

**ESTUDIO COMPARATIVO
ENTRE DISTINTOS MÉTODOS
DE ESTIMACIÓN DE LA
PROBABILIDAD DE ERROR**

Autor: José Luis Flores Aznar

Tutor: Francisco José Simois Tirado

Ingeniería de Telecomunicaciones

*A mi familia y amigos que han
apoyado mis estudios durante
todos estos años con su paciencia*

1	<i>Introducción</i>	3
1.1	Planteamiento del problema	3
1.2	Objetivos del Proyecto	5
1.3	Organización de la memoria	6
2	<i>Modelo de un sistema de transmisión</i>	8
2.1	Esquema general	8
2.2	Transmisor	10
2.3	Canal	11
2.4	Receptor	13
2.5	Filtro de Wiener	14
3	<i>Métodos de estimación de la probabilidad de error</i>	21
3.1	Algunas definiciones previas	21
3.2	El método de Monte Carlo	23
3.2.1	Fundamento teórico.....	24
3.2.2	Realización del Método	25
3.3	El muestreo enfatizado	27
3.3.1	Fundamento teórico.....	28
3.3.2	Muestreo enfatizado convencional (CIS)	30
3.3.3	Muestreo enfatizado mejorado (IIS).....	31
3.3.4	Muestreo enfatizado eficiente (EIS).....	32
3.4	Extrapolación de cola	33
4	<i>Simulaciones</i>	37
4.1	Simulación del método de Monte Carlo	37
4.1.1	Monte Carlo con canal AWGN.....	37
4.1.2	Monte Carlo con modelo de canal con eco simple.....	41
4.2	Simulación del método CIS	47

4.2.1	CIS con canal AWGN	47
4.2.2	CIS con modelo de canal con eco simple	49
4.3	Simulación del método IIS	62
4.4	Simulación del método EIS	69
4.5	Simulación del método de extrapolación de cola.....	80
5	<i>Conclusiones y futuras líneas de investigación.....</i>	<i>103</i>
	<i>Anexo.....</i>	<i>105</i>
	<i>Referencias</i>	<i>124</i>

1 INTRODUCCIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA

En la actualidad las transmisiones digitales están presentes en muchas de las actividades cotidianas de nuestras vidas. Progresivamente, y en la mayoría de las aplicaciones, las transmisiones digitales están sustituyendo a las analógicas por sus mayores ventajas generales.

Pero, ¿cuáles son las diferencias entre una transmisión digital y una analógica? Principalmente, la diferencia radica en el lenguaje, por así llamarlo, en el que se produce la comunicación. Mientras que una transmisión analógica podría asemejarse a la comunicación vocal que usamos los seres humanos, la digital podría decirse que es mucho más primitiva, ya que su funcionamiento se basa en sólo dos letras, por así llamarlas: el 1 y el 0.

A primera vista, esta sencillez es contraria a los cada vez más complicados sistemas ya no sólo de comunicación, sino también de información. Pero su sencillez no debe verse como una limitación, sino como una potente ventaja, según vamos a ir detallando a continuación.

En las transmisiones de información el principal escollo a salvar suele ser el ruido que interfiere en la comunicación y que puede degradar la información hasta el punto de hacer ininteligibles los datos que se reciben. Este ruido no está presente exclusivamente en las transmisiones inalámbricas, sino que también hace acto de presencia en medios como cables, guíasondas, etc.

Por lo tanto, interesa tener un método de transmisión que sea inmune al ruido, o al menos lo menos sensible posible a sus efectos. Esta cualidad la cumple un sistema digital, pues debido a su sencillo lenguaje sólo existen 2 símbolos posibles, el 1 y el 0. Bastaría con transmitir dichos unos y ceros con niveles de energía lo suficientemente altos como para que, por muy fuerte que fuera el ruido reinante en el ambiente, no se

podieran confundir entre sí. A esta propiedad de poder obtener la misma señal en recepción que la que fue transmitida se la denomina **capacidad de regeneración**.

Por otro lado, el diseño modular que permiten los sistemas digitales redundan en una mayor **flexibilidad** así como en un menor **precio**. Además, los sistemas digitales son, por lo general, más **fiables**.

Mientras que las señales analógicas tienen características muy distintas según la información a transmitir, las de una señal digital son iguales independientemente del tipo de información transmitida; es lo que se denomina la **uniformidad del formato** de las señales digitales. Esto supone que un mismo sistema se puede usar para diversas situaciones, con el consecuente ahorro de recursos económicos y flexibilizando al máximo el uso del equipo.

Por último, una ventaja muy útil en el uso compartido de los medios de transmisión es la sencilla **multiplexación** en el tiempo y frecuencia que permiten los sistemas digitales. Esta ventaja permite la transmisión de varias señales como si fueran una sola compartiendo el medio de transmisión, con el único requisito de que el receptor sea capaz de discriminar las señales a la perfección.

Pues bien, a la hora de implantar un nuevo sistema de comunicación digital es necesario hacer un análisis previo de las condiciones en las que va a funcionar. Mediante el estudio de las condiciones del canal, las características de los equipos transmisores y, en general, de las particularidades de cada caso, pueden ser desechadas múltiples opciones no recomendables.

La probabilidad de error es uno de los parámetros clave en casi todos los sistemas de transmisión, así que en el estudio previo que se realiza es importante conocer el valor que se quiere conseguir de la misma para el sistema en concreto. Dependiendo del valor deseado el sistema deberá ser más o menos estricto; a menor probabilidad de error deseada, más rigurosas deberán ser las condiciones a imponer. Por ejemplo, las características del canal pueden desaconsejar el uso de determinadas tecnologías y tipos de modulación. Debemos recordar que hay muchos tipos de canales

y muy diferentes condiciones externas dentro incluso de un mismo tipo de canal. Las condiciones atmosféricas, interferencias externas, naturaleza del propio medio, etc, pueden ser determinantes a la hora de diseñar el sistema.

Una vez que el sistema a instalar ha sido definido, llega la hora de la simulación. La simulación es una herramienta muy útil y potente a la hora de saber cómo se va a comportar el sistema. Para ello se suelen usar potentes computadores y aplicaciones diseñadas específicamente; permiten constatar de forma fiable que el sistema va a funcionar tal y como el diseñador ha previsto. En nuestro caso, la simulación nos va a servir para comprobar que la probabilidad de error en los datos transmitidos no supera un cierto margen determinado. Para ello es necesario poder caracterizar adecuadamente el tipo de canal con el que se va a trabajar.

La principal limitación viene impuesta por el número de muestras necesarias. Cuando se simulan probabilidades de error muy pequeñas con los métodos tradicionales es preciso un gran número de muestras. Existe una regla general y muy extendida (que ya tendremos tiempo de comentar más adelante en mayor profundidad) según la cual para obtener probabilidades del orden de P_e es necesario usar al menos $10/P_e$ muestras. Por ejemplo, en el caso de $P_e = 10^{-8}$ se necesitarían 10^9 muestras, cantidad excesiva incluso para los potentes ordenadores actuales.

1.2 OBJETIVOS DEL PROYECTO

En este proyecto se muestran diversas técnicas de simulación que, frente a las tradicionales, reducen el número de muestras necesarias para la estimación de las probabilidades de error, con los consecuentes ahorros en tiempo y dinero.

Uno de estos métodos es el llamado **muestreo enfatizado** o *importance sampling*, que acelera el proceso de la simulación mediante la introducción de fuentes de ruido no deseadas que hacen que los errores ocurran con una mayor frecuencia que en las condiciones normales. Para que este método no falsee los resultados es necesario ponderar adecuadamente los errores.

Se pueden distinguir varias formas de llevar a cabo el método del muestreo enfatizado:

- ✓ *Conventional Importance Sampling* (IIS): modificando la media del ruido introducido en el canal.
- ✓ *Improved Importance Sampling* (CIS): modificando la desviación típica del ruido introducido.
- ✓ *Efficient Importance Sampling* (EIS): modificando la desviación típica del ruido introducido, pero trabajando con dicho ruido considerado justo antes del decisor existente en el receptor.

Otro método posible es el método de **extrapolación de colas** o *tail extrapolation*. Este método consiste en estimar una probabilidad de error mediante la extrapolación (de ahí su nombre) de los resultados obtenidos en la estimación de probabilidades de error mayores, las cuales se corresponden con simulaciones más rápidas debido a que requieren menos muestras.

1.3 ORGANIZACIÓN DE LA MEMORIA

El presente documento está dividido en 5 capítulos:

En el capítulo 2 se realiza una introducción teórica a los sistemas de comunicación. En este apartado se realizará una descripción de un modelo de sistema de transmisión, detallando los bloques que en él intervienen, así como una definición de las características que determinan el funcionamiento del mismo, como son el tipo de modulación y el modelo de canal empleado. Por último se realiza una breve introducción a los filtros de Wiener, empleados en el desarrollo del proyecto.

En el capítulo 3 se detalla el fundamento teórico de cada uno de los métodos implementados para la estimación de la probabilidad de error, así como el funcionamiento de cada uno de ellos.

En el capítulo 4 se especifican las condiciones de las simulaciones llevadas a cabo así como los resultados de las mismas.

Por último se realiza un resumen de las conclusiones obtenidas y se sugieren futuras líneas de investigación a desarrollar en este campo.

Se incluye un anexo en el que se listan los principales códigos implementados para la realización de las simulaciones.

2 MODELO DE UN SISTEMA DE TRANSMISIÓN

2.1 ESQUEMA GENERAL

Cualquier sistema de transmisión, bien sea analógico o digital, puede ser representado por el modelo de la figura 2.1. En esta figura se pueden apreciar tres bloques bien diferenciados:



Figura 2.1: modelo de un sistema de transmisión.

- ✓ Transmisor: consta de la fuente de información así como de los diversos procesos a los que es sometida dicha información para poder ser transmitida, adaptándola óptimamente al canal por el que va a propagarse.
- ✓ Canal: representa al medio por el que se propaga la información y que provoca todas las posibles transformaciones que ésta puede sufrir durante la propagación.
- ✓ Receptor: se compone de todos los bloques que extraen la información de la señal transmitida y que tienen como objetivo proporcionar dicha información al destinatario en un formato inteligible para el mismo.

Cada bloque se puede dividir en otros con funciones específicas que simplifican el estudio del sistema completo. En la figura 2.2 se refleja un modelo más detallado [1], que a continuación explicamos.

La **fuente de información** es el origen de los datos, los cuales pueden ser de naturaleza analógica o digital.

Si la señal es analógica, en el **codificador de fuente** se realiza el paso a digital. Además, a la señal digital se la somete a un proceso en el cual se elimina la redundancia

con el fin de reducir el volumen de la información no necesaria (por ejemplo se quitaría la letra u que en castellano siempre sigue a la consonante q, dado que no aporta ninguna información). Se realiza a veces también una encriptación de los datos para evitar que posibles intrusos puedan entender la información transmitida.

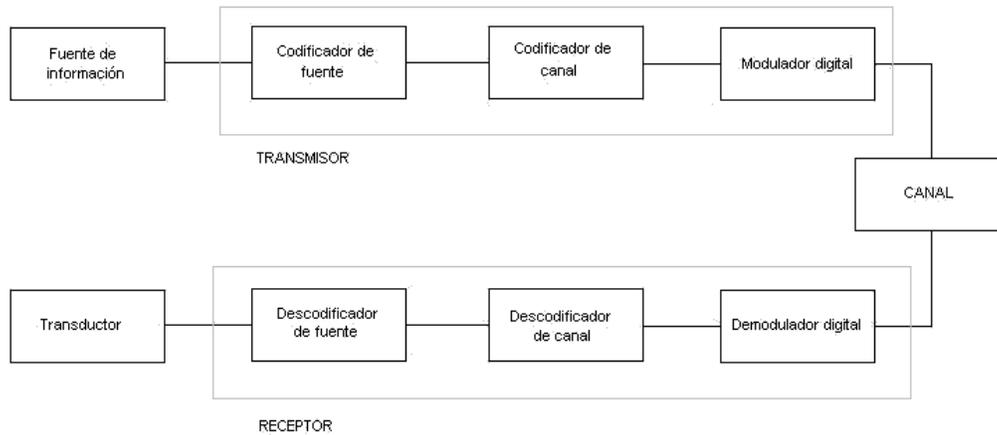


Figura 2.2: modelo detallado de un sistema de transmisión.

El **codificador de canal** introduce información redundante de forma controlada con el fin de garantizar una correcta recepción de los datos.

Para poder transmitir la secuencia digital (compuesta exclusivamente de unos y ceros) por un canal físico es necesario asociar la información a una señal, que será la que posteriormente se transmita. Ésta es la función del **modulador digital**.

En recepción, el proceso es el contrario: se va realizando un procesado de la señal física recibida hasta que la información es ofrecida al usuario en el formato deseado.

El destino de la información necesitará normalmente un dispositivo, como por ejemplo un transductor, que transforme la señal recibida en una secuencia entendible por él.

2.2 TRANSMISOR

El tipo de modulación elegida en nuestro proyecto es la **BPSK** (*Binary PSK*). Esta modulación es muy sencilla y no realiza ningún tipo de agrupación de los bits en símbolos más complejos. En ellas sólo se transmiten dos símbolos, los correspondientes al 1 y al 0.

En esta modulación, al símbolo 1 le corresponde la señal

$$s_1(t) = \sqrt{\frac{2E_b}{T_b}} \cdot \cos(2\pi f_c t), \quad (2.1)$$

mientras que, para el símbolo 0,

$$s_2(t) = \sqrt{\frac{2E_b}{T_b}} \cdot \cos(2\pi f_c t + \pi) = -\sqrt{\frac{2E_b}{T_b}} \cdot \cos(2\pi f_c t). \quad (2.2)$$

En estas expresiones, E_b es la **energía transmitida por bit** y T_b es la **duración del bit**, mientras que f_c es la frecuencia de la portadora.

Tomando como función base

$$\phi_1(t) = \sqrt{\frac{2}{T_b}} \cos(2\pi f_c t), \quad (2.3)$$

podemos describir las señales asociadas a los símbolos 1 y 0 como

$$s_1(t) = \sqrt{E_b} \phi_1, \quad (2.4)$$

$$s_2(t) = -\sqrt{E_b} \phi_1. \quad (2.5)$$

2.3 CANAL

Para la realización de las simulaciones se usarán dos modelos de canal: uno más sencillo (canal AWGN) en el que sólo se considera la influencia del ruido para comprobar que los algoritmos están bien diseñados, y otro modelo en el que además del ruido se tiene en cuenta el efecto de la **memoria** de los canales con eco simple. En las figuras 2.5 y 2.6 se pueden observar ambos modelos de canal.

En ambos tipos de canal el ruido que se añade a la señal se supone blanco, gaussiano y de media cero. El ruido blanco se caracteriza por poseer una densidad espectral de potencia plana; es decir, la densidad espectral de potencia del ruido es la misma para toda las frecuencias y vendrá dada por

$$S_w = \frac{N_o}{2}, \quad (2.6)$$

donde N_o está referido a la entrada del receptor, y se mide en W/Hz.

Por otra parte, que el ruido sea gaussiano implica que su función de densidad de probabilidad (también denominada pdf, *probability density function*) es gaussiana, en cuyo caso se tiene que

$$f_N(n) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(n-m_n)^2}{2\sigma^2}\right), \quad (2.7)$$

donde m_n es la media de la variable aleatoria y σ es su desviación típica.

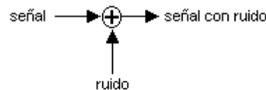


Figura 2.3: modelo de canal AWGN.

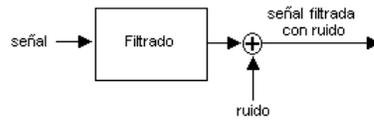


Figura 2.4: modelo de canal con eco simple.

Las razones por la que escogemos funciones de densidad de probabilidad gaussianas son dos principalmente. Por un lado, permiten un fácil manejo matemático y, por otro lado, gran parte de los procesos aleatorios provocados por fenómenos físicos se pueden aproximar con bastante exactitud a este tipo de funciones [2].

Por otra parte, el filtrado que se realiza en el canal con eco simple se hace a través de un filtro de orden uno cuya expresión es

$$H(z) = 1 + \alpha \cdot z^{-1}, \quad (2.8)$$

siendo α un parámetro que haremos variar desde 0 hasta 1 con el fin de hacer comparaciones, y que es el causante del eco.

El efecto de la memoria supone que en la decisión que se hace en recepción no sólo influya el símbolo de ese instante, sino que también afecte el anterior. Posibles causas de este efecto son las reflexiones de la señal en el interior del medio físico (cable, aire, etc.) que hacen que la información llegue múltiples veces (en nuestro caso sólo dos), cada vez más atenuada.

Es evidente que estos efectos no son deseables, pues dificultan el entendimiento entre el transmisor y el receptor, pero vienen impuestos por la naturaleza del canal y la única opción que existe para reducir sus efectos negativos es introducir filtrados adecuados en el bloque receptor.

2.4 RECEPTOR

En realidad la elección del receptor viene impuesta por el transmisor que se haya escogido. En este caso, en el que se ha optado por una modulación BPSK, el receptor también será muy sencillo, pues sólo tiene que distinguir entre dos tipos de símbolos transmitidos: los que representan al nivel alto y los que lo hacen al nivel bajo.

En el receptor, la forma de decidir el símbolo que se ha transmitido es sencilla y se basa en la división en regiones del espacio en el que pueden encontrarse las señales recibidas. En nuestro caso existen dos símbolos, por lo que se distinguen dos regiones asociadas, una a cada uno de los símbolos que pueden ser transmitidos.

Por tanto, para poder distinguir entre el 1 y el 0 es necesario imponer en recepción un **umbral de decisión**. Dicho umbral no es más que la frontera que separa la región correspondiente a cada símbolo. Si los símbolos son equiprobables (no siempre ocurrirá esto, pero en nuestro caso vamos a suponer que sí), este umbral estará en la mitad del intervalo que hay entre los mismos.

Si la señal recibida está dentro de la zona asociada al símbolo que fue transmitido, la recepción será correcta. En caso de que, por el contrario, dicha señal sea recibida en la región contraria, el bloque decisor interpretará que el símbolo transmitido ha sido el asociado a la zona en cuestión, produciéndose así un error. El que un símbolo transmitido se desplace hasta ser recibido en la región contraria será debido al efecto del canal.

Veámoslo con un ejemplo: si en transmisión se ha asociado una señal de 3V con el '1' y una de -3V con el '0', es evidente que el umbral de decisión teórico debería estar en el nivel 0V. De esta manera, si se ha transmitido un 1 pero debido al canal dicho nivel disminuye hasta caer por debajo del umbral (-0.2V por ejemplo), entonces en recepción se interpretará como un 0, produciéndose así una decisión errónea en la secuencia proporcionada al destino de la información.

Cuanto mayor sea la diferencia entre los niveles de tensiones asociados al 1 y 0, menor probabilidad de error tendremos. El problema es que esta solución supone un aumento de la potencia necesaria en transmisión, opción que no siempre es posible.

Cuando el canal utilizado es AWGN (sin eco), la probabilidad de error teórica viene dada por [1]

$$P_e = \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right). \quad (2.9)$$

Podemos ver así que la probabilidad de que se produzca un error es función de la energía de bit y del ruido presente en el canal. Evidentemente, a mayor ruido, mayor será la probabilidad de error, y en cambio ésta será menor cuanto mayor sea la energía con la que se haga la transmisión.

En la figura (2.5) se representa la probabilidad de error en función del cociente E_b/N_0 . En ella se observa cómo a medida que crece dicho cociente la probabilidad de error disminuye.

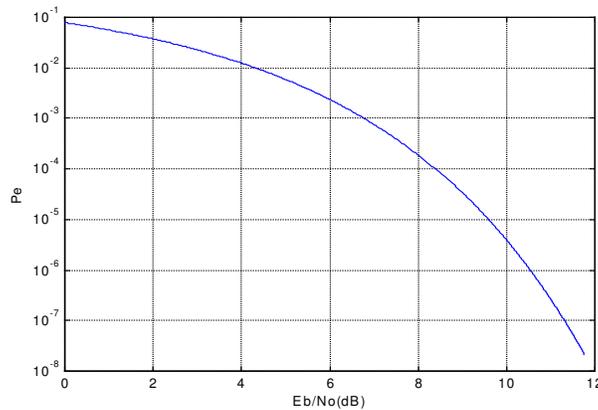


Figura 2.5: curva teórica de la probabilidad de error en la modulación BPSK.

2.5 FILTRO DE WIENER

Un filtro de Wiener es un filtro lineal óptimo que nosotros usaremos para contrarrestar los efectos no deseables causados por el modelo de canal con eco simple.

Como ya hemos visto, el modelo de este tipo de canal incluye un filtrado cuyo efecto es necesario paliar en la medida de lo posible para que la calidad de la transmisión no empeore.

Una forma de conseguir este objetivo es incluyendo un filtro de Wiener en el receptor, con el cual se buscará que la convolución de su respuesta impulsiva con la del filtro incluido en el modelo del canal sea lo más parecida posible a la función impulso unidad, puesto que esto supondría que el efecto conjunto de ambos filtros es nulo.

Consideremos el diagrama de bloques de la figura 2.6.

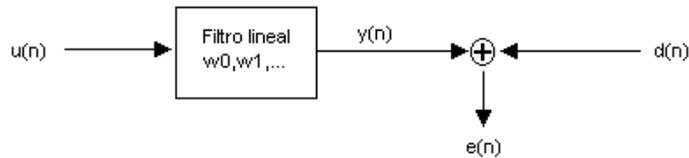


Figura 2.6: diagrama de bloques del problema estadístico del filtrado.

La entrada del filtro es la secuencia $u(n)$, y el filtro está caracterizado por la respuesta impulsiva w_0, w_1, w_2, \dots . La salida del filtro será la secuencia $y(n)$. Denominando $d(n)$ a la respuesta deseada (que en nuestro caso particular, como hemos dicho, será la función impulso unidad), podremos calcular el error cometido, representado por la secuencia $e(n)$. Nuestro objetivo es hacer que dicho error sea lo menor posible [3].

Es posible expresar la salida del filtro en un instante determinado n mediante la suma de convolución, de la forma

$$y(n) = \sum_{k=0}^{\infty} w_k^* \cdot u(n-k), \quad (2.10)$$

donde el asterisco significa conjugación compleja. Hemos visto que la estimación de la respuesta deseada conlleva un error

$$e(n) = d(n) - y(n). \quad (2.11)$$

Este error es una muestra de una variable aleatoria en un instante determinado.

Con el fin de optimizar el diseño del filtro es necesario minimizar este error. Aunque existen muchos criterios para ello, el más habitual, y el que vamos a escoger nosotros, es el de minimizar el valor cuadrático medio [3], por lo que podemos definir una función de coste de la forma

$$J = E[e(n)e^*(n)] = E[|e(n)|^2], \quad (2.12)$$

donde $E[\cdot]$ representa la esperanza matemática. Es necesario hallar las condiciones para las cuales se minimiza la función de coste definida, pues serán éstas las condiciones que se deban cumplir para que el filtro sea óptimo.

En general, los datos de entrada al filtro serán complejos, por lo que los coeficientes del filtro también lo serán. Podemos expresar por tanto el coeficiente k -ésimo del filtro de la forma

$$w_k = a_k + jb_k. \quad (2.13)$$

De acuerdo con ello, definimos el operador gradiente como

$$\nabla_k = \frac{\partial}{\partial a_k} + j \frac{\partial}{\partial b_k}, \quad k = 0, 1, 2, \dots \quad (2.14)$$

Aplicando este operador a la función de coste definida en (2.12) obtenemos que

$$\nabla_k J = \frac{\partial J}{\partial a_k} + j \frac{\partial J}{\partial b_k}, \quad k = 0, 1, 2, \dots \quad (2.15)$$

Para que esta definición sea válida es necesario que J sea estrictamente real.

Para que la función de coste tome un valor mínimo es necesario imponer que la función gradiente de la misma sea igual a cero, es decir,

$$\nabla_k J = 0, \quad k = 0, 1, 2, \dots \quad (2.16)$$

Dicha igualdad se puede separar en dos ecuaciones, una para la parte real y otra para la parte imaginaria, de la forma

$$\begin{aligned} \frac{\partial J}{\partial a_k} &= 0, \\ \frac{\partial J}{\partial b_k} &= 0. \end{aligned} \quad (2.17)$$

Imponiendo estas condiciones obtendremos un filtro óptimo en el sentido del mínimo error cuadrático medio.

Por otro lado, podemos escribir el gradiente de la función de coste como

$$\nabla_k J = E \left[\frac{\partial e(n)}{\partial a_k} e^*(n) + \frac{\partial e^*(n)}{\partial a_k} e(n) + j \frac{\partial e(n)}{\partial b_k} e^*(n) + j \frac{\partial e^*(n)}{\partial b_k} e(n) \right]. \quad (2.18)$$

A partir de las ecuaciones (2.10), (2.11) y (2.13) podemos obtener los términos en derivadas parciales de la anterior expresión. Así, llegamos a que

$$\begin{aligned} \frac{\partial e(n)}{\partial a_k} &= -u(n-k), \\ \frac{\partial e(n)}{\partial b_k} &= ju(n-k), \\ \frac{\partial e^*(n)}{\partial a_k} &= -u^*(n-k), \\ \frac{\partial e^*(n)}{\partial b_k} &= -ju^*(n-k). \end{aligned} \quad (2.19)$$

Sustituyendo estas relaciones en (2.18) obtenemos que

$$\nabla_k J = -2E[u(n-k)e^*(n)]. \quad (2.20)$$

Si denominamos $e_o^*(n)$ al valor de la estimación del error en el caso óptimo, es decir, cuando toma su valor mínimo, entonces podemos afirmar que

$$E[u(n-k)e_0^*(n)] = 0, \quad k = 0, 1, 2, \dots \quad (2.21)$$

En otras palabras, la condición necesaria y suficiente para que la función de coste definida tome su valor mínimo es que el error sea ortogonal a la secuencia de entrada al filtro de Wiener. Esta conclusión es el llamado **principio de ortogonalidad**. Haciendo uso de él obtenemos que

$$E[y(n)e_0^*(n)] = E\left[\sum_{k=0}^{\infty} w_k^* u(n-k)e_0^*(n)\right] = \sum_{k=0}^{\infty} w_k^* \cdot E[u(n-k)e_0^*(n)] = 0. \quad (2.22)$$

Es decir, el error también es ortogonal a la salida del filtro de Wiener.

Por otra parte, usando (2.10) y (2.11) podemos describir el principio de ortogonalidad como

$$E\left[u(n-k)\left(d^*(n) - \sum_{i=0}^{\infty} w_{0i} u^*(n-i)\right)\right] = 0, \quad (2.23)$$

donde w_{0i} es el i -ésimo coeficiente de la respuesta impulsiva del filtro óptimo. Reordenando esta ecuación obtenemos que

$$\sum_{i=0}^{\infty} w_{0i} E[u(n-k)u^*(n-i)] = E[u(n-k)d^*(n)] \quad k = 0, 1, 2, \dots \quad (2.24)$$

Las esperanzas que aquí aparecen pueden ser interpretadas de la siguiente forma:

- ✓ La primera de ellas es la función autocorrelación de la entrada del filtro para un retraso de $i - k$. Entonces puede ser expresada como

$$r(i-k) = E[u(n-k)u^*(n-i)]. \quad (2.25)$$

- ✓ El término $E[u(n-k)d^*(n)]$ es la correlación cruzada entre la entrada del filtro, $u(n-k)$, y la respuesta deseada, $d(n)$, para un retraso de $-k$. Podemos expresarla entonces de la forma

$$p(-k) = E[u(n-k)d^*(n)]. \quad (2.26)$$

Utilizando las ecuaciones (2.25) y (2.26) podemos escribir (2.24) como

$$\sum_{i=0}^{\infty} w_{oi} r(i-k) = p(-k), \quad k = 0, 1, 2, \dots \quad (2.27)$$

Este es un sistema de ecuaciones que permite calcular los coeficientes del filtro óptimo. Reciben el nombre de **ecuaciones de Wiener-Hopf**.

Tal y como están escritas en (2.27), las ecuaciones de Wiener-Hopf proporcionan un filtro óptimo de longitud infinita (IIR), es decir, con un número infinito de coeficientes. En nuestro caso, en cambio, vamos a usar un filtro de longitud finita (FIR), en concreto con M coeficientes, es decir, de orden $M-1$, por lo que las ecuaciones de Wiener-Hopf serán

$$\sum_{i=0}^{M-1} w_{oi} r(i-k) = p(-k), \quad k = 0, 1, 2, \dots, M-1. \quad (2.28)$$

Estas ecuaciones pueden ser formuladas en forma matricial. Sea \mathbf{R} la matriz de correlaciones de la entrada al filtro, definida como

$$\mathbf{R} = E[\mathbf{u}(n)\mathbf{u}^H(n)], \quad (2.29)$$

donde el superíndice H denota una matriz hermítica, es decir, traspuesta y conjugada, y $\mathbf{u}(n)$ es el vector de entrada de dimensión $M \times 1$, definido por

$$\mathbf{u}(n) = [u(n), u(n-1), \dots, u(n-M+1)]^T, \quad (2.30)$$

donde T indica la operación de trasposición. Denotando con el vector $\mathbf{p}(n)$ a la correlación cruzada entre la entrada del filtro y la respuesta deseada, obtenemos que

$$\mathbf{p}(n) = E[\mathbf{u}(n)d^*(n)] = [p(0), p(-1), \dots, p(1-M)]^T. \quad (2.31)$$

Con todo esto las ecuaciones de Wiener-Hopf pueden ser rescritas de la forma

$$\mathbf{R}\mathbf{w}_o = \mathbf{p}, \quad (2.32)$$

donde $\mathbf{w}_o = [w_{o0}, w_{o1}, \dots, w_{oM-1}]^T$ es el vector que contiene los coeficientes óptimos del filtro de Wiener. Finalmente, despejando obtenemos que

$$\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}. \quad (2.33)$$

Este sencillo cálculo permite conseguir un filtro cuya característica permitirá anular, o al menos disminuir en gran parte, los efectos del eco que se introduce en el canal. Con él conseguiremos que la respuesta impulsiva conjunta del filtro y el canal sea lo más parecida posible a una función impulso unidad, lo que significaría que el efecto del eco ha sido contrarrestado.

3 MÉTODOS DE ESTIMACIÓN DE LA PROBABILIDAD DE ERROR

3.1 ALGUNAS DEFINICIONES PREVIAS

A lo largo del proyecto utilizaremos como esquema el presentado en la figura (3.1). En ella podemos ver la nomenclatura que usaremos a lo largo de la memoria

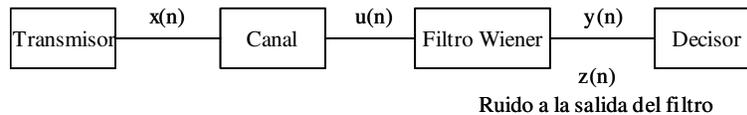


Figura 3.1: esquema de trabajo.

Como ya hemos comentado, la probabilidad de error es el número de errores que se producen en una determinada secuencia. Usualmente lo que interesa es, en concreto, la probabilidad o tasa de error de bit (BER, *bit error rate*), pero no es la única forma de medir los errores; en determinados sistemas puede ser de más utilidad la caracterización del sistema en función del número de símbolos o de otro tipo de bloques recibidos libres de errores. En cualquier caso, en nuestro proyecto sólo nos ocuparemos de la BER, que además coincidirá con la probabilidad de error de símbolo dado que la modulación utilizada (BPSK) es binaria.

La probabilidad de error está determinada por dos factores: la probabilidad de que se produzca un error suponiendo que se ha transmitido un 1, y la de que se produzca si el símbolo que se ha transmitido es un 0. Este razonamiento podemos expresarlo matemáticamente mediante la ecuación

$$p = \pi_1 \cdot p_1 + \pi_0 \cdot p_0, \quad (3.2)$$

donde p es la probabilidad de error total, p_i la probabilidad de que se produzca el error al transmitirse el símbolo i y π_i la **probabilidad a priori** de que se transmita el símbolo i . En condiciones normales, las probabilidades a priori son iguales para los dos símbolos

posibles (1,0), pues en principio ninguno de ellos es más probable que el otro. En ese caso $\pi_i = 0.5$.

En general, la probabilidad de que exista un error cuando se ha transmitido un cero es la misma que cuando se ha transmitido un uno (hablamos en este caso de que tenemos un **canal binario simétrico**), aunque en algunas situaciones esta regla no es válida. En nuestro caso, no obstante, sí vamos a suponer que se cumple, por lo que la probabilidad de error total coincide con la probabilidad de que ocurra un error al transmitir un 1 o un 0, como se puede apreciar en

$$p = \pi_1 \cdot p_1 + \pi_0 \cdot p_0 = 0.5 \cdot (p_1 + p_0) = p_i, \quad i = 1, 2. \quad (3.3)$$

De acuerdo con esto, y a la hora de evaluar la bondad de los distintos métodos de estimación de la probabilidad de error, será necesario considerar dos aspectos:

- ✓ La probabilidad de error estimada debe ser lo más próxima posible a la teórica. Aquí, en general, podemos distinguir métodos **insesgados**, que son aquellos en los que la probabilidad estimada es más cercana a la teórica a medida que aumenta el número de muestras usado en la estimación; y métodos **sesgados**, que no cumplen dicha regla. A lo largo del proyecto veremos ejemplos de ambos tipos.
- ✓ La *varianza* de los valores estimados debe ser baja. O dicho, de otro modo, la dispersión de las probabilidades de error obtenidas en cada simulación alrededor del valor promedio debe ser lo más pequeña posible, pues esto aumenta la confianza en que nuestra estimación ha sido buena.

Para poder establecer comparaciones en lo que respecta a la varianza de la estimación vamos a definir el **producto tiempo-fiabilidad** (*time-reliability product*) [2], que viene dado por

$$\xi = N \cdot \sigma^2, \quad (3.4)$$

siendo N el número de muestras utilizado en la simulación y σ la desviación típica de la estimación. Por tanto, aquel método que presente un menor valor del producto tiempo-

fiabilidad será el mejor de todos, si además es insesgado. Más tarde veremos que la elección del método óptimo no es inmediata dado que dependerá de muchos factores.

Con objeto de apreciar las mejoras introducidas por un método respecto de otro definimos el cociente [6]

$$\eta = \frac{\xi_1}{\xi_2}, \quad (3.5)$$

que intuitivamente podemos interpretar como la **ganancia** que introduce el método 2 respecto del 1. Por tanto, cuanto mayor sea esta ganancia, menor será el producto tiempo-fiabilidad del segundo método frente al primero, lo cual siempre será positivo.

En general, las comparaciones las estableceremos respecto del método de Monte Carlo (que analizamos en el apartado 3.2), por lo que en la definición (3.5) tendremos que ξ_1 será siempre la correspondiente a dicho método.

En resumen, en los métodos insesgados el parámetro que toma mayor relevancia a la hora de establecer comparaciones será el producto tiempo-fiabilidad, mientras que en los sesgados habrá que valorar también la exactitud de la estimación.

3.2 EL MÉTODO DE MONTE CARLO

Se trata de un método muy sencillo y de contrastado funcionamiento. No obstante, como ya se apuntó al principio del presente proyecto, tiene un inconveniente importante: el elevado número de muestras necesarias para estimar de forma precisa probabilidades de error pequeñas, del orden de $10/P_e$. Teniendo en cuenta que en los sistemas actuales no es extraño tratar de conseguir probabilidades de error de 10^{-8} o menores, vemos que dicho inconveniente limita la utilización de este sencillo método. Sin embargo, para probabilidades de error no demasiado pequeñas el método es completamente recomendable.

3.2.1 FUNDAMENTO TEÓRICO

Por definición, la probabilidad de que se produzca un error habiéndose transmitido un símbolo en concreto viene dada por [2]

$$p_i = \int_R f_Y(y) dy, \quad (3.6)$$

donde $f_Y(y)$ es la función de densidad de probabilidad de la señal recibida; y la región R a la que se extiende la integral es la correspondiente al símbolo contrario al transmitido, que es la región que determina el error.

En el caso de que se haya transmitido un 1 y suponiendo símbolos equiprobables y canal simétrico, la probabilidad de error vendrá determinada por

$$p = \int_{R_1} f_Y(y) dy = \int_{-\infty}^0 f_Y(y) dy, \quad (3.7)$$

siendo R_1 la zona correspondiente a la transmisión de un 0, es decir, la región del error. Esta ecuación puede describirse como

$$p = \int_{-\infty}^{\infty} H(y) f_Y(y) dy, \quad (3.8)$$

donde la función $H(y)$ se denomina **función indicadora de error**, y toma el valor 1 si y pertenece a la región R_1 (en este ejemplo) y 0 si no pertenece a ella. Como su propio nombre indica, esta función se encarga de señalar cuándo se ha producido un error.

La ecuación (3.8) puede expresarse de la forma

$$p = E[H(Y)], \quad (3.9)$$

siendo $E[\cdot]$ el operador esperanza, evaluado sobre una función de la variable aleatoria Y , con función de densidad de probabilidad $f_Y(y)$.

De esta última ecuación se desprende que la probabilidad de error es la media de la función indicadora de error $H(y)$. Un estimador de la media es

$$\hat{p} = \frac{1}{N} \sum_{i \in I_o} H(y_i), \quad (3.10)$$

donde I_o es el conjunto de los i para los cuales $H(y_i)$ toma valor 1 (es decir, ha habido error) y N es el número de elementos de la muestra.

En resumen, el método de Monte Carlo obtiene la probabilidad de error simplemente promediando el número de errores ocurridos sobre el número total de muestras con el que se hace la simulación. Es interesante hacer notar que este método es insesgado, es decir, se cumple que $E(\hat{p}) = p$ [2].

La limitación que presenta Monte Carlo deriva de la función indicadora de error, ya que si su valor es cero para todas las muestras la probabilidad estimada será nula, lo cual puede ocurrir con relativa facilidad cuando se estiman probabilidades de error pequeñas con pocas muestras. Para resolver esta limitación se introducen métodos que modifican la función indicadora de error para que tome valores distintos de cero con mayor frecuencia. Más tarde hablaremos de ellos con detenimiento.

3.2.2 REALIZACIÓN DEL MÉTODO

Para llevar a cabo este método, en el receptor se debe realizar una comparación entre la señal transmitida y la recibida. En aquellos símbolos en los que no exista coincidencia se habrá producido un error. Contando los símbolos que no son iguales en ambas secuencias se conoce el número de errores ocurridos. Si dicha cantidad es dividida entre el número total de símbolos transmitidos (la longitud de la secuencia) obtendremos la probabilidad de error estimada.

Ya hemos comentado anteriormente que existe una regla aproximada, cuyo uso está muy extendido, por la cual se puede conocer de antemano el número aproximado de muestras que es necesario usar en las simulaciones para estimar una probabilidad de error determinada con el método de Monte Carlo. A continuación vamos a mostrar la justificación de dicha regla [2].

Cuando el número de muestras es tal que $N \rightarrow \infty$ el estimador tiende a una distribución normal de media p , siendo p el valor de la probabilidad de error. Entonces, se demuestra [2], [4] que podemos expresar el intervalo de confianza de la forma

$$P[y_+ \leq p \leq y_-] = 1 - \alpha, \quad (3.11)$$

siendo los extremos del intervalo

$$y_{\pm} = 10^{-v} \left\{ \pm (d_{\alpha}^2 / 2\eta) \left[\pm (4\eta / d_{\alpha}^2 + 1)^{0.5} \right] \right\}, \quad (3.12)$$

donde $\hat{p} = 10^{-v}$ es la probabilidad de error estimada, $\eta = N / 10^{-v}$, con N el número de muestras empleadas en la simulación, y d_{α} es tal que

$$\frac{1}{(2\pi)^{0.5}} \int_{-d_{\alpha}}^{d_{\alpha}} e^{-t^2/2} dt = 1 - \alpha, \quad (3.13)$$

siendo α un parámetro que representa el intervalo de confianza deseado, en tanto por uno.

Si representamos este intervalo de confianza para el 90%, 95% y 99% obtenemos la figura 3.2. En ella podemos apreciar que a medida que aumenta el número de símbolos usados en la simulación más exacta es la probabilidad de error estimada, puesto que el intervalo de confianza es menor.

Pues bien, consideremos ahora la regla que establece que para estimar una probabilidad de error $P_e = 10^{-v}$ es necesario usar como mínimo $10/P_e$ muestras en las simulaciones. El cociente $10/P_e$ coincide en la figura 3.2 con la vertical situada en 10^{v+1} . Si nos fijamos vemos que esta vertical provoca una desviación en la probabilidad de error, para un intervalo de confianza del 95 %, de $(1.8 \hat{p}, 0.55 \hat{p})$, aproximadamente. Es decir, la incertidumbre en la estimación está limitada por un factor de alrededor de 2, lo cual está considerado como aceptable en general [2].

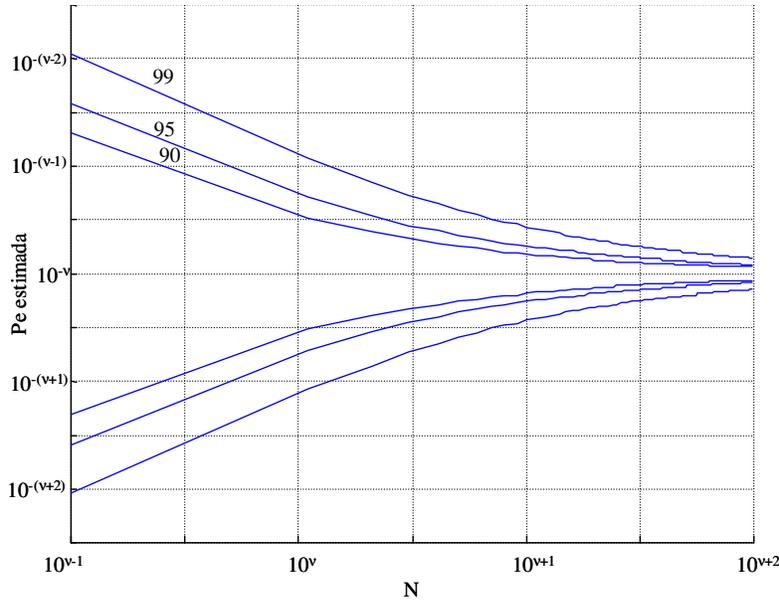


Figura 3.2: intervalos de confianza para $Pe=10^{-v}$ y el 90%, 95% y 99%.

Si el número de muestras es aumentado a 10^{v+2} , el intervalo de confianza obtenido es $(1.25 \hat{p}, 0.80 \hat{p})$ aproximadamente, menor que en el caso anterior, pero la disminución no es lo suficientemente importante teniendo en cuenta que el número de muestras es diez veces mayor, con el consecuente aumento en el tiempo de computación que ello conlleva.

Es necesario, por tanto, llegar a un compromiso entre el tiempo de computación y la incertidumbre de la probabilidad de error estimada, ya que el aumento del número de muestras usados en la estimación no siempre produce una mejora notable en la probabilidad de error estimada. Este compromiso es el que suele llevar a la regla ya comentada.

3.3 EL MUESTREO ENFATIZADO

Como comentamos en el apartado anterior, en el método de Monte Carlo la función indicadora de error toma valores muy cercanos a cero cuando la probabilidad a estimar es pequeña. Para solucionar este inconveniente se introduce una nueva función

de densidad de probabilidad que lleve asociada una función indicadora de error que tome valores mayores [2].

Para conseguirlo se introduce una función de densidad de probabilidad de ruido distinta a la original. Recordemos que existen tres versiones del muestreo enfatizado: CIS, en el que se modifica la varianza del ruido, IIS si el parámetro a modificar es la media y EIS si las variaciones se realizan con las muestras de ruido a la salida del filtro de Wiener existente en recepción.

3.3.1 FUNDAMENTO TEÓRICO

Si tomamos la ecuación (3.8) y la expresamos en función de la entrada al sistema (en nuestro caso el filtro de Wiener, tal como se indica en la figura 3.2, tendremos que

$$p = \int_{-\infty}^{\infty} H[g(\mathbf{u})]f_U(\mathbf{u})d\mathbf{u}, \quad (3.14)$$

donde $g(\cdot)$ es la función de transferencia del sistema y $f_U(\mathbf{u})$ es la función de densidad de probabilidad de la entrada del mismo, que en general será M -dimensional. Esta ecuación puede ser rescrita como

$$p = \int_{-\infty}^{\infty} H[g(\mathbf{u})]w(\mathbf{u})f_U^*(\mathbf{u})d\mathbf{u}, \quad (3.15)$$

donde $f_U^*(\mathbf{u})$ es la **función de densidad de probabilidad sesgada** y $w(\mathbf{u})$ es el **peso**, que se define por tanto como [7]

$$w(\mathbf{u}) = \frac{f_U(\mathbf{u})}{f_U^*(\mathbf{u})}. \quad (3.16)$$

Esta función de densidad enfatizada es la que introducimos en sustitución de la original. Su misión es provocar más errores de los que se producían originalmente, de forma que la función indicadora de error tome valores no tan cercanos a cero como en Monte Carlo. Ahí está la esencia del método del muestreo enfatizado. Por supuesto, es

necesario hacer una ponderación para que el resultado final no se vea falseado; esta ponderación se realiza mediante el peso definido en (3.16).

Al igual que en Monte Carlo, la probabilidad de error se puede relacionar con el operador esperanza por medio de la expresión

$$p = E_* \{H[g(\mathbf{u})]w(\mathbf{u})\}, \quad (3.17)$$

donde $E_*[\cdot]$ indica el operador esperanza evaluado sobre una función de la variable aleatoria U , con función de densidad de probabilidad $f_U^*(\mathbf{u})$.

Dado que la i -ésima muestra de salida la podemos escribir como

$$y_i = g(\mathbf{u}_i), \quad (3.18)$$

donde \mathbf{u}_i es el vector de entrada correspondiente, podemos estimar (3.17) como

$$\hat{p} = \frac{1}{N} \cdot \sum_{i \in I_o} H(y_i)w(\mathbf{u}_i), \quad (3.19)$$

donde, como en Monte Carlo, I_o es el conjunto de los i para los cuales $H(y_i)$ toma valor 1. A su vez, (3.19) puede ser rescrita como

$$\hat{p} = \frac{1}{N} \cdot \sum_{i=1}^N H(y_i) \cdot w(\mathbf{u}_i), \quad (3.20)$$

expresión que es completamente general.

Llegados a este punto es importante hacer hincapié en la diferencia que supone el uso de una función de densidad de probabilidad de error distinta respecto de la de Monte Carlo. Mientras que en éste la probabilidad de error se estimaba como una simple media aritmética, en el muestreo enfatizado se tiene una media ponderada. Esa ponderación es la que compensa el hecho de que la función indicadora de error tome valores mayores debido a la introducción de una función de densidad de probabilidad distinta. El factor de ponderación es el peso, definido como el cociente de ambas funciones de densidad, original y modificada (sesgada).

Hacemos notar también que el método del muestreo enfatizado, al igual que el de Monte Carlo, es insesgado, por lo que $E(\hat{p}) = p$ [2].

Por otra parte, como hemos visto, la función de densidad que hay que modificar es la de la señal de entrada al receptor. Pero esa señal de entrada está formada por dos procesos aleatorios: la de la señal transmitida y la del ruido. Aunque en principio podría modificarse cualquiera de los dos, o incluso ambos, realmente sólo suele hacerse con el ruido. Ello se debe a que, como ya comentamos en el capítulo 2, en nuestro proyecto vamos a suponer que el ruido se puede aproximar a una distribución gaussiana, con la ventaja de que este tipo de distribuciones son relativamente fáciles de generar. En concreto, asumiremos que el ruido es gaussiano de media cero y desviación típica σ , es decir, sigue una distribución $f_N(n) = N(0, \sigma^2)$.

Esto significa entonces que (supuesto que el ruido y la señal transmitida son independientes entre sí), la ecuación (3.20) se va a convertir en

$$\hat{p} = \frac{1}{N} \cdot \sum_{i=1}^N H(y_i) \cdot w(\mathbf{n}_i), \quad (3.21)$$

siendo \mathbf{n}_i el vector de ruido correspondiente, mientras que el peso se obtendrá como

$$w(\mathbf{n}) = \frac{f_N(\mathbf{n})}{f_N^*(\mathbf{n})}, \quad (3.22)$$

donde $f_N(n)$ y $f_N^*(n)$ son las funciones de densidad del ruido original y sesgada, respectivamente.

3.3.2 MUESTREO ENFATIZADO CONVENCIONAL (CIS)

Pues bien, en el muestreo enfatizado convencional (CIS) la característica de la función de densidad de probabilidad que se modifica para introducir una mayor cantidad de errores es la varianza. Se define así el parámetro γ a partir de las desviaciones típicas que intervienen, de la forma

$$\gamma = \frac{\sigma^*}{\sigma}, \quad (3.23)$$

siendo σ^* la varianza de la función de densidad modificada, que cumplirá que $\sigma^* > \sigma$, pues de esta manera el ruido introducido en el canal producirá una mayor cantidad de errores, lo que permitirá estimar probabilidades de error menores con el mismo número de muestras, como ya se ha explicado previamente.

En principio, como ya hemos dicho, las funciones de densidad de probabilidad que intervienen son M -dimensionales. Sin embargo, teniendo en cuenta que el ruido lo suponemos blanco, todas sus muestras son independientes, lo que nos permite obtener la función de densidad conjunta como el producto de las funciones de densidad individuales de cada una de M las variables que intervienen en cada momento. Esto supone que para cada punto es necesario calcular $2M$ funciones de densidad de probabilidad (M originales y M sesgadas).

El problema que presenta este método es la fuerte dependencia de la ganancia obtenida respecto de la memoria del sistema, como podremos comprobar en el capítulo de simulaciones. Esto limitará en gran medida la conveniencia del método. Este inconveniente se trata de subsanar mediante el método EIS, que veremos más adelante.

3.3.3 MUESTREO ENFATIZADO MEJORADO (IIS)

En este caso es la media del ruido el parámetro modificado para producir una mayor cantidad de errores. Podremos tomar medias tanto positivas como negativas, dado que no existen restricciones en ese sentido ya que en ambos casos el efecto es el mismo: un aumento en el número de errores.

Dependiendo del valor de la media escogido se obtendrán distintos valores de ganancia, por lo que es importante hacer una correcta elección. Se ha demostrado [2] que en general no es conveniente tomar una media constante, sino que es mejor opción que ésta sea distinta en cada muestra. En el caso particular de un sistema lineal se

obtienen resultados óptimos tomando como media un valor proporcional a la respuesta impulsiva del sistema en cada instante, es decir,

$$\mu(kT_s) = C \cdot h(kT_s), \quad (3.24)$$

siendo C una constante y T_s el periodo de muestreo.

Así pues, mientras que el método CIS permite una implementación secuencial, en la que los símbolos son tratados individualmente, IIS requiere en general una implementación en bloques, dado que la media es dependiente de la respuesta impulsiva del sistema. No obstante, en el caso de un canal AWGN la respuesta impulsiva del sistema en recepción es la función impulso unidad (pues no existe filtro de Wiener), por lo que la media será entonces un valor constante. El valor de dicha constante será el que determine la ganancia introducida.

3.3.4 MUESTREO ENFATIZADO EFICIENTE (EIS)

Como se ha señalado con anterioridad y comprobaremos más adelante, la mejora introducida mediante CIS es muy modesta para valores de M medianamente grandes, por lo que su eficiencia se ve muy mermada. El método EIS trata de eliminar dicha dependencia para obtener así una ganancia cuyo valor sea independiente del orden del sistema que se esté estudiando.

El método EIS deriva del CIS, con la diferencia de que en este caso se aprovecha una propiedad muy importante de nuestro sistema: como un filtro de Wiener es lineal, el ruido seguirá siendo gaussiano a la salida del mismo. En este caso, las funciones de densidad conjuntas, M -dimensionales, podrán ser sustituidas por funciones de densidad unidimensionales, lo cual equivale a aplicar CIS a un sistema sin memoria, funciones que además son muy sencillas de calcular.

Así pues, en este caso la ecuación (3.21) es sustituida por [5]

$$\hat{p} = \frac{1}{N} \cdot \sum_{i=1}^N H(y_i) w(z_i), \quad (3.25)$$

donde z_i es la muestra i -ésima del ruido a la salida del filtro de recepción.

Por su parte, el peso se calculará ahora como

$$w(z_i) = \frac{f_Z(z_i)}{f_Z^*(z_i)}, \quad (3.26)$$

siendo $f_Z(z)$ y $f_Z^*(z)$ las funciones de densidad de probabilidad de ruido original y sesgada, respectivamente. La diferencia de ambas respecto a las funciones conjuntas $f_N(n)$ y $f_N^*(n)$ es que han sido trasladadas a la salida del filtro de Wiener del receptor, por lo que sus nuevas varianzas son:

$$\sigma_Z^2 = \sigma_N^2 \cdot \sum_{n=0}^{M-1} |h(n)|^2 \quad (3.27)$$

$$(\sigma_Z^*)^2 = (\sigma_Y^*)^2 \cdot \sum_{n=0}^{M-1} |h(n)|^2, \quad (3.28)$$

donde $h(n)$ son los coeficientes del filtro de Wiener de recepción, y el sumatorio está extendido a todos los coeficientes de dicho filtro. En este caso serán M sumandos, puesto que la longitud del filtro es igual que la memoria del sistema.

Para este caso el cociente γ es

$$\gamma = \frac{\sigma_Y^*}{\sigma_Y} = \frac{\sigma_Z^*}{\sigma_Z}. \quad (3.29)$$

3.4 EXTRAPOLACIÓN DE COLA

Esta técnica está basada en la suposición de que la función de densidad de probabilidad a la entrada del decisor viene dada, al menos en sus valores más extremos, por lo que se denomina una clase exponencial generalizada, caracterizada por [2]

$$f_Y^{(\nu)}(y) = \frac{\nu}{2\sqrt{2} \cdot \sigma \cdot \Gamma(1/\nu)} \exp\left(-\left|\frac{y-\mu}{\sqrt{2} \cdot \sigma}\right|^\nu\right), -\infty < y < \infty \quad (3.30)$$

donde $\Gamma(\cdot)$ es la función gamma. La función de densidad gaussiana pertenece a esta clase exponencial, pues se corresponde con $\nu = 2$, por lo que no tendremos problema en aplicar este método a nuestro caso.

La media de esta distribución es μ , mientras que su varianza, que denotamos como V_ν , no coincide con σ sino que viene dada por

$$V_\nu = 2\sigma^2 \frac{\Gamma(3/\nu)}{\Gamma(1/\nu)}. \quad (3.31)$$

En general, los parámetros ν , μ y σ pueden ser distintos para el símbolo 1 y el 0. No obstante, en nuestro caso consideraremos que son iguales en ambos tipos de símbolos.

Aunque la función de densidad de probabilidad ha sido definida desde menos infinito hasta infinito, sólo sus extremos, es decir, la porción de la cola, son usados en el cálculo de la probabilidad de error. En efecto, suponiendo por ejemplo que el símbolo transmitido es el cero, la probabilidad de error será

$$p(t) = \int_{t+\mu}^{\infty} f_Y^{(\nu)}(y) dy, \quad (3.32)$$

Donde a t se le denomina umbral, y es la diferencia entre el umbral de decisión real y la media μ . Dado que transmitimos el símbolo 0 y la modulación es BPSK, suponiendo que la amplitud del símbolo es $-A$ y que el umbral de decisión está en el origen, tendremos que $t - A = 0$, por lo que el umbral t será igual a A .

Para valores suficientemente grandes de t y realizando una expansión asintótica de la función $\Gamma(\cdot)$ en la ecuación (3.30), puede obtenerse que [2]

$$p(t) \approx \exp \left[- \left(\frac{t}{\sqrt{2}\sigma} \right)^{\nu} [1 - \varepsilon(t)] \right]. \quad (3.33)$$

donde $\varepsilon(t)$ es un parámetro que representa en cierta forma el error cometido en la aproximación. Pero aún es posible simplificar más esta expresión. En efecto, en los casos en lo que $\nu = 1$, $\varepsilon(t) = 0$; y cuando $\nu > 1$, podemos suponer que $\varepsilon(t) \ll 1$ si t es lo suficientemente grande. Así pues, despreciando $\varepsilon(t)$ en (3.33) obtenemos que la probabilidad de error cumple que

$$\ln[-\ln p(t)] \approx \nu \ln \left(\frac{t}{\sqrt{2}\sigma} \right). \quad (3.34)$$

A partir de esta expresión vemos que, bajo las suposiciones realizadas, el doble logaritmo de la probabilidad de error es proporcional al logaritmo de $t/(\sqrt{2}\sigma)$.

La expresión (3.34) no es más entonces que la ecuación de una recta, y, como es bien sabido, cualquier recta está completamente definida a partir de dos de sus puntos, basta con obtener dos puntos cualesquiera obtenidos mediante estimaciones para poder obtener otro punto cualquiera de dicha recta.

Por tanto, podemos emplear como generadores de la recta un par de valores de t correspondientes a probabilidades de error grandes, pues entonces el tiempo que se usará en sus respectivas estimaciones será muy pequeño. Una vez conocida la ecuación de la recta, y mediante sencillas operaciones matemáticas, se puede calcular cualquier probabilidad de error. Esto supone un gran ahorro de tiempo y de recursos, pues las probabilidades de error grandes que se usan en los cálculos requieren muchas menos muestras que las que se realmente quieren estimar.

Los puntos a partir de los cuales se obtiene la recta se denominan **pseudoumbrales**, t_i . Asociado a cada pseudoumbrales existe una probabilidad de error, $p(t_i)$.

En realidad sólo son necesarios dos pseudoumbrales, pero las estimaciones resultantes serán mejores si se utilizan más, según tendremos oportunidad de comprobar en el capítulo de simulaciones. En el caso de que trabajemos con tres o más pseudoumbrales, será necesario obtener la recta de mejor ajuste para los mismos.

Vemos ahora el porqué del ahorro operacional: a partir de dos (como mínimo) probabilidades de error de valores relativamente grandes y, en consecuencia, fácilmente estimables con el método de Monte Carlo, por ejemplo, es posible hallar una probabilidad de error más pequeña y que requeriría por tanto un tiempo de computación mucho mayor.

El problema es que éste es un método sesgado, frente al resto de métodos que analizamos en el proyecto, que son insesgados [2]. Esto supone que la probabilidad de error estimada se aleja de los valores reales, de forma más acusada cuanto menor es dicha probabilidad, como veremos en apartados posteriores.

4 SIMULACIONES

En este capítulo detallamos las condiciones impuestas en las simulaciones y presentamos los resultados de las mismas mediante diversas tablas y gráficas.

Todas las simulaciones han sido realizadas con 10^6 muestras, y repetidas 20 veces. Por tanto, las probabilidades de error obtenidas son el resultado de promediar los resultados parciales de dichas repeticiones. Además, se calculará en cada experimento la varianza de la probabilidad de error, evaluada sobre el total de las 20 realizaciones.

Los algoritmos han sido implementados en Matlab 5. Puesto que Matlab está diseñado especialmente para el cálculo matricial, una de las principales premisas a la hora de implementar los algoritmos ha sido restringir lo más posible el uso de bucles con objeto de reducir el tiempo de computación.

Como ya se adelantó, se han realizado dos tipos de simulaciones: una con un modelo de canal AWGN y otro con un modelo de canal con eco simple, en el que se incluye el efecto de la memoria. En los próximos epígrafes se describen cada uno de ellos.

4.1 SIMULACIÓN DEL MÉTODO DE MONTE CARLO

Puesto que éste es el método más extendido y del cual partimos, todas las comparaciones las realizaremos con el mismo, siempre teniendo en cuenta el modelo de canal que se usa en la simulación.

4.1.1 MONTE CARLO CON CANAL AWGN

Para realizar una simulación lo primero que se necesita es una información a transmitir. Dicha información la generamos creando una secuencia aleatoria de unos y ceros con igual probabilidad.

Para realizar la transmisión de la información hay que asociar a cada símbolo una tensión. En este caso hemos optado por asignar una amplitud 0 al símbolo 0 y una

amplitud A (variable) al 1. Se ha optado por esta posibilidad por simple comodidad a la hora de implementar el código. Pero, por supuesto, los resultados obtenidos de esta forma son los mismos que se lograrían si hubiéramos asociado $-A/2$ al 0 y $+A/2$ al 1, puesto que lo importante es la diferencia de amplitudes entre ambos niveles, no los niveles en sí.

Pero, ¿cuál debe ser el valor de A ? Para calcularlo es necesario hacer uso de la expresión teórica de la probabilidad de error para un sistema de transmisión BPSK. Como ya vimos anteriormente, dicha probabilidad de error, en presencia de un canal AWGN, es

$$P_e = \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right). \quad (4.1)$$

Despejando de esta expresión obtenemos que

$$\frac{E_b}{N_0} = \left[\operatorname{erfc}^{-1}(2P_e) \right]^2, \quad (4.2)$$

donde $\operatorname{erfc}^{-1}(\cdot)$ es la función inversa de la función complementaria de error. Vemos que por medio de esta última expresión podemos obtener el cociente E_b/N_0 necesario para conseguir una cierta probabilidad de error.

Lo único que nos falta es conocer la relación entre A y E_b . Dicha relación viene dada por [3]

$$A = 2\sqrt{E_b}, \quad (4.3)$$

por lo que, sustituyendo en (4.2),

$$A = 2\sqrt{N_0} \cdot \operatorname{erfc}^{-1}(2P_e). \quad (4.4)$$

Esta ecuación nos proporciona la amplitud necesaria para conseguir una probabilidad de error determinada. Puesto que la hemos obtenido a partir de la expresión teórica de la probabilidad de error para un canal AWGN, sólo será válida en los casos en los que consideremos dicho modelo de canal.

A continuación se simula el efecto del canal AWGN. En este caso, el canal sólo introduce una fuente de ruido gaussiana de media cero y desviación típica σ . El valor de la varianza está determinado por el valor de N_0 tomado, pues la potencia de la señal discreta de ruido empleada en la simulación es [2]

$$P = \sigma^2 = \frac{N_0}{2}. \quad (4.5)$$

Finalmente, el receptor comprueba en qué región está el nivel recibido y le asocia como símbolo el correspondiente a dicha región.

Por otra parte, y aunque los resultados no dependen de ello, aclararemos que se ha tomado para N_0 un valor cualquiera, como por ejemplo $N_0 = 0.05$, obteniendo así el valor necesario de E_b para una probabilidad de error determinada. Por tanto, despejando de (4.5) obtenemos que

$$\sigma = \sqrt{\frac{N_0}{2}} = 0.1581, \quad (4.6)$$

y las amplitudes necesarias, calculadas según (4.4), son las reflejadas en la tabla 4.1.

	$P_e = 10^{-1}$	$P_e = 10^{-2}$	$P_e = 10^{-3}$	$P_e = 10^{-4}$	$P_e = 10^{-5}$	$P_e = 10^{-6}$
A	0.4052	0.7356	0.9772	1.1760	1.3486	1.5031

Tabla 4.1: amplitudes necesarias para MC con canal AWGN.

En la tabla 4.2 se muestran los resultados. En ella se detallan las probabilidades medias obtenidas promediando sobre las 20 repeticiones realizadas, así como su varianza.

MC	$P_e = 10^{-1}$	$P_e = 10^{-2}$	$P_e = 10^{-3}$	$P_e = 10^{-4}$	$P_e = 10^{-5}$	$P_e = 10^{-6}$
P_e media	0.999e-001	1.000e-002	0.994e-003	1.004e-004	1.135e-005	1.200e-006
Varianza	8.9763e-008	1.2191e-008	9.1592e-010	8.7410e-011	7.8185e-012	5.8947e-013

Tabla 4.2: resultados de MC con modelo simple de canal.

En la figura 4.1 podemos ver la representación de las probabilidades de error simuladas (representadas mediante círculos verdes) frente a la curva teórica (dibujada en azul). Como era de esperar, los resultados son muy similares, si bien para una probabilidad de error de 10^{-6} empieza ya a apreciarse una pequeña desviación.

Como hemos dicho anteriormente, el método de Monte Carlo tiene como limitación el número de muestras que necesita para estimar probabilidades de error pequeñas. En este caso las simulaciones se han realizado con 10^6 muestras, por lo que según la regla aproximada se podrían calcular de forma medianamente fiable hasta probabilidades del orden de 10^{-5} , lo que concuerda con nuestros resultados.

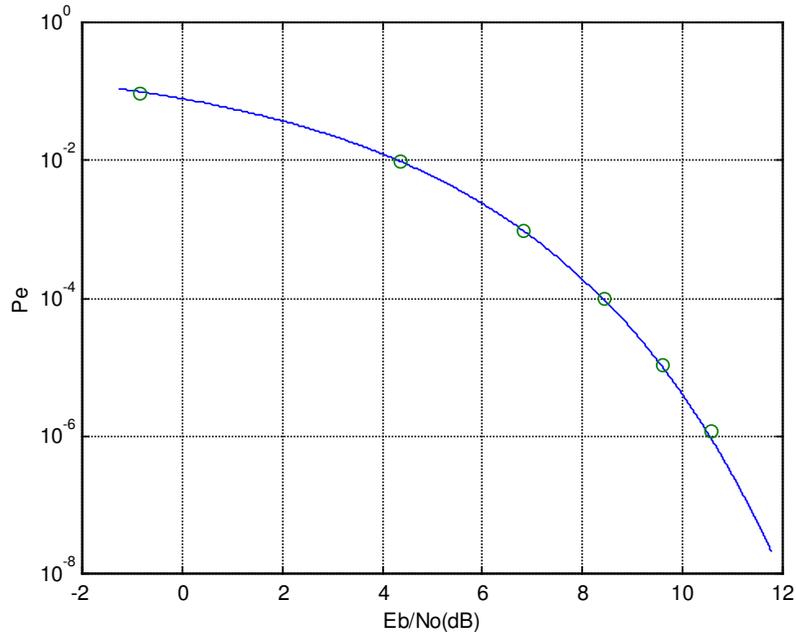


Figura 4.1: P_e obtenidas en Monte Carlo frente a la curva teórica.

En la figura 4.2 podemos ver representada la evolución de la varianza en función de la probabilidad de error obtenida mediante el método de Monte Carlo. Podemos observar así que la varianza de la estimación disminuye conforme lo hace la probabilidad de error que se está estimando, de forma prácticamente lineal, lo cual concuerda con el análisis efectuado en el capítulo 3. Debemos recordar que el canal usado es el AWGN, por lo que cuando usemos un canal con eco simple obtendremos un empeoramiento de estos resultados.

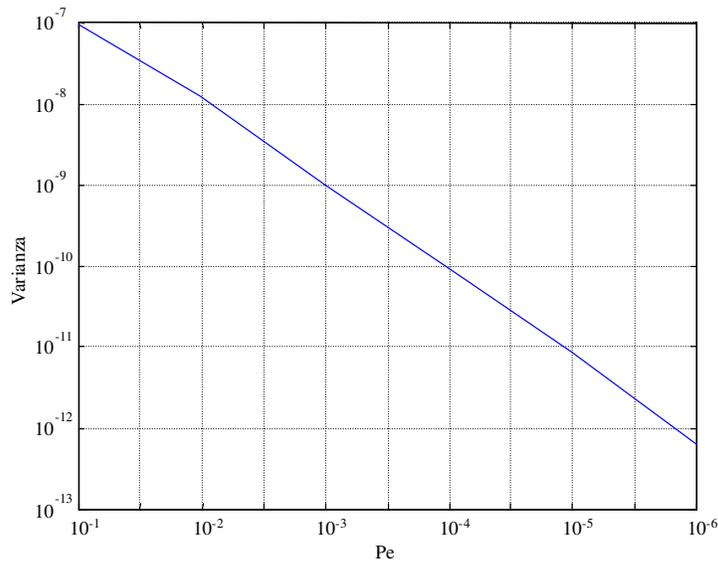


Figura 4.2: evolución de la varianza en función de la P_e en MC con canal AWGN.

4.1.2 MONTE CARLO CON MODELO DE CANAL CON ECO SIMPLE

En este caso utilizaremos un modelo de canal un poco más complejo. Además del ruido gaussiano introducido, tendremos en cuenta un filtrado de orden uno dado por

$$H(z) = 1 + \alpha \cdot z^{-1}, \quad (4.7)$$

donde α representa al eco y es un parámetro de diseño. En el caso ideal (que coincidiría con un canal AWGN), $\alpha = 0$; cuanto mayor sea α peores serán las características del

canal. Nosotros tomaremos como valor límite $\alpha = 1$, es decir, un eco del mismo nivel que la señal directa.

Al usar este modelo de canal es necesario introducir un filtrado adaptativo en recepción que contrarreste el efecto del eco. Dicho objetivo se consigue por medio de un filtro de Wiener de orden $M-1$. Sus coeficientes son calculados una sola vez para cada valor de M y cada probabilidad de error, y se usan en todas las simulaciones en las que se mantengan dichos parámetros.

Es importante hacer notar que el orden completo del sistema es M , dado que se compone de un eco de orden 1 y un filtrado de Wiener de orden $M-1$ (M coeficientes). Dado que el orden del sistema y el del filtro de Wiener guardan una relación directa entre sí, hablaremos indistintamente de ambos a lo largo del proyecto.

Para el cálculo de los coeficientes del filtro sólo se usan 1000 muestras debido a que usar el total de las mismas (es decir, un millón) es inviable e innecesario, ya que la precisión en la obtención de los coeficientes no mejora sustancialmente.

Puesto que la finalidad del filtro de Wiener es contrarrestar la acción del eco del canal, cuando se elimina la fuente de ruido gaussiano debe cumplirse idealmente que

$$h_{canal}(n) * h_{filtro}(n) = \delta(n). \quad (4.8)$$

Esta propiedad ha sido usada para comprobar que el diseño del filtro es correcto. Nunca podrá verificarse de forma exacta, ya que nuestro filtro de Wiener es FIR, pero sí de manera bastante aproximada. Para ello se ha calculado la convolución de las funciones de transferencia del canal (sin ruido) y del filtro de Wiener, y se ha impuesto que un 95% de la energía del resultado esté concentrada en el origen.

Sabemos que si la función que se obtiene es un impulso unidad su acción no afecta al sistema en nada, pero puesto que este tipo de función no se puede obtener realmente hemos impuesto esta condición de la energía con objeto de llegar a un compromiso entre la opción de tener una función delta lo más perfecta posible y la necesidad de trabajar con sistemas reales. De esta forma aseguramos que la función

delta es lo suficientemente pura como para asegurar que el filtrado de Wiener se está haciendo de forma correcta.

Por todo esto, para cada M (orden total del sistema) existirá un valor máximo del parámetro α para el cual se sigue cumpliendo la condición de la energía que ha sido impuesta. Esto se refleja en la tabla 4.3, donde podemos ver los valores máximos de α obtenidos para cada M . Vemos que, a medida que aumenta el orden del filtro de Wiener, mayores valores de α puede corregir mediante su acción en el receptor. O, desde otro punto de vista, cuanto mayor es α , aumenta el orden necesario del filtro que permite cumplir la condición impuesta para la energía.

M	α	M	α
2	0.3	16	0.723
4	0.46	18	0.746
6	0.531	20	0.763
8	0.585	25	0.797
10	0.633	30	0.827
12	0.672	35	0.839
14	0.695	40	0.860

Tabla 4.3: valores obtenidos del parámetro α .

En la figura 4.3 vemos los mismos resultados pero representados de forma gráfica. Podemos observar así que, para valores bajos del parámetro α , los incrementos de éste suponen un aumento pequeño en el orden necesario del filtro de Wiener. Sin embargo, a medida que α aumenta son necesarios incrementos de M cada vez mayores para que el 95% de la energía siga concentrada en el origen. Es decir, que para valores de α cercanos a 1 el orden necesario en el filtro de Wiener para que el sistema siga funcionando correctamente crece a un ritmo mucho más elevado.

Dado que el canal que usamos en este caso ya no es AWGN, las amplitudes

asociadas a los símbolos necesarias para obtener una cierta P_e ya no son las mismas que en el caso del canal simple (tabla 4.1). En este caso será necesario emplear amplitudes mayores para poder equilibrar los efectos introducidos por el eco del canal. Los cálculos han sido efectuados por el método de prueba y error, buscando cuál es la amplitud necesaria para obtener cada probabilidad de error deseada para un cierto valor de α . Los resultados obtenidos se reflejan en la tabla 4.4.

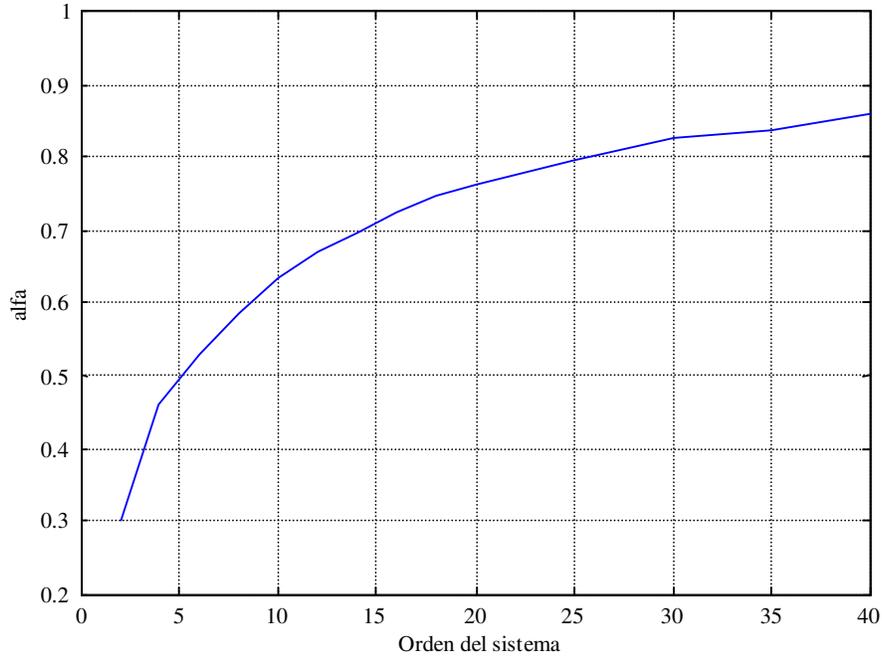


Figura 4.3: valores máximos de α obtenidos para cada orden del filtro.

Hay que decir que, ya que no se ha empleado ninguna fórmula teórica, los valores de amplitud anteriores son sólo aproximados, siendo dicha aproximación tanto peor cuanto menor sea P_e , por los problemas inherentes al método de Monte Carlo que ya hemos comentado. Pese a todo, esto no afecta a los resultados del producto tiempo-fiabilidad, que es lo que realmente nos interesa.

Lo resultados de las simulaciones del método de Monte Carlo se presentan en la tabla 4.5. En ella se muestran los valores medios y varianzas obtenidos para cada

probabilidad de error y para cada valor de M . Como se puede observar, las probabilidades de error estimadas son bastante precisas para valores de P_e iguales o menores a 10^{-5} , siendo en cambio bastante peores para 10^{-6} , como era de esperar según lo que ya hemos venido explicando. Y, lo que también es importante, no existe influencia alguna del valor de M en los resultados obtenidos.

M	α	$P_e=10^{-1}$	$P_e=10^{-2}$	$P_e=10^{-3}$	$P_e=10^{-4}$	$P_e=10^{-5}$	$P_e=10^{-6}$
2	0.300	0.420	0.840	1.180	1.460	1.690	1.900
6	0.531	0.445	0.920	1.300	1.620	1.900	2.150
10	0.633	0.465	0.970	1.400	1.750	2.070	2.400
14	0.695	0.485	1.020	1.475	1.870	2.200	2.550
18	0.746	0.500	1.090	1.610	2.090	2.500	2.850
25	0.797	0.520	1.150	1.720	2.250	2.750	3.100
30	0.827	0.543	1.210	1.880	2.520	3.080	3.530
35	0.839	0.550	1.220	1.900	2.520	3.110	3.582
40	0.860	0.564	1.265	2.020	2.730	3.400	4.000

Tabla 4.4: amplitudes necesarias para cada orden y probabilidad de error, en un canal con eco simple.

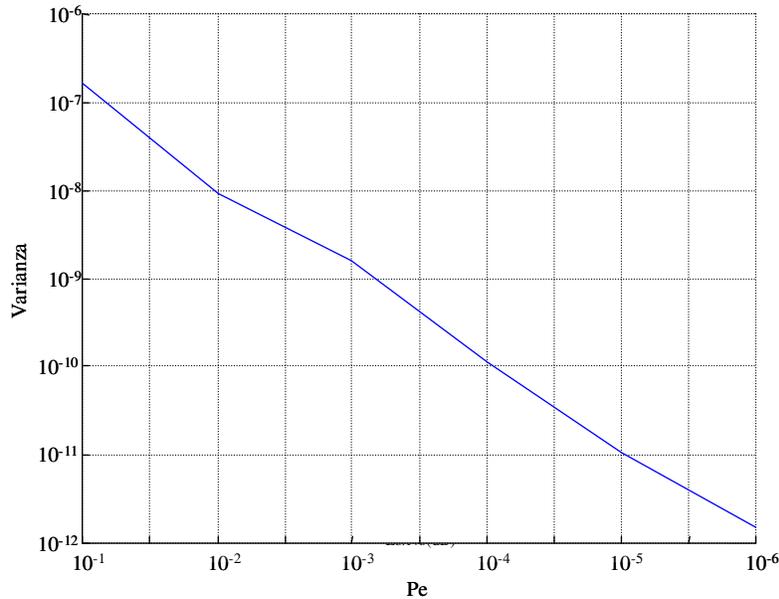


Figura 4.4: varianzas obtenidas en MC con canal con eco simple y $M=40$.

En la figura 4.4 se presentan las varianzas obtenidas usando el canal con eco simple y tomando como orden del sistema $M=40$. Vemos que, tal y como adelantamos en el apartado 4.2.1, los valores de las varianzas son algo peores a los obtenidos para el canal AWGN, pero el empeoramiento no es excesivo gracias al filtro de Wiener que introducimos en recepción y que nos permite contrarrestar el efecto del eco introducido en el canal.

M		$P_e=10^{-1}$	$P_e=10^{-2}$	$P_e=10^{-3}$	$P_e=10^{-4}$	$P_e=10^{-5}$	$P_e=10^{-6}$
2	Pe media	$1.0276 \cdot 10^{-1}$	$1.0730 \cdot 10^{-2}$	$1.0850 \cdot 10^{-3}$	$1.1015 \cdot 10^{-4}$	$1.2900 \cdot 10^{-5}$	$1.5000 \cdot 10^{-6}$
	Varianza	$2.0168 \cdot 10^{-7}$	$2.0184 \cdot 10^{-8}$	$1.3296 \cdot 10^{-9}$	$1.2950 \cdot 10^{-10}$	$7.780 \cdot 10^{-12}$	$1.1100 \cdot 10^{-12}$
6	Pe media	$1.0237 \cdot 10^{-1}$	$1.0637 \cdot 10^{-2}$	$1.1242 \cdot 10^{-3}$	$1.1610 \cdot 10^{-4}$	$1.215 \cdot 10^{-5}$	$1.3000 \cdot 10^{-6}$
	Varianza	$1.2186 \cdot 10^{-7}$	$1.4393 \cdot 10^{-8}$	$1.2789 \cdot 10^{-9}$	$8.9460 \cdot 10^{-11}$	$1.5610 \cdot 10^{-11}$	$8.5000 \cdot 10^{-13}$
10	Pe media	$1.0204 \cdot 10^{-1}$	$1.0496 \cdot 10^{-2}$	$1.0054 \cdot 10^{-3}$	$1.0755 \cdot 10^{-4}$	$1.1600 \cdot 10^{-5}$	$0.6000 \cdot 10^{-6}$
	Varianza	$1.2481 \cdot 10^{-7}$	$9.1910 \cdot 10^{-9}$	$9.623 \cdot 10^{-10}$	$9.1420 \cdot 10^{-11}$	$1.7620 \cdot 10^{-11}$	$4.6000 \cdot 10^{-13}$
14	Pe media	$1.0115 \cdot 10^{-1}$	$1.0233 \cdot 10^{-2}$	$1.0101 \cdot 10^{-3}$	$0.9675 \cdot 10^{-4}$	$1.1550 \cdot 10^{-5}$	$1.2500 \cdot 10^{-6}$
	Varianza	$1.1670 \cdot 10^{-7}$	$1.8232 \cdot 10^{-8}$	$1.1449 \cdot 10^{-9}$	$1.2070 \cdot 10^{-10}$	$9.8400 \cdot 10^{-12}$	$9.3000 \cdot 10^{-13}$
18	Pe media	$1.0190 \cdot 10^{-1}$	$1.0064 \cdot 10^{-2}$	$1.0531 \cdot 10^{-3}$	$0.9965 \cdot 10^{-4}$	$1.0850 \cdot 10^{-5}$	$1.3000 \cdot 10^{-6}$
	Varianza	$0.8542 \cdot 10^{-7}$	$1.1434 \cdot 10^{-8}$	$1.0288 \cdot 10^{-9}$	$5.2800 \cdot 10^{-11}$	$1.1080 \cdot 10^{-11}$	$1.2700 \cdot 10^{-12}$
25	Pe media	$1.0279 \cdot 10^{-1}$	$1.0116 \cdot 10^{-2}$	$1.0498 \cdot 10^{-3}$	$1.0980 \cdot 10^{-4}$	$0.9800 \cdot 10^{-5}$	$1.2500 \cdot 10^{-6}$
	Varianza	$0.6430 \cdot 10^{-7}$	$1.3450 \cdot 10^{-8}$	$1.4749 \cdot 10^{-9}$	$1.3760 \cdot 10^{-10}$	$6.300 \cdot 10^{-12}$	$1.9900 \cdot 10^{-12}$
30	Pe media	$1.0032 \cdot 10^{-1}$	$1.0279 \cdot 10^{-2}$	$1.0533 \cdot 10^{-3}$	$0.9840 \cdot 10^{-4}$	$1.0450 \cdot 10^{-5}$	$1.7500 \cdot 10^{-6}$
	Varianza	$1.2514 \cdot 10^{-7}$	$1.3716 \cdot 10^{-8}$	$1.0518 \cdot 10^{-9}$	$1.0180 \cdot 10^{-10}$	$7.5200 \cdot 10^{-12}$	$1.5700 \cdot 10^{-12}$
35	Pe media	$1.0010 \cdot 10^{-1}$	$1.0287 \cdot 10^{-2}$	$1.0290 \cdot 10^{-3}$	$1.0655 \cdot 10^{-4}$	$1.1250 \cdot 10^{-5}$	$1.4000 \cdot 10^{-6}$
	Varianza	$1.1873 \cdot 10^{-7}$	$1.8081 \cdot 10^{-8}$	$1.1872 \cdot 10^{-9}$	$1.4400 \cdot 10^{-10}$	$1.5040 \cdot 10^{-11}$	$1.4100 \cdot 10^{-12}$
40	Pe media	$1.0016 \cdot 10^{-1}$	$1.0295 \cdot 10^{-2}$	$1.0006 \cdot 10^{-3}$	$1.0995 \cdot 10^{-4}$	$1.0900 \cdot 10^{-5}$	$1.6500 \cdot 10^{-6}$
	Varianza	$1.6358 \cdot 10^{-7}$	$9.1424 \cdot 10^{-9}$	$1.6102 \cdot 10^{-9}$	$1.1460 \cdot 10^{-10}$	$1.0520 \cdot 10^{-11}$	$1.5000 \cdot 10^{-12}$

Tabla 4.5: resultados de MC para un canal con eco simple.

4.2 SIMULACIÓN DEL MÉTODO CIS

Como ya ha sido explicado anteriormente, en este método se introduce una fuente de ruido con una varianza distinta de la original. Para ello se utiliza una función de densidad de probabilidad cuya varianza es mayor que la correspondiente al método de Monte Carlo.

4.2.1 CIS CON CANAL AWGN

El parámetro variado respecto a la función de densidad de probabilidad original es la varianza. Pero, ¿cuál es el valor óptimo que debe tomar la nueva varianza? Para poder averiguarlo se han realizado simulaciones para varios valores del parámetro γ definido como

$$\gamma = \frac{\sigma^*}{\sigma}. \quad (4.9)$$

En concreto, se va a probar para $\gamma = 1, 2, 3, 4$ y 5 . Cuando dicho parámetro toma el valor 1 , entonces $\sigma^* = \sigma$, y por tanto este caso es el mismo que el método de Monte Carlo.

En la tabla 4.6 podemos observar los resultados obtenidos. En ella se muestran los valores medios y varianzas de la probabilidad de error estimada para cada probabilidad de error teórica y cada valor del parámetro γ . Como era de esperar, los resultados para $\gamma = 1$ son muy parecidos a los de Monte Carlo con el canal AWGN, los cuales se mostraban en la tabla 4.2.

En las figura 4.5 se representa la ganancia η , definida en (3.5) para las distintas probabilidades de error simuladas.

Se puede comprobar que el método CIS introduce grandes mejoras respecto al de Monte Carlo, tanto mayores cuanto menor es la probabilidad de error a obtener. Era previsible, puesto que el método de Monte Carlo es muy bueno para probabilidades de

error moderadas, pero malo cuando se quieren estimar probabilidades muy pequeñas, dado que necesita muchas muestras.

		$\gamma=1$	$\gamma=2$	$\gamma=3$	$\gamma=4$	$\gamma=5$
$P_e=10^{-1}$	P_e media	$1.000*10^{-1}$	$0.999*10^{-1}$	$1.000*10^{-1}$	$1.000*10^{-1}$	$0.999*10^{-1}$
	Varianza	$12.663*10^{-8}$	$4.1140*10^{-8}$	$7.0571*10^{-8}$	$8.2021*10^{-8}$	$8.4475*10^{-8}$
$P_e=10^{-2}$	P_e media	$1.0000*10^{-2}$	$0.9900*10^{-2}$	$0.0100*10^{-2}$	$0.9900*10^{-2}$	$1.0000*10^{-2}$
	Varianza	$7.6251*10^{-9}$	$1.3125*10^{-9}$	$1.3345*10^{-9}$	$1.1068*10^{-9}$	$2.6650*10^{-9}$
$P_e=10^{-3}$	P_e media	$0.9977*10^{-3}$	$0.9993*10^{-3}$	$0.9999*10^{-3}$	$0.9998*10^{-3}$	$1.0000*10^{-3}$
	Varianza	$88.819*10^{-11}$	$2.4547*10^{-11}$	$2.5801*10^{-11}$	$4.5836*10^{-11}$	$2.0796*10^{-11}$
$P_e=10^{-4}$	P_e media	$1.0210*10^{-4}$	$0.9989*10^{-4}$	$0.9992*10^{-4}$	$0.9986*10^{-4}$	$1.0013*10^{-4}$
	Varianza	$803.05*10^{-13}$	$7.0917*10^{-13}$	$4.1020*10^{-13}$	$4.5820*10^{-13}$	$3.9713*10^{-13}$
$P_e=10^{-5}$	P_e media	$11.350*10^{-6}$	$10.004*10^{-6}$	$9.9842*10^{-6}$	$9.9776*10^{-6}$	$9.9762*10^{-6}$
	Varianza	$7818.4*10^{-15}$	$16.186*10^{-15}$	$5.0350*10^{-15}$	$3.9084*10^{-15}$	$4.9054*10^{-15}$
$P_e=10^{-6}$	P_e media	$7.0000*10^{-7}$	$9.9861*10^{-7}$	$9.9657*10^{-7}$	$9.9914*10^{-7}$	$9.9630*10^{-7}$
	Varianza	$5368.4*10^{-16}$	$1.5280*10^{-16}$	$1.0777*10^{-16}$	$0.4262*10^{-16}$	$0.3727*10^{-16}$

Tabla 4.6: resultados de CIS con canal AWGN.

Por otra parte, vemos en las gráficas que la bondad del nuevo método es dependiente del valor de γ , aunque dicha dependencia no es demasiado acusada así que, en primera aproximación, podríamos pensar que cualquier γ mayor o igual a 2 podría darnos unos resultados bastante buenos. Eso sí, la dependencia con γ es distinta según sea el valor de P_e .

En resumen, hemos podido comprobar que el método CIS introduce grandes mejoras respecto al de Monte Carlo para probabilidades de error muy pequeñas. Para probabilidades de error grandes, el método de Monte Carlo sigue siendo una buena opción. Además, dependiendo del valor de γ escogido la ganancia con el método CIS será mayor o menor.

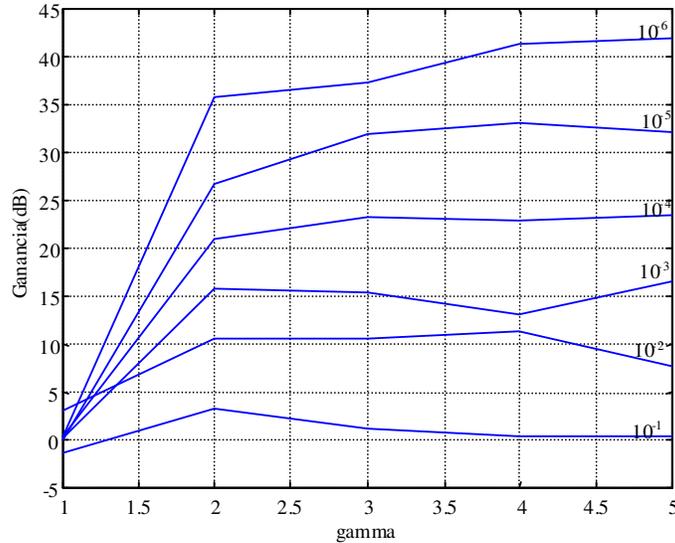


Figura 4.5: ganancia de CIS en función de gamma.

4.2.2 CIS CON MODELO DE CANAL CON ECO SIMPLE

En este caso usaremos el modelo de canal que incluye filtrado y, por tanto, que tiene memoria. Para ello usaremos las amplitudes obtenidas y mostradas en la tabla 4.4.

La forma de funcionamiento de la simulación es similar a la explicada para el método de Monte Carlo. Por tanto, introduciremos un filtro de Wiener en recepción para compensar al máximo los perjuicios introducidos por el eco en el modelo del canal. Los coeficientes del filtro serán los mismos que fueron calculados en el apartado 4.1.2.

La diferencia con el método de Monte Carlo es que en este caso es necesario el cálculo de pesos para obtener la probabilidad de error, según veíamos en la ecuación (3.21).

El problema es que la función de densidad de probabilidad conjunta no es fácil de obtener en general, pero dado que en nuestro caso hemos considerado que las muestras de ruido son independientes entre sí, el cálculo se simplifica notablemente, ya que será igual al producto de las funciones de densidad de probabilidad individuales. Bajo esta condición la función de densidad conjunta es

$$f_{TOTAL} = f_0 \cdot f_1 \cdot f_2 \cdot \dots \cdot f_{M-1}, \quad (4.10)$$

donde

$$f_i = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{r_i^2}{2\sigma^2}\right), \quad (4.11)$$

siendo r_i la i -ésima muestra de ruido original.

Vemos que la única diferencia en el receptor respecto al método CIS sin tener en cuenta el efecto de la memoria es el número de factores (M en total) que intervienen en el cálculo del peso. Los valores de M estudiados son 2, 6, 10, 18, 30 y 40.

Los valores de γ tomados inicialmente fueron los mismos que para simulaciones anteriores. El problema es que la dependencia que presenta este método con γ es tal que a partir de un cierto orden no demasiado alto del sistema el empeoramiento era demasiado elevado, resultando el método poco eficaz. (Más adelante se podrá comprobar esto gráficamente con mayor claridad). Por tanto, se ha optado por realizar las simulaciones para distintos γ dependiendo del orden del sistema en cada caso, según se detalla en la tabla 4.7.

M	Gamma1	Gamma2	Gamma3	Gamma4	Gamma5
2	1	2	3	4	5
6	1	2	3	4	5
10	1	2	3	4	5
18	1	1.4	1.8	2.2	2.5
30	1	1.3	1.6	1.9	2.2
40	1	1.1	1.2	1.3	1.4

Tabla 4.7: valores del parámetro γ

En las tablas 4.8 a 4.13 podemos observar los valores medios obtenidos mediante las simulaciones para todas las probabilidades de error, órdenes y valores de γ

M	$P_e = 10^{-1}$	Gamma1	Gamma2	Gamma3	Gamma4	Gamma5
2	Pe media	1.0288e-001	1.0283e-001	1.0278e-001	1.0271e-001	1.0254e-001
	Varianza	6.2138e-008	8.9177e-008	1.6751e-007	2.0076e-007	5.4073e-007
6	Pe media	1.0227e-001	1.0272e-001	1.0184e-001	1.0233e-001	1.0088e-001
	Varianza	9.4933e-008	7.1644e-007	1.0857e-005	4.2963e-005	1.1060e-004
10	Pe media	1.0208e-001	1.0208e-001	0.9516e-001	0.9037e-001	0.6104e-001
	Varianza	8.0582e-008	4.0654e-006	1.1098e-004	1.8474e-003	1.2584e-003
18	Pe media	1.0193e-001	1.0178e-001	1.0039e-001	0.9756e-001	1.2960e-001
	Varianza	1.6603e-007	1.4050e-006	3.4874e-005	5.6162e-004	1.2443e-002
30	Pe media	0.9999e-001	0.9989e-001	1.0109e-001	1.3011e-001	1.5455e-001
	Varianza	7.8369e-008	2.4390e-006	1.1390e-004	1.3805e-002	1.3765e-001
40	Pe media	1.0019e-001	1.0010e-001	0.9993e-001	1.0249e-001	1.1252e-001
	Varianza	9.1200e-008	1.0453e-006	1.0032e-005	1.4776e-004	7.9925e-003

Tabla 4.8: resultados de CIS con modelo de canal con eco simple, $P_e=10^{-1}$.

M	$P_e = 10^{-2}$	Gamma1	Gamma2	Gamma3	Gamma4	Gamma5
2	Pe media	1.0779e-002	1.0735e-002	1.0785e-002	1.0751e-002	1.0746e-002
	Varianza	1.3098e-008	2.8360e-009	5.8362e-009	9.0933e-009	1.0893e-008
6	Pe media	1.0676e-002	1.0636e-002	1.0591e-002	1.0438e-002	1.1288e-002
	Varianza	1.3904e-008	1.9591e-008	1.8355e-007	5.5416e-007	3.1080e-006
10	Pe media	1.0442e-002	1.0542e-002	1.1249e-002	8.6724e-003	7.0678e-003
	Varianza	1.6484e-008	1.9713e-007	2.5326e-006	2.8440e-005	6.4228e-005
18	Pe media	1.0098e-002	1.0126e-002	1.0184e-002	9.9767e-003	8.2816e-003
	Varianza	1.3903e-008	5.7399e-008	9.8372e-007	2.4331e-005	1.4522e-005
30	Pe media	1.0296e-002	1.0379e-002	1.0866e-002	1.1126e-002	1.9509e-002
	Varianza	1.0611e-008	1.0999e-007	3.7157e-006	1.0478e-004	4.2493e-003
40	Pe media	1.0303e-002	1.0279e-002	1.0200e-002	9.9604e-003	1.2209e-002
	Varianza	1.1322e-008	2.3864e-008	1.6490e-007	1.8828e-006	8.2012e-005

Tabla 4.9: resultados de CIS con modelo de canal con eco simple, $P_e=10^{-2}$.

M	$P_e = 10^{-3}$	Gamma1	Gamma2	Gamma3	Gamma4	Gamma5
2	Pe media	1.0827e-003	1.0870e-003	1.0852e-003	1.0881e-003	1.0850e-003
	Varianza	1.5423e-009	1.1057e-010	5.2724e-011	1.6219e-010	2.5349e-010
6	Pe media	1.1115e-003	1.1123e-003	1.1449e-003	1.0964e-003	1.1327e-003
	Varianza	1.5055e-009	9.8833e-010	3.3514e-009	7.7746e-009	5.9227e-008
10	Pe media	1.0113e-003	9.9896e-004	1.1115e-003	1.2410e-003	6.9252e-004
	Varianza	6.1308e-010	4.1103e-009	1.0393e-007	1.9763e-006	4.8037e-007
18	Pe media	1.0603e-003	1.0552e-003	9.9368e-004	9.9659e-004	1.1902e-003
	Varianza	1.2539e-009	2.5436e-009	3.0066e-008	3.0767e-007	1.0732e-006
30	Pe media	1.0551e-003	1.0320e-003	9.9903e-004	7.4233e-004	2.5659e-003
	Varianza	1.2416e-009	4.0619e-009	8.3900e-008	2.9353e-007	3.4494e-005
40	Pe media	1.0083e-003	1.0272e-003	1.0258e-003	1.0060e-003	1.1253e-003
	Varianza	1.0449e-009	3.3107e-009	1.2027e-008	5.5132e-008	1.0261e-006

Tabla 4.10: resultados de CIS con modelo de canal con eco simple, $P_e=10^{-3}$.

M	$P_e = 10^{-4}$	Gamma1	Gamma2	Gamma3	Gamma4	Gamma5
2	Pe media	1.0995e-004	1.0767e-004	1.0790e-004	1.0758e-004	1.0730e-004
	Varianza	1.1163e-010	1.6534e-012	1.7006e-012	1.3732e-012	2.3035e-012
6	Pe media	1.1850e-004	1.1658e-004	1.1773e-004	1.2040e-004	1.1622e-004
	Varianza	1.0921e-010	1.2129e-011	4.2501e-011	3.5700e-010	7.5979e-010
10	Pe media	1.1075e-004	1.0919e-004	1.2916e-004	1.5292e-004	1.3234e-004
	Varianza	6.1250e-011	9.3474e-011	4.0419e-009	3.3966e-008	8.4011e-008
18	Pe media	9.9300e-005	1.0145e-004	1.1324e-004	1.2181e-004	8.4195e-005
	Varianza	1.3443e-010	1.5576e-010	2.1184e-009	1.7717e-008	8.0474e-009
30	Pe media	9.8700e-005	1.0727e-004	8.8525e-005	1.1787e-004	7.5049e-006
	Varianza	1.0633e-010	1.3592e-010	1.0201e-009	6.7444e-008	1.0619e-010
40	Pe media	1.0635e-004	1.0563e-004	1.1227e-004	8.5931e-005	7.3771e-005
	Varianza	1.1466e-010	1.2078e-010	5.9555e-010	1.1384e-009	5.2033e-009

Tabla 4.11: resultados de CIS con modelo de canal con eco simple, $P_e=10^{-4}$.

M	$P_e = 10^{-5}$	Gamma1	Gamma2	Gamma3	Gamma4	Gamma5
2	Pe media	1.2000e-005	1.1910e-005	1.1912e-005	1.1855e-005	1.1860e-005
	Varianza	9.7895e-012	5.1835e-014	3.0050e-014	3.7800e-014	5.8686e-014
6	Pe media	1.1850e-005	1.1916e-005	1.1617e-005	1.1829e-005	1.3049e-005
	Varianza	8.1342e-012	6.4523e-013	1.0177e-012	7.6379e-012	2.5164e-011
10	Pe media	1.1700e-005	1.1556e-005	1.1642e-005	6.3943e-006	3.2355e-006
	Varianza	9.0632e-012	2.5085e-012	4.0114e-011	4.3531e-011	1.6893e-011
18	Pe media	1.0250e-005	1.0666e-005	9.6845e-006	1.1875e-005	1.0047e-005
	Varianza	1.1671e-011	4.2453e-012	1.1819e-011	2.5212e-010	9.7027e-011
30	Pe media	1.0500e-005	1.1610e-005	8.9246e-006	5.5692e-006	1.4380e-006
	Varianza	6.5789e-012	7.3991e-012	5.8149e-011	4.8853e-011	5.0686e-012
40	Pe media	1.0250e-005	1.2457e-005	9.2643e-006	9.2226e-006	1.0908e-005
	Varianza	1.2092e-011	1.1855e-011	1.5745e-011	7.3001e-011	4.2768e-010

Tabla 4.12: resultados de CIS con modelo de canal con eco simple, $P_e=10^{-5}$.

M	$P_e=10^{-6}$	Gamma1	Gamma2	Gamma3	Gamma4	Gamma5
2	Pe media	1.4500e-006	1.2609e-006	1.2642e-006	1.2698e-006	1.2575e-006
	Varianza	1.1026e-012	8.3480e-016	8.7334e-016	2.8635e-016	7.3806e-016
6	Pe media	1.1500e-006	1.3426e-006	1.2749e-006	1.1705e-006	1.1780e-006
	Varianza	1.0816e-012	1.1157e-014	1.8805e-014	4.0535e-014	3.8308e-013
10	Pe media	6.5000e-007	7.3374e-007	7.3994e-007	4.2560e-007	6.9381e-007
	Varianza	6.6053e-013	1.4993e-014	1.0730e-013	3.3438e-013	3.9633e-012
18	Pe media	1.4500e-006	1.2094e-006	1.2518e-006	1.2753e-006	6.4047e-007
	Varianza	2.3658e-012	1.7371e-013	3.1160e-013	2.9804e-012	7.0195e-013
30	Pe media	1.0500e-006	1.4766e-006	1.4110e-006	4.5069e-007	6.0695e-007
	Varianza	7.8684e-013	7.9438e-013	4.0340e-012	2.5524e-013	2.3292e-012
40	Pe media	1.0500e-006	1.1180e-006	6.8645e-007	1.7662e-006	2.2932e-007
	Varianza	2.2605e-012	1.0499e-012	4.2418e-013	1.4335e-011	5.2768e-014

Tabla 4.13: resultados de CIS con modelo de canal con eco simple, $P_e=10^{-6}$.

En las tablas mostradas en las páginas anteriores podemos apreciar que las probabilidades de error simuladas son muy parecidas a los valores teóricos esperados.

Al igual que en el caso del canal AWGN, para comparar la eficacia de CIS respecto de Monte Carlo se recurre a la representación de la ganancia para los distintos valores del parámetro γ . En las figuras presentadas a continuación se muestran los resultados obtenidos.

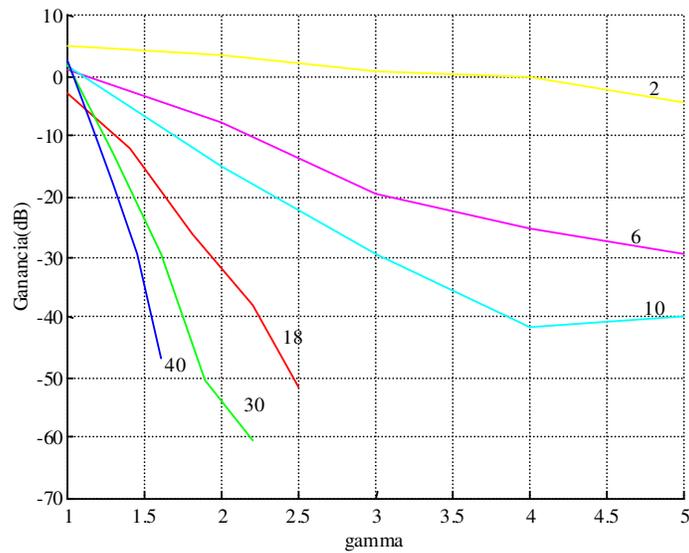


Figura 4.6: ganancia en función de gamma, $P_e=10^{-1}$.

En la figura 4.5 vemos que el método implementado no introduce mejoras apreciables para probabilidades de error grandes, del orden de 10^{-1} . Además, cuanto mayor es la memoria del sistema peor comportamiento presenta CIS respecto a Monte Carlo, como ya se sugería en el apartado 3.3.2. Y, por otra parte, ahora sí existe una fuerte dependencia respecto de γ , tanto mayor cuanto mayor es la memoria del sistema.

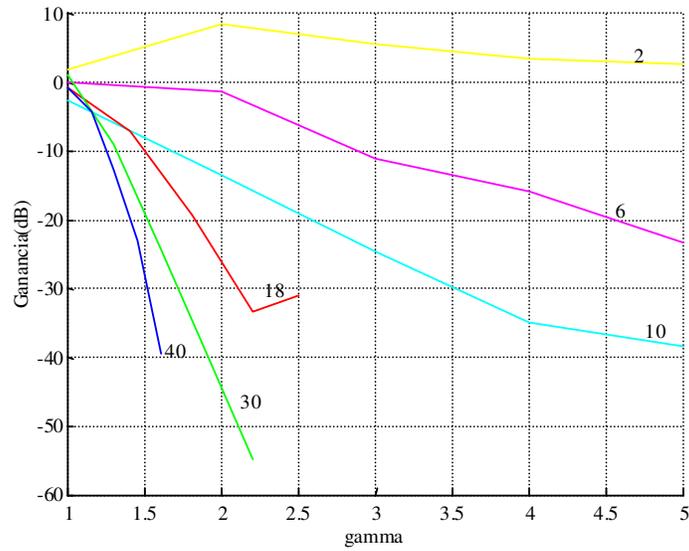


Figura 4.7: ganancia en función de gamma, $P_e=10^{-2}$.

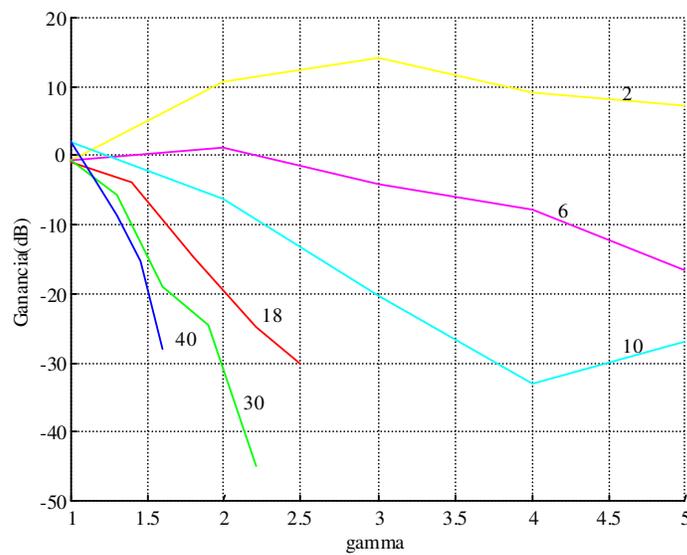


Figura 4.8: ganancia en función de gamma, $P_e=10^{-3}$.

En las figuras 4.6 y 4.7 podemos observar cómo la mejora introducida por el método CIS es cada vez mayor cuando aumentamos P_e , aunque aún está por debajo del método de Monte Carlo para la mayoría de los casos.

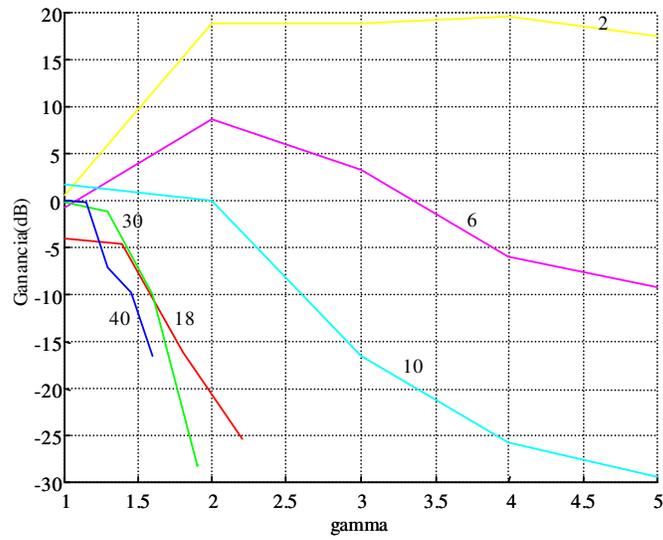


Figura 4.9: ganancia en función de gamma, $P_e = 10^{-4}$

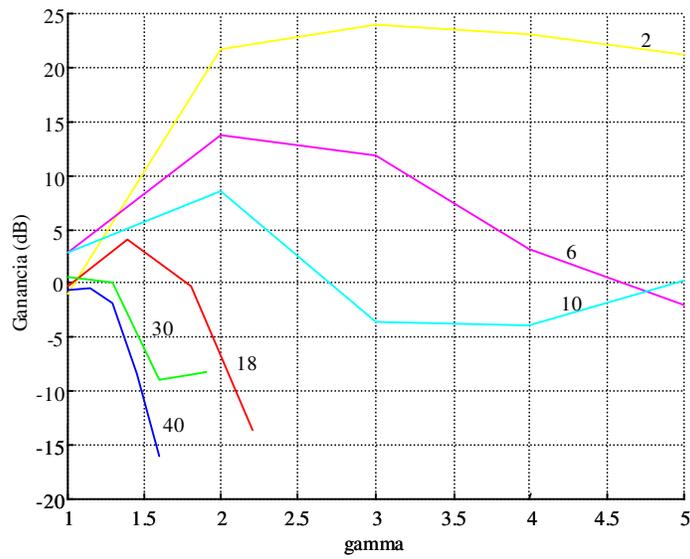


Figura 4.10: ganancia en función de gamma, $P_e = 10^{-5}$

En las figuras 4.9 y 4.10 podemos corroborar la creciente mejora introducida al disminuir la probabilidad de error.

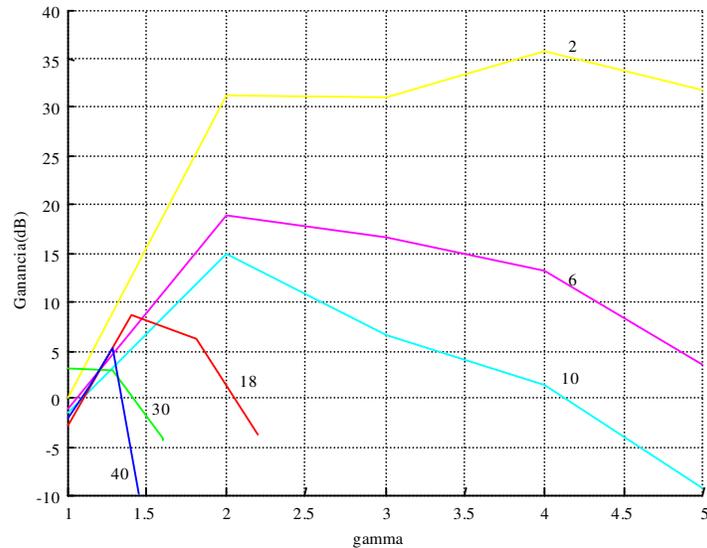


Figura 4.11: ganancia en función de gamma, $P_e = 10^{-6}$

Por último, en la figura 4.11 vemos la gráfica para $P_e = 10^{-6}$, donde apreciamos que la mejora introducida por el método CIS es la mayor de entre todas las probabilidades de error simuladas. Es de suponer que dicha mejora seguiría creciendo si se realizaran estimaciones para probabilidades de error incluso menores.

Observamos, no obstante, que sigue existiendo una fuerte dependencia respecto de la memoria del sistema. Se aprecia así una marcada disminución de la ganancia al aumentar el orden, lo que supone que para valores medianamente altos del mismo este método no resulte interesante.

En este caso también sigue apareciendo una dependencia acusada respecto a γ , mayor para valores altos del orden del sistema. Podemos advertir que la mejora introducida disminuye al aumentar el valor de γ por encima de un cierto valor óptimo, el cual depende del orden. Es por ello que para valores de M mayores que 10 ha sido

necesaria la elección de un rango de valores de γ más reducido para poder reflejar la evolución de la ganancia correctamente, como ya se comentó con anterioridad.

En definitiva, hemos podido ver que con el método CIS se introducen mejoras muy notables respecto a Monte Carlo, mayores cuanto menor es la probabilidad de error. Sin embargo, presenta una dependencia con la memoria del sistema que hace que su uso esté muy restringido. Sería interesante, por tanto, encontrar otro método que elimine dicha dependencia, para poder usarlo cualquiera que sea la memoria. Esto se intentará conseguir, como sabemos, con el muestreo enfatizado eficiente, que analizaremos más adelante.

4.3 SIMULACIÓN DEL MÉTODO IIS

En este método se plantea la variación de la media del ruido para conseguir mejoras respecto al método de Monte Carlo. A diferencia de CIS, la varianza del ruido permanece invariable. En ningún caso se plantea la variación de ambos parámetros simultáneamente, ya que se supone que esto no introduce ninguna mejora [2].

El problema que se plantea a la hora de implementar este método es la elección de la media, ya que la mejora introducida será fuertemente dependiente de ésta, según más tarde comprobaremos. Como ya vimos en el apartado 3.3.3, los resultados obtenidos son mejores tomando valores variables de la media en lugar de constantes; sin embargo, en nuestro caso, y dado que la respuesta impulsiva de un canal AWGN es la función impulso unidad, vimos que la media sí podía ser fija. Queda abierta la elección del valor óptimo de la misma, lo cual trataremos de determinar en lo que sigue.

Por otra parte, el valor de la constante C en la ecuación (3.24), y por tanto de la media, se ha parametrizado en función del cociente entre la media y la varianza, dado que ésta es un valor fijo. En la tabla 4.14 podemos observar los valores adoptados en las simulaciones. En el caso en el que la media tome valor nulo éste método es equivalente al de Monte Carlo, pues la ponderación se hace con un peso que toma valor uno.

μ_i/σ	μ_i	μ_i/σ	μ_i
-8	-1,2648	4	0,6324
-6	-0,9486	3,5	0,55335
-4	-0,6324	3	0,4743
-3.5	-0,55335	2,5	0,39525
3	0,4743	2	0,3162
-2.5	-0,39525	1,5	0,23715
-2	-0,3162	0,4	0,06324
-1.6	-0,25296	0,2	0,03162
-1.2	-0.18972	0	0
-0.8	-0.12648		
-0.6	-0.09486		
-0.4	-0.06324		
-0.2	-0.03162		

Tabla 4.14: valores de la media usados en la simulación.

En la tabla 4.15 se muestran los resultados de las probabilidades de error medias y varianzas obtenidas en las simulaciones de este método. En ella podemos apreciar que las probabilidades halladas son muy similares a los valores teóricos.

Por otra parte, al igual que en CIS, para poder evaluar las mejoras introducidas frente al método de Monte Carlo, se define la ganancia como

$$\eta = \frac{\xi_{MC}}{\xi_{IIS}}. \quad (4.12)$$

En la figura 4.12 podemos observar la representación de la ganancia conseguida en función de la media del ruido. Si nos fijamos en concreto en los valores máximos y los comparamos con los obtenidos en CIS, (figura 4.5), vemos que los de IIS son algo mejores para las probabilidades de error más pequeñas. No obstante, a medida que la

probabilidad de error aumenta las ganancias son menores, así como más parecidas a las obtenidas en el método CIS.

Sin embargo, estas mayores ganancias de IIS no son tan ventajosas, pues presentan una marcada dependencia con el valor de la media, de forma que una mala elección de la misma puede incluso dar lugar a una ganancia negativa, es decir, empeorarían los resultados respecto de los obtenidos con el método de Monte Carlo. Esta dependencia afecta más a las probabilidades de error mayores.

Por otro lado, se comprueba que existe una notable simetría de las gráficas respecto del origen. Esto quiere decir que a la hora de variar la media de la función de densidad, es equivalente hacerlo en sentido positivo o negativo.

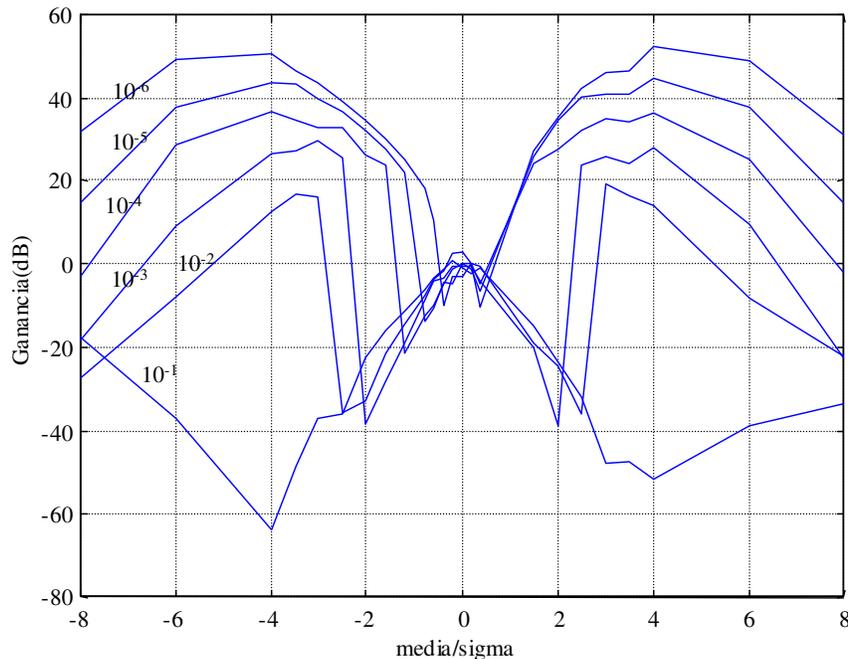


Figura 4.12: ganancia en función de la media.

En resumen, con este método se obtienen ganancias muy importantes para algunos valores de la media, pero fuera del rango adecuado las ganancias sufren una acusada disminución, por lo que este método no es recomendable si no se es cuidadoso

a la hora de elegir el valor de la media de la función de densidad de probabilidad sesgada.

		$\mu/\sigma =6$	$\mu/\sigma =4$	$\mu/\sigma =3.5$	$\mu/\sigma =3$	$\mu/\sigma =2.5$	$\mu/\sigma =2$	$\mu/\sigma =1.5$
$Pe=10^{-1}$	Pe media	5.5632e-002	1.2086e-001	9.7782e-002	1.1271e-001	1.0268e-001	9.9052e-002	1.0004e-001
	Varianza	6.7408e-004	5.0864e-002	5.1011e-003	5.4834e-003	1.4194e-004	1.8710e-005	2.6786e-006
$Pe=10^{-2}$	Pe media	4.9195e-003	4.9916e-003	4.9918e-003	4.9923e-003	9.9679e-003	9.0784e-003	1.0115e-002
	Varianza	8.1199e-008	4.9851e-010	2.8020e-010	1.4177e-010	5.1459e-005	3.4392e-006	8.9010e-007
$Pe=10^{-3}$	Pe media	4.9813e-004	4.9941e-004	4.9953e-004	4.9948e-004	4.9978e-004	1.5104e-003	7.9091e-004
	Varianza	1.0539e-010	1.4449e-012	3.4665e-012	2.2846e-012	3.6617e-012	6.9905e-006	9.1884e-008
$Pe=10^{-4}$	Pe media	4.9979e-005	4.9970e-005	4.9919e-005	4.9920e-005	1.8326e-002	4.9926e-005	4.9743e-005
	Varianza	2.7108e-013	2.0922e-014	3.3738e-014	2.7683e-014	6.6806e-003	1.5857e-013	3.3408e-013
$Pe=10^{-5}$	Pe media	5.0039e-006	4.9915e-006	4.9880e-006	4.9968e-006	4.9832e-006	4.9772e-006	5.0293e-006
	Varianza	1.3266e-015	2.6845e-016	6.7846e-016	6.5896e-016	7.6014e-016	2.8009e-015	2.0095e-014
$Pe=10^{-6}$	Pe media	4.9912e-007	4.9770e-007	4.9993e-007	4.9961e-007	5.0100e-007	5.0663e-007	4.9138e-007
	Varianza	7.5637e-018	3.4703e-018	1.3866e-017	1.4685e-017	3.4571e-017	1.7883e-016	1.1714e-015

Tabla 4.15: resultados obtenidos en el método IIS

		$\mu/\sigma = 0.4$	$\mu/\sigma = 0.2$	$\mu/\sigma = 0$	$\mu/\sigma = -0.2$	$\mu/\sigma = -0.4$	$\mu/\sigma = -0.6$	$\mu/\sigma = -0.8$	$\mu/\sigma = -1.2$
$Pe=10^{-1}$	Pe media	9.9919e-002	1.0002e-001	9.9942e-002	9.9952e-002	1.0003e-001	1.0008e-001	1.0014e-001	9.9805e-002
	Varianza	1.0822e-007	1.6065e-007	1.0469e-007	7.4546e-008	1.1970e-007	1.8932e-007	3.7863e-007	1.2464e-006
$Pe=10^{-2}$	Pe media	1.0013e-002	1.0025e-002	9.9813e-003	9.9839e-003	1.0018e-002	9.9948e-003	9.8979e-003	1.0008e-002
	Varianza	1.2716e-008	1.1894e-008	5.9323e-009	7.0372e-009	1.6787e-008	2.8181e-008	6.4600e-008	3.3342e-007
$Pe=10^{-3}$	Pe media	9.9601e-004	9.9764e-004	9.9655e-004	1.0111e-003	1.0142e-003	1.0172e-003	1.0272e-003	9.4763e-004
	Varianza	2.3873e-009	9.5420e-010	9.9416e-010	9.5522e-010	1.9139e-009	2.3709e-009	6.7704e-009	6.9210e-008
$Pe=10^{-4}$	Pe media	9.9193e-005	1.0199e-004	1.0255e-004	1.0428e-004	9.9197e-005	1.0595e-004	9.6939e-005	1.0918e-004
	Varianza	2.8109e-010	8.6669e-011	9.0155e-011	1.1881e-010	2.6448e-010	7.9567e-010	1.4085e-009	1.1672e-008
$Pe=10^{-5}$	Pe media	9.4220e-006	8.768e-006	8.7500e-006	1.0718e-005	8.0647e-006	1.1405e-005	9.5804e-006	5.0411e-006
	Varianza	3.6357e-011	9.8568e-012	7.8816e-012	2.3264e-011	2.1083e-011	9.3512e-011	2.0114e-010	4.8817e-014
$Pe=10^{-6}$	Pe media	1.2196e-006	5.9708e-007	1.0500e-006	1.1767e-006	1.2796e-006	4.6317e-007	4.8183e-007	5.0375e-007
	Varianza	6.0297e-012	5.9836e-013	1.1026e-012	1.1079e-012	5.5787e-012	5.2066e-014	9.3640e-015	1.8681e-015

Tabla 4.16: resultados obtenidos en el método IIS

		$\mu/\sigma = -1.6$	$\mu/\sigma = -2$	$\mu/\sigma = -2.5$	$\mu/\sigma = -3$	$\mu/\sigma = -3.5$	$\mu/\sigma = -4$	$\mu/\sigma = -6$
$Pe=10^{-1}$	Pe media	1.0063e-001	1.0153e-001	1.0056e-001	9.9445e-002	8.7298e-002	4.9915e-002	4.6730e-002
	Varianza	3.6642e-006	1.6399e-005	3.6765e-004	4.4638e-004	6.3047e-003	2.5056e-007	4.4806e-004
$Pe=10^{-2}$	Pe media	1.0159e-002	1.1049e-002	7.3928e-003	4.9948e-003	4.9959e-003	5.0008e-003	4.9473e-003
	Varianza	1.6614e-006	2.3342e-005	4.4099e-005	2.9652e-010	2.3914e-010	6.1747e-010	6.9404e-008
$Pe=10^{-3}$	Pe media	1.1623e-003	1.0780e-003	4.9931e-004	4.9963e-004	4.9928e-004	4.9925e-004	4.9776e-004
	Varianza	5.7224e-007	6.7005e-006	2.5613e-012	1.0290e-012	1.8010e-012	1.8901e-012	1.1538e-010
$Pe=10^{-4}$	Pe media	4.9879e-005	4.9911e-005	5.0027e-005	4.9952e-005	4.9947e-005	4.9889e-005	4.9824e-005
	Varianza	3.7219e-013	2.0014e-013	4.6496e-014	4.4481e-014	3.0962e-014	1.9992e-014	1.1681e-013
$Pe=10^{-5}$	Pe media	4.9656e-006	5.0014e-006	4.9886e-006	4.9880e-006	4.9869e-006	4.9941e-006	4.9914e-006
	Varianza	1.4153e-014	5.0277e-015	1.6687e-015	7.9357e-016	3.8620e-016	3.5361e-016	1.2678e-015
$Pe=10^{-6}$	Pe media	4.9849e-007	4.9822e-007	4.9789e-007	4.9846e-007	4.9914e-007	4.9878e-007	4.9894e-007
	Varianza	5.9296e-016	2.0587e-016	7.3586e-017	2.4678e-017	1.2883e-017	5.1856e-018	6.7683e-018

Tabla 4.17: resultados obtenidos en el método IIS

4.4 SIMULACIÓN DEL MÉTODO EIS

En la simulación de este método sólo consideraremos el uso del modelo de canal con eco simple, en el que incluimos el efecto de la memoria, ya que en el caso del canal AWGN este método es equivalente al CIS, el cual ya hemos estudiado.

Dado que un filtro de Wiener siempre es lineal, podemos hacer uso de una de las propiedades que poseen los sistemas lineales: si su entrada es gaussiana, la salida también lo será. Por tanto, ya que el ruido que introduce el canal es gaussiano, a la salida del filtro de Wiener en recepción obtendremos un ruido que también lo es. La única diferencia estará en su varianza, pero este hecho no supone ningún problema, pues la misma es completamente predecible una vez conocida la respuesta impulsiva del filtro de Wiener.

Gracias a esta propiedad podemos calcular fácilmente los pesos a la salida del filtro, por lo que no necesitaremos obtener $2M$ funciones de densidad de probabilidad en cada iteración, como en CIS (M para cada una de las desviaciones típicas, la original y la modificada), sino sólo dos. Esto supone un gran ahorro computacional y una simplificación notable en los códigos empleados. De hecho, los tiempos de computación que han sido necesarios para la realización de las simulaciones del método EIS han sido mucho menores que los de CIS.

En las tablas 4.18 a 4.23 podemos observar los valores medios y las varianzas obtenidos en las simulaciones de EIS. Las probabilidades estimadas están de nuevo cercanas a las teóricas, sobre todo para valores de P_e grandes.

Para poder tener una idea de la mejora introducida por medio de EIS es necesario, tal y como hemos hecho con los métodos anteriores, calcular su ganancia respecto de Monte Carlo, que estará definida en este caso de la forma

$$\eta = \frac{\xi_{MC}}{\xi_{EIS}}. \quad (4.13)$$

En las figuras 4.13 a 4.18 podemos observar la representación de la ganancia en función del valor del parámetro γ para los distintos órdenes simulados.

M	$P_e=10^{-1}$	Gamma1	Gamma2	Gamma3	Gamma4	Gamma5
2	P_e media	1.0288e-001	1.0281e-001	1.0279e-001	1.0275e-001	1.0288e-001
	Varianza	6.2138e-008	6.2787e-008	7.6621e-008	8.8808e-008	1.6780e-007
6	P_e media	1.0227e-001	1.0245e-001	1.0229e-001	1.0231e-001	1.0227e-001
	Varianza	9.4933e-008	4.3469e-008	6.7707e-008	6.1277e-008	1.9785e-007
10	P_e media	1.0208e-001	1.0211e-001	1.0199e-001	1.0222e-001	1.0202e-001
	Varianza	8.0582e-008	1.0331e-007	6.2619e-008	1.2637e-007	1.4058e-007
18	P_e media	1.0193e-001	1.0185e-001	1.0198e-001	1.0194e-001	1.0192e-001
	Varianza	1.6603e-007	5.7230e-008	1.0559e-007	1.0521e-007	2.0682e-007
30	P_e media	9.9995e-002	9.9974e-002	9.9996e-002	1.0010e-001	1.0011e-001
	Varianza	7.8369e-008	8.9409e-008	1.3014e-007	1.5604e-007	1.6760e-007
40	P_e media	1.0019e-001	1.0029e-001	1.0016e-001	1.0020e-001	1.0029e-001
	Varianza	9.1200e-008	1.3601e-007	1.1529e-007	1.8639e-007	2.5473e-007

Tabla 4.18: resultados de EIS con modelo de canal con eco simple, $P_e=10^{-1}$.

M	$P_e=10^{-2}$	Gamma1	Gamma2	Gamma3	Gamma4	Gamma5
2	P_e media	1.0779e-002	1.0746e-002	1.0757e-002	1.0753e-002	1.0751e-002
	Varianza	1.3098e-008	1.9022e-009	1.7116e-009	3.4956e-009	5.1895e-009
6	P_e media	1.0676e-002	1.0651e-002	1.0659e-002	1.0656e-002	1.0663e-002
	Varianza	1.3904e-008	2.3488e-009	3.1824e-009	2.3256e-009	4.1992e-009
10	P_e media	1.0442e-002	1.0460e-002	1.0479e-002	1.0488e-002	1.0481e-002
	Varianza	1.6484e-008	2.8419e-009	5.5211e-009	3.5674e-009	6.5003e-009
18	P_e media	1.0098e-002	1.0076e-002	1.0090e-002	1.0096e-002	1.0127e-002
	Varianza	1.3903e-008	6.6099e-009	3.5730e-009	5.1074e-009	7.8892e-009
30	P_e media	1.0296e-002	1.0287e-002	1.0287e-002	1.0317e-002	1.0335e-002
	Varianza	1.0611e-008	4.0980e-009	5.3906e-009	6.8887e-009	8.1935e-009
40	P_e media	1.0303e-002	1.0304e-002	1.0303e-002	1.0320e-002	1.0316e-002
	Varianza	1.1322e-008	4.8215e-009	4.8875e-009	7.0382e-009	8.6288e-009

Tabla 4.19: resultados de EIS con modelo de canal con eco simple, $P_e = 10^{-2}$.

M	$P_e=10^{-3}$	Gamma1	Gamma2	Gamma3	Gamma4	Gamma5
2	P_e media	1.0827e-003	1.0869e-003	1.0873e-003	1.0859e-003	1.0862e-003
	Varianza	1.5423e-009	6.8904e-011	5.2139e-011	4.4827e-011	3.1602e-011
6	P_e media	1.1115e-003	1.1225e-003	1.1249e-003	1.1224e-003	1.1251e-003
	Varianza	1.5055e-009	1.3039e-010	5.7125e-011	8.8383e-011	1.1769e-010
10	P_e media	1.0113e-003	1.0140e-003	1.0120e-003	1.0151e-003	1.0113e-003
	Varianza	6.1308e-010	9.4662e-011	1.0680e-010	6.4884e-011	4.6693e-011
18	P_e media	1.0603e-003	1.0480e-003	1.0464e-003	1.0463e-003	1.0457e-003
	Varianza	1.2539e-009	1.2346e-010	1.8068e-010	1.9082e-010	2.2249e-010
30	P_e media	1.0551e-003	1.0559e-003	1.0502e-003	1.0550e-003	1.0557e-003
	Varianza	1.2416e-009	2.9760e-010	1.4693e-010	1.5927e-010	3.3373e-010
40	P_e media	1.0083e-003	1.0128e-003	1.0075e-003	1.0053e-003	1.0073e-003
	Varianza	1.0449e-009	1.8724e-010	3.7955e-010	3.5042e-010	2.7479e-010

Tabla 4.20: resultados de EIS con modelo de canal con eco simple, $P_e = 10^{-3}$.

M	$P_e=10^{-4}$	Gamma1	Gamma2	Gamma3	Gamma4	Gamma5
2	P_e media	1.0995e-004	1.0744e-004	1.0781e-004	1.0759e-004	1.0741e-004
	Varianza	1.1163e-010	1.4116e-012	7.7922e-013	6.7470e-013	1.1366e-012
6	P_e media	1.1850e-004	1.1615e-004	1.1525e-004	1.1557e-004	1.1560e-004
	Varianza	1.0921e-010	1.6024e-012	1.3976e-012	1.1150e-012	9.4701e-013
10	P_e media	1.1075e-004	1.0914e-004	1.0954e-004	1.0887e-004	1.0923e-004
	Varianza	6.1250e-011	1.5608e-012	1.6251e-012	1.7409e-012	1.7227e-012
18	P_e media	9.9300e-005	1.0011e-004	1.0110e-004	1.0170e-004	1.0087e-004
	Varianza	1.3443e-010	3.6376e-012	4.5895e-012	2.3499e-012	4.4964e-012
30	P_e media	9.8700e-005	1.0143e-004	1.0180e-004	1.0132e-004	1.0201e-004
	Varianza	1.0633e-010	6.6116e-012	5.8679e-012	3.1576e-012	6.1703e-012
40	P_e media	1.0635e-004	1.0549e-004	1.0768e-004	1.0826e-004	1.0840e-004
	Varianza	1.1466e-010	8.0002e-012	6.9905e-012	7.1021e-012	1.1716e-011

Tabla 4.21: resultados de EIS con modelo de canal con eco simple, $P_e=10^{-4}$.

M	$P_e=10^{-5}$	Gamma1	Gamma2	Gamma3	Gamma4	Gamma5
2	P_e media	1.2000e-005	1.1909e-005	1.1944e-005	1.1885e-005	1.1883e-005
	Varianza	9.7895e-012	3.7443e-014	1.5216e-014	1.7055e-014	1.0720e-014
6	P_e media	1.1850e-005	1.2053e-005	1.1934e-005	1.1965e-005	1.1986e-005
	Varianza	8.1342e-012	6.3005e-014	2.1432e-014	2.5602e-014	2.1708e-014
10	P_e media	1.1700e-005	1.1058e-005	1.1000e-005	1.1101e-005	1.1020e-005
	Varianza	9.0632e-012	6.4473e-014	3.3759e-014	3.3552e-014	3.0382e-014
18	P_e media	1.0250e-005	1.1036e-005	1.1030e-005	1.0958e-005	1.1023e-005
	Varianza	1.1671e-011	5.4669e-014	5.8721e-014	8.8440e-014	1.4709e-013
30	P_e media	1.0500e-005	1.0459e-005	1.0382e-005	1.0366e-005	1.0365e-005
	Varianza	6.5789e-012	2.4610e-013	9.2832e-014	1.2941e-013	9.3057e-014
40	P_e media	1.0250e-005	1.1173e-005	1.1300e-005	1.1179e-005	1.1223e-005
	Varianza	1.2092e-011	1.8893e-013	1.7062e-013	1.6291e-013	3.1986e-013

Tabla 4.22: resultados de EIS con modelo de canal con eco simple, $P_e = 10^{-5}$.

M	$P_e=10^{-6}$	Gamma1	Gamma2	Gamma3	Gamma4	Gamma5
2	P_e media	1.4500e-006	1.2627e-006	1.2657e-006	1.2676e-006	1.2638e-006
	Varianza	1.1026e-012	5.6738e-016	5.4094e-016	3.2403e-016	1.6442e-016
6	P_e media	1.1500e-006	1.2855e-006	1.2729e-006	1.2779e-006	1.2717e-006
	Varianza	1.0816e-012	1.4338e-015	4.8809e-016	2.6348e-016	2.1713e-016
10	P_e media	6.5000e-007	7.9525e-007	8.0482e-007	7.9491e-007	7.9426e-007
	Varianza	6.6053e-013	7.2261e-016	2.4916e-016	2.0701e-016	2.6858e-016
18	P_e media	1.4500e-006	1.3592e-006	1.3881e-006	1.3816e-006	1.3686e-006
	Varianza	2.3658e-012	2.2546e-015	2.7739e-015	2.2058e-015	9.7680e-016
30	P_e media	1.0500e-006	1.4104e-006	1.4154e-006	1.4140e-006	1.4419e-006
	Varianza	7.8684e-013	6.8032e-015	3.1764e-015	3.5360e-015	4.9740e-015
40	P_e media	1.0500e-006	1.2768e-006	1.2861e-006	1.2823e-006	1.2423e-006
	Varianza	2.2605e-012	1.5015e-014	3.7518e-015	8.3616e-015	9.8625e-015

Tabla 4.23: resultados de EIS con modelo de canal con eco simple, $P_e = 10^{-6}$.

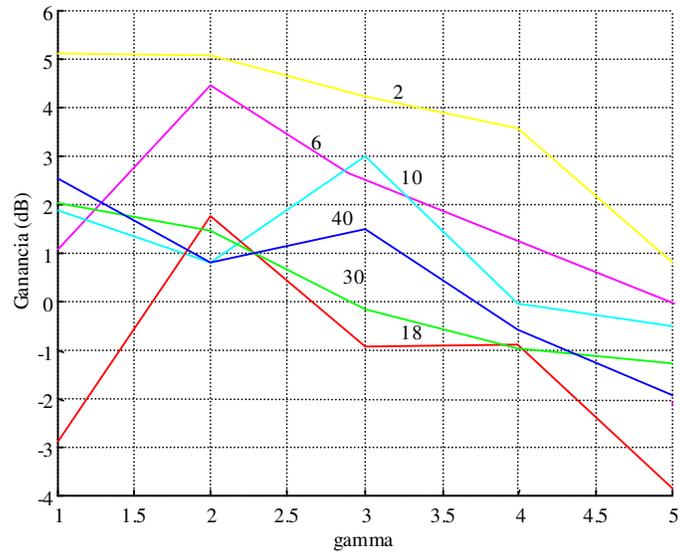


Figura 4.13: ganancia en función de γ , $P_e = 10^{-1}$.

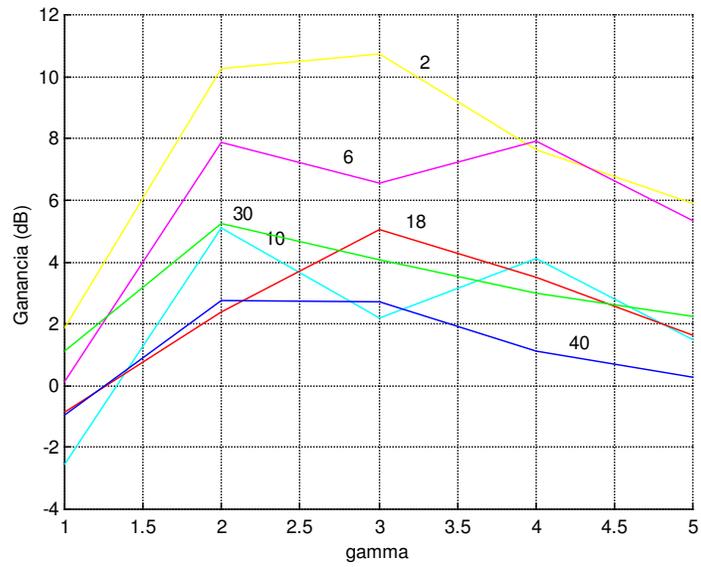


Figura 4.14: ganancia en función de γ , $P_e = 10^{-2}$.

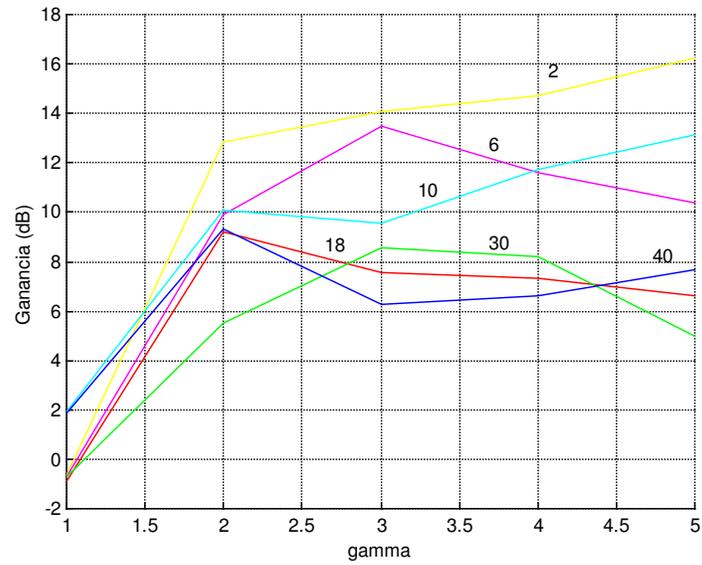


Figura 4.15: ganancia en función de γ , $P_e = 10^{-3}$.

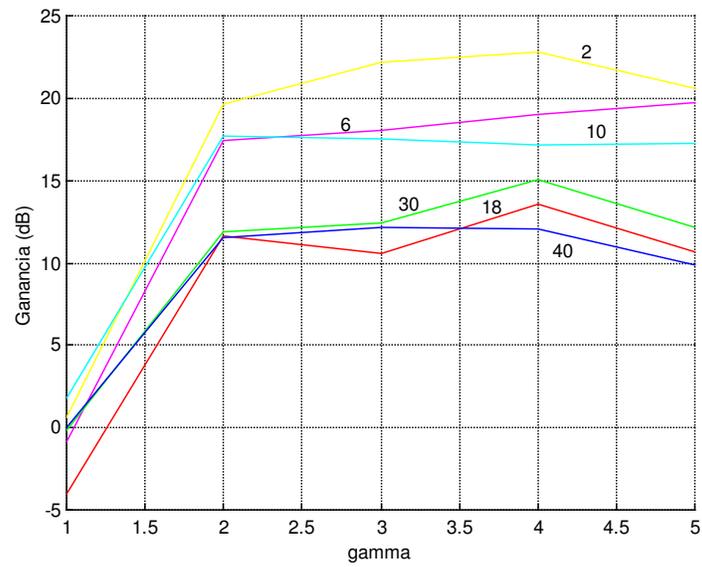


Figura 4.16: ganancia en función de γ , $P_e = 10^{-4}$.

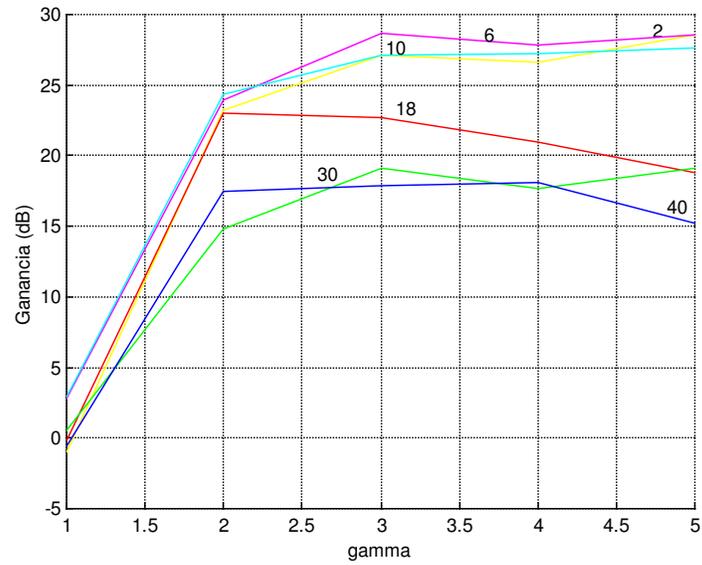


Figura 4.17: ganancia en función de γ , $P_e = 10^{-5}$.

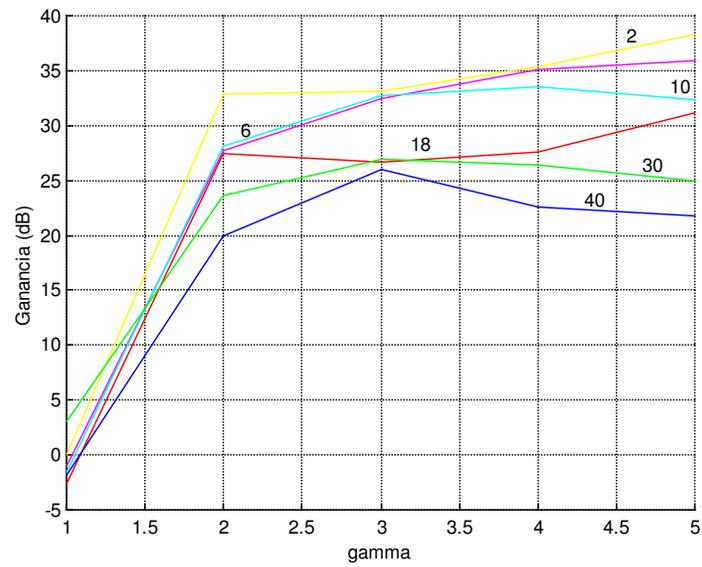


Figura 4.18: ganancia en función de γ , $P_e = 10^{-6}$.

Observando las figuras mostradas podemos notar que las ganancias introducidas son mayores cuanto menor es la probabilidad de error estimada.

En comparación con el método CIS presentado anteriormente, los valores máximos de la ganancia obtenidos son muy parecidos. La principal diferencia entre ambos es que mientras que en CIS la ganancia era fuertemente dependiente del orden del sistema, M , en este caso no existen variaciones notables de su valor. Esta independencia respecto a M es más significativa cuanto menor es la probabilidad de error que está siendo simulada.

Por otra parte, EIS también demuestra ser mucho menos dependiente de γ de lo que lo era CIS aplicado sobre un sistema con memoria. De hecho, esa dependencia es similar a la que tenía CIS cuando lo empleábamos en un canal AWGN, por lo que en este aspecto podemos afirmar que EIS es equivalente a CIS, aplicado éste en un sistema sin memoria, que era el mejor caso posible.

En resumen, el método EIS es muy similar al CIS, pero con dos ventajas muy importantes: por un lado se consigue una importante independencia de la efectividad del método respecto a la memoria del canal, y por otro los cálculos necesarios son mucho menores.

4.5 SIMULACIÓN DEL MÉTODO DE EXTRAPOLACIÓN DE COLA

Como vimos en el apartado 3.4, este método se basa en una expresión obtenida bajo ciertas suposiciones y aproximaciones, según la cual el doble logaritmo de la probabilidad de error es lineal con el logaritmo del umbral de decisión.

Dado que cualquier recta es determinada por la posición de dos de sus puntos, bastarán dos probabilidades de error para determinar dicha recta. Una vez conocida la misma, bastará con sustituir en ella el umbral que nos interesa en cada caso para obtener la probabilidad de error deseada.

Los pseudoumbrales deben ser equiespaciados entre sí en decibelios [2], y para cada uno de ellos se realiza una simulación de Monte Carlo con la cual se obtiene la probabilidad de error requerida. Luego se extrapolarán linealmente estos resultados para

estimar la P_e que realmente se desee. Estos cálculos se realizarán 20 veces partiendo de los mismos valores de pseudumbrales, y finalmente se promediarán los resultados obtenidos.

Se necesitan por tanto dos puntos o pseudumbrales como mínimo, pero es posible usar más de dos, en cuyo caso será necesario hallar la recta por algún método estadístico de aproximación. En las simulaciones realizadas ha sido usado el método de los mínimos cuadrados, para calcular la recta de regresión que mejor se ajusta a las probabilidades de error asociadas a los pseudumbrales.

Dado que cualquier recta se puede representar de la forma

$$y = \alpha + \beta x, \quad (4.14)$$

es necesario calcular los parámetros α y β para cada grupo de pseudumbrales escogidos, pues cada uno de ellos determinarán rectas de mejor ajustes distintas. De acuerdo con el método de los mínimos cuadrados [5], los parámetros α y β se obtienen a partir de

$$\alpha = \frac{FE - DC}{nF - D^2}, \quad \beta = \frac{nC - DE}{nF - D^2} \quad (4.15)$$

donde

$$C = \sum_{i=1}^n x_i y_i, \quad D = \sum_{i=1}^n x_i, \quad E = \sum_{i=1}^n y_i, \quad F = \sum_{i=1}^n x_i^2. \quad (4.16)$$

En estas ecuaciones, x_i, y_i son las coordenadas de los puntos a partir de los cuales se va a calcular la recta de regresión. Por lo tanto, x_i representa los pseudumbrales e y_i las probabilidades de error.

Una vez calculados los coeficientes C, D, E y F basta con sustituir en (4.15) y posteriormente en (4.14) para obtener la ecuación de la recta, a partir de la cual es inmediato obtener la probabilidad de error buscada para cualquier umbral x . Para ello simplemente usamos que, según lo visto en el apartado 3.4,

$$\ln[-\ln p(t)] = \alpha + \beta x. \quad (4.17)$$

Despejando de esta ecuación se obtiene el valor de la probabilidad de error asociado al umbral impuesto en las simulaciones.

En nuestro caso hemos considerado que el cociente $\sqrt{2}\sigma$ está incluido en t_k , siendo t_k el k -ésimo pseudoumbra, según la ecuación (3.34). Dicha suposición no supone ningún cambio sustancial y los resultados no se ven falseados por ella [2].

Nombre	DISTANCIAS AL UMBRAL (dB)			
21	2	1		
22	4	2		
23	6	3		
24	8	4		
25	3	2		
26	4	3		
27	5	4		
31	3	2	1	
32	6	4	2	
33	9	6	3	
34	4	3	2	
35	5	4	3	
36	9	8	7	
41	4	3	2	1
42	8	6	4	2
43	9	8	7	6

44	8	7	6	5
----	---	---	---	---

Tabla 4.24: grupos de pseudoumbrales usados en las simulaciones.

Para poder establecer comparaciones se han realizado simulaciones para grupos de umbrales distintos, tanto en número como en distancia entre sí. En la tabla 4.24 podemos ver los distintos pseudoumbrales usados en las simulaciones, expresados en dB respecto al valor del umbral requerido realmente. Dichos pseudoumbrales han sido tomados a igual distancia entre sí. Se ha tratado de contemplar la mayor variedad de casos posible. La numeración que identifica ha cada uno de los grupos de pseudoumbrales consta de dos dígitos: el primero de ellos representa el número de pseudoumbrales y el segundo identifica los distintos valores de pseudoumbrales escogidos.

Dado el gran número de valores medios obtenidos en las simulaciones de este método se ha optado por mostrar sólo aquellos casos que, de acuerdo con los resultados, se han considerado los mejores.

Para la comparación de los casos escogidos se ha tomado como criterio el valor obtenido para la varianza. En las tablas 4.25 a 4.28 se muestran los resultados para todos los casos planteados. Nótese que el menor valor ha sido sombreado con el fin de resaltar cuál de los grupos de pseudoumbrales resulta óptimo en cada caso.

Vemos que la elección óptima es muy dependiente del orden y de la probabilidad de error. Pero, dado que la finalidad del método es estimar probabilidades de error pequeñas, nos fijaremos en ellas para determinar los valores óptimos de los pseudoumbrales.

En las tablas podemos apreciar que para la estimación de una probabilidad de error de 10^{-6} las menores desviaciones típicas se obtienen para los casos que cuentan con tres y cuatro pseudoumbrales. Dependiendo del orden del sistema el grupo de pseudoumbrales óptimo es diferente, pero el caso 42 es el que más se repite. De cualquier forma, todos ellos tienen en común que los pseudoumbrales son equidistantes entre sí (como impusimos como requisito) y con el umbral original

Por tanto, la mejor elección será dependiente de las condiciones que se van a imponer en la simulación, pero como norma general se deben escoger pseudoumbrales equidistantes entre sí y equidistantes también con el umbral real.

$P_e=10^{-1}$	$M=2$	$M=6$	$M=10$	$M=18$	$M=30$	$M=40$
21	30.911e-004	43.462e-004	49.305e-004	34.316e-004	28.676e-004	32.201e-004
22	13.579e-004	14.506e-004	12.497e-004	12.201e-004	11.409e-004	11.762e-004
23	4.4129e-004	6.2442e-004	6.7871e-004	4.4289e-004	3.4470e-004	3.7178e-004
24	2.8590e-004	2.7262e-004	2.3587e-004	2.7698e-004	2.8624e-004	2.3306e-004
25	33.803e-004	30.619e-004	36.648e-004	30.898e-004	17.933e-004	17.485e-004
26	26.105e-004	18.091e-004	19.186e-004	20.565e-004	20.535e-004	17.776e-004
27	17.765e-004	12.476e-004	14.182e-004	13.318e-004	9.8297e-004	8.8698e-004
31	17.766e-004	15.470e-004	12.494e-004	16.074e-004	10.966e-004	16.288e-004
32	6.1357e-004	6.1887e-004	4.4138e-004	4.3562e-004	4.2132e-004	4.5584e-004
33	2.7856e-004	2.1360e-004	2.1307e-004	2.4112e-004	1.9645e-004	2.5891e-004
34	12.049e-004	16.525e-004	13.185e-004	9.8845e-004	9.9246e-004	8.6271e-004
35	6.5750e-004	9.7036e-004	9.7518e-004	6.9877e-004	7.4247e-004	6.0405e-004
36	2.4806e-004	1.4725e-004	2.7233e-004	3.4368e-004	4.1980e-004	5.9300e-004
41	13.293e-004	9.5838e-004	9.8579e-004	8.7709e-004	6.9733e-004	5.2626e-004
42	2.2966e-004	2.7750e-004	1.9494e-004	1.9329e-004	2.3647e-004	1.6448e-004
43	1.8218e-004	1.8566e-004	1.4036e-004	2.1213e-004	2.4063e-004	2.8598e-004
44	2.4632e-004	1.9774e-004	2.2325e-004	1.3463e-004	1.2943e-004	1.4365e-004

Tabla 4.25: varianzas obtenidas para $P_e=10^{-1}$.

$P_e = 10^{-2}$	$M=2$	$M=6$	$M=10$	$M=18$	$M=30$	$M=40$
21	7.8850e-004	3.6524e-004	3.1498e-004	1.8051e-004	1.1070e-004	0.8153e-004
22	0.7043e-004	0.9580e-004	0.6971e-004	1.1457e-004	1.0687e-004	1.5628e-004
23	0.8764e-004	1.0006e-004	0.8977e-004	98.323e-004	1.0324e-004	1.3424e-004
24	1.0097e-004	1.2724e-004	0.9298e-004	1.4796e-004	1.3462e-004	1.8028e-004
25	3.1137e-004	1.3088e-004	1.0910e-004	54.778e-004	1.8173e-004	2.0727e-004
26	0.7650e-004	0.8177e-004	0.8594e-004	1.5359e-004	2.4743e-004	3.4038e-004
27	0.9857e-004	1.7280e-004	1.9106e-004	3.1202e-004	2.6046e-004	4.9262e-004
31	1.5822e-004	1.5738e-004	1.1335e-004	84.360e-004	0.6083e-004	0.6051e-004
32	0.5433e-004	0.7791e-004	0.7637e-004	70.468e-004	1.0643e-004	1.1129e-004
33	0.6466e-004	0.9417e-004	1.0169e-004	1.1167e-004	1.0874e-004	1.2964e-004
34	1.2773e-004	0.6159e-004	0.7365e-004	96.381e-004	0.7410e-004	1.4643e-004
35	0.6892e-004	0.7755e-004	0.7440e-004	1.3893e-004	1.5445e-004	2.2524e-004
36	2.0614e-004	2.4559e-004	3.2555e-004	2.8641e-004	4.6975e-004	3.9100e-004
41	0.9160e-004	0.7484e-004	0.6315e-004	36.225e-004	1.0165e-004	0.8734e-004
42	0.6078e-004	0.9091e-004	0.7912e-004	55.792e-004	0.9905e-004	1.0666e-004
43	1.3476e-004	1.7744e-004	1.5000e-004	2.3273e-004	2.2593e-004	2.9270e-004
44	0.9915e-004	1.0871e-004	1.4954e-004	1.3043e-004	1.7191e-004	2.2559e-004

Tabla 4.26: varianzas obtenidas para $P_e=10^{-2}$.

$P_e = 10^{-3}$	$M=2$	$M=6$	$M=10$	$M=18$	$M=30$	$M=40$
21	3.2501e-005	2.1916e-005	3.5735e-005	6.3830e-005	5.9982e-005	9.2221e-005
22	4.3035e-005	3.2457e-005	4.5637e-005	4.4730e-005	5.0803e-005	6.4303e-005
23	3.2394e-005	4.8499e-005	4.3410e-005	5.1034e-005	7.2462e-005	5.5653e-005
24	4.5007e-005	3.4840e-005	4.3473e-005	4.5555e-005	4.7955e-005	3.8998e-005
25	3.6097e-005	4.4935e-005	6.4195e-005	6.0986e-005	7.9306e-005	12.328e-005
26	4.2685e-005	6.1668e-005	8.3926e-005	10.943e-005	15.370e-005	13.563e-005
27	5.3005e-005	10.470e-005	11.923e-005	12.915e-005	12.295e-005	17.331e-005
31	2.1936e-005	2.7478e-005	3.3473e-005	4.0487e-005	5.4602e-005	6.5803e-005
32	2.1143e-005	2.5929e-005	2.9116e-005	4.7656e-005	3.2111e-005	2.7621e-005
33	2.9892e-005	3.3517e-005	3.0436e-005	4.3084e-005	3.7965e-005	2.7187e-005
34	2.8271e-005	3.1578e-005	3.1982e-005	5.0758e-005	5.1763e-005	6.1536e-005
35	3.8593e-005	4.5807e-005	4.2653e-005	5.2795e-005	5.5089e-005	7.4217e-005
36	7.1190e-005	10.571e-005	8.6539e-005	10.725e-005	10.121e-005	10.124e-005
41	1.7802e-005	3.3401e-005	2.8243e-005	3.2362e-005	4.4094e-005	3.7128e-005
42	2.2741e-005	2.8051e-005	3.2955e-005	3.4629e-005	2.6926e-005	3.4085e-005
43	6.4154e-005	6.3144e-005	6.0842e-005	6.5030e-005	6.7466e-005	7.0683e-005
44	4.0933e-005	4.5206e-005	4.7107e-005	5.4892e-005	7.6128e-005	5.7801e-005

Tabla 4.27: varianzas obtenidas para $P_e=10^{-3}$.

$P_e = 10^{-4}$	$M=2$	$M=6$	$M=10$	$M=18$	$M=30$	$M=40$
21	0.9924e-005	1.3075e-005	1.5300e-005	2.9108e-005	1.6625e-005	1.9792e-005
22	1.0632e-005	0.7833e-005	1.1424e-005	1.4850e-005	1.6207e-005	1.7584e-005
23	1.0023e-005	1.3446e-005	1.0982e-005	1.5375e-005	1.0310e-005	1.2625e-005
24	1.0250e-005	1.0462e-005	0.7661e-005	1.1425e-005	0.9548e-005	1.0348e-005
25	1.1945e-005	2.2822e-005	1.7212e-005	2.7391e-005	2.7645e-005	3.4696e-005
26	1.7615e-005	2.4074e-005	2.5964e-005	1.6328e-005	2.5908e-005	3.1613e-005
27	1.8175e-005	2.2202e-005	2.1330e-005	2.9284e-005	4.3459e-005	3.8431e-005
31	0.8063e-005	1.0614e-005	0.8181e-005	1.2180e-005	1.5556e-005	1.3431e-005
32	0.7502e-005	0.8908e-005	0.9319e-005	1.1759e-005	1.1243e-005	0.8637e-005
33	0.6755e-005	0.7909e-005	0.8605e-005	1.0213e-005	0.5678e-005	0.7935e-005
34	0.9081e-005	1.1599e-005	1.0670e-005	1.1356e-005	1.4587e-005	1.8110e-005
35	1.2605e-005	1.4379e-005	1.0474e-005	1.5934e-005	1.8579e-005	2.2885e-005
36	1.6942e-005	3.3062e-005	2.6496e-005	3.5758e-005	2.8072e-005	1.6010e-005
41	0.7246e-005	0.5718e-005	0.6180e-005	0.6841e-005	0.8447e-005	1.1969e-005
42	1.0747e-005	0.9319e-005	0.5626e-005	0.7113e-005	0.7.10e-005	0.8776e-005
43	1.2055e-005	1.9349e-005	2.1243e-005	1.5753e-005	1.5872e-005	1.3014e-005
44	1.3259e-005	1.3586e-005	1.7030e-005	1.6011e-005	1.1805e-005	1.3551e-005

Tabla 4.26: varianzas obtenidas para $P_e=10^{-4}$.

$P_e=10^{-5}$	$M=2$	$M=6$	$M=10$	$M=18$	$M=30$	$M=40$
21	4.6344e-006	4.0935e-006	3.5674e-006	6.5991e-006	5.1567e-006	5.9453e-006
22	3.3071e-006	2.4122e-006	3.2228e-006	3.5816e-006	4.2832e-006	5.0345e-006
23	2.7001e-006	2.5463e-006	3.0346e-006	2.6509e-006	2.7273e-006	2.7664e-006
24	2.9925e-006	3.2932e-006	1.9536e-006	2.6693e-006	4.0873e-006	3.0586e-006
25	2.3113e-006	4.1262e-006	3.3611e-006	5.4388e-006	4.3528e-006	8.2208e-006
26	5.4827e-006	7.0411e-006	5.2998e-006	4.9636e-006	8.2052e-006	8.5858e-006
27	4.5019e-006	8.0093e-006	5.2022e-006	7.7384e-006	6.8204e-006	7.1039e-006
31	1.9270e-006	2.1402e-006	2.3898e-006	3.7502e-006	3.8004e-006	3.4943e-006
32	2.0517e-006	2.4994e-006	2.0004e-006	2.4164e-006	2.6881e-006	1.9735e-006
33	2.4622e-006	1.9345e-006	3.0653e-006	2.4033e-006	2.1362e-006	1.7496e-006
34	2.5553e-006	2.7584e-006	2.6201e-006	2.3256e-006	3.9669e-006	5.4344e-006
35	3.0396e-006	2.3193e-006	3.2110e-006	2.6158e-006	4.7995e-006	4.5199e-006
36	8.3392e-006	1.1011e-005	9.9637e-006	6.3259e-006	6.5167e-006	5.0001e-006
41	2.3047e-006	1.9396e-006	2.2736e-006	1.6841e-006	3.1189e-006	2.0976e-006
42	2.0920e-006	2.8471e-006	1.7500e-006	1.8827e-006	1.6766e-006	1.7976e-006
43	4.2083e-006	5.1064e-006	4.3175e-006	4.4069e-006	3.7097e-006	2.8630e-006
44	3.2173e-006	3.6930e-006	3.1761e-006	3.5185e-006	3.8391e-006	3.0575e-006

Tabla 4.27: varianzas obtenidas para $P_e=10^{-5}$.

$P_e = 10^{-6}$	$M=2$	$M=6$	$M=10$	$M=18$	$M=30$	$M=40$
21	8.7194e-007	12.210e-007	15.983e-007	8.0449e-007	21.178e-007	22.703e-007
22	5.7180e-007	6.8371e-007	4.7012e-007	11.192e-007	11.239e-007	18.428e-007
23	4.8558e-007	6.7161e-007	5.2582e-007	8.9136e-007	9.4380e-007	5.8936e-007
24	6.9152e-007	6.4748e-007	5.8923e-007	7.0103e-007	9.1864e-007	5.5916e-007
25	12.128e-007	11.702e-007	9.4754e-007	15.380e-007	23.392e-007	26.056e-007
26	7.7858e-007	12.565e-007	10.184e-007	14.958e-007	22.958e-007	17.977e-007
27	15.755e-007	14.069e-007	11.750e-007	20.697e-007	29.275e-007	17.259e-007
31	7.0123e-007	6.9133e-007	4.7849e-007	4.9736e-007	11.970e-007	9.2325e-007
32	5.1171e-007	3.6747e-007	3.2993e-007	6.6200e-007	8.0356e-007	5.2158e-007
33	5.7140e-007	5.6791e-007	4.3407e-007	8.9152e-007	5.5097e-007	5.1373e-007
34	6.4077e-007	6.5621e-007	6.2699e-007	10.589e-007	14.631e-007	8.4259e-007
35	5.1573e-007	6.3339e-007	5.1390e-007	8.2952e-007	11.331e-007	9.0368e-007
36	18.372e-007	32.320e-007	18.860e-007	27.538e-007	20.150e-007	11.202e-007
41	6.5204e-007	4.8594e-007	2.3455e-007	5.4448e-007	7.2459e-007	10.311e-007
42	4.4536e-007	4.4283e-007	3.0780e-007	3.8164e-007	7.0924e-007	4.3029e-007
43	9.7935e-007	18.292e-007	8.3729e-007	15.418e-007	15.161e-007	5.7417e-007
44	7.2961e-007	11.777e-007	9.5508e-007	13.623e-007	13.810e-007	9.8889e-007

Tabla 4.28: varianzas obtenidas para $P_e=10^{-6}$.

Caso 42		$P_e=10^{-1}$	$P_e=10^{-2}$	$P_e=10^{-3}$	$P_e=10^{-4}$	$P_e=10^{-5}$	$P_e=10^{-6}$
$M=2$	P_e media	1.0395e-001	1.0347e-002	1.1678e-003	1.4719e-004	2.2658e-005	3.5571e-006
	Varianza	5.2745e-008	3.6953e-009	0.51715e-009	11.550e-011	4.3766e-012	1.9835e-013
$M=6$	P_e media	1.0263e-001	1.0367e-002	1.2779e-003	1.7335e-004	2.6569e-005	4.1816e-006
	Varianza	7.7005e-008	8.2660e-009	0.78684e-009	8.6855e-011	8.1057e-012	1.9610e-013
$M=10$	P_e media	1.0197e-001	1.0388e-002	1.1864e-003	1.6808e-004	2.4470e-005	2.7912e-006
	Varianza	3.8001e-008	6.2603e-009	1.0861e-009	3.1652e-011	3.0627e-012	0.94739e-013
$M=18$	P_e media	1.0139e-001	1.0104e-002	1.2239e-003	1.5420e-004	2.4326e-005	4.7631e-006
	Varianza	3.7361e-008	3.1127e-009	1.1992e-009	5.0605e-011	3.5444e-012	1.4565e-013
$M=30$	P_e media	0.9955e-001	1.0411e-002	1.1743e-003	1.4760e-004	2.3889e-005	5.6492e-006
	Varianza	5.5919e-008	9.8108e-009	7.2502e-010	5.1986e-011	2.8108e-012	5.0302e-013
$M=40$	P_e media	0.9957e-001	1.0307e-002	1.0765e-003	1.3881e-004	2.0398e-005	3.8727e-006
	Varianza	2.7054e-008	11.377e-009	1.1618e-009	7.7035e-011	3.2315e-012	1.8515e-013

Tabla 4.29: resultados de la simulación para el caso 42 en extrapolación de colas.

En la tabla 4.29 se presentan los valores medios del caso 42. (Las tablas correspondientes al resto de casos no se han incluido en la presente memoria debido al considerable número de las mismas y a que no aportan información relevante). Por último, en la tabla 4.30 podemos observar los valores de ganancia obtenidos respecto al método de Monte Carlo para $P_e = 10^{-6}$. Vemos que es con el caso 42 para el que se obtienen por lo general mayores ganancias, como era de esperar. No obstante, estos valores de ganancia están en cualquier caso por debajo de los conseguidos con el muestreo enfatizado.

$G(\text{dB})$	M						G media
Nombre	2	6	10	18	30	40	
21	1,6249	-2,4267	-7,4160	2,9402	-4,5705	-5,3530	-2,5335
22	5,2898	2,6101	3,2131	0,0725	0,9330	-3,5410	1,4296
23	6,7094	2,7653	2,2406	2,0495	2,4498	6,3609	3,7626
24	3,6386	3,0830	1,2516	4,1359	2,6844	6,8178	3,6019
25	-1,2414	-2,0576	-2,8746	-2,6888	-5,4339	-6,5497	-3,4743
26	2,6086	-2,6756	-3,5012	-2,4468	-5,2715	-3,3257	-2,4354
27	-3,5140	-3,6576	-4,7438	-5,2675	-7,3825	-2,9719	-4,5896
31	3,5175	2,5139	3,0599	7,1172	0,3851	2,4622	3,1760
32	6,2542	8,0032	6,2888	4,6334	3,8469	7,4220	6,0748
33	5,2958	4,2220	3,9061	2,0480	7,1247	7,5538	5,0251
34	4,3006	2,9667	0,7121	0,5533	-1,3583	3,2562	1,7384
35	6,1862	3,2742	2,4398	2,6741	0,8619	2,6482	3,0141
36	-4,8486	-10,8817	-8,8534	-7,7481	-4,1383	0,7826	-5,9479
41	4,1492	5,5760	9,2525	6,3309	4,7455	1,5029	5,2595
42	7,4604	6,3828	6,8920	9,4175	4,9315	9,0933	7,3629
43	0,6159	-5,9377	-1,8002	-2,7100	-1,6672	6,5878	-0,8186
44	3,1728	-2,1127	-2,9435	-1,6346	-0,8568	1,8655	-0,4182

Tabla 4.30: ganancia del método de extrapolación de cola respecto a MC (dB) para $P_e=10^{-6}$.

Uno de los inconvenientes que presenta este método [2] es la inexactitud de la probabilidad de error estimada para valores reducidos de la misma. Esto sucede porque

se utiliza una recta que es una aproximación en sí misma, y además se extiende su validez a valores de probabilidades de error más restrictivas.

Para apreciar este hecho se representa el cociente entre la probabilidad de error estimada y el valor hallado con el método de Monte Carlo (que será más cercano al teórico ya que se sabe que la estimación es insesgada en este caso). Si el método no presentara el efecto antes comentado el resultado del cociente debería ser aproximadamente la unidad.

En las figuras 4.19 a 4.35 podemos observar dichas gráficas representadas logarítmicamente en ambos ejes. Cada una de las figuras corresponde a uno de los casos presentados en la tabla 4.24 y en cada una de ellas se reflejan las curvas obtenidas para cada orden simulado. Nótese que como norma general la desviación del valor estimado respecto al teórico es tanto más fuerte cuanto menor es la probabilidad que se está estimando.

El comportamiento al variar el orden dentro de una misma elección de pseudumbrales es dispar, pero en general existe una mayor desviación cuanto mayor es el orden del sistema.

Haciendo una comparación entre los distintos pseudumbrales escogidos vemos que las menores desviaciones se obtienen en los casos 21, 31 y 41. En ellos los puntos escogidos como pseudumbrales están más cercanos al punto de estudio que en el resto de las posibilidades. Por tanto, desde el punto de vista de la desviación de la probabilidad estimada respecto de la teórica será conveniente escoger valores de pseudumbrales lo más cercanos posible al punto de estudio, siempre equidistantes entre sí.

En comparación con la valoración de los pseudumbrales que hicimos anteriormente, vemos que en la mayor parte de las elecciones óptimas de los mismos desde el punto de vista de la desviación típica, el valor obtenido de la probabilidad de error presentará una importante desviación respecto del teórico.

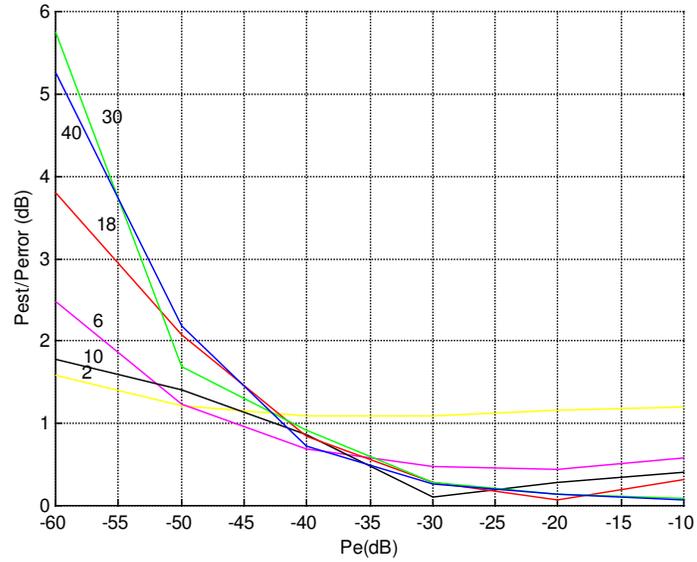


Figura 4.19: cociente entre las probabilidades estimada y teórica para el caso 21.

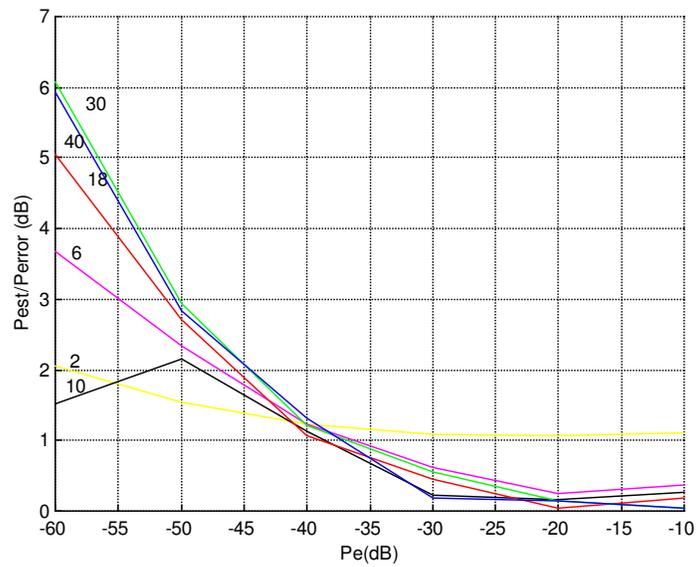


Figura 4.20: cociente entre las probabilidades estimada y teórica para el caso 22.

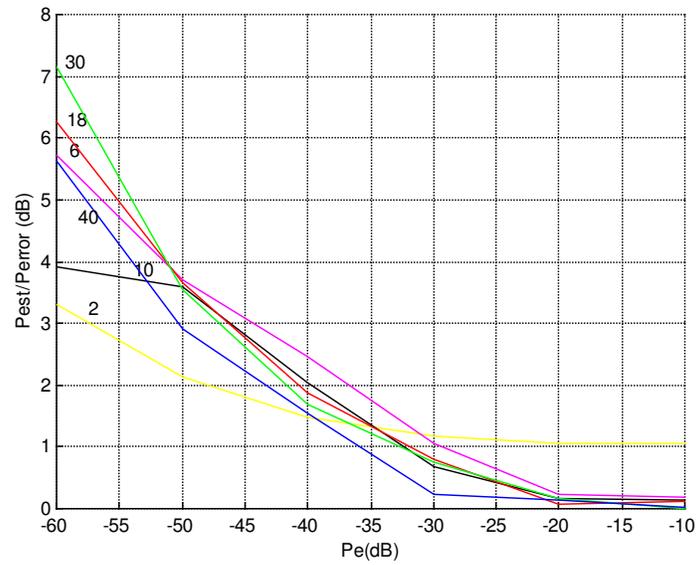


Figura 4.21: cociente entre las probabilidades estimada y teórica para el caso 23.

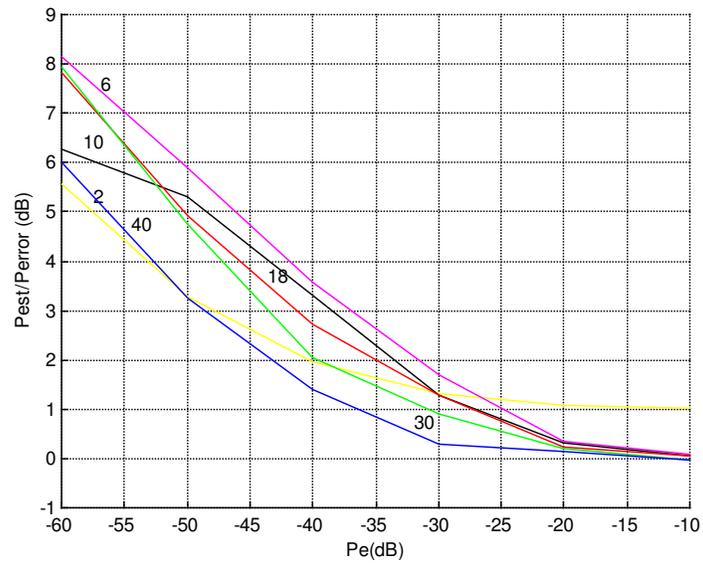


Figura 4.22: cociente entre las probabilidades estimada y teórica para el caso 24.

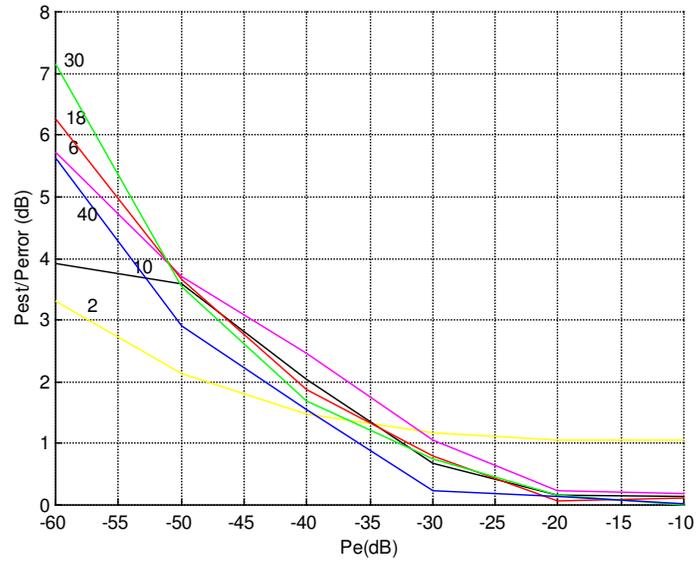


Figura 4.23: cociente entre las probabilidades estimada y teórica para el caso 25.

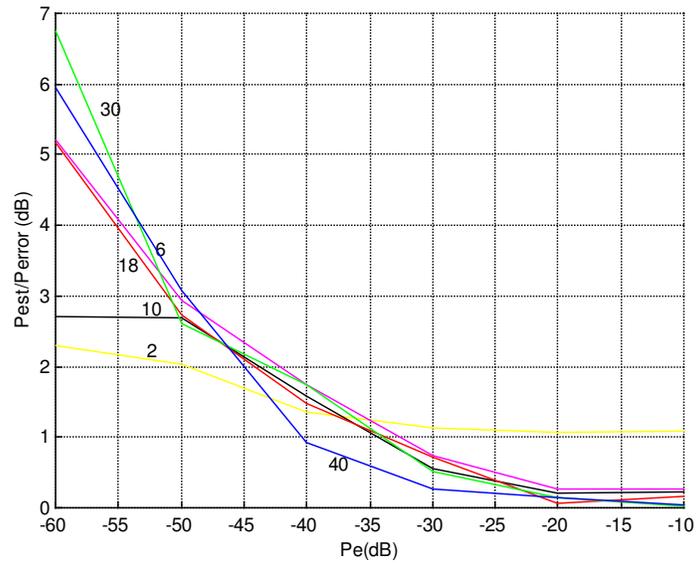


Figura 4.24: cociente entre las probabilidades estimada y teórica para el caso 26.

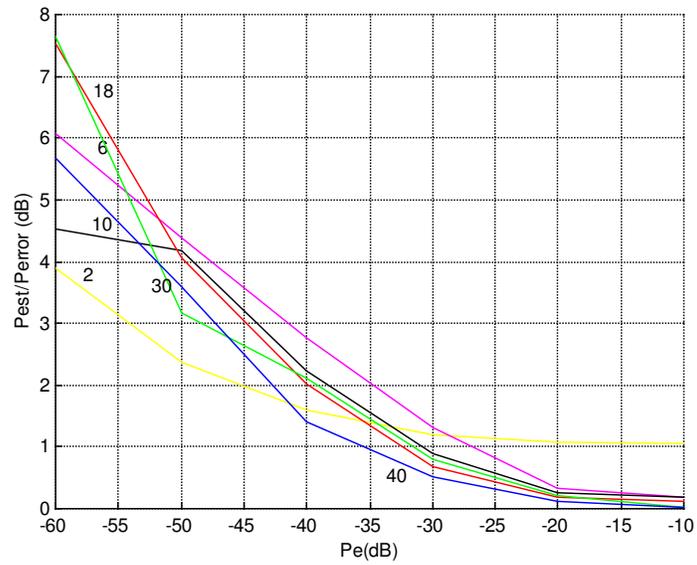


Figura 4.25: cociente entre las probabilidades estimada y teórica para el caso 27.

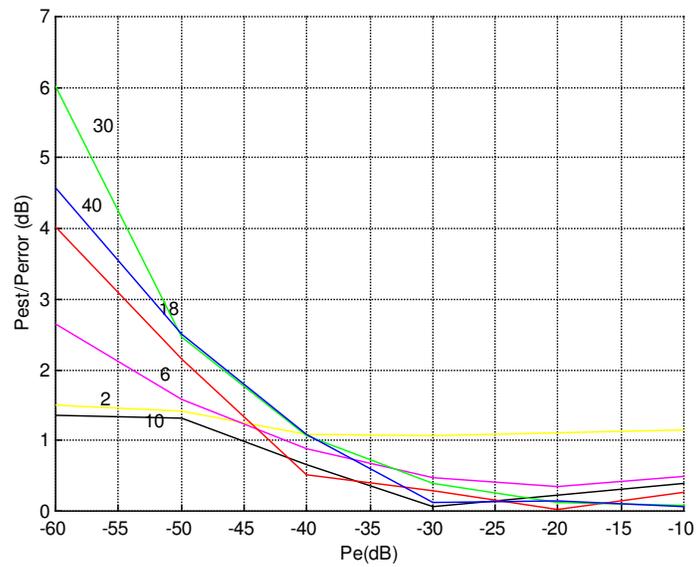


Figura 4.26: cociente entre las probabilidades estimada y teórica para el caso 31.

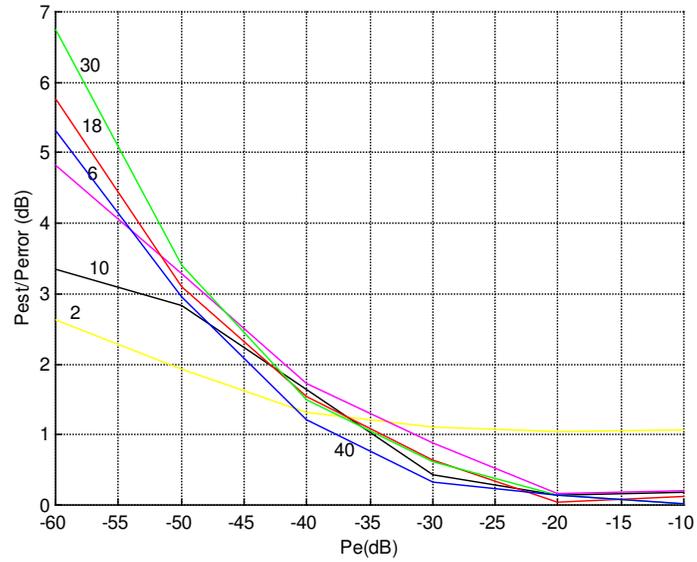


Figura 4.27: cociente entre las probabilidades estimada y teórica para el caso 32.

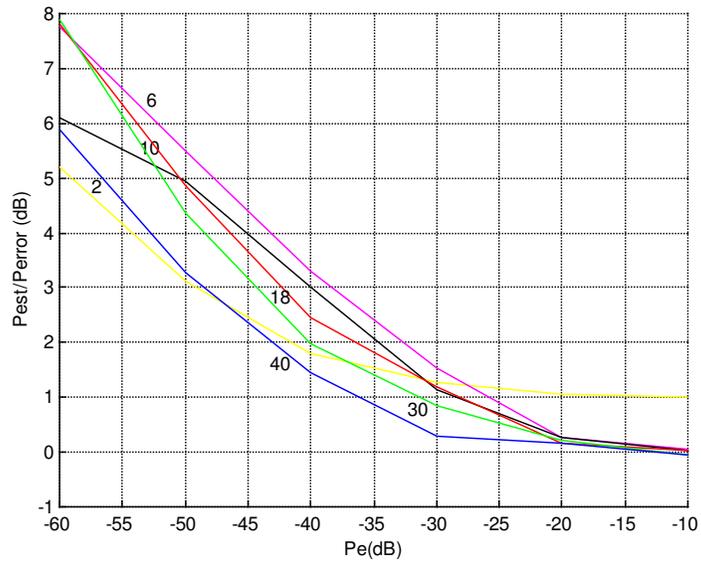


Figura 4.28: cociente entre las probabilidades estimada y teórica para el caso 33.

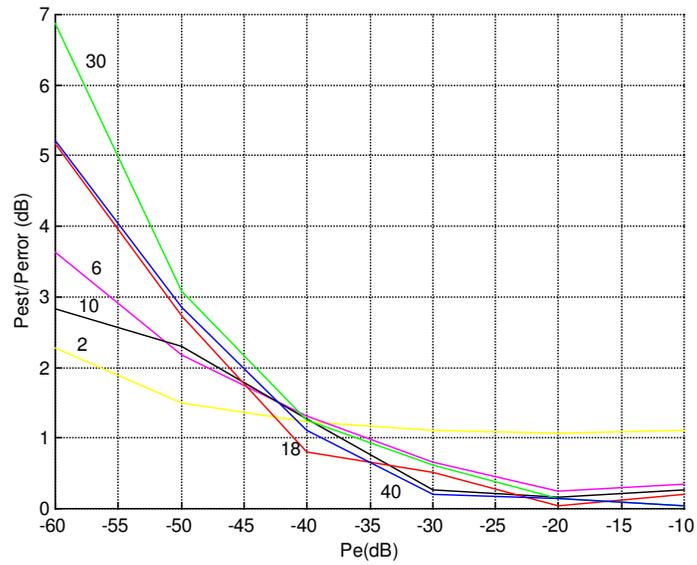


Figura 4.29: cociente entre las probabilidades estimada y teórica para el caso 34.

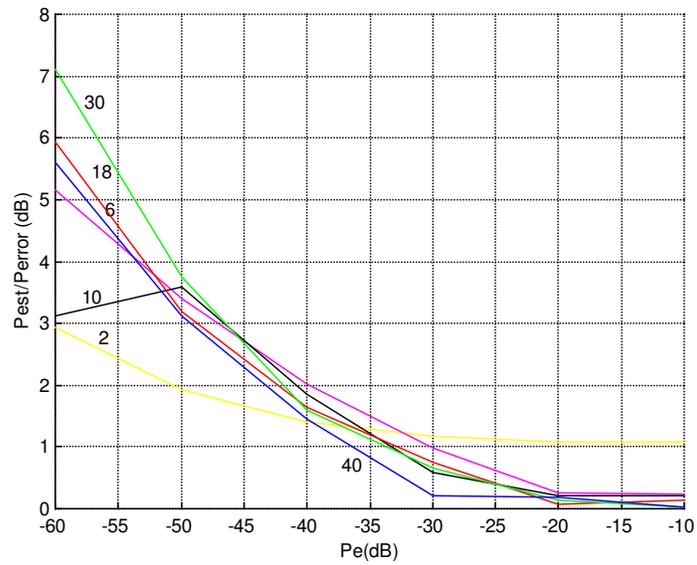


Figura 4.30: cociente entre las probabilidades estimada y teórica para el caso 35.

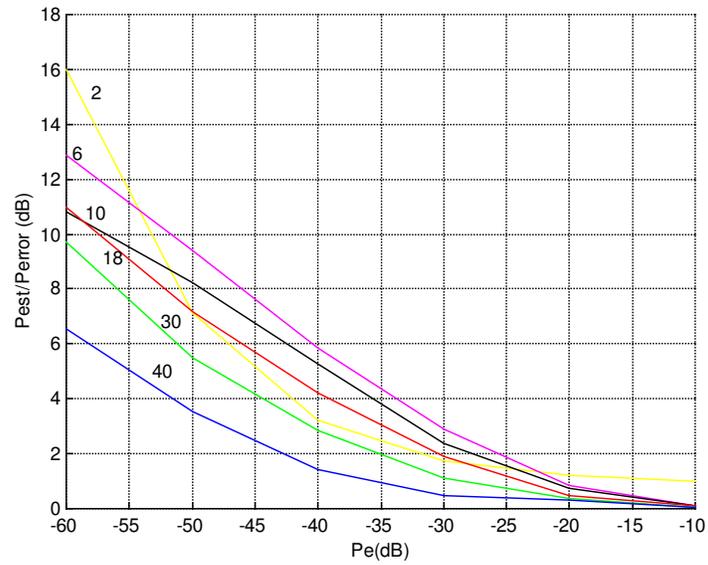


Figura 4.31: cociente entre las probabilidades estimada y teórica para el caso 36.

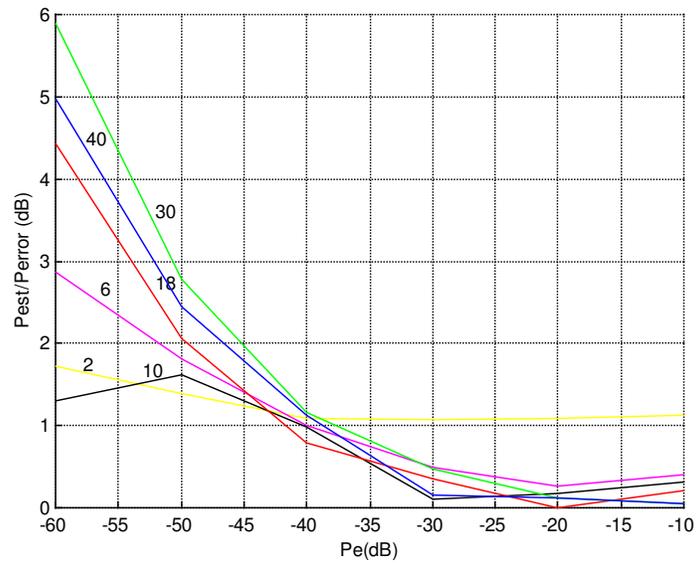


Figura 4.32: cociente entre las probabilidades estimada y teórica para el caso 41.

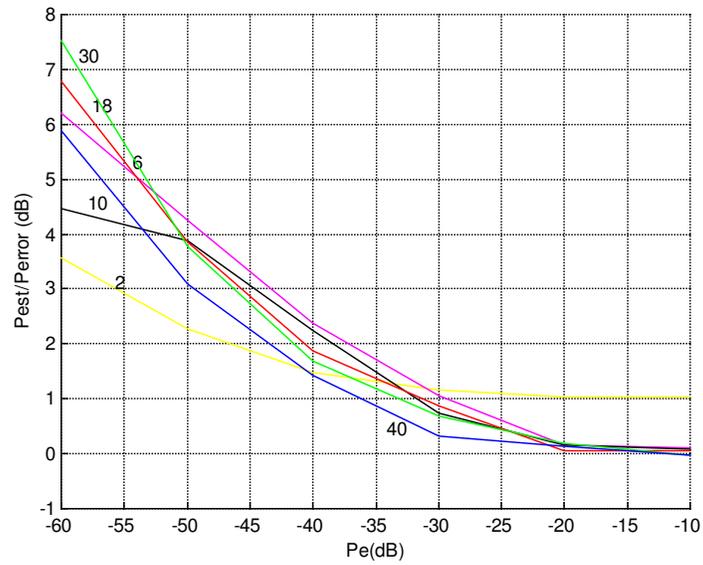


Figura 4.33: cociente entre las probabilidades estimada y teórica para el caso 42.

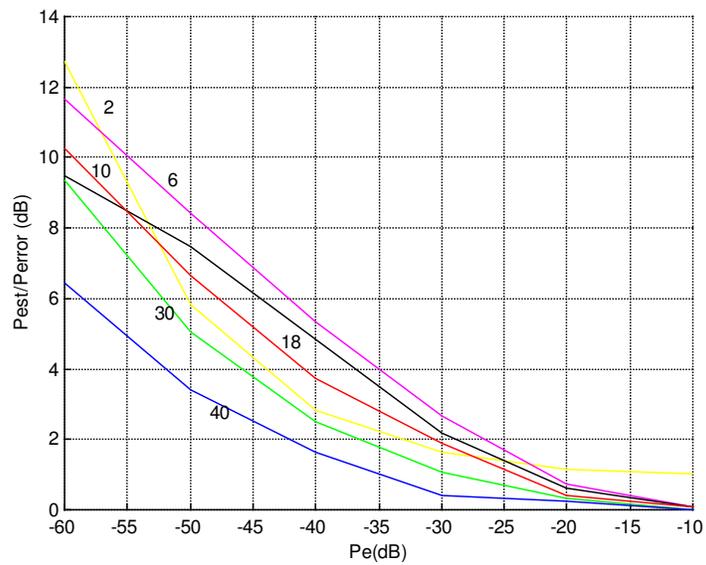


Figura 4.34: cociente entre las probabilidades estimada y teórica para el caso 43.

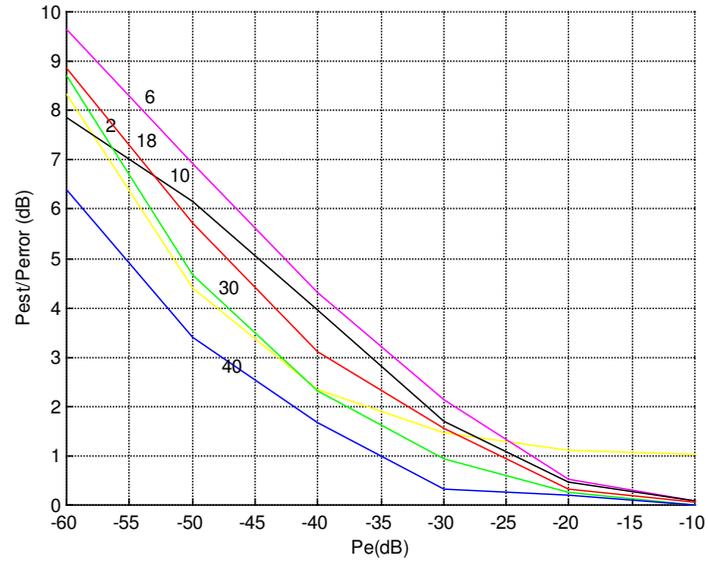


Figura 4.35: cociente entre las probabilidades estimada y teórica para el caso 44.

5 CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN

Hemos comprobado que la sencillez del método de Monte Carlo no es aprovechable en los casos en los que la probabilidad de error es demasiado pequeña, dado que requiere manejar secuencias de datos demasiados largas que ralentizan las simulaciones en exceso. Es por ello que se plantea la necesidad de encontrar nuevos métodos que permitan simular dichos valores de probabilidad sin la necesidad de usar demasiadas muestras.

Con el método CIS, en el que se modificaba la varianza del ruido introducido en el canal, hemos comprobado que se introducen notables mejoras. El problema es que dichas mejoras son fuertemente dependientes del orden del sistema, de forma que cuando éste toma valores grandes el método resulta del todo ineficaz. Otro de los problemas de este método es la carga computacional que conlleva el cálculo de las múltiples funciones de densidad de probabilidad para órdenes grandes.

En la misma línea, también hemos estudiado el método IIS, en el que es la media del ruido el parámetro modificado. Con este método se han obtenido unas ganancias algo superiores a las del método CIS, pero con una fuerte dependencia de la media. Por tanto, la ganancia obtenida decrece muy rápidamente cuando el valor escogido para la media de la función de densidad de probabilidad no es el óptimo, por lo que es necesaria una gran precisión absoluta a la hora de determinar dicho valor óptimo.

Una variación del método CIS se consigue aprovechando la linealidad del sistema, puesto que una secuencia gaussiana sigue siéndolo a la salida de un filtrado lineal. Este método implementado es el EIS, con el que se han conseguido muy buenos resultados en las mejoras introducidas, además de conseguir una independencia importantísima respecto al orden del sistema. Además de ello es importante señalar que la carga computacional es mucho menor que la de CIS, pues en este caso sólo es necesario calcular dos funciones de densidad de probabilidad independientemente de cuál sea el orden del sistema.

Por último, se ha implementado el método de extrapolación de colas, en el que se realiza una aproximación lineal. Dicha aproximación permite estimar una probabilidad de error muy pequeña a partir de dos o más puntos obtenidos mediante la simulación de probabilidades mayores. La ventaja de este método es su sencillez y rapidez. El problema es que la aproximación que es necesario realizar limita mucho la validez del método. Así, hemos visto que la probabilidad de error sufre desviaciones importantes respecto de su valor teórico cuando este es un valor pequeño.

En resumen, no es posible determinar a priori cuál es el método que supera en prestaciones a todos los demás, dado que cada uno de ellos presenta unas características que pueden hacer recomendables su uso allí donde todos los demás fallan. La elección del mejor método debe ser realizada una vez conocidas las condiciones de la simulación. De cualquier forma, es evidente que el método que presenta mejores características generales es el EIS, por lo que en caso de duda lo mejor es elegir dicho método para la realización de las simulaciones.

En la implementación de todos los métodos estudiados se ha usado un tipo de modulación muy sencilla, la BPSK. Podría estudiarse el comportamiento de estos métodos con otros modelos de modulación más complejos.

En las situaciones reales los sistemas no siempre son aproximables a sistemas lineales. Es por ello que resultaría interesante el estudio de la validez de los métodos implementados para sistemas no lineales.

Resulta interesante también la introducción de sistemas adaptativos, que vayan acomodando su comportamiento a las características instantáneas del sistema, son lo que presumiblemente se introducirán mejoras en los resultados obtenidos en este proyecto.

Por último, el uso de la extrapolación de cola obteniendo los pseudoumbrales con el muestreo enfatizado permitiría su comparación con el método usado en el presente proyecto.

ANEXO

A continuación se muestran los listados de los ficheros que han sido desarrollados durante la implementación en Matlab de los distintos métodos estudiados en el presente Proyecto. Sólo se incluyen los listados más relevantes en las simulaciones. No obstante, todos los ficheros usados pueden encontrarse en el disco compacto adjunto a la presente memoria.

En cada uno de los listados que se muestran a continuación se describen los parámetros de entrada de las funciones, así como comentarios que facilitan la comprensión de las órdenes más importantes.

Para cada uno de los métodos se incluye el fichero principal y todos los ficheros a los que se realizan llamadas desde él, bien directa o indirectamente.

```

function proyMC(Pe,path1,path2)

%Esta funcion simula el proceso de transmisión con ruido y recepción
%segun el analisis de Monte Carlo.
%Parametros:
%      Pe:Probabilidad de error a estimar
%      Path1: donde quiero que guarde los resultados de simular,
incluyendo el nombre del fichero
%      Path2: donde guarda los valores medios, incluyendo los
nombres de los ficheros

%En x la señal original. En w el ruido del canal(su varianza dependera
%de la relacion var=pot=No/2). En y la suma del ruido y de la señal.
%Despues se hace un bucle para comparar con el umbral de decision.
%Tomo No=0.05

clc
format long
t0=clock;
No=0.05;
Eb=No*Qinv(Pe);
amp=2*sqrt(Eb);           %Depende del valor de Eb: amp=2*sqrt(Eb)
sigma=0.1581
N=1000000;

for n=1:20

    x=rand(1,N);           %Genera numeros aleatorios uniformemente
    x=amp*round(x);       %Los convierto en ceros y "unos"(la
    amplitud depende de Eb/No
    w=sigma*randn(1,N);   %Genero ruido gaussiano media0 var No/2
    y=x+w;                %Sumo señal y ruido
    clear w

    rec=zeros(1,N);       %Inicializo el vector rec
    aux=find(y>(amp/2));  %aux tiene los indices en los que creo
    que hay "1"
    rec(aux)=amp*ones(1,length(aux)); %Pongo dichos indices a "1"
    clear aux

    z=x-rec;              %Matriz en la que si !=0, error
    clear rec y
    zerr=find(z);         %Matriz con los indices en los que hay
    errores
    errores=length(zerr)  %Numero de errores ocurridos
    clear zerr z x

    Pe=errores/N          %Probabilidad de error segun Monte Carlo
    eval(['save ..\' path1 ' num2str(n) '.mat']);

end

calc_est(path1,path2)    %Para calcular Pe media y varianza
Tiempo=etime(clock,t0)  %Tiempo que tarda

```

```

function proyCIS (Pe, numero, gamma, path1, path2)

%CONVENTIONAL IMPORTANCE SAMPLING---> Vario la varianza del ruido

%Parametros:
%      Pe:probabilidad de error
%      numero: es el exponente de la probabilidad de error (PANÁ)
%      gamma:cociente de sigmas
%      path1:donde quiero que se guarden los resultados de los
experimentos
%      (incluyendo hasta el nombre del fichero)
%      path2:donde quiero que se guarde la media y sigma de los
20 experimentos

%En x la señal original. En w el ruido del canal. En y la suma del
%ruido y de la señal. Despues se hace un bucle para comparar con el
%umbral de decision. Este bucle es muy parecido al de Monte Carlo
%pero antes de calcular la prob de error se multiplica por un peso.

%El parametro gamma es el valor que toma el cociente entre sigma y
sigmaestrella(sigmaster).
%Los valores de amp se calculan automaticamente haciendo llamadas
%a otros programas.

format long
t0=clock;

sigma=0.1581;
No=0.05;
Eb=No*Qinv (Pe);
amp=2*sqrt (Eb);           %Depende del valor de Eb: amp=2*sqrt (Eb)
sigmaster=sigma*gamma    %sigma varia segun gamma.Para generar f*
N=1000000;

for n=1:20

    x=rand (1,N);           %Genera numeros aleatorios uniformemente
    x=amp*round (x);        %Los convierto en ceros y "unos"
    w=sigmaster*randn (1,N); %f*:Genero ruido gaussiano con std
sigmaster
    y=x+w;                 %Sumo señal y ruido

    rec=zeros (1,N);       %Inicializo el vector rec
    aux=find (y>(amp/2));  %aux tiene los indices en los que creo
que hay "1"
    rec (aux)=amp*ones (1,length (aux)); %Pongo dichos indices a "1"
    clear aux

    f=1/(sigma*sqrt (2*pi))*exp (- (w).^2/(2*(sigma)^2));
                                %Numerador del peso(es la original:f)

    fstar=1/(sigmaster*sqrt (2*pi))*exp (- (w).^2/(2*(sigmaster)^2));
                                %Denominador del peso(es la
modificada:f*)

```

```

peso=abs(f./fstar);           %Estoy generando el peso(peso=f/f*)
clear num den

z=x-rec;                      %Matriz en la que si !=0, error
clear rec y x
zerr=find(z);                 %Matriz con los indices en los que hay
errores

errores=sum(peso(zerr));      %"numero" de errores que hay
clear peso w z i zerr f fstar

Pe=errores/N                  %Probabilidad de error
eval(['save ..\' path1 '' num2str(n) '.mat']);
clear errores
end

calc_est(path1,path2)         %Para calcular Pe media y varianza
%dibujaNvar(numero)          %Dibuja la grafica del N*varianza
Tiempo=etime(clock,t0)       %Tiempo que tarda

function calc_est(path1,path2)

%Este programa me sirve para calcular los valores medios y desv tipica
%de cada 20 ensayos repetidos. EL path1 y path2 se pasan como
parametros

clc

for i=1:20
    eval(['load ..\' path1 '' num2str(i) '.mat']);           %Cargo los
    ficheros
    eval(['Pevec(' num2str(i) ')=Pe']);                       %Almaceno la Pe
end

clear N errores n i Pe
Pemed=mean(Pevec)                                             %Calculo la
media
sigma=std(Pevec)                                              %Calculo la
sigma
eval(['save ..\' path2 '.mat'])                               %Salvo los
resultados

```

```

function proyIIS(Pe,mu_sigma,path1,path2)

%IMPROVED IMPORTANCE SAMPLING---> Vario la media del ruido
%Parametros:
%      Pe:probabilidad de error
%      mu_sigma:cociente entre la media y la varianza(lo miro en la
gráfica que quiero representar)
%      path1:donde quiero que se guarden los resultados de los
experimentos(incluyendo
%      hasta el nombre del fichero)
%      path2:donde quiero que se guarde la media y sigma de los 20
experimentos

%En x la señal original. En w el ruido del canal. En y la suma del
%ruido y de la señal. Despues se hace un bucle para comparar con el
%umbral de decision. Este bucle es muy parecido al de Monte Carlo
%pero antes de calcular la prob de error se multiplica por un peso.

%El valor de sigma es 0.1581. El parametro mu es el
%valor de la media de la f*
%Los valores de amp se calculan automaticamente haciendo llamadas
%a otros programas.

format long
t0=clock;
sigma=0.1581;
mu=sigma*mu_sigma;           %Media;la hallo con los parámetros
pasados
No=0.05;
Eb=No*Qinv(Pe);             %Nivel de energia necesario para la Pe
amp=2*sqrt(Eb);             %Depende del valor de Eb: amp=2*sqrt(Eb)
N=1000000;

for n=1:20
    %clc
    x=rand(1,N);             %Genera numeros aleatorios uniformemente
    x=amp*round(x);          %Los convierto en ceros y "unos"
    w=(sigma*randn(1,N))+mu; %f*:Genero ruido gaussiano de media mu
    y=x+w;                   %Sumo señal y ruido

    rec=zeros(1,N);         %Inicializo el vector rec
    aux=find(y>(amp/2));     %aux tiene los indices en los que creo
que hay "1"
    rec(aux)=amp*ones(1,length(aux)); %Pongo dichos indices a "1"
    clear aux

    f=1/(sigma*sqrt(2*pi))*exp(-(w).^2/(2*(sigma)^2));
    %Numerador del peso(es la original:f)

    fstar=1/(sigma*sqrt(2*pi))*exp(-(w-mu).^2/(2*(sigma)^2));
    %Denominador del peso(es la
modificada:f*)

    peso=abs(f./fstar);     %Estoy generando el peso(peso=f/f*)

```

```
clear num den
z=x-rec; %Matriz en la que si !=0, error
clear rec y x
zerr=find(z); %Matriz con los indices en los que hay
errores

errores=sum(peso(zerr)); %"numero" de errores que hay
clear peso w z i zerr f fstar

Pe=errores/N %Probabilidad de error
eval(['save ..\' path1 '' num2str(n) '.mat']);
end

calc_est(path1,path2) %Para calcular Pe media y varianza
Tiempo=etime(clock,t0) %Tiempo que tarda
```

```

function proyWienmc_saving(amp1,alfa,M,path1)

%Este programa calcula la probabilidad de error segun el metodo de
%Monte Carlo para un canal con memoria y ruido.Lo hace para las seis
%probabilidades de error (M y alfa dadas). Llama al programa del
%canal (canal) y al del filtro de wiener en recepción (wiener)
%Parametros:
%       amp1:vector de amplitudes necesarias (halladas probing)
%       alfa:coeficientes del canal: [1 alfa1 alfa2...]
%       M:orden del sistema
%       path1:es donde quiero que guarde los resultados (sin
incluir el fichero)

%clc
format long
tic
N=1000000;                               %Para calcular el filtro de wiener
solo usare las 1000 primeras

for exp=1:6
    Pevec=[];
    disp 'Buscamos obtener: ';
    No=0.05;
    eval(['Perr=1e-' num2str(exp) '']);
    amp=amp1(exp);
    clear Eb No Perr

    eval(['load ..\Pfc\Coeficientes\coef' num2str(M) '.mat w;'])
    b=w;
    a=1;

    for n=1:20

        x=amp*randint(1,N);                %Genera numeros aleatorios
uniformemente                             %y los convierto en ceros y "unos"(la
                                            %amplitud depende de Eb/No)
        xf=canal(x,alfa);                  %Salida del canal con ruido y memoria

        %b=w;                               %Si lo hago fuera ahorro operaciones
        %a=1;
        %keyboard
        y=filter(b,a,xf);                  %Salida del filtro de Wiener
        clear xf

        rec=zeros(1,N);                    %Inicializo el vector rec
        aux=find(y>(amp/2));                %aux tiene los indices en los que
creo que hay "1"
        rec(aux)=amp*ones(1,length(aux));   %Pongo dichos indices a "1"
        clear aux y                         %Comprobar que puedo borrar y

        z=x-rec;                            %Matriz en la que si !=0, error
        clear rec

```

```

        zerr=find(z);           %Matriz con los indices en los que
hay errores
        errores=length(zerr);   %Numero de errores ocurridos
        keyboard
        clear zerr z x         %Comprobar que puedo borrar x

        Pe=errores/N           %Probabilidad de error segun Monte
Carlo
        Pevec=[Pevec Pe];
        clear errores

        eval(['save ' path1 '\mcwien' num2str(exp) '\mcwien'
num2str(exp) '_' num2str(n) '.mat M Pe alfa amp w']);
        clear Pe

        end
        Pevec
        Pemediam=mean(Pevec)
        sigma=std(Pevec)
        clear Pevec

        eval(['save ' path1 '\mcwien' num2str(exp) '\pemed' num2str(exp)
'.mat M Pemediam sigma alfa amp w']);

end
Tiempo=toc                    %Tiempo que tarda

function y=canal(in,alfa)

%Este fichero me implementa un canal con memoria y ruido.Devuelve la
%salida del canal
%Parametros:
%       alfa:es el coeficiente (o vector) de la ecuacion del
canal:
%       H(1/z)=1+alfa(1/z)+alfa2(1/z^2)+...
%       in:entrada del canal

M=length(alfa);               %Es la memoria del canal
N=length(in);
sigma=0.1581;
w=sigma*randn(1,N);          %f:Genero ruido gaussiano con std sigma
x=in;

a=ones(1,M);
a(2:M)=0;                    %Coeficientes de y(n)
b=alfa;                      %Coeficientes de x(n)

y=filter(b,a,x);             %Salida del canal
y=y+w;                       %Sumo señal y ruido(se elimina cuando
simulamos sin ruido)

```

```

function coef(M, alfa)

%Este programa calcula los coeficientes del filtro que posteriormente
%se usan en las simulaciones de MC,CIS y EIS.
%Parametros:
%       M:orden del sistema
%       alfa:coeficientes del canal: [1 alfa1 alfa2...]
%
x=randint(1,5000);           %Genera numeros aleatorios uniformemente
                             %y los convierto en ceros y "unos"(la
                             %amplitud depende de Eb/No)
xf=canal(x, alfa);          %Salida del canal con ruido y memoria

xprima=x(1:2000);          %Partes que sirven para calcular el
filtro

xfprima=xf(1:2000);
w=wiener(xprima,xfprima,M); %Coeficientes del filtro de wiener
(calculados para las 1000 primeras muestras)
clear xprima xfprima x xf

eval(['save ..\Pfc\Coeficientes\coef' num2str(M) '.mat;'])

function w=wiener(dprima,uprima,M)

%Este programa calcula los coeficientes del filtro de wiener de
%recepcion. Hay que pasarle la señal original y la señal recibida.
%Hace los cálculos para N=1000 y devuelve el resultado. En esta
%version el filtrado debe hacerse externamente, porque el filtro
%se calcula para 1000 muestras del total (1000000).
%Parametros:
%       dprima:es la señal original
%       uprima:es la señal recibida
%       alfa: parametros caracteristicos del canal
%       M:es el orden del sistema que quiero

aux=zeros(1,length(uprima)+M-1); %Añado ceros en los extremos
aux(1:length(uprima))=uprima;

aux2=zeros(1,length(dprima)+M-1); %Añado ceros en los extremos
aux2(1:length(dprima))=dprima;

R=zeros(M,M);                 %Inicializo las matrices
p=zeros(M,1);

```

```
for n=1:length(uprima)

    u=aux(n:n+M-1);           %Porcion de entrada que uso en el
    bucle

    temp=xcorr(u,u);          %Autocorrelacion (sirve para
    hallar R)
    temp2=temp.';

    temp3=temp(M:2*M-1);      %Genero la matriz R
    eval(['R' num2str(n) '=toeplitz(temp3);']);
    clear temp3

    d=aux2(n:n+M-1);         %Porcion de señal deseada que se
    usa

    temp3=xcorr(u,d).';      %Correlacion recibida-deseada

    t=ceil(length(temp3)/2);  %Genero el vector columna p
    eval(['p' num2str(n) '=temp3(t:t+M-1,1);']);
    clear temp3

    eval(['R=R+R' num2str(n) '']); %Actualizacion
    eval(['p=p+p' num2str(n) '']);
end

R=R./length(uprima);        %Matrices "medias"
p=p./length(uprima);

w=inv(R)*p;                 %Coeficientes del filtro
```

```

function proyWienCIS(Perr, amp, alfa, M, gammas, path)           %VERSIÓN
DESPLAZANDO EL RUIDO

%Este programa calcula la probabilidad de error segun el metodo CIS
%para una gamma y un canal con memoria y ruido. Llama al programa del
%canal(canalwiencis)
%Parametros:
%       Perr:probabilidad de error deseada
%       amp:amplitud usada en MC para la Perr
%       alfa:coeficientes del canal: [1 alfa1 alfa2...]
%       M:orden del sistema
%       gammas:vector con las gammas a usar en la simulación
%       path:es donde quiero que guarde los resultados (sin
incluir el fichero)

%clc
format long
tic
N=1000000;

for r=1:5

    disp('Buscamos obtener')
    Perr
    eval(['gama=gammas(' num2str(r) ')'])
    M
    Pevec=[];
    No=0.05;
    sigma=0.1581;
    sigmastar=sigma*gama;           %sigma para generar f*

    eval(['load ..\Pfc\Coefficientes\coef' num2str(M) '.mat w;'])
    b=w;
    a=1;

    for n=1:20

        x=amp*randint(1,N);           %Genera numeros aleatorios
uniformemente y los convierto en   %ceros y "unos"(la
                                     %amplitud depende de Eb/No)

        [xf,ruido]=canalwiencis(x,alfa,gama); %Salida del canal con
ruido y memoria y ruido generado   %aleatoriamente

        y=filter(b,a,xf);           %Salida del filtro de Wiener
clear xf

        rec=zeros(1,N);           %Inicializo el vector rec
aux=find(y>(amp/2));           %aux tiene los indices en los que
creo que hay "1"
rec(aux)=amp*ones(1,length(aux)); %Pongo dichos indices a "1"
clear aux

```

```

    f=1/(sigma*sqrt(2*pi))*exp(-(ruido).^2/(2*(sigma)^2));
                                %Numerador del peso(es la original:f)
    fstar=1/(sigmastar*sqrt(2*pi))*exp(-
(ruido).^2/(2*(sigmastar)^2));
                                %Denominador del peso(es la modificada:f*)
    peso=f./fstar;

    for i=1:M-1                    %Este bucle sirve para calcular la
fi,fstari debido a la memoria

        ruidoaux=ones(1,N);
        eval(['ruidoaux(' num2str(i) '+1:N)=ruido(1:N-' num2str(i)
');']);

        f=1/(sigma*sqrt(2*pi))*exp(-(ruidoaux).^2/(2*(sigma)^2));
        %Calculo fi
        fstar=1/(sigmastar*sqrt(2*pi))*exp(-
(ruidoaux).^2/(2*(sigmastar)^2)); %Calculo fstari

        peso=peso.*f./fstar;      %Multiplico por el producto de ls
f anteriores

        clear f fstar ruidoaux

    end
    clear ruido

    z=x-rec;                      %Matriz en la que si !=0, error
    clear rec
    zerr=find(z);                 %Matriz con los indices en los
que hay errores
    errores=sum(peso(zerr));      %"numero" de errores que hay
    clear peso ruido z zerr f fstar
    Pe=errores/N                 %Probabilidad de error segun
Monte Carlo
    Pevec=[Pevec Pe];
    clear errores

    eval(['save ' path '\gamma' num2str(r) '\wiencis' num2str(r) '_'
num2str(n) '.mat M Pe alfa amp w gammas']);
    clear Pe
end

    Pemediam=mean(Pevec)
    sigma=std(Pevec)
    clear Pevec

    eval(['save ' path '\gamma' num2str(r) '\pemed' num2str(r) '.mat M
Pemediam sigma alfa amp w gammas']);
    clear Pe

end
Tiempo=toc                       %Tiempo que tarda

```

```
function [y,w]=canalwiencis(in,alfa,gamma)

%Este fichero implementa un canal con memoria y ruido.Devuelve la
%salida del canal y el vector de ruido generado. Este se usa en el
%fichero proywiencis para hallar el peso.
%Parametros:
%       alfa:es el coeficiente (o vector) de la ecuacion del
canal:
%       H(1/z)=1+alfa(1/z)+alfa2(1/z^2)+...
%       in:entrada del canal
%       gamma:cociente entre sigmas

M=length(alfa);           %Es la memoria del canal
N=length(in);
sigma=0.1581;
sigmastar=sigma*gamma;
w=sigmastar*randn(1,N);   %f*:Genero ruido gaussiano con std
sigmastar
x=in;

a=ones(1,M);
a(2:M)=0;                %Coeficientes de y(n)
b=alfa;                  %Coeficientes de x(n)

y=filter(b,a,x);         %Salida del canal
y=y+w;
```

```

function proyWienEIS(Perr, amp, alfa, M, gammas, path)

%Este programa calcula la probabilidad de error segun el metodo CIS
%para una gamma y un canal con memoria y ruido. Llama al programa del
%canal (canalwienEIS)
%Parametros:
%       Perr:probabilidad de error deseada
%       amp:amplitud usada en MC para la Perr
%       alfa:coeficientes del canal: [1 alfa1 alfa2...]
%       M: orden del sistema
%       gammas:vector con las gammas a usar en la simulación
%       path:es donde quiero que guarde los resultados (sin
incluir el fichero)

%clc

format long
tic
N=1000000;

for r=1:5

    disp('Buscamos obtener')
    Perr
    eval(['gama=gammas(' num2str(r) ')'])

    M
    Pevec=[];
    No=0.05;
    sigma=0.1581;

    sigmastar=sigma*gama;           %sigma para generar f*

    eval(['load ..\Pfc\Coeficientes\coef' num2str(M) '.mat w;'])
    b=w;
    a=1;

    sumatorio=sum(w.^2)             %Calcula el sumatorio
    sigmaz=sigma*sqrt(sumatorio);
    sigmastarz=sigmastar*sqrt(sumatorio); %Es la desv a usar
    clear sumatorio sigma %sigmastar

    for n=1:20

        x=amp*randint(1,N);         %Genera numeros aleatorios
uniformemente y los convierto en ceros y "unos"(la
                                     %amplitud depende de Eb/No)

        [xf,ruido]=canalwienEIS(x, alfa, sigmastar); %Salida del canal
con ruido y memoria y ruido generado aleatoriamente

        y=filter(b,a,xf);           %Salida del filtro de Wiener
clear xf

```

```

ruido=filter(b,a,ruido); %Ruido a la salida del filtro
clear ruido

rec=zeros(1,N); %Inicializo el vector rec
aux=find(y>(amp/2)); %aux tiene los indices en los que
creo que hay "1"
rec(aux)=amp*ones(1,length(aux)); %Pongo dichos indices a "1"
clear aux

f=1/(sigmaz*sqrt(2*pi))*exp(-(ruido).^2/(2*(sigmaz)^2));
%Numerador del peso(es la original:f)
fstar=1/(sigmastarz*sqrt(2*pi))*exp(-
(ruido).^2/(2*(sigmastarz)^2));
%Denominador del peso(es la modificada:f*)

clear ruidoz
peso=f./fstar;

z=x-rec; %Matriz en la que si !=0, error
clear rec
zerr=find(z); %Matriz con los indices en los
que hay errores
errores=sum(peso(zerr)); %"numero" de errores que
hay
clear peso ruido z zerr f fstar
Pe=errores/N %Probabilidad de error segun
Monte Carlo
Pevec=[Pevec Pe];
clear errores

eval(['save ' path '\gamma' num2str(r) '\wieneis' num2str(r) '_'
num2str(n) '.mat M Pe alfa amp w gammas']);
clear Pe
end

Pemediam=mean(Pevec)
sigma=std(Pevec)
clear Pevec

eval(['save ' path '\gamma' num2str(r) '\pemed' num2str(r) '.mat M
Pemediam sigma alfa amp w gammas']);
clear Pe

end
Tiempo=toc %Tiempo que tarda

```

```
function [y,w]=canalwienEis(in,alfa,sigmastar)
```

```

%Este fichero me implementa un canal con memoria y ruido.Devuelve la
%salida del canal y el vector de ruido generado. Este se usa en el
%fichero proywieneis para hallar el peso.
%Parametros:

```

```
%      alfa:es el coeficiente (o vector) de la ecuacion del
canal:
%      H(1/z)=1+alfa(1/z)+alfa2(1/z^2)+...
%      in:entrada del canal
%      sigmastar:sigma del ruido

M=length(alfa);      %Es la memoria del canal
N=length(in);

w=sigmastar*randn(1,N); %f*:Genero ruido gaussiano con std
sigmastar
x=in;

a=ones(1,M);
a(2:M)=0;      %Coeficientes de y(n)
b=alfa;      %Coeficientes de x(n)

y=filter(b,a,x);      %Salida del canal
y=y+w;
```

```

function proyTAIL(amp, alfa, M, umbrales, path, umb)

%Este programa calcula la probabilidad de error segun el metodo de
%la extrapolacion de colas para un canal con memoria y ruido.
%Para cada probabilidad de error tiene que llamar 3 veces al
%metodo de Monte Carlo
%Parametros:
%      amp:vector de amplitudes necesarias (halladas probing)
%      alfa:coeficientes del canal: [1 alfa1 alfa2...]
%      M:orden del sistema
%      umbrales:distancias de los umbrales a la Pe buscada
%      path:donde quiero que guarde los resultados
%      umb:necesario para automatizar (nombre de un directorio)

u=amp/2;           %Definimos el umbral de decision
udb=10*log10(u)    %Umbral de decision en decibelios
l=length(umbrales);

for i=1:l          %Definimos los pseudoumbrales

    eval(['aux' num2str(i) '=udb-umbrales(' num2str(i) ');'])
    eval(['t' num2str(i) '=exp((aux' num2str(i) '-1)/10);'])
    eval(['clear aux' num2str(i) ' '])

end

p=[];

for n=1:20
    n
    x=[];
    y=[];

    for i=1:l

        eval(['p' num2str(i) '=proywienMcTAIL(t' num2str(i) '*2, alfa, M,
        ' 'path' ' ');'])

        eval(['y(' num2str(i) ')=log(-log(p' num2str(i) '));'])

        eval(['clear p' num2str(i) ' '])

        eval(['x(' num2str(i) ')=log(t' num2str(i) ');']) %Obtenemos
los terminos que intervienen

    end

    x0=log(u);

    C=sum(x.*y);
    D=sum(x);
    E=sum(y);
    F=sum(x.^2);           %Para obtener la recta de regresion y =
a + bx

```

```

a=((F*E)-(D*C))/((1*F)-(D^2));
b=((1*C)-(D*E))/((1*F)-(D^2));

y0=exp(-exp(a+(b*x0)));
p=[p y0];
clear aux x y y0 C D E F a b

end

eval(['clear t' num2str(i) ' '])

pmedia=mean(p)
sigma=std(p)

eval(['save ' path '\Pe' num2str(M) '_' num2str(umb) '.mat M pmedia p
sigma amp alfa umbrales']);

function Pe=proyWienWcTAIL(amp, alfa, M, path1)

%Este programa calcula la probabilidad de error segun el metodo de
%Monte Carlo para un canal con memoria y ruido.Lo hace para las seis
%probabilidades de error (M y alfa dadas). Llama al programa del
%canal (canal) y al del filtro de wiener en recepción (wiener)
%Parametros:
%       amp:amplitudes necesarias (halladas probing)
%       alfa:coeficientes del canal: [1 alfa1 alfa2...]
%       M:orden del sistema
%       path1:es donde quiero que guarde los resultados (sin
incluir el fichero)

%clc
format long
N=1000000;                               %Para calcular el filtro de wiener
solo usare las 1000 primeras

eval(['load ..\Pfc\Coeficientes\coef' num2str(M) '.mat w;'])
b=w;
a=1;

x=amp*randint(1,N);                       %Genera numeros aleatorios uniformemente
                                           %y los convierto en ceros y "unos"(la
                                           %amplitud depende de Eb/No)
xf=canal(x, alfa);                         %Salida del canal con ruido y memoria

y=filter(b, a, xf);                       %Salida del filtro de Wiener
clear xf

rec=zeros(1,N);                           %Inicializo el vector rec
aux=find(y>(amp/2));                       %aux tiene los indices en los que creo que
hay "1"

```

```
rec(aux)=amp*ones(1,length(aux)); %Pongo dichos indices a "1"
clear aux y %Comprobar que puedo borrar y

z=x-rec; %Matriz en la que si !=0, error
clear rec
zerr=find(z); %Matriz con los indices en los que hay
errores
errores=length(zerr); %Numero de errores ocurridos
clear zerr z x %Comprobar que puedo borrar x

Pe=errores/N; %Probabilidad de error segun Monte Carlo
clear errores

eval(['save ' path1 '\mcwien' num2str(exp) '\pemed' num2str(exp) '.mat
M Pemed sigma alfa amp w']);
```

REFERENCIAS

1. John G. Proakis, *Digital Communications*, Mc Graw Hill (1995).
2. Michel C. Jeruchim, Philip Balaban and K. Sam Shanmugan, *Simulation of Communications Systems*, Plenum (1992).
3. Simon Haykin, *Adaptive Filter Theory*, Pentice Hall (1996).
4. Jialin Zou, V. K. Bhargava, *On the Optimum Biasing of Importance Sampling for Simulation of Communications Systems*, IEEE
5. Michel C. Jeruchim, Peter M. Hahn, Kevyn P. Smyntek, *An Experimental Investigation of Conventional and Efficient Importance Sampling*, IEEE Transactions on Communications
6. Michel C. Jeruchim, Peter M. Hahn, *Developments in the Theory and Application of Importance Sampling*, IEEE Transactions on Communications.
7. Philip Balaban, K. Sam Shanmugan, *A Modified Monte Carlo Technique for the Evaluation of Error Rate in Digital Communications Systems*, IEEE Transactions on Communications.