

Simulación y Análisis del Algoritmo de Codificación Turbo en Sistemas de Comunicaciones Móviles: Un Estudio Comparativo

Fernando Corteggiano¹, Marcelo Gioda², Esteban Carranza, M. Luciana Medina, Diego Bazán, Gerardo Di Claudio

¹*Dto. de Electricidad y Electrónica, Universidad Nacional de Río Cuarto, Argentina.
fcorteggiano@ing.unrc.edu.ar*

²*Dto. de Electricidad y Electrónica, Universidad Nacional de Río Cuarto, Argentina.
mgioda@ing.unrc.edu.ar*

Resumen: El objetivo de un sistema de comunicación es transmitir información a la mayor velocidad con el menor consumo de energía y con la mínima cantidad de errores; aún cuando el medio o canal de comunicación empleado sea muy ruidoso. La máxima capacidad de transmisión de datos de un canal de comunicación está dada por el límite de Shannon [1]. Debido a su performance cercana a ese límite, los Códigos Turbo han recibido mucha atención desde su aparición [2] y han sido incluidos en las especificaciones de los futuros sistemas de comunicaciones móviles de tercera generación (3G) como el Universal Mobile Telecommunication System (UMTS)[3]. Este trabajo presenta la implementación de un algoritmo de codificación turbo, basado en las especificaciones de UMTS, mostrando la mejora de la performance de error frente a un sistema no codificado y a un codificador convolucional basado en el algoritmo de Viterbi. Se presenta un ejemplo concreto donde se aprecian las diferencias de la mejora en la tasa de error de bits en un canal con ruido y con modulación BPSK.

Palabras Clave— Códigos Turbo, Performance de Error, Codificador Convolucional, BER, Viterbi.

Workshop— No se considera encuadrar en ninguno de los cinco Workshop descriptos, aunque si en los temas 7: Comunicaciones y Redes y 14: Optimización y Simulación.

1. Introducción

La tercera generación (3G) de comunicaciones móviles, afronta una serie de desafíos tecnológicos. Entre otros: tasa de datos para el usuario de hasta 2 Mbps en su celular, continuidad de servicios multimedia aún bajo techo, cobertura en todo momento y en todo lugar, aprovechamiento de rebotes y multicaminos de la señal transmitida, y esquemas de codificación que combatan el desvanecimiento de la señal recibida [4].

En estas condiciones tan exigentes, el medio de comunicación obliga al uso de técnicas de codificación robustas y modernas, para poder transmitir a altas tasas de datos a través de la interfaz de aire.

La relación entre la cantidad de bits de información interpretados erróneamente por el receptor y la cantidad de bits de información enviados por el transmisor, se conoce como *Bit Error Rate* (BER) o tasa de error de bit [5].

Normalmente se requiere que a los bits de información que se vayan a transmitir, se les agregue bits de redundancia (sin información) para que, llegado el caso de alterarse algún bit, se tengan en el receptor los bits de redundancia para poder deducir el valor real del bit de información alterado. Los llamados “esquemas de codificación de canal” están pensados para recuperar el valor correcto de cada bit de información.

Se podría tomar cada bit de información y transmitirlo tres veces seguidas, de tal manera que si uno de los tres bits se altera, están los otros dos (supuestamente inalterados) para recuperar el valor correcto del bit. Por ejemplo, si se desea transmitir “1011” se debería transmitir “111000111111” (el subrayado es sólo para distinguir un bit de información de otro), de tal manera que si llega “111000101011”, el receptor pueda deducir que el mensaje era “1011”.

Se observa que el ejemplo del esquema de codificación anterior con redundancia de bits, no es eficiente por cuanto requiere tres veces más velocidad (o tiempo) comparado con el envío del mensaje sin redundancia. Pero lo que además lo hace ineficiente, es que en muchos casos, el ruido que produce el canal o medio de comunicación puede alterar el valor de varios bits seguidos y, de todas maneras, ser interpretado el bit de manera errónea en el receptor [5].

Existe un límite en la eficiencia para aprovechar la capacidad de un canal, llamado límite de Shannon. Los esquemas de codificación con Código Turbo, son lo que más se aproximan a dicho límite. Por su complejidad matemática y de implementación en los transmisores y receptores, los Códigos Turbo son empleados sólo cuando las condiciones del canal así lo requieren.

La 3G de comunicaciones móviles representada por UMTS (Universal Mobile Telecommunication System), que es una continuación de la tecnología actual de comunicación celular GSM (Global System for Mobile communication), ha seleccionado un esquema de codificación con Código Turbo [3].

2. Objetivos

El objetivo del presente trabajo es, a través del análisis y la simulación de un algoritmo de codificación turbo, mostrar las mejoras de la performance de error que este ofrece frente a dos sistemas de comunicaciones: uno sin codificación de canal y otro con codificación convolucional basado en el algoritmo de Viterbi.

3. Marco Teórico

Como se muestra en la figura 1, el codificador turbo de UMTS esta compuesto de dos codificadores convolucionales sistemáticos recursivos (RSC) en paralelo. El generador feedforward es 15 y el generador feedback es 13, ambos en octal. El número de bits de datos en la entrada del codificador turbo es K , con: $40 \leq K \leq 5114$.

Los datos son codificados por el primer codificador (el de arriba) en su orden normal y por el segundo codificador (el inferior) después del interleaver. Inicialmente ambos switch están en la posición superior [6].

Los bits de datos son transmitidos junto con los bits de paridad generados por los dos codificadores (la salida sistemática del codificador inferior con interleaver no es utilizada y no se muestra en la figura). Así la tasa de código es $r=1/3$, no incluyendo los bits de cola. Los primeros $3K$ bits de salida del codificador son de la forma:

$$X_1, Z_1, Z_1', X_2, Z_2, Z_2', \dots, X_K, Z_K, Z_K'$$

donde X_k es el bit sistemático de datos k-esimo, Z_k es la salida de paridad del codificador de arriba sin interleaver y Z'_k es la salida de paridad del codificador inferior con interleaver.

La tasa de código también puede ser $r=1/2$ si se sacan todos los bits de paridad par del codificador superior y todos los bits de paridad impar del codificador inferior este proceso es denominado puncturing.

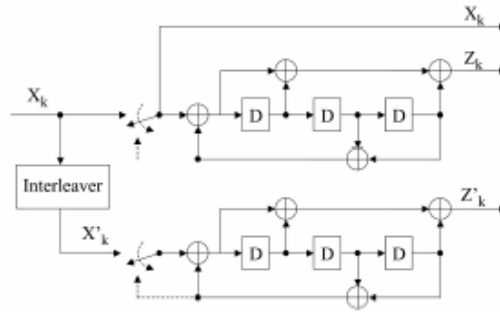


Figura 1: Codificador Turbo

Después que los K bits de datos han sido codificados, los trellis de ambos codificadores son forzados al estado de todos cero por la selección apropiada de bits de cola. Mientras que en los códigos convolucionales convencionales se agregan terminaciones de cola de todos ceros, los bits de colas de un RSC dependerán del estado del codificador. Debido a que el estado de los dos codificadores RSC usualmente serán diferentes después que los datos han sido codificados, las colas para cada codificador deben ser calculadas separadamente y transmitidas. Los bits de cola son generados para cada codificador llevando los dos switch a la posición inferior, de esa manera la entrada a los dos codificadores será la indicada por las líneas de punto. Los bits de cola son entonces transmitidos al final de la trama codificada de acuerdo a

$$X_{K+1}, Z_{K+1}, X_{K+2}, Z_{K+2}, X_{K+3}, Z_{K+3}, X'_{K+1}, Z'_{K+1}, X'_{K+2}, Z'_{K+2}, X'_{K+3}, Z'_{K+3},$$

donde X representa los bits de cola del codificador de arriba, Z representa los bits de paridad correspondientes a la cola del codificador de arriba, X' representa los bits de cola del codificador inferior y Z' representa los bits de paridad correspondientes a la cola del codificador inferior.

Así cuando los bits de cola son tenidos en cuenta el numero de bits codificados es $3K+12$, y la tasa de código es $K/(3K+12)$

Los datos codificados se asumen que serán enviados a través de un canal con AWGN, flat fading y modulación bpsk. La salida del filtro acoplado del receptor es $Y_k = a_k S_k + n_k$, donde $S_k = 2X_k - 1$ para los bits sistemáticos, $S_k = 2Z_k - 1$ para los bits de paridad del codificador de arriba, $S_k = 2Z'_k - 1$ para los bits de paridad del codificador inferior, a_k es la ganancia del canal ($a_k = 1$ para AWGN y es una variable aleatoria con distribución de Rayleigh para flat fading), n_k es el ruido gaussiano con varianza $\sigma^2 = 1/(2E_s / N_0) = (3K + 12)/(2K(E_b / N_0))$, E_s es la energía por bit de código, E_b es la energía por bit de dato y N_0 es la densidad espectral de ruido.

La entrada al decodificador se asume que será de la forma LLR (log-likelihood ratio), lo cual asegura que la ganancia del canal y la varianza del ruido han sido adecuadamente tomadas en cuenta. Así la entrada al decodificador será:

$$R_k = \log \left[\frac{P[S_k = +1|Y_k]}{P[S_k = -1|Y_k]} \right]$$

Aplicando la regla de Bayes y asumiendo que $P[S_k = +1] = P[S_k = -1]$

$$R_k = \log \left[\frac{f_y[Y_k|S_k = +1]}{f_y[Y_k|S_k = -1]} \right]$$

donde $f_y[Y_k|S_k]$ es la función de densidad de probabilidad condicional pdf de Y_k dado S_k , que es gaussiana con media $a_k S_k$ y varianza σ^2 .

Sustituyendo en la expresión de la pdf gaussiana y simplificando llegamos a:

$$R_k = \frac{2a_k}{\sigma^2} Y_k$$

Así los coeficientes del filtro acoplado deben ser multiplicados por un factor $\frac{2a_k}{\sigma^2}$ antes de ser enviado al decodificador. De acá en más, la notación $R(X_k)$ denotará la LLR recibida correspondiente al bit sistemático X_k ; $R(Z_k)$ denotará la LLR recibida correspondiente al bit de paridad superior Z_k , $R(Z'_k)$ denotará la LLR recibida correspondiente al bit de paridad inferior Z'_k .

La arquitectura del decodificador se muestra en la figura.

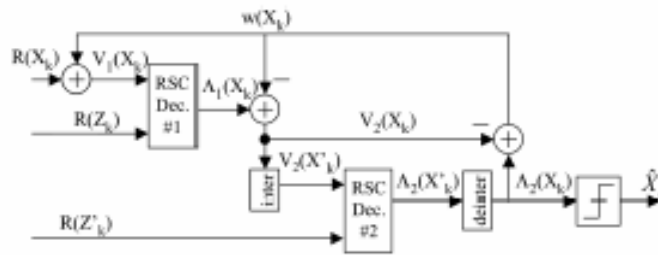


Figura 2: Decodificador Turbo

Como es indicado por la presencia del camino de realimentación el decodificador opera de manera iterativa. Cada iteración completa consiste en dos medias iteraciones, una para cada código RSC. El decodificador RSC #1 opera durante la primera media iteración, y RSC #2 opera durante la segunda media iteración.

El valor $w(X_k)$, con $1 \leq k \leq K$, es la información extrínseca producida por el decodificador #2 y luego introducida a la entrada del decodificador #1. Después de la primera iteración $w(X_k)$ es inicializada a todos ceros (puesto que el decodificador #2 aun no ha actuado sobre los datos). Después de cada iteración completa, los valores de $w(X_k)$ serán actualizados para reflejar la confianza depositada en los datos que se propagan del decodificador #2 al #1. Debido a que los

dos decodificadores tienen cola independiente, solo la información depositada en los bits de datos actuales es pasada entre los decodificadores. Así $w(X_k)$ no está definida para $K+1 \leq k \leq K+3$ (si estuviera definida sería simplemente igual a cero después de cada iteración).

La información extrínseca debe ser tenida en cuenta por el decodificador #1. Sin embargo debido a la forma en que las métricas de las ramas son obtenidas, es suficiente simplemente sumar $w(X_k)$ al LLR sistemático recibido $R(X_k)$, cuyo resultado es una nueva variable denotada por $V_1(X_k)$. Para $1 \leq k \leq K$, la entrada al decodificador #1 es una combinación de los bits de paridad recibidos en la forma LLR $R(Z_k)$, y los bits $V_1(X_k)$ obtenidos de la suma de los datos sistemáticos y la información extrínseca. Para $w(X_k)$, no hay información extrínseca disponible y por lo tanto la entrada al decodificador #1 son los bits de cola recibidos y escalados del codificador superior, $V_1(X_k) = R(X_k)$, y los bits de paridad recibidos y escalados, $R(Z_k)$. La salida del decodificador #1 es el LLR $\Lambda_1(X_k)$, donde $1 \leq k \leq K$ puesto que el LLR de los bits de cola no es compartido con el otro decodificador [7].

Sustrayendo $w(X_k)$ de $\Lambda_1(X_k)$, se obtiene una nueva variable denotada como $V_2(X_k)$. Similar a $V_1(X_k)$, $V_2(X_k)$ contiene la suma del LLR sistemático del canal y la información extrínseca producida por el decodificador #1 (observe sin embargo que la información extrínseca para el decodificador #1 nunca es explícitamente calculada. Para $1 \leq k \leq K$, las entradas al decodificador #2 son $V_2(X'_k)$, que es la versión con interleaving de $V_2(X_k)$, y $R(Z'_k)$, que es el LLR del canal correspondiente a los bits de paridad del segundo codificador. Para $K+1 \leq k \leq K+3$, la entrada al decodificador #2 son los bits de cola recibidos y escalados del codificador inferior, $V_2(X'_k) = R(X'_k)$, y los bits de paridad recibidos y escalados, $R(Z'_k)$. La salida del decodificador #2 es el LLR $\Lambda_2(X'_k)$, $1 \leq k \leq K$, el cual es de-interleaving para formar $\Lambda_2(X_k)$. La información extrínseca $w(X_k)$ es obtenida por sustracción de $\Lambda_2(X_k)$ menos $V_2(X_k)$ y es realimentada para usarla durante la siguiente iteración por el decodificador #1.

Una vez que las iteraciones han sido completadas, una decisión hard es tomada usando $\Lambda_2(X_k)$, $1 \leq k \leq K$, donde $\hat{X}_k = 1$ cuando $\Lambda_2(X_k) > 0$ y $\hat{X}_k = 0$ cuando $\Lambda_2(X_k) \leq 0$.

4. Metodología

En esta sección se describen sucintamente las estrategias involucradas en la construcción del modelo de simulación de un sistema codificador – decodificador turbo basado en las especificaciones de UMTS, cuyos resultados (performance de error) serán utilizados para realizar la comparación con los obtenidos mediante el algoritmo de Viterbi [8].

4.1 Implementación del Codificador Turbo

Se modelaron dos codificadores convolucionales sistemáticos recursivos (RSC) en paralelo. El generador feedforward es 15 (G1=1101) y el generador feedback es 13 (G0=1011), ambos en octal. Se llevo adelante una simulación con $K=5114$, respetando las especificaciones de UMTS que sugieren valores de K entre 40 y 5114. La tasa de código empleada es $r=1/2$ (puncturing).

4.2 Modelado del canal de salida soft.

Se modeló un canal con AWGN y modulación BPSK. La salida del filtro acoplado del receptor es $Y_k = a_k S_k + n_k$, donde a_k es la ganancia del canal representada por una variable aleatoria con distribución de Rayleigh para flat fading. Se utiliza $a_k = 1$ pues se considera que sólo hay presencia de AWGN, n_k es el ruido gaussiano con varianza $\sigma^2 = 1/(2E_s/N_0) = (3K+12)/(2K(E_b/N_0))$, E_s es la energía por bit de código o símbolo, E_b es la energía por bit de dato y N_0 es la densidad espectral de ruido.

Así los coeficientes del filtro acoplado deben ser multiplicados por un factor $\frac{2a_k}{\sigma^2}$ antes de ser enviados al decodificador.

4.3 Implementación del Decodificador Turbo

Se utilizó en la decodificación, una implementación del algoritmo Log-MAP [9].

El proceso de decodificación de turbo código comienza con la formación de “probabilidades a posteriori” (APPs) para cada bit de datos, la cual es usada para elegir el valor del bit que corresponde a la probabilidad máxima a posteriori (MAP) para ese bit de dato [10].

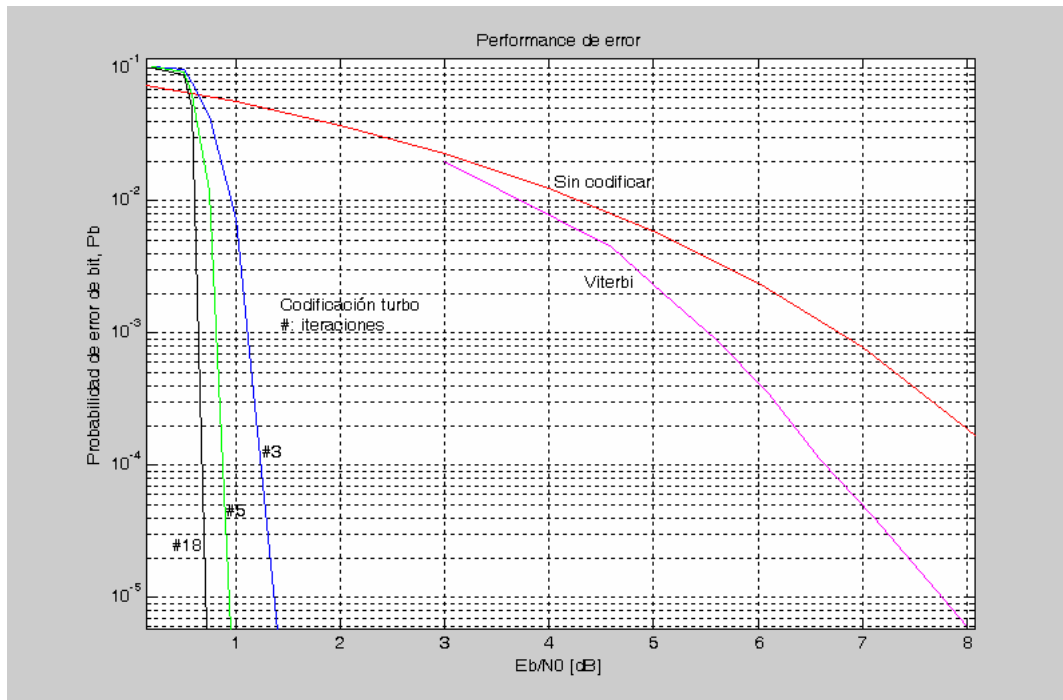
En la recepción de una secuencia de código de bit errónea, el proceso de decisión que se hace con APPs, permite al algoritmo log-MAP determinar el valor del bit más probable a ser transmitido. Esto es distinto al algoritmo Viterbi (VA), donde el APP para cada bit de dato no se puede alcanzar. En cambio, el VA encuentra la secuencia mas probable a ser transmitida. Hay, sin embargo, similitudes en la implementación de los dos algoritmos. Para la implementación del algoritmo log-MAP, se procede de la misma manera que el algoritmo de Viterbi en dos direcciones sobre un bloque de códigos de bits. Una vez que este cómputo bi-direccional es realizado, se obtienen los estados y las métricas de las ramas para el bloque.

5. Resultados de la simulación

Las simulaciones fueron corridas para ilustrar tres variantes del algoritmo de codificación Turbo, las cuales representan el resultado de la performance de error para #3, #5 y #18 iteraciones en un canal con AWGN y modulación BPSK. Además se simuló el comportamiento de un sistema de comunicación con codificación convolucional basado en el algoritmo de Viterbi, aplicado sobre el mismo canal. Finalmente se realizó la simulación del sistema sin ningún tipo de codificación. En la gráfica se aprecia la gran ventaja de los códigos Turbo, en cuanto a su mejor performance de error frente al algoritmo de Viterbi y frente a un esquema sin codificación.

Para un mismo valor de tasa de error de bits, como por ejemplo 1×10^{-4} , se observa que los codificadores Turbo necesitan de 5 a 6 dB menos de Eb/No (Energía de bit sobre densidad espectral de potencia de ruido), con respecto a los esquemas convolucionales de Viterbi (empleados habitualmente en la 2G de comunicaciones móviles, como por ejemplo GSM) y de hasta 8 dB con respecto a un sistema sin codificación.

Además se aprecia cómo mejora la performance de los esquemas con Código Turbo al aumentar la cantidad de iteraciones.



6. Conclusiones

Se ha analizado el algoritmo de Decodificación Turbo, el que fue implementado para simular un sistema de comunicaciones en un canal con AWGN y modulación BPSK.

Para proveer un ejemplo concreto esta simulación se enfocó en las especificaciones de los sistemas 3G de UMTS sobre Códigos Turbo. Se observa en los resultados cómo la codificación turbo logra mejorar iterativamente la performance de error del sistema propuesto, superando ampliamente al algoritmo de Viterbi y a los esquemas sin codificación.

8. Bibliografía

- [1] Proakis J., Digital Communication, Prentice Hall.
- [2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes", in Proc., IEEE Int. Conf. on Communications (Geneva, Switzerland, May 1993) pp. 1064-1070.
- [3] www.3GPP.org, Satellite Component of UMTS/IMT-2000; W-CDMA Radio interface for Multimedia Broadcast/Multicast Service (MBMS) ETSI TR 102227
- [4] Clint Smith, Daniel Collins, "3G Wireless Network CDMA2000 WCDMA/UMTS Network Architecture and Protocol 3G Network Design Methodologie, ISBN 0-07-136381-5.
- [5] Sklar Bernard, Digital Communications Fundamentals and Applications, Prentice Hall
- [6] M. C. Valenti, and J. Sun, "The UMTS Turbo Code and an Efficient Decoder Implementation suitable for Software-Defined Radios", International Journal of Wireless Information Networks, vol. 8, Nro 4, Oct 2001.
- [7] J. Hagenauer, E. Offer, and L. Papke, "Iterative Decoding of Binary Block and Convolutional Codes", IEEE Trans. Inform. Theory, vol 42, No 2, pp. 429-445, March 1996.
- [8] G.D Forney Jr. "The Viterbi Alghoritm", Proc IEEE Trans. On Inf Theory, +pp, 268-278, May.

- [9] Pietrobon, S., "Implementation and Performance of a Turbo/MAP Decoder," *Int'l. J. Satellite Commun.* Vol. 15, Jan-Feb 1998, pp. 23-46.
- [10] Bahl, L. R., Cocke J., Lelinek, F. and Raviv, J. "Optimal decoding of Linear Codes for Minimizing Symbol Error Rate," *Trans. Inform. Theory*, vol. IT-20, March 1974, pp 248-287.