmo iterativo/adaptativo de bajo coste computacional, recurriendo a la técnica clásica conocida como *aproximación estocástica* [Hyvari2001]. La idea consiste en usar estimas instantáneas del error de aproximación y plantear un algoritmo iterativo/adaptativo basado en el descenso de gradiente similar al algoritmo LMS:

$$\begin{aligned} \hat{\boldsymbol{\theta}}^n &= \hat{\boldsymbol{\theta}}^{n-1} - \frac{1}{2} \mu^n \nabla_{\boldsymbol{\theta}} \varepsilon_c^n[k]^2 \\ &= \hat{\boldsymbol{\theta}}^{n-1} + \mu^n (y[k] - \mathbf{v}_c^n[k]^T \hat{\boldsymbol{\Phi}}^{n-1}) \mathbf{u}_c^n[k] \\ &= \hat{\boldsymbol{\theta}}^{n-1} + \mu^n \varepsilon_c^n[k] \mathbf{u}_c^n[k], \end{aligned} \quad (6.60)$$

donde se ha sustituido c_k por c para simplificar la notación. Comparando (6.60) con el LMS y otros algoritmos iterativos/adaptativos estándar, la principal diferencia se encuentra en que, puesto que se desconoce el intervalo real al que pertenece cada observación, previamente a la actualización de $\hat{\boldsymbol{\theta}}$ existe una competición entre los distintos modelos (intervalos) y $\hat{\boldsymbol{\theta}}$ se ajusta tratando de minimizar la distancia entre la observación k -ésima, $y[k]$, y la predicción realizada por la hipótesis ganadora con la estima anterior de $\boldsymbol{\theta}$, $\mathbf{v}_c^n[k]^T \hat{\boldsymbol{\Phi}}^{n-1}$.

Para esta clase de técnicas resulta fundamental realizar un análisis de su convergencia, que en este caso resulta muy complicado debido a que $\varepsilon_c^n[k]$ no es una función continua ni de $\hat{\boldsymbol{\theta}}$ ni de las hipótesis. No obstante, a pesar de su discontinuidad $\varepsilon_c^n[k]$ es una función bien definida y única en todos los puntos excepto quizás en aquellos en los que $\varepsilon_i^n[k] = \varepsilon_j^n[k]$ para algún $i \neq j$. Afortunadamente, asumiendo una FDP continua para $y[k-1]$, $y[k]$ y $\boldsymbol{\theta}$, la probabilidad de que $\varepsilon_c^n[k]$ se encuentre en alguno de dichos puntos de discontinuidad es nula, de modo que en general (6.60) es también una función bien definida cuya convergencia se va a poder analizar. Para llevar a cabo este análisis, nótese que (6.57) puede considerarse una aproximación determinista, basada en la media muestral, de la siguiente función de coste estocástica,

$$J_e(\boldsymbol{\theta}) = \mathbf{E}((y[k] - \boldsymbol{\Phi}^T \mathbf{v}_c^n[k])^2) = \mathbf{E}(\varepsilon_c^n[k]^2). \quad (6.61)$$

Nuevamente (6.61) es una función discontinua de $y[k]$ y $\mathbf{v}_c[k]$, pero en esta ocasión, siguiendo un razonamiento paralelo al de [Kohon1991], en el Apéndice G se demuestra que su gradiente exacto con respecto a $\boldsymbol{\theta}$ es

$$\nabla_{\boldsymbol{\theta}}(J_e(\boldsymbol{\theta})) = -2\mathbf{E}((y[k] - \boldsymbol{\Phi}^T \mathbf{v}_c^n[k])\mathbf{u}_c^n[k]) = -2\mathbf{E}(\varepsilon_c^n[k]\mathbf{u}_c^n[k]). \quad (6.62)$$

Esta ecuación es fundamental, ya que implica que la dirección de mayor pendiente de (6.61) con respecto a $\boldsymbol{\theta}$ (dada por su gradiente) coincide con la de la mejor hipótesis (esto es, el vector ganador). Por consiguiente, la relación entre (6.60) y el algoritmo de descenso de gradiente exacto para (6.61),

$$\begin{aligned} \hat{\boldsymbol{\theta}}^n &= \hat{\boldsymbol{\theta}}^{n-1} + \mu^n \mathbf{E}((y[k] - \mathbf{v}_c^n[k]^T \hat{\boldsymbol{\Phi}}^{n-1})\mathbf{u}_c^n[k]) \\ &= \hat{\boldsymbol{\theta}}^{n-1} + \mu^n \mathbf{E}(\varepsilon_c^n[k]\mathbf{u}_c^n[k]), \end{aligned} \quad (6.63)$$

es la misma que para el LMS: (6.60) es una aproximación estocástica de (6.63) obtenida sustituyendo la esperanza matemática de (6.63) por su estima muestral instantánea. En consecuencia, este estimador, mostrado en el Algoritmo 6.6, se ha denominado LMS competitivo (“Competitive” LMS, CLMS) [Luengo2001a, Luengo2004a].

Hasta ahora se ha demostrado que el CLMS actualiza $\boldsymbol{\theta}$ siguiendo la dirección del mínimo de la función de coste, pero no se ha considerado el valor de la tasa de aprendizaje, μ^n . Para un algoritmo de gradiente estocástico como el CLMS se puede demostrar que (6.60) converge a un mínimo de (6.61) cuando n tiende a infinito si μ^n

es una secuencia decreciente que cumple dos condiciones [Robbin1951, Kushne1978]:

$$\sum_{n=1}^{\infty} \mu^n = \infty, \quad (6.64)$$

$$\sum_{n=1}^{\infty} (\mu^n)^2 < \infty. \quad (6.65)$$

Un ejemplo de regla de adaptación de μ que satisface tanto (6.64) como (6.65) es [Hyvari2001]

$$\mu^n = \frac{C}{C + n - 1} \mu^1, \quad (6.66)$$

donde C es una constante cuyo valor determina la velocidad de decrecimiento de μ^n : cuanto mayor sea C más lentamente disminuye el valor de μ^n . Sin embargo, en la práctica con frecuencia se utiliza un valor de μ^n constante ($C = \infty$ en (6.66)) para problemas “on-line” (imponiéndose las mismas restricciones a μ que para el LMS), o se decreta ligeramente su valor al principio y se mantiene fijo posteriormente con el fin de permitir una rápida adaptación del algoritmo [Hyvari2001].

1. Inicializar el parámetro de actualización, μ^1 , a un valor prefijado, y el vector θ a un valor aleatorio, $\hat{\theta}_0$, dentro del rango de parámetros admisible.
2. Para $1 \leq n \leq N_i$, siendo N_i un número de iteraciones prefijado (en problemas “on-line” $N_i = N$) o el número de iteraciones necesario para que el algoritmo converja:
 - 2.1. Elegir una observación: para el caso “on-line”, $y[k] = y[n]$, mientras que para el caso “batch” $y[k]$ se escoge aleatoriamente entre las N disponibles ($1 \leq k \leq N$).
 - 2.2. Construir el conjunto de los M posibles patrones de referencia para $y[k]$, $\{\mathbf{v}_0^n[k], \dots, \mathbf{v}_{M-1}^n[k]\}$, a partir del conjunto de M posibles vectores q -arios, $\{\mathbf{s}_0^n[k], \dots, \mathbf{s}_{M-1}^n[k]\}$.
 - 2.3. Calcular el error de aproximación para cada patrón mediante (6.59), y utilizar el que proporcione un menor error para obtener $\hat{\theta}^n$ mediante (6.60).
 - 2.4. Actualizar μ , esto es, obtener $\mu^{n+1} = f(\mu^n)$ usando una regla preestablecida.
3. La estima final del parámetro buscada es $\hat{\theta}^{N_i}$.

Algoritmo 6.6: Algoritmo LMS Competitivo (CLMS) para un problema genérico tanto “batch” como “on-line”.

6.5.4. Algoritmo Competitivo para Mapas Caóticos: SK-TM, BSK-TM y Mapas de Bernouilli

Para ilustrar la aplicación del CLMS en la estimación de parámetros de señales caóticas se van a considerar tres mapas cuya partición natural depende de sus paráme-

tros: el SK-TM, el BSK-TM, y los mapas de Bernouilli. En los tres casos la iteración hacia delante no presenta una dependencia lineal con los parámetros, de modo que se utiliza la iteración hacia atrás, al igual que en la Sección 6.4.4, denominándose al algoritmo resultante “Backward” CLMS (BCLMS). A efectos de comparación se va a considerar también la aplicación del algoritmo CLMS con el error de aproximación y los patrones de referencia de la iteración adelante (“Forward” CLMS, FCLMS), aunque en esta ocasión los resultados obtenidos van a ser siempre peores.

En primer lugar se va a desarrollar el algoritmo CLMS para el SK-TM y BSK-TM, en cuyo caso sólo hay dos patrones de referencia válidos y un parámetro a estimar. Las expresiones para la iteración hacia atrás del SK-TM y del BSK-TM vienen dadas por (6.46) y (6.47) respectivamente, de donde se obtienen fácilmente los dos posibles vectores de referencia para cada uno: $\mathbf{v}_0[k] = [0, y[k]]^T$ y $\mathbf{v}_1[k] = [1 - y[k], y[k]]^T$ para el SK-TM, mientras que $\mathbf{v}_0[k] = [-(1 - y[k])/2, (1 + y[k])/2]^T$ y $\mathbf{v}_1[k] = [(1 - y[k])/2, (1 + y[k])/2]^T$ para el BSK-TM. Nótese que en este caso (iteración hacia atrás), se cumple que

$$\mathbf{v}_i^n[k] = \mathbf{h}(y[k]; s[k - 1] = i),$$

estando \mathbf{h} definida como en la Sección 6.4.4 para $s[k - 1] = i$, con $i = 1, \dots, M$.

Por otra parte, el CLMS viene dado siempre por (6.60), siendo el menor error de aproximación ahora $\varepsilon_c^n[k] = \min(|\varepsilon_1^n[k]|, |\varepsilon_2^n[k]|)$, el vector ganador en cada iteración

$$\mathbf{v}_c^n[k] = \begin{cases} \mathbf{v}_1^n[k], & |\varepsilon_1^n[k]| \leq |\varepsilon_2^n[k]|; \\ \mathbf{v}_2^n[k], & |\varepsilon_1^n[k]| > |\varepsilon_2^n[k]|; \end{cases}$$

y los errores asociados a cada patrón

$$\begin{aligned} \varepsilon_1^n[k] &= y[k - 1] - \hat{c}^{n-1}y[k], \\ \varepsilon_2^n[k] &= y[k - 1] - (1 - (1 - \hat{c}^{n-1})y[k]), \end{aligned}$$

para el SK-TM, y para el BSK-TM,

$$\begin{aligned} \varepsilon_1^n[k] &= y[k - 1] - \left(\frac{(1 + \hat{c}^{n-1})(1 + y[k])}{2} - 1 \right), \\ \varepsilon_2^n[k] &= y[k - 1] - \left(1 - \frac{(1 - \hat{c}^{n-1})(1 + y[k])}{2} \right). \end{aligned}$$

Para un mapa de Bernouilli la situación se complica ligeramente, ya que existen múltiples intervalos. Considerando un mapa lo más general posible, dado por (2.15), su iteración hacia atrás es

$$f_{s[k-1]}^{-1}(x[k]; e_0, \dots, e_M) = \frac{x[k] - e_0}{e_M - e_0} e_{s[k-1]} + \frac{e_M - x[k]}{e_M - e_0} e_{s[k-1]-1}, \quad (6.67)$$

de modo que, asumiendo que el rango de la señal (esto es, los valores de e_0 y e_M) es conocido o ha sido estimado previamente, en la relación entre cada par de observaciones

intervienen como mucho dos parámetros del mapa. En consecuencia, los vectores de referencia tienen dimensión $M \times 1$ (incluyendo el término de “offset”), pudiéndose obtener para un mapa de Bernoulli genérico del modo siguiente:

$$\mathbf{v}_i^n[k] = \begin{bmatrix} v_0^{i,n}[k] \\ v_1^{i,n}[k] \\ v_2^{i,n}[k] \\ \vdots \\ v_{M-2}^{i,n}[k] \\ v_{M-1}^{i,n}[k] \end{bmatrix} = \begin{bmatrix} e_M \delta[i-M] & e_0 \delta[i-1] \\ \delta[i-1] & \delta[i-2] \\ \delta[i-2] & \delta[i-3] \\ \vdots & \vdots \\ \delta[i-(M-2)] & \delta[i-(M-1)] \\ \delta[i-(M-1)] & \delta[i-M] \end{bmatrix} \cdot \begin{bmatrix} \frac{y[k]-e_0}{e_M-e_0} \\ \frac{e_M-y[k]}{e_M-e_0} \end{bmatrix}, \quad (6.68)$$

donde $v_m^{i,n}[k]$ indica el componente m -ésimo de $\mathbf{v}_i^n[k]$, y $\delta[\cdot]$ es la función impulso o delta de Dirac.

Analizando (6.68) se observa que nunca van a existir más de dos elementos no nulos. En consecuencia, las ecuaciones de actualización de los parámetros en función del vector ganador se pueden escribir como sigue:

1. Si el patrón ganador es el primero (es decir, si $c = 1$),

$$\hat{e}_1^n = \hat{e}_1^{n-1} + \mu^n \left(y[k-1] - \frac{e_M - y[k]}{e_M - e_0} e_0 - \frac{y[k] - e_0}{e_M - e_0} \hat{e}_1^{n-1} \right) \frac{y[k] - e_0}{e_M - e_0}.$$

2. Si el patrón ganador es el último (esto es, $c = M$),

$$\hat{e}_{M-1}^n = \hat{e}_{M-1}^{n-1} + \mu^n \left(y[k-1] - \frac{y[k] - e_0}{e_M - e_0} e_M - \frac{e_M - y[k]}{e_M - e_0} \hat{e}_{M-1}^{n-1} \right) \frac{e_M - y[k]}{e_M - e_0}.$$

3. Finalmente, si el ganador es el intervalo intermedio i -ésimo, $2 \leq i \leq M-1$,

$$\begin{aligned} \hat{e}_{i-1}^n &= \hat{e}_{i-1}^{n-1} + \mu^n \left(y[k-1] - \frac{e_M - y[k]}{e_M - e_0} \hat{e}_{i-1}^{n-1} - \frac{y[k] - e_0}{e_M - e_0} \hat{e}_i^{n-1} \right) \frac{e_M - y[k]}{e_M - e_0}, \\ \hat{e}_i^n &= \hat{e}_i^{n-1} + \mu^n \left(y[k-1] - \frac{e_M - y[k]}{e_M - e_0} \hat{e}_{i-1}^{n-1} - \frac{y[k] - e_0}{e_M - e_0} \hat{e}_i^{n-1} \right) \frac{y[k] - e_0}{e_M - e_0}. \end{aligned}$$

Por último, como ya se ha mencionado anteriormente, también se va a considerar el CLMS obtenido a partir de la iteración hacia delante (FCLMS). En este caso el error de aproximación hacia delante para el intervalo i -ésimo y un mapa caótico cualquiera se define como

$$\varepsilon_i^n[k] = y[k] - f_i(y[k-1]; \hat{\boldsymbol{\theta}}^{n-1}), \quad (6.69)$$

de modo que, desarrollando su gradiente para el intervalo ganador, $i = c$, se puede llegar fácilmente a una ecuación muy similar a (6.60):

$$\begin{aligned} \hat{\boldsymbol{\theta}}^n &= \hat{\boldsymbol{\theta}}^{n-1} - \frac{1}{2} \mu^n \nabla_{\boldsymbol{\theta}} \varepsilon_c^n[k]^2 \\ &= \hat{\boldsymbol{\theta}}^{n-1} + \mu^n (y[k] - f_c(y[k-1]; \hat{\boldsymbol{\theta}}^{n-1})) \nabla_{\boldsymbol{\theta}} f_c(y[k-1]; \boldsymbol{\theta}) \Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}^{n-1}} \\ &= \hat{\boldsymbol{\theta}}^{n-1} + \mu^n \varepsilon_c^n[k] \nabla_{\boldsymbol{\theta}} f_c(y[k-1]; \boldsymbol{\theta}) \Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}^{n-1}}. \end{aligned} \quad (6.70)$$

Obviamente $\nabla_{\boldsymbol{\theta}} f_c(y[k-1]; \boldsymbol{\theta})|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}^{n-1}}$ desempeña el mismo papel en (6.70) que $\mathbf{u}_c^n[k]$ en (6.60), aunque con dos importantes diferencias debido a la dependencia no lineal de $f_c(y[k-1]; \boldsymbol{\theta})$ con $\boldsymbol{\theta}$:

1. La forma del gradiente va a ser específica para cada mapa, y va a depender en general de los parámetros a estimar.
2. La demostración del Apéndice G no es válida para esta clase de mapas, de modo que no se puede garantizar su convergencia desde un punto de vista teórico, aunque sí se ha comprobado experimentalmente la convergencia del algoritmo competitivo mediante simulaciones.

En la práctica se van a considerar los mismos tres mapas que para la iteración hacia atrás: el SK-TM, el BSK-TM y los mapas de Bernouilli. Definiendo los vectores de referencia en esta ocasión como

$$\mathbf{u}_i^n[k] = \nabla_{\boldsymbol{\theta}} f_i(y[k-1]; \boldsymbol{\theta})|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}^{n-1}} = \mathbf{g}(y[k-1]; s[k-1] = i), \quad (6.71)$$

con $1 \leq i \leq M$ y \mathbf{g} definido como en (6.3), la ecuación para el FCLMS es idéntica a la del BCLMS:

$$\hat{\boldsymbol{\theta}}^n = \hat{\boldsymbol{\theta}}^{n-1} + \mu^n \varepsilon_c^n[k] \mathbf{u}_c^n[k],$$

donde $\varepsilon_c^n[k]$ y $\mathbf{u}_c^n[k]$ vienen dados respectivamente por (6.69) y (6.71) para $i = c$. En las tablas 6.3 y 6.4 se muestran los posibles valores del error y los patrones de referencia para el SK-TM, el BSK-TM y los mapas de Bernouilli. Nótese que en este caso no existe término de “offset” para el SK-TM ni para el BSK-TM, de modo que los vectores de referencia tienen dimensión uno. En cuanto a los mapas de Bernouilli, al igual que ocurría con el BCLMS, en cada iteración únicamente se actualizan 1 ó 2 parámetros, en función de cual sea el patrón ganador. Por último, también debe resaltarse que en todos los casos el valor actual de los parámetros a estimar aparece como un divisor en las ecuaciones de $\mathbf{v}_c^n[k]$ y $\varepsilon_c^n[k]$, lo que da lugar a problemas numéricos que afectan al rendimiento del algoritmo, como se muestra en la Sección 6.6.

6.6. Resultados

En esta sección se presentan los principales resultados obtenidos para la estimación de los parámetros de diversos mapas caóticos unidimensionales. En primer lugar, en la Figura 6.4 se muestra un ejemplo del rendimiento de los estimadores HC-ML y ML aproximados para el S-TM. El estimador HC-ML se obtiene usando los algoritmos 6.4 y 6.2, mientras que la aproximación al estimador ML se lleva a cabo mediante un algoritmo de búsqueda en una rejilla dividido en dos fases:

1. Búsqueda inicial del mínimo usando una rejilla con 1.000 puntos distribuidos uniformemente a lo largo del espacio de parámetros, lo que evita el elevado coste computacional derivado de encontrar los límites de las regiones.

Mapa	$\varepsilon_i^n[k]$
SK-TM	$\varepsilon_1^n[k] = y[k] - \frac{y[k-1]}{\hat{c}^{n-1}}, \quad \varepsilon_2^n[k] = y[k] - \frac{1-y[k-1]}{1-\hat{c}^{n-1}}.$
BSK-TM	$\varepsilon_1^n[k] = y[k] - \left(2\frac{1+y[k-1]}{1+\hat{c}^{n-1}} - 1\right), \quad \varepsilon_2^n[k] = y[k] - \left(2\frac{1-y[k-1]}{1-\hat{c}^{n-1}} - 1\right).$
Bernouilli	$\varepsilon_i^n[k] = \begin{cases} y[k] - e_0 - \frac{(e_M - e_0)(y[k-1] - e_0)}{\hat{e}_1^{n-1} - e_0}, & i = 1; \\ y[k] - e_0 - \frac{(e_M - e_0)(y[k-1] - \hat{e}_{i-1}^{n-1})}{\hat{e}_i^{n-1} - \hat{e}_{i-1}^{n-1}}, & 2 \leq i \leq M - 1; \\ y[k] - e_0 - \frac{(e_M - e_0)(y[k-1] - \hat{e}_{M-1}^{n-1})}{e_M - \hat{e}_{M-1}^{n-1}}, & i = M. \end{cases}$

Tabla 6.3: Errores de aproximación para el algoritmo FCLMS del SK-TM, del BSK-TM y de un mapa de Bernouilli genérico.

Mapa	$\mathbf{u}_i^n[k]$
SK-TM	$u_1^n[k] = -\frac{y[k-1]}{(\hat{c}^{n-1})^2}, \quad u_2^n[k] = \frac{1-y[k-1]}{(1-\hat{c}^{n-1})^2}.$
BSK-TM	$u_1^n[k] = -2\frac{1+y[k-1]}{(1+\hat{c}^{n-1})^2}, \quad u_2^n[k] = 2\frac{1-y[k-1]}{(1-\hat{c}^{n-1})^2}.$
Bernouilli	$\mathbf{u}_1^n[k] = \left[\mathbf{0}_{M-2}^T, -\frac{(e_M - e_0)(y[k-1] - e_0)}{(\hat{e}_1^{n-1} - e_0)^2} \right]^T.$ $\mathbf{u}_i^n[k] = \left[\mathbf{0}_{i-2}^T, -\frac{(e_M - e_0)(\hat{e}_i^{n-1} - y[k-1])}{(\hat{e}_i^{n-1} - \hat{e}_{i-1}^{n-1})^2}, -\frac{(e_M - e_0)(y[k-1] - \hat{e}_{i-1}^{n-1})}{(\hat{e}_i^{n-1} - \hat{e}_{i-1}^{n-1})^2}, \mathbf{0}_{M-i-1}^T \right]^T,$ $2 \leq i \leq M - 1.$ $\mathbf{u}_M^n[k] = \left[-\frac{(e_M - e_0)(e_M - y[k-1])}{(e_M - \hat{e}_{M-1}^{n-1})^2}, \mathbf{0}_{M-2}^T \right]^T.$

Tabla 6.4: Vectores de referencia para el algoritmo FCLMS del SK-TM, del BSK-TM y de un mapa de Bernouilli genérico.

2. Refinamiento de la búsqueda usando una segunda rejilla más fina alrededor del mínimo de la función de coste encontrado mediante la primera rejilla.

Como puede apreciarse, el MSE del HC-ML se encuentra muy cercano al del ML, alcanzando ambos el CRLB a partir de una relación señal a ruido de 20 dB aproximadamente, y superando ampliamente el rendimiento del estimador bloque basado en el TLS, que no alcanza el CRLB, pero cuyo coste computacional es muy inferior tanto al del ML como al del HC-ML. El comportamiento de estos tres estimadores para otros ejemplos (no mostrados) del TM y S-TM con distintos valores de $x[0]$ y β es similar, aunque con la variabilidad característica de las señales caóticas ya mostrada en los capítulos anteriores: la ganancia del HC-ML con respecto al TLS y el punto exacto en que alcanza el CRLB dependen en gran medida de la condición inicial, el valor del parámetro de bifurcación, y la longitud de la secuencia caótica.

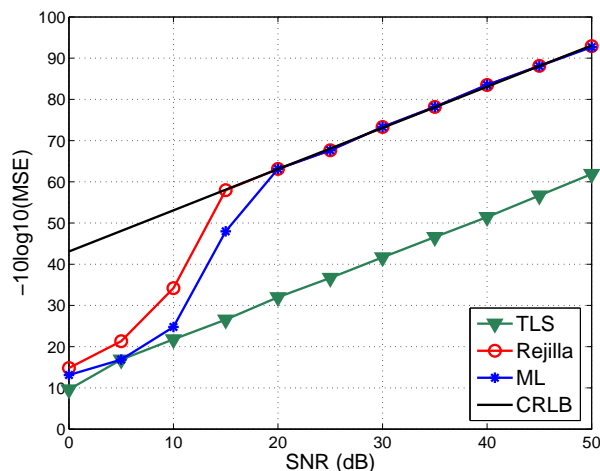


Figura 6.4: Comparación del rendimiento del estimador HC-ML con el de búsqueda en una rejilla (ML aproximado) para el S-TM con $\beta = 1,45$, $x[0] = -0,61$ y $N = 10$.

Además, al igual que ocurría con los estimadores óptimos y subóptimos de la secuencia, los estimadores ML aproximado y HC-ML del parámetro de bifurcación son inconsistentes. Como prueba de esta afirmación, en la Figura 6.5 se muestra un ejemplo del MSE del HC-ML para el S-TM con $\beta = 1,9$, $x[0] = -0,63$ y tres valores diferentes de SNR. Puede observarse claramente cómo el MSE del estimador alcanza un punto de saturación a partir de un cierto valor de N que depende de la SNR ($N = 5$ para una SNR = 12 dB, $N = 9$ para una SNR = 15 dB, y $N = 11$ para una SNR = 18 dB aproximadamente).

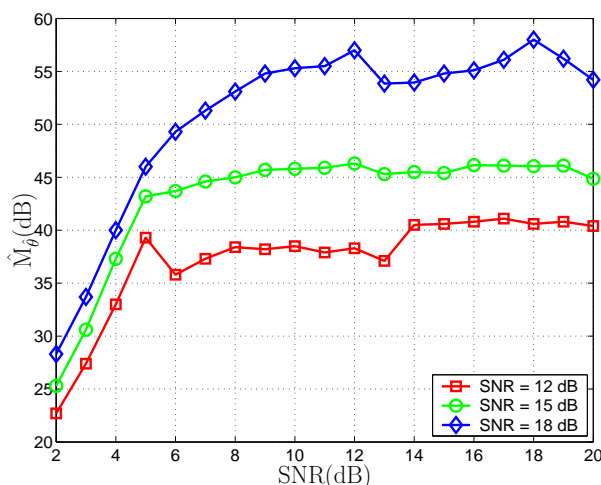


Figura 6.5: MSE del estimador HC-ML en función de la longitud de la secuencia y distintos valores de SNR para el S-TM con $\beta = 1,9$ y $x[0] = -0,63$.

A continuación se estudia el comportamiento de los diversos estimadores bloque propuestos en la Sección 6.4, cuyo rendimiento es menor que el de los estimadores ML y HC-ML, pero cuyo coste computacional es también muy inferior. En la Figura 6.6 se muestra el MSE de $\hat{\theta}$ para cuatro de los seis mapas de la Tabla 6.1 (mapas logístico, de Singer, del seno y exponencial) usando los tres estimadores bloque de la Sección 6.4 (MBE, LS y TLS) para secuencias “largas” ($N = 100$). Excepto en el caso del mapa exponencial, para el cual el MBE con $r = 1$ presenta un comportamiento muy malo, en el resto de los casos todos los estimadores proporcionan un MSE similar, especialmente para valores de SNR medios/altos. El MBE con $r = 2$ presenta un rendimiento ligeramente superior al resto, y en esta ocasión el TLS no parece que proporcione una ventaja significativa con respecto al LS.

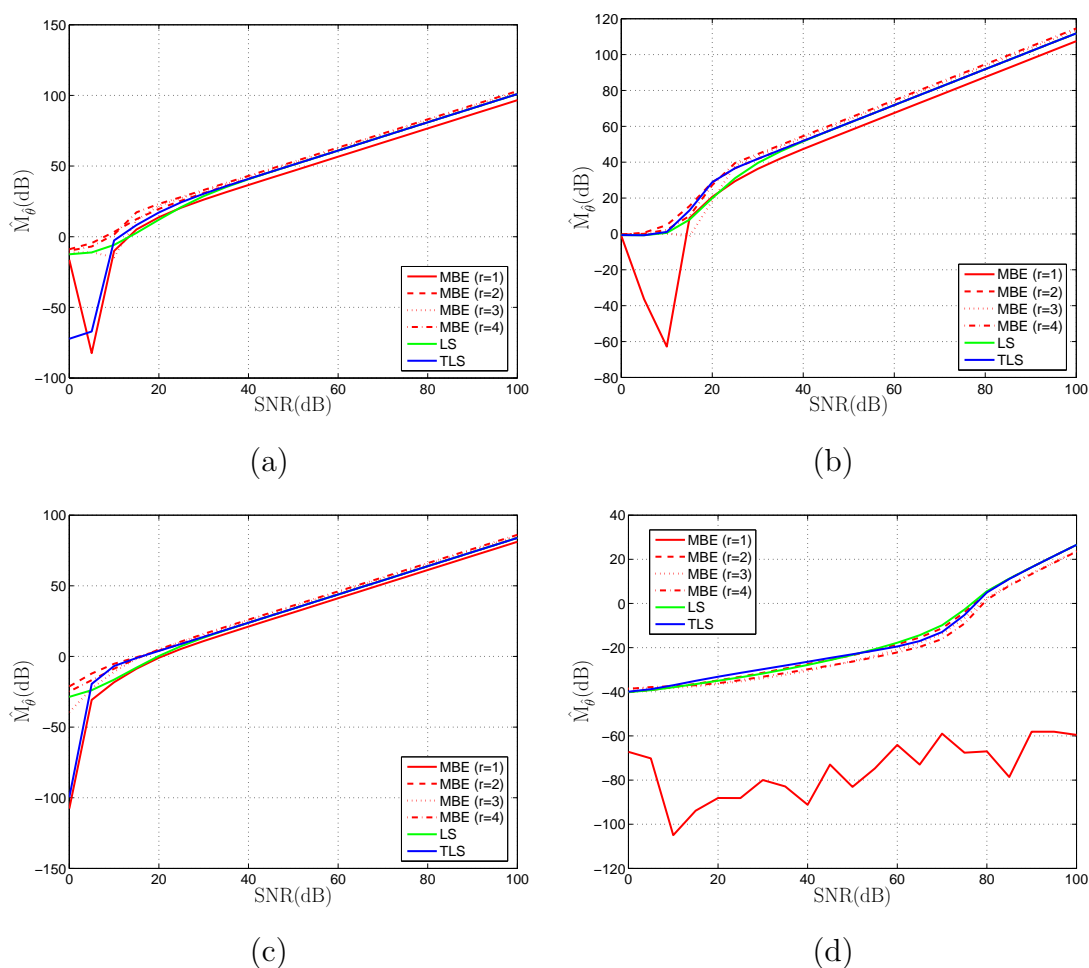


Figura 6.6: Comparación del rendimiento de los estimadores MBE ($r = 1, 2, 3$ y 4), LS y TLS para secuencias generadas por distintos mapas caóticos unimodales con $N = 100$ y 1.000 condiciones iniciales aleatorias. (a) Mapa Logístico con $\lambda = 4$. (b) Mapa de Singer con $c = 1,07$. (c) Mapa del seno con $c = 0,95$. (d) Mapa exponencial con $c = 3,5$.

El resto de esta sección se dedica a analizar el comportamiento del principal algoritmo propuesto en este capítulo: el CLMS. En primer lugar, en la Figura 6.7 se muestra la convergencia del BCLMS tanto para el SK-TM como para el BSK-TM: en un caso sin ruido, con un parámetro de adaptación constante ($\mu = 0,01$) y una inicialización aleatoria del parámetro de bifurcación, las estimas dadas por el BCLMS siempre convergen al valor real de c .

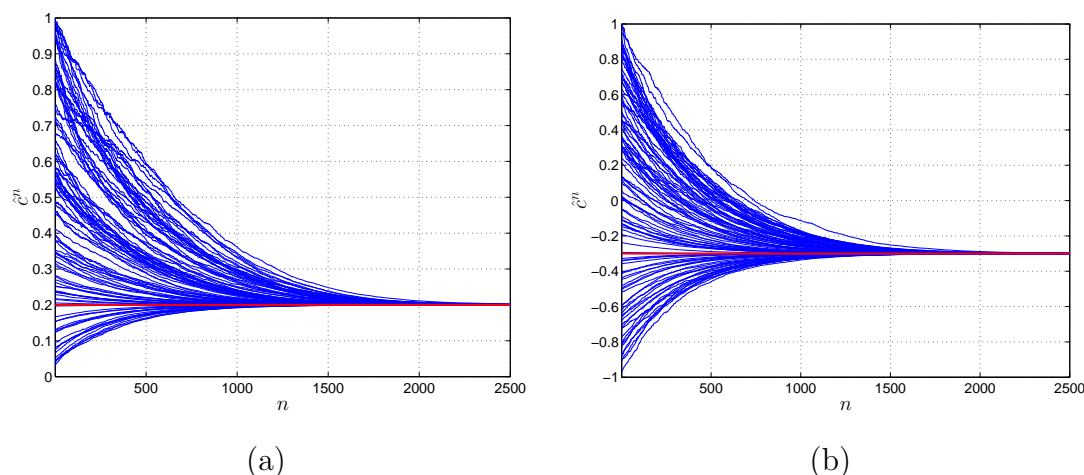


Figura 6.7: Convergencia del algoritmo BCLMS con una inicialización aleatoria, $N = 100$, $\mu = 0,01$ y $C = \infty$. (a) SK-TM con $c = 0,2$. (b) BSK-TM con $c = -0,3$.

Esta ausencia de mínimos locales y convergencia garantizada para un valor de μ suficientemente bajo del BCLMS contrasta con la divergencia ocasional observada en el caso del FCLMS, que se muestra en la Figura 6.8. Como se aprecia en las figuras 6.8(a) y (b), para un mismo valor del parámetro de adaptación ($\mu = 0,01$) el FCLMS presenta una velocidad de convergencia superior a la del BCLMS en general. No obstante, para una inicialización aleatoria algunas realizaciones del algoritmo divergen hacia valores absurdos del parámetro de bifurcación. Este problema se puede resolver en parte utilizando un valor de μ menor, como se muestra en las figuras 6.8(c) y (d). Sin embargo, una solución mejor consiste en utilizar una inicialización fija en un punto intermedio del espacio de parámetros: $\hat{c}^0 = 0,5$ para el SK-TM, y $\hat{c}^0 = 0$ para el BSK-TM. De este modo, en el caso sin ruido se garantiza la convergencia (no mostrada) de \hat{c}^n hacia el valor real de c incluso con $\mu = 0,01$.

Desafortunadamente, este buen comportamiento de los algoritmos BCLMS y FCLMS no puede garantizarse para mapas con un mayor número de intervalos. En las figuras 6.9 y 6.10 se muestra la convergencia de \hat{e}_i^n para dos mapas de Bernoulli distintos con cuatro intervalos. En este caso se usa el BCLMS con una inicialización fija de sus parámetros de tal modo que se obtenga una partición uniforme en el espacio de parámetros: $\hat{e}_1^0 = 0,25$, $\hat{e}_2^0 = 0,5$ y $\hat{e}_3^0 = 0,75$. En estas dos figuras puede apreciarse cómo, mientras que para el primer mapa el vector de parámetros estimados siempre converge hacia el real, en el segundo caso aparecen mínimos locales que pueden dar

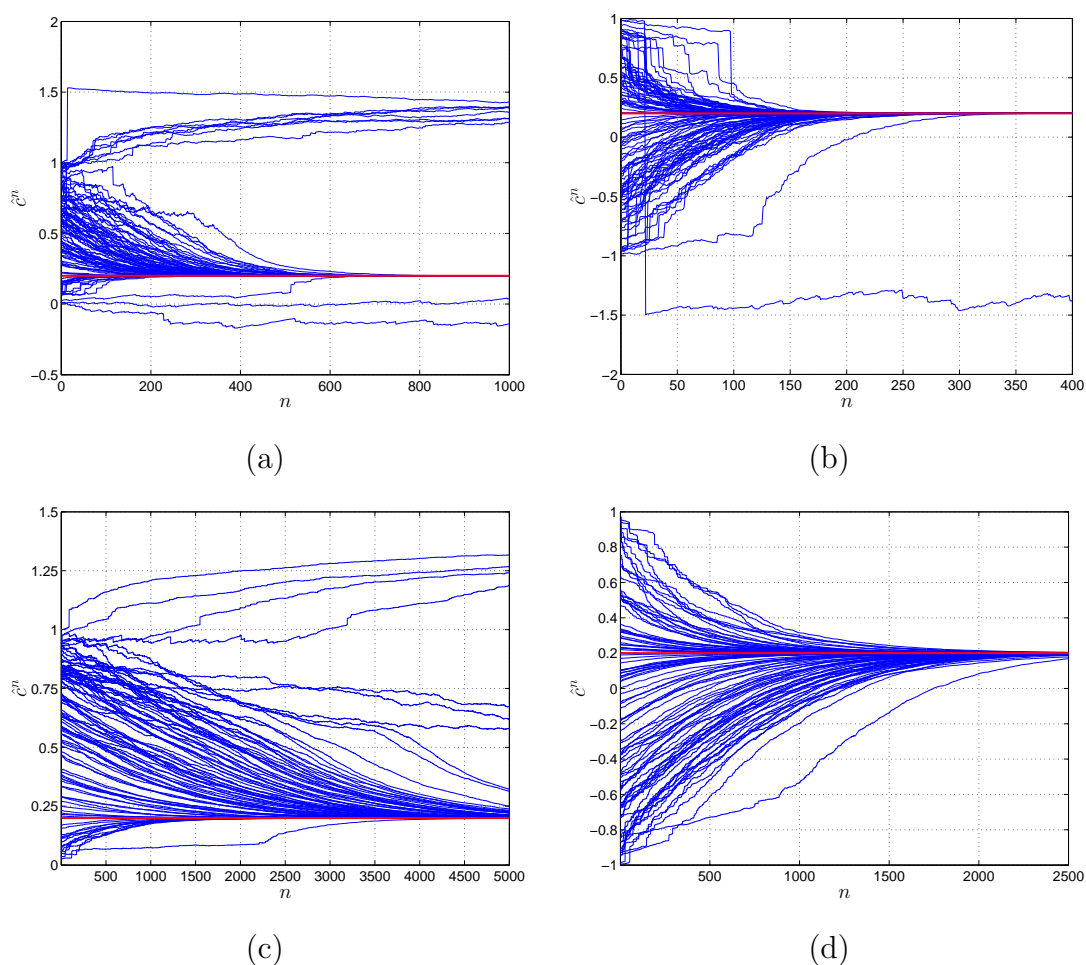


Figura 6.8: Convergencia del algoritmo FCLMS con una inicialización aleatoria, $N = 100$ y $C = \infty$. (a) SK-TM con $c = 0,2$ y $\mu = 0,01$. (b) BSK-TM con $c = 0,2$ y $\mu = 0,01$. (c) SK-TM con $c = 0,2$ y $\mu = 0,001$. (d) BSK-TM con $c = 0,2$ y $\mu = 0,001$.

lugar a particiones distintas de la deseada. Aunque en cualquier caso la convergencia del BCLMS se encuentra asegurada, tal y como se demuestra en el Apéndice G. No obstante, la presencia de estos mínimos locales va a provocar un empeoramiento del MSE promedio de la secuencia estimada, siendo necesario estudiar con mayor detalle su frecuencia de aparición y sus efectos, así como considerar posibles soluciones de este problema. Por lo tanto, en lo sucesivo únicamente se presentan resultados de MSE para el SK-TM y el BSK-TM.

En primer lugar, en la Figura 6.11 se compara el MSE del BCLMS y del FCLMS para el SK-TM. Las gráficas se han obtenido, para cada algoritmo y valor de SNR, a partir de 1.000 simulaciones de Monte Carlo llevando a cabo 5.000 iteraciones del algoritmo y utilizando la estima final obtenida. Puede apreciarse la aparición de un efecto umbral en el comportamiento del FCLMS para $c = 0,5$: es necesaria una SNR mínima de apro-

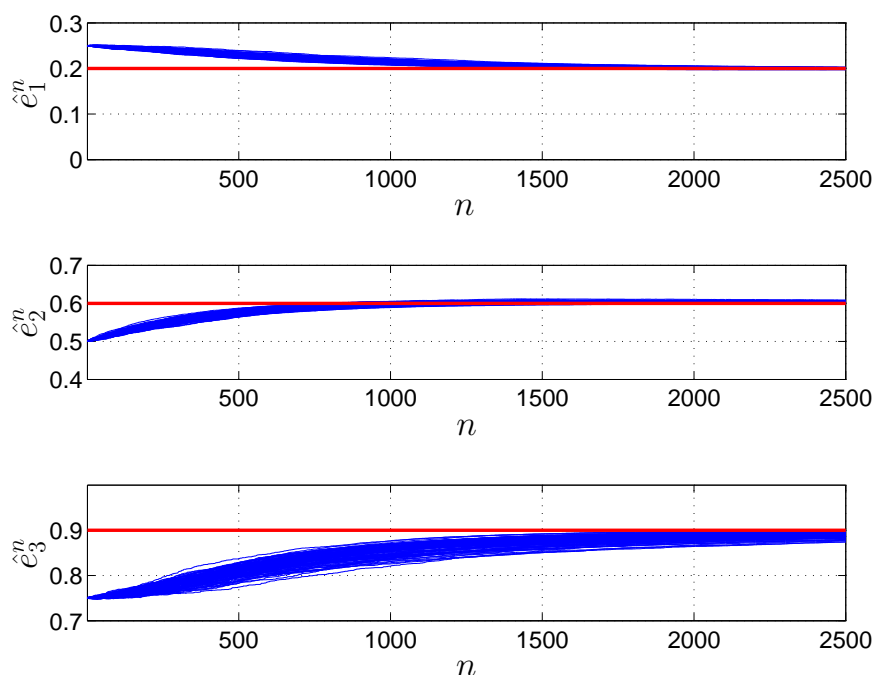


Figura 6.9: Convergencia del algoritmo BCLMS para un mapa de Bernoulli con cuatro intervalos ($e_0 = 0$, $e_1 = 0,2$, $e_2 = 0,6$, $e_3 = 0,9$ y $e_4 = 1$), una inicialización fija ($\hat{e}_1^0 = 0,25$, $\hat{e}_2^0 = 0,5$ y $\hat{e}_3^0 = 0,75$), $N = 100$, $\mu = 0,01$ y $C = \infty$.

ximadamente 15 dB para que empiece a funcionar correctamente. Además, cuando c se encuentra cercano al punto intermedio de su espacio de parámetros ($c = 0,5$ en este caso) el MSE de ambos algoritmos es esencialmente el mismo a partir de dicho umbral, mientras que en los extremos ($c = 0,1$ por ejemplo) el BCLMS consigue una ganancia importante (aproximadamente 5 dB en este caso) con respecto al FCLMS para valores altos de SNR. El pobre MSE observado en el caso del BCLMS con $\mu = 0,001$ se debe a que el algoritmo aún no ha sido capaz de converger después de 5.000 iteraciones. Las mismas conclusiones son aplicables al BSK-TM para el valor de c correspondiente, ya que ambos mapas son topológicamente conjugados.

A continuación, en la Figura 6.12 se compara el rendimiento de los dos algoritmos competitivos con el del estimador bloque correspondiente: el HCLS. Como puede verse, el BCLMS presenta básicamente el mismo rendimiento que el HCLS, mientras que el FCLMS sólo consigue dicho MSE para valores de c cercanos a 0,5, proporcionando un MSE mucho menor para valores muy bajos o muy altos de c .

Por último, en la Figura 6.13 se muestra el MSE de la estima de la secuencia completa en función del valor real de c para distintos valores de SNR. Puede verse como el algoritmo BCLMS siempre consigue eliminar parte del ruido presente en la entrada, obteniéndose una SNR a la salida superior. Por el contrario, el FCLMS consigue un rendimiento tan bueno como el del BCLMS para valores de c cercanos al punto medio

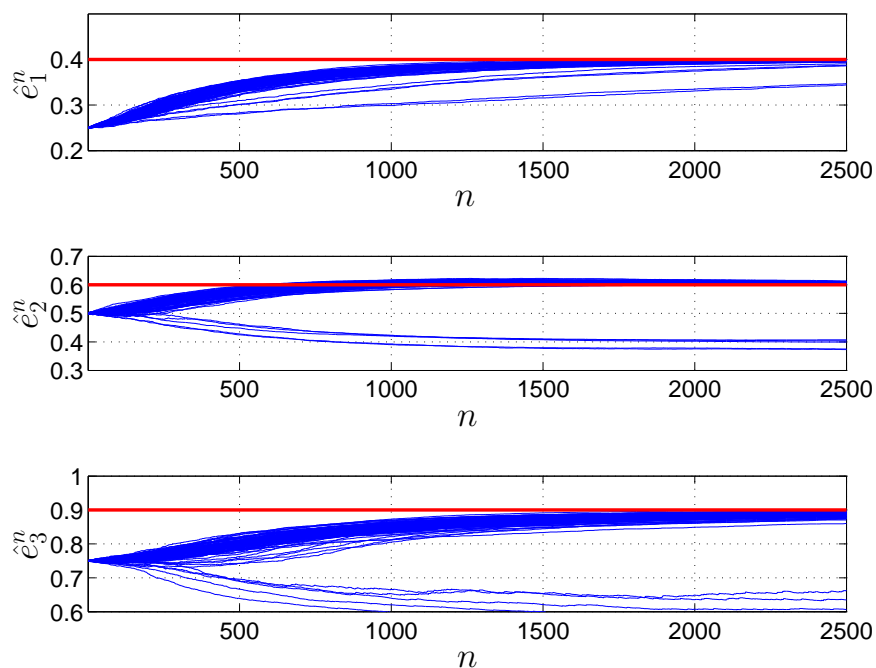


Figura 6.10: Convergencia del algoritmo BCLMS para otro mapa de Bernoulli con cuatro intervalos ($e_0 = 0$, $e_1 = 0,4$, $e_2 = 0,6$, $e_3 = 0,9$ y $e_4 = 1$), una inicialización fija ($\hat{e}_1^0 = 0,25$, $\hat{e}_2^0 = 0,5$ y $\hat{e}_3^0 = 0,75$), $N = 100$, $\mu = 0,01$ y $C = \infty$.

del espacio de parámetros, pero mucho peores conforme c se acerca a los extremos del mismo, no siendo capaz en estos casos de filtrar el ruido presente en las observaciones.

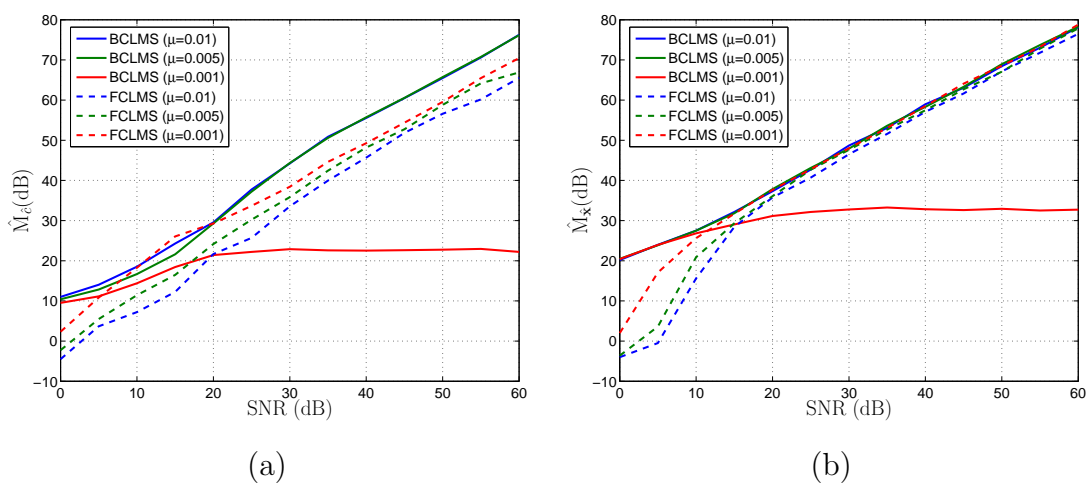


Figura 6.11: MSE de \hat{c} para los algoritmos BCLMS y FCLMS en función de la SNR para el SK-TM. (a) $c = 0,1$. (b) $c = 0,5$.

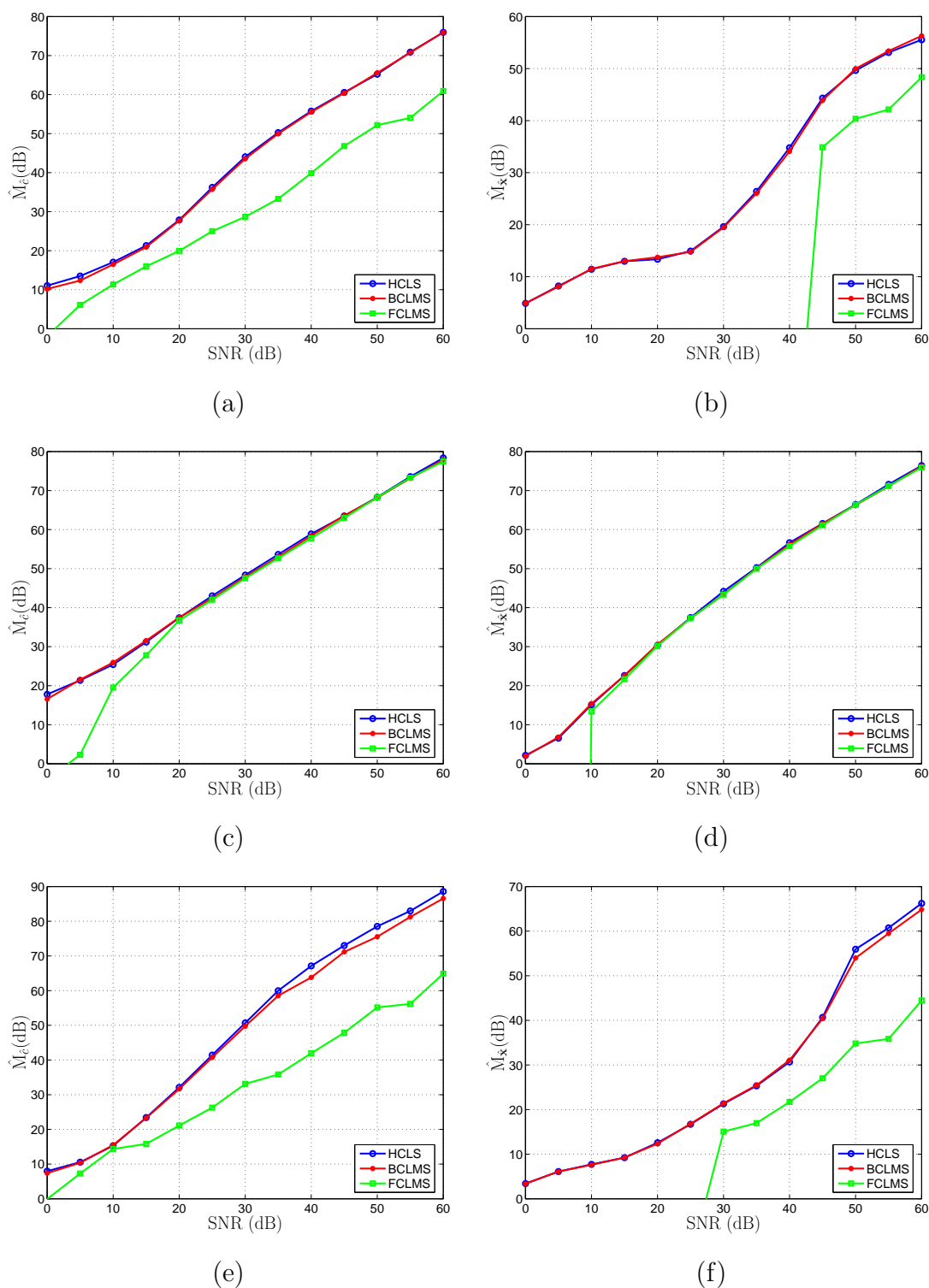


Figura 6.12: Comparación del MSE del algoritmo HCLS con el de los algoritmos BCLMS y FCLMS para el SK-TM. (a) MSE de \hat{c} para $c = 0,05$. (b) MSE de la secuencia caótica completa, $\hat{\mathbf{x}}$, para $c = 0,05$. (c) MSE de \hat{c} para $c = 0,5$. (d) MSE de la secuencia caótica completa, $\hat{\mathbf{x}}$, para $c = 0,5$. (e) MSE de \hat{c} para $c = 0,85$. (f) MSE de la secuencia caótica completa, $\hat{\mathbf{x}}$, para $c = 0,85$.

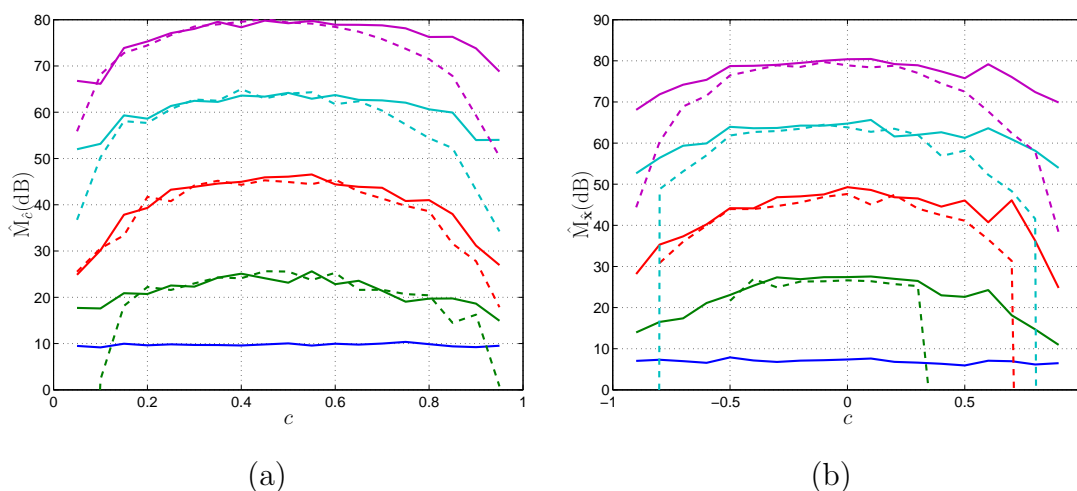


Figura 6.13: MSE de la secuencia caótica completa para los algoritmos BCLMS y FCLMS con distintos valores de SNR (SNR = 0, 15, 30, 45 y 60 dB), en función del parámetro de bifurcación. (a) SK-TM. (b) BSK-TM.

6.7. Discusión

En este capítulo se ha estudiado el problema de la estimación de los parámetros de un mapa caótico cuando su forma es conocida. Este problema es mucho más complejo que el de la estimación de la señal caótica, no pudiéndose aplicar técnicas de estimación Bayesiana ni de máxima verosimilitud. En consecuencia, a lo largo del capítulo se han considerado dos clases de técnicas subóptimas basadas en la relación determinista existente entre dos muestras consecutivas de la señal caótica: métodos bloque (MBE, LS y TLS) y estimadores iterativos/adaptativos locales (algoritmo CLMS). Este algoritmo CLMS resulta especialmente relevante, ya que puede utilizarse en otras aplicaciones relacionadas con el “clustering” o la identificación ciega por ejemplo.

La evaluación del rendimiento de los distintos estimadores se ha realizado del mismo modo que en los capítulos precedentes: mediante simulaciones de Monte Carlo. Las principales conclusiones extraídas a la vista de los resultados son las siguientes:

1. Aunque es posible plantear el estimador ML de los parámetros siguiendo una metodología similar a la del estimador ML de la secuencia (división del espacio de parámetros en función del itinerario, cálculo de hasta $P(N)$ estimas ML locales y selección de la mejor como estima ML global), en esta ocasión resulta mucho más complicado obtener cada una de las estimas ML locales. Por lo tanto, la aplicación del estimador ML sólo resulta factible para mapas PWL muy sencillos con un único parámetro (TM y S-TM por ejemplo) y secuencias muy cortas.
2. En la práctica, el modo de encontrar las estimas de los parámetros para mapas con dependencia lineal de los mismos consiste en explotar dicha relación lineal

existente entre dos muestras consecutivas. En este sentido, los métodos bloque propuestos (MBE, LS y TLS) proporcionan buenos resultados para SNRs medias/altas, aunque sin llegar a alcanzar el CRLB aparentemente.

3. Las prestaciones de los métodos bloque se encuentran limitadas para bajas relaciones señal a ruido por su sesgo. Aunque resulta difícil corregir completamente este problema, debido a la correlación entre el ruido y la señal introducida por la transformación no lineal del mapa caótico, sí que es posible minimizar su influencia aplicando un umbral antes de la estimación.
4. Como alternativa a los métodos bloque se pueden utilizar algoritmos iterativos/adaptativos locales estándar basados en el método del gradiente, el LMS o el algoritmo de Newton-Raphson, cuyo rendimiento es similar al de aquellos.
5. Los métodos bloque e iterativos convencionales no se pueden aplicar directamente a aquellos mapas cuya partición natural depende de los parámetros a estimar (como el SK-TM, el BSK-TM, o los mapas de Bernouilli), ya que habría que encontrar una estima para cada uno de los $P(N)$ posibles itinerarios, como en el caso del estimador ML. Para esta clase de mapas se han propuesto un método bloque, HCLS, y un algoritmo iterativo/adaptativo local, CLMS, que permiten estimar sus parámetros con un bajo coste computacional y un rendimiento similar al del resto de los mapas.
6. Este último algoritmo, el CLMS, es probablemente el más relevante de todos los presentados a lo largo del capítulo, ya que resulta útil en otras aplicaciones como el “clustering” o la identificación ciega. Además, se ha demostrado que converge hacia un mínimo de la esperanza matemática del cuadrado del error de aproximación siguiendo la dirección de su gradiente.
7. En el caso de la estimación de señales caóticas se ha comparado el rendimiento del CLMS obtenido a partir de la iteración hacia delante, FCLMS, y hacia atrás, BCLMS, constatándose que el BCLMS proporciona un mejor rendimiento que el FCLMS, tanto desde el punto de vista de la convergencia como del MSE.

Por último, debe considerarse la extensión de los estimadores propuestos a mapas d -dimensionales, al igual que en los capítulos anteriores. En primer lugar, dada la complejidad y el coste computacional de los estimadores HC-ML y ML para mapas unidimensionales con un único parámetro, parece evidente que no resulta factible su utilización para mapas multidimensionales. En consecuencia, cualquier problema que implique la estimación de los parámetros de un mapa caótico d -dimensional va a tener que recurrir a métodos subóptimos. En este sentido, la extensión de los métodos bloque o iterativos/adaptativos considerados a lo largo del capítulo a mapas d -dimensionales resulta inmediata siempre que se mantenga la linealidad (posiblemente a tramos) con los parámetros, y teniendo en cuenta que el coste computacional va a aumentar con el número de parámetros, así como con la dimensión de los mapas usados.