



Responsabilidad Corporativa y Sostenibilidad  
Cuaderno Red de  
Cátedras Telefónica



## Estudio de Técnicas de Codificación de Canal en Redes Celulares OFDM

Cátedra Telefónica de la Universidad de Zaragoza

El objetivo del presente trabajo es analizar las estructuras de codificación de canal empleadas en los sistemas LTE y WiMAX, proponiendo algoritmos eficientes para su decodificación.

Jorge Ortín Gracia  
Enero 2012

## Biografía

Jorge Ortín

Nacido en Zaragoza en 1981. Recibió los títulos de ingeniero de telecomunicación y doctor ingeniero de telecomunicación por la Universidad de Zaragoza en 2005 y 2011 respectivamente. Sus áreas de interés son los sistemas de comunicaciones móviles, especialmente en los campos de la codificación de canal y de la gestión de recursos radio.

## Índice

1. Introducción
2. Códigos convolucionales *tailbiting*
3. Turbocódigos
4. Precodificación independiente de canal
5. Conclusiones

## 1. Introducción

Los últimos años están siendo testigos de un avance sin precedentes en la implantación de una gran variedad de servicios multimedia en los sistemas de telecomunicaciones móviles. Dichos servicios tienen una estructura de tráfico muy distinta a los servicios tradicionales de voz ofrecidos hasta la fecha, con unos parámetros de calidad de servicio y ancho de banda muy exigentes. Por ello, es de especial importancia alcanzar una elevada eficiencia espectral que permita ofrecer servicio al mayor número posible de usuarios empleando un recurso limitado como es el espectro y poder gestionar y multiplexar servicios con requerimientos de ancho de banda y retardo variados. De todos los esquemas de modulación presentes hasta la fecha, OFDM (*Orthogonal Frequency Division Multiplexing*) es el que presenta las características más adecuadas para ofrecer estos servicios por su flexibilidad para la asignación de recursos, su capacidad para obtener tasas de transmisión elevadas y su eficiente implementación. Por esta razón, se empleará en la interfaz radio de los sistemas móviles WiMAX (*Worldwide Interoperability for Microwave Access*) y LTE (*Long Term Evolution*).

Si bien OFDM presenta múltiples ventajas frente a las modulaciones que emplean una única portadora, también presenta una serie de vulnerabilidades que degradan su funcionamiento en entornos inalámbricos móviles. En esta clase de sistemas es habitual la pérdida de la información contenida en parte de las portadoras al transmitirse en canales con desvanecimientos, lo que hace necesario el empleo de estrategias de corrección de errores que recuperen los errores introducidos en aquellas portadoras atenuadas. Entre estas estrategias destacan tanto los esquemas de codificación de canal como los de precodificación, cuyo estudio se analiza en este escrito.

Tanto LTE como WiMAX emplean fundamentalmente dos esquemas de codificación de canal para recuperar los errores producidos durante la transmisión: los códigos convolucionales *tailbiting* y los turbocódigos, siendo los primeros empleados para codificar información de control, puesto que requieren unos algoritmos de decodificación más sencillos desde un punto de vista computacional, mientras que los turbocódigos se utilizan para codificar información de usuario, ya que alcanzan una mayor eficiencia pese a ser más complejos que los códigos convolucionales. Otro método para corregir errores en la transmisión y que al contrario de la codificación de canal no disminuye la tasa de transmisión de información útil es el empleo de técnicas de precodificación en el transmisor que compensan los efectos del canal antes de pasar por él y no después, como ocurre cuando se emplea únicamente ecualización en el sistema. Estos mecanismos aprovechan las propiedades estadísticas del canal para pre-ecualizar la señal antes de ser transmitida o aumentar la diversidad en tiempo y frecuencia.

En las siguientes secciones se va a analizar varias técnicas de codificación de canal y precodificación para sistemas móviles OFDM, proponiendo soluciones que permitan mejorar las prestaciones del sistema.

## 2. Códigos convolucionales *tailbiting*

Los códigos convolucionales constituyen el esquema de codificación de canal más básico que se implementa en los sistemas móviles de nueva generación. El principio de funcionamiento de los códigos convolucionales se basa en la alimentación de una serie de registros de desplazamiento con una secuencia de símbolos de entrada a codificar, los cuales pueden ser bits o grupos de bits. En cada instante de tiempo, el codificador calcula las salidas mediante sumas módulo-2 del contenido de parte de los registros de desplazamiento y de la entrada en ese instante. En función del método empleado para inicializar la memoria (los registros de desplazamiento) del codificador, los códigos convolucionales se dividen en códigos *zero tail* y los códigos *tailbiting*.

En los códigos convolucionales *zero tail*, el contenido de la memoria del codificador se inicializa a cero, añadiendo al final de cada bloque de datos otro grupo de ceros para resetear la memoria del codificador. Este hecho provoca que el *trellis* del código comience y finalice en un estado conocido, lo que facilita el proceso de decodificación mediante el algoritmo de Viterbi. No obstante, la tasa efectiva de codificación va a disminuir, debido a la transmisión de los bits que se emplean para resetear la memoria del codificador. Por el contrario, los códigos convolucionales *tailbiting* inicializan la memoria del codificador con los últimos bits del bloque que se está codificando, lo que provoca que el estado inicial y final del *trellis* sea el mismo, si bien este es desconocido por el decodificador. En este caso, la tasa de codificación no empeora al evitarse los bits de reseteo.

La figura 1.2 muestra los *trellis* del codificador de la figura 1.1 cuando se realiza una codificación *zero tail* y otra *tailbiting*. En el caso *zero tail* (izquierda) es necesario añadir dos ceros al final de la secuencia de datos, por lo que la información útil transmitida corresponderá a tres bits, que codificados se transforman en diez. Como se ha señalado anteriormente, todos los posibles caminos en el *trellis* empiezan y acaban en el estado cero. En azul se representa uno de ellos, el correspondiente a la codificación de la secuencia 111. Al contrario, cuando se emplea codificación *tailbiting* no se añade ningún bit al final de la secuencia de datos, por lo que en este caso se podrán transmitir cinco bits de información útil, que se convierten en diez tras la codificación. Este hecho provoca que el *trellis* no tenga un estado inicial y final definido *a priori*, si bien dicho estado ha de ser el mismo. En el *trellis* de la derecha se representa en azul el camino correspondiente a la codificación de la secuencia 11110. El estado inicial corresponde a los dos últimos bits codificados, 01, puesto que este será el contenido de la memoria cuando se alimenten los dos últimos bits. Es importante recalcar que solo están permitidos en el *trellis* aquellos caminos que empiecen y acaben en el mismo estado. Así, el camino marcado en rojo no pertenece realmente al *trellis tailbiting* representado.

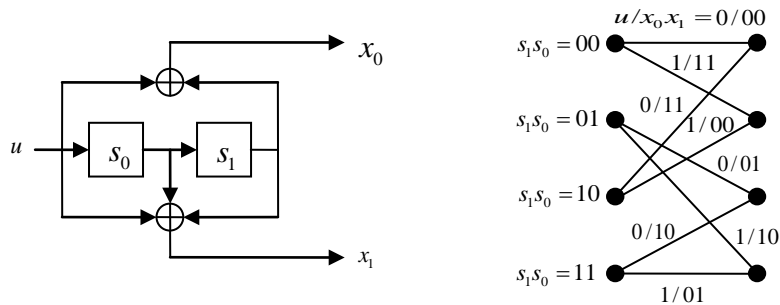


Figura 1.1. Codificador convolucional de tasa 1/2 y memoria  $m=2$  y transición del *trellis* asociada

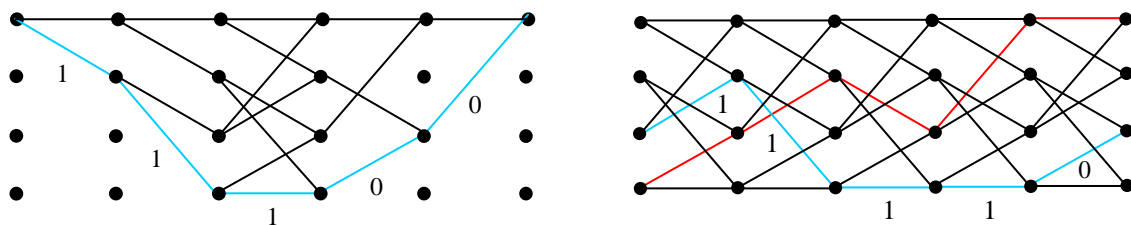


Figura 1.2. *Trellis zero tail* y camino correspondiente a la secuencia 111 (izquierda) y *trellis tailbiting*, camino correspondiente a la secuencia 11110 (azul) y camino no perteneciente al *trellis*.

La decodificación de códigos *zero tail* se realiza directamente mediante el algoritmo de Viterbi, aplicándolo desde el estado inicial cero hacia delante y calculando en cada iteración el camino superviviente de mejor métrica que llega a cada uno de los estados intermedios del *trellis*. Una vez que se alcanza el final del *trellis*, se realiza el *traceback* del camino superviviente que termina en el estado cero, decodificando la secuencia de datos correspondiente a dicho camino. Sin embargo, en el caso de emplear códigos *tailbiting* el decodificador no sabe cuál es el estado inicial y final del *trellis*, por lo que no sabrá desde qué estado iniciar el algoritmo de Viterbi y desde qué estado realizar el *traceback*.

Una solución evidente para resolver este problema sería aplicar el algoritmo de Viterbi para cada uno de los posibles estados iniciales y finales del *trellis*, seleccionando finalmente aquella secuencia con la mejor métrica. Si bien esta estrategia de decodificación por fuerza bruta es óptima, su implementación es inviable en sistemas reales al crecer exponencialmente el número de realizaciones del algoritmo de Viterbi con el tamaño de la memoria. Por ello, se han propuesto hasta la fecha diversos algoritmos subóptimos con una carga computacional inferior a la de esta estrategia. Estos algoritmos aprovechan la circularidad de los códigos *tailbiting* aplicando el algoritmo de Viterbi al mismo bloque concatenado varias veces hasta que se cumple alguna condición de convergencia. Esto permite reutilizar la información de las métricas de los caminos supervivientes al final de cada iteración del algoritmo de Viterbi para emplearlas como métricas iniciales de la siguiente iteración, lo que disminuye la carga computacional de la decodificación [1].

Si bien todos estos algoritmos presentan condiciones de terminación para evitar la presencia de situaciones de no convergencia, el número de operaciones necesario para decodificar un bloque es generalmente variable y dependiente de las condiciones del canal. Este hecho se agrava cuando se emplean en sistemas móviles en los que las condiciones de propagación cambian bruscamente, por lo que se produce a su vez un aumento de la variabilidad del tiempo necesario para la decodificación. Para estas situaciones, el desarrollo de un algoritmo de decodificación cuyo tiempo de procesado sea conocido y fijo y cuyos resultados sean cercanos a los óptimos simplificaría el diseño e implementación del receptor mejorando su rendimiento [2].

Como se ha señalado anteriormente, en los códigos *tailbiting* el estado inicial y final del *trellis* ha de ser el mismo, si bien es desconocido por el receptor. Un método de decodificación consistiría en buscar un estado candidato para comenzar y finalizar el *trellis* desde él y aplicar de este modo el algoritmo clásico de Viterbi. Este método no estaría basado en la concatenación sucesiva del mismo bloque de entrada, por lo que se evitarían problemas de convergencia. Para elegir este estado, la solución más lógica consiste en buscar el estado del *trellis* cuya verosimilitud sea máxima. Una vez encontrado y gracias a las propiedades circulares del *trellis*, se puede desplazar circularmente la secuencia de datos codificados para comenzar y finalizar el *trellis* en el estado elegido. El principal problema del método es el referente a la elección de este estado inicial, puesto que si se elige adecuadamente, la decodificación posterior mediante el algoritmo de Viterbi será óptima. Para elegirlo, se propone emplear una versión modificada del algoritmo SOVA (*Soft Output Viterbi Algorithm*) [3] que estima las probabilidades de los estados de un camino determinado del *trellis*. La decodificación por tanto se compone de dos fases diferenciadas: en la primera de ellas se aplica el algoritmo SOVA para elegir el estado que servirá de inicio y final del *trellis*. Una vez encontrado este estado, se aplica el algoritmo clásico de Viterbi en la segunda fase de la decodificación. Puesto que la complejidad del SOVA es similar a la del algoritmo de Viterbi, la carga computacional máxima es dos veces la del algoritmo de Viterbi.

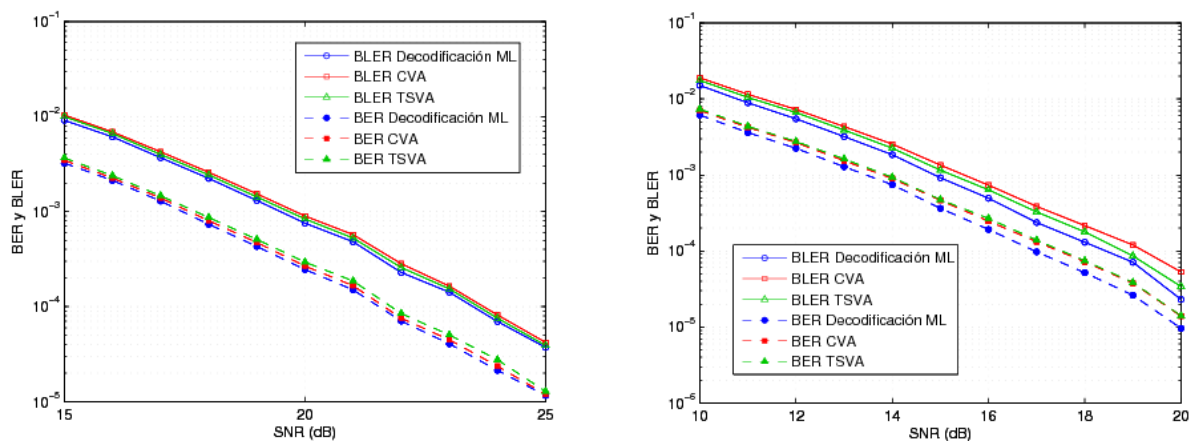


Figura 2. BER y BLER frente a SNR para distintos algoritmos de decodificación y canal móvil ITU Vehicular A a 30 km/h. Código *tailbiting* de LTE a tasa 1/3 (derecha) y de WiMAX a tasa 1/2 (izquierda).

La figura 2 compara los resultados en términos de BER (*Bit Error Rate*) y BLER (*Block Error Rate*) de este algoritmo propuesto (denominado TSVA – *Two Step Viterbi Algorithm*) en un canal móvil con los obtenidos mediante decodificación óptima (ML) y el algoritmo presentado en [1] (CVA – *Circular Viterbi Algorithm*). Los códigos analizados corresponden al código *tailbiting* empleado en WiMAX a tasa 1/2 y al código *tailbiting* empleado en LTE (tasa 1/3). El tamaño del bloque de datos es el mínimo posible recogido en cada sistema y corresponde a 48 bits para WiMAX y 40 bits para LTE. Los datos codificados se entrelazan según el método empleado en cada estándar y se modulan en símbolos QPSK, los cuales se transmiten en una señal OFDM de 5 MHz, con una frecuencia de portadora de 2,5 GHz y 512 subportadoras. Los resultados muestran un comportamiento muy cercano al óptimo tanto en términos de BER como de BLER, y una mejora respecto el algoritmo CVA en términos de BLER.

Anteriormente se ha señalado que el algoritmo de fuerza bruta consistente en realizar una decodificación de Viterbi para cada posible estado inicial y final es el que obtiene una decodificación óptima. Otro modo de obtener una decodificación de máxima verosimilitud con una carga computacional inferior consiste en aplicar al *trellis* algoritmos de búsqueda de rutas óptimas en grafos [4]. Este algoritmo propuesto está basado en el algoritmo A\*, dividiéndose la decodificación en dos fases. La primera de ellas consiste en aplicar el algoritmo de Viterbi con el objeto de obtener información referente al *trellis*. Esta información se usa en la segunda fase de la decodificación, en la que se aplica el algoritmo A\* para encontrar el camino de máxima verosimilitud del *trellis*.

La tabla 1 muestra la carga computacional total del algoritmo de fuerza bruta, del algoritmo subóptimo TSVA, del algoritmo óptimo propuesto (denominado EA\*) y del algoritmo de Viterbi clásico para decodificar el código *tailbiting* de WiMAX a tasa 1/2 y canal móvil. Los parámetros de transmisión son los mismos que los empleados en la figura 2. En paréntesis se ha añadido el incremento de carga computacional que supone cada algoritmo frente a la del algoritmo de Viterbi clásico. Como se puede observar, el algoritmo óptimo propuesto tiene una carga computacional ligeramente superior a la del algoritmo subóptimo TSVA y muy inferior a la del algoritmo de fuerza bruta. No obstante, es importante señalar que la carga computacional máxima del algoritmo subóptimo estaba limitada a dos veces la del algoritmo de Viterbi, mientras que la del EA\* puede ser mayor. Asimismo, los requerimientos de memoria del EA\* son muy superiores.

SNR	Fuerza bruta	TSVA	EA*	Algoritmo Viterbi
10 dB	589824 (6400%)	10994 (19,3%)	11437 (24,1%)	9216
15 dB	589824 (6400%)	9371 (1,68%)	9560 (3,74%)	9216
20 dB	589824 (6400%)	9297 (0,88%)	9301 (0,92%)	9216

Tabla 1. Número medio de operaciones del EA\*, el TSVA y el algoritmo de fuerza bruta para decodificar el código *tailbiting* de WiMAX (tasa 1/2, tamaño de bloque 6 bytes, canal ITU Veh. A ext.)



### 3. Turbocódigos

Los turbocódigos se emplean en la gran mayoría de los sistemas móviles de nueva generación, entre los que se incluyen LTE y WiMAX, debido a su elevada capacidad correctora de errores y su moderada complejidad. Los turbocódigos, tal y como se plantearon originalmente, están formados en el transmisor por dos codificadores convolucionales trabajando en paralelo y separados por un entrelazador. De este modo cada bloque de datos se codifica dos veces: la primera de ellas alimentando el codificador convolucional con la secuencia de bits original y la segunda alterando el orden de la misma con el entrelazador. La decodificación se realiza con dos decodificadores SISO (*Soft Input Soft Output*), uno para cada una de las versiones codificadas de los datos, los cuales se van pasando iterativamente las estimaciones *soft* de los bits de información codificados.

Los principales parámetros que van a definir el comportamiento de los turbocódigos son fundamentalmente cuatro: la estructura del código convolucional componente, el método de terminación del *trellis*, el entrelazador empleado y el algoritmo de decodificación implementado por el receptor. De estos cuatro parámetros, únicamente el algoritmo de decodificación es el que se puede elegir libremente a la hora de implementar el receptor de un sistema determinado, estando los otros tres fijados por el propio sistema. Por ello, la elección del algoritmo de decodificación influirá en los resultados específicos que se obtengan con cada código y determinará la complejidad del receptor.

El algoritmo que consigue realizar una decodificación *soft* óptima es el MAP (*Maximum A Posteriori*), el cual obtiene todas las probabilidades asociadas a cada uno de los estados y transiciones del *trellis*. No obstante, el principal inconveniente de este algoritmo es su elevada complejidad, debido al empleo de funciones no lineales y la realización de gran cantidad de multiplicaciones. Para resolver dichos problemas en [5] se propone pasar el algoritmo MAP al dominio logarítmico, obteniéndose el algoritmo max-log-MAP, el cual hace desaparecer todas las no-linealidades, transformando las multiplicaciones en sumas y comparaciones. Otro algoritmo de decodificación que presenta a su salida valores *soft* es el SOVA [3], el cual está basado en el algoritmo clásico de Viterbi. El SOVA emplea las métricas de los caminos descartados en cada uno de los estados que conforman el camino de máxima verosimilitud para estimar la probabilidad de los bits decodificados, los cuales están asociados a las distintas transiciones del camino de máxima verosimilitud. El principal problema de este algoritmo es que únicamente tiene en cuenta la información asociada a los caminos descartados a lo largo del camino de máxima verosimilitud, perdiéndose la del resto de caminos que puede haber en el *trellis* (otros caminos descartados en las transiciones intermedias de los caminos descartados).

Debido a la importancia de esta elección, en [6] se analiza el funcionamiento de los algoritmos de decodificación SOVA y max-log-MAP con las dos clases de turbocódigos que más se emplean en la

actualidad en sistemas móviles: los turbocódigos convolucionales binarios (empleado en LTE) y los turbocódigos convolucionales duobinarios (empleado en WiMAX). Para ello, se cuantifica mediante simulación cuál es la mejora en términos de BLER que logra un algoritmo sobre otro así como la carga computacional de cada uno de ellos, lo que permitirá obtener en cada caso la combinación de codificador-decodificador óptima. Una elección correcta del algoritmo de decodificación permitirá maximizar las prestaciones del código, disminuyendo la potencia transmitida requerida para obtener una BLER determinada mientras se limita la complejidad de la decodificación y por tanto el coste de la electrónica digital necesaria para implementarla.

La figura 3 muestra el comportamiento del turbocódigo binario de LTE y del turbocódigo duobinario de WiMAX decodificados con el SOVA y el max-log-MAP en el canal móvil ITU Vehicular A a 30 km/h. La tasa de código en ambos casos es 1/2 y los tamaños de bloque son 48 (izquierda) y 488 bits (derecha). Los datos codificados se entrelazan según el método empleado en cada estándar y se modulan en símbolos QPSK, los cuales se transmiten en una señal OFDM de 5 MHz, con una frecuencia de portadora de 2,5 GHz y 512 subportadoras. La mejora del max-log-MAP respecto al SOVA para el turbocódigo de LTE es despreciable en los dos tamaños de bloque, mientras que para el código de WiMAX varía entre los 2 dB para un tamaño de bloque de 48 bits y 0,5 dB cuando el bloque es de 488 bits. Por otro lado, el comportamiento de ambos códigos es muy similar cuando se emplea el algoritmo max-log-MAP para decodificar, no produciéndose mejora apreciable entre un código y otro.

Respecto a la carga computacional asociada a la decodificación, la tabla 2 muestra el número medio de operaciones realizadas en función del algoritmo de decodificación y del turbocódigo empleados para distintas BLER, con un tamaño de bloque de 48 bits y canal ITU Vehicular A. Si se comparan los distintos algoritmos de decodificación, la carga computacional del SOVA es un 60% inferior a la del max-log-MAP para el turbocódigo de LTE. Para el turbocódigo de WiMAX, el ahorro en el número de operaciones del SOVA respecto al max-log-MAP varía entre el 30% y el 40% en función de la BLER.

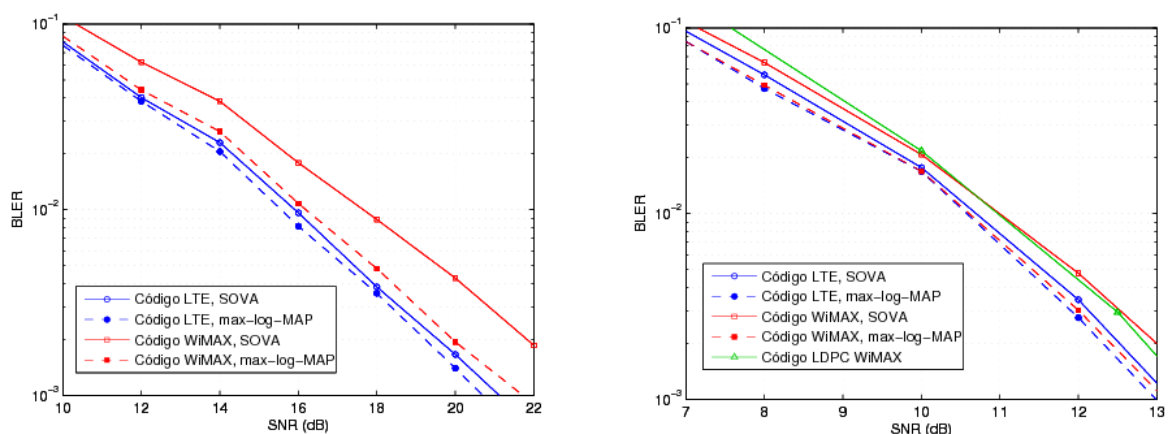


Figura 3. BLER frente para los algoritmos SOVA y max-log-MAP y los turbocódigos de WiMAX y LTE a tasa 1/2 en canal ITU Vehicular A a 30 km/h. Tamaños de bloque 48 bits (izquierda) y 488 bits (derecha).

BLER	SOVA LTE	Max-log-MAP LTE	SOVA WiMAX	Max-log-MAP WiMAX
$10^{-1}$	10668	26991	25743	34581
$10^{-2}$	3590	9298	7964	11690
$10^{-3}$	3072	8090	6146	9901

Tabla 2. Número medio de operaciones de los algoritmos SOVA y max-log-MAP para los turbocódigos de WiMAX y LTE a tasa 1/2 y tamaño de bloque 48 bits.

De los resultados obtenidos, se concluye que el SOVA es preferible al max-log-MAP para el turbocódigo de LTE, al tener unas prestaciones similares pero con una carga computacional considerablemente inferior. Para el código de WiMAX, el algoritmo max-log-MAP es preferible, puesto que si bien el número de operaciones ejecutado por cada algoritmo es cercano, el max-log-MAP tiene una ganancia en torno a los 2 dB sobre el SOVA.

## 4. Precodificación independiente de canal

Otro modo de mejorar el comportamiento de un sistema OFDM frente a la variabilidad que presentan los canales móviles es el empleo de estrategias de precodificación. Aunque estas técnicas aún no se emplean en los estándares móviles LTE y WiMAX, su uso podrá mejorar la eficiencia de los mismos en un futuro. La base de la precodificación consiste en preconformar la información a transmitir en cada portadora con el objetivo de mejorar algún aspecto de la transmisión, como puede ser obtener una BER más baja en el receptor sin necesidad de aumentar la información redundante enviada. Si bien hasta la fecha se han propuesto múltiples técnicas de precodificación para sistemas OFDM con este último propósito, la gran mayoría de ellas presuponen un conocimiento *a priori* total o parcial del canal por el transmisor. Este tipo de esquemas es inviable desde un punto de vista práctico en sistemas móviles, donde se requeriría una gran cantidad de información de *feedback* de muchos usuarios diferentes que observan canales distintos para poder realizar la precodificación en la estación base correctamente. Asimismo, la latencia inherente a este *feedback* unida a la propia variabilidad del canal provocaría con frecuencia que la estimación del canal realizada en el receptor ya no sea útil cuando la recibe el transmisor. Por ello, resulta de mucha utilidad emplear técnicas de precodificación que no necesiten disponer de ninguna información *a priori* del canal en el transmisor.

El objetivo principal de los precodificadores independientes del canal es evitar la pérdida completa de la información contenida en las portadoras que sufren desvanecimientos profundos. Una solución para tratar de resolver esta pérdida consiste en redistribuir la información contenida en varios símbolos QPSK / QAM

entre varias portadoras, consiguiéndose aumentar la diversidad de la señal transmitida sin comprometer la eficiencia.

En [7] se ha propuesto un sistema de precodificación para sistemas OFDM independiente del canal en el que se multiplica cada bloque a precodificar por una matriz de Hadamard que distribuye la energía de cada símbolo de entrada entre todos los de salida. Tras este bloque se realiza un entrelazado tiempo-frecuencia de longitud superior a los tiempos anchos de banda de coherencia del canal cuya misión es separar los elementos constituyentes de un mismo bloque de precodificación a la hora de realizar el mapeo posterior a las portadoras OFDM. En recepción se realiza una ecualización *zero forcing* con *clipping* que limita la amplificación del ruido en aquellas portadoras más atenuadas por el canal y la posterior redistribución de este ruido al resto de símbolos cuando se realiza la precodificación inversa. Este sistema propuesto va a obtener una mejora sustancial de la probabilidad de error en el símbolo cuando se compara con un sistema sin precodificación.

La figura 4 muestra el comportamiento del sistema de precodificación propuesto comparado con otros sistemas de precodificación [8] sobre una señal OFDM de 5 MHz, con una frecuencia de portadora de 2,5 GHz y 512 subportadoras. Esta señal se transmite en el canal ITU Vehicular A con el móvil desplazándose a 120 km/h. Los parámetros empleados para el entrelazador del precodificador son 64 portadoras de coherencia y 12 símbolos de coherencia. Esta elección asegura que la separación entre elementos de un mismo bloque precodificado sea suficiente para que el canal visto por cada elemento se pueda considerar independiente del canal visto por el resto de elementos. El tamaño del bloque de precodificación es variable. Los datos de entrada al precodificador corresponden a símbolos complejos QPSK. En el receptor la demodulación es *hard*, refiriéndose por tanto la BER a los bits contenidos en los símbolos QPSK. Se ha supuesto asimismo un conocimiento total del canal por parte del receptor, por lo que el error en su estimación es nulo. Como se puede observar, la BER obtenida es inferior en el sistema propuesto a la del sistema OFDM sin precodificador para todos los tamaños del bloque simulados.

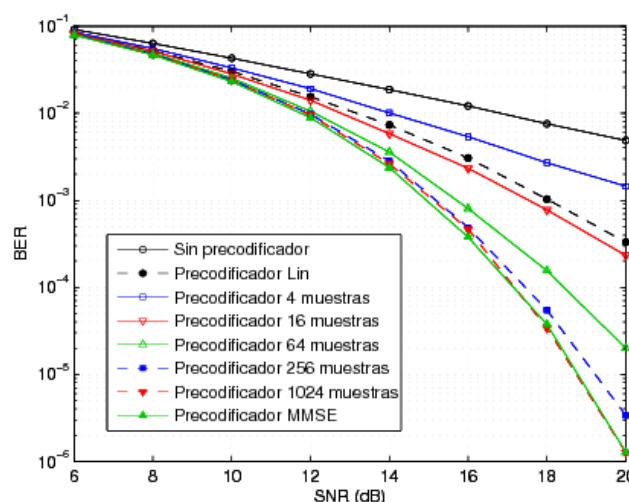


Figura 4. BER frente a SNR para distintos bloques del tamaño de precodificación

## 5. Conclusiones

En este cuaderno se han explicado los principales esquemas de codificación de canal que se emplean en los sistemas móviles de nueva generación WiMAX y LTE. Para uno de estos esquemas, los códigos convolucionales *tailbiting*, se han mostrado los resultados de dos nuevos algoritmos de decodificación, TSVA y EA\*, propuestos recientemente y que mejoran las prestaciones de la mayoría de algoritmos de decodificación presentes hasta la fecha. Asimismo, se ha hecho una comparativa de los principales algoritmos de decodificación que se pueden aplicar a los turbocódigos, el segundo esquema de codificación empleado en WiMAX y LTE, eligiendo el algoritmo más adecuado para cada sistema. Finalmente, se han presentado los resultados de un esquema de precodificación independiente del canal para sistemas OFDM que, de ser implementado en WiMAX o LTE, conseguiría disminuir la tasa de error sin necesidad de disminuir la eficiencia de la transmisión.

## 6. Bibliografía

- [1] H. R. Y. Shao, S. Lin y M. P. C. Fossorier: "Two Decoding Algorithms for Tailbiting Codes". *IEEE Transactions on Communications*, 2003, vol. 51, no. 10, pp. 1658-1665.
- [2] J. Ortin, P. García, F. Gutiérrez y A. Valdovinos: "Two Step SOVA-based Decoding Algorithm for Tailbiting Codes". *IEEE Communications Letters*, 2009, vol. 13, no. 7, pp. 510-512.
- [3] J. Hagenauer, E. Offer y L. Papke: "Iterative Decoding of Binary Block and Convolutional Codes". *IEEE Transactions on Information Theory*, 1996, vol. 42, no. 2, pp. 429-445.
- [4] J. Ortin, P. García, F. Gutiérrez y A. Valdovinos: "A\*-based Algorithm for Reduced Complexity ML Decoding of Tailbiting Codes". *IEEE Communications Letters*, 2010, vol. 14, no. 9, pp. 854-856.
- [5] J. Erfanian, S. Pasupathy y G. Gulak: "Reduced Complexity Symbol Detectors with Parallel Structures for ISI Channels". *IEEE Transactions on Communications*, 1994, vol. 42, no. 2, pp. 1661-1671.
- [6] J. Ortin, P. García, F. Gutiérrez y A. Valdovinos: "Performance Analysis of Turbo Decoding Algorithms in Wireless OFDM Systems". *IEEE Transactions on Consumer Electronics*, 2009, vol. 55, no. 3, pp. 1149-1154.



# Cuaderno Red de Cátedras Telefónica

## Estudio de Técnicas de Codificación de Canal en Redes Celulares OFDM

14

- [7] J. Ortin, P. García, F. Gutiérrez y A. Valdovinos: "*Channel Independent Precoder for OFDM Systems over Fading Channels*". *IEEE Transactions on Broadcasting*, 2009, vol. 55, no. 4, pp. 818-825.
- [8] Y. P. Lin y F. S. Phoong: "*BER Minimized OFDM Systems with Channel Independent Precoders*". *IEEE Transactions on Signal Processing*, 2003, vol. 51, no. 9, pp. 2369-2380.