

# Ecualización de canal

Francesc Rey Micolau  
Javier Villares Pera

PID\_00197110



Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia de Reconocimiento-NoComercial-SinObraDerivada (BY-NC-ND) v.3.0 España de Creative Commons. Podéis copiarlos, distribuirlos y transmitirlos públicamente siempre que citéis el autor y la fuente (FUOC. Fundació para la Universitat Oberta de Catalunya), no hagáis de ellos un uso comercial y ni obra derivada. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.es>

# Índice

|  |    |
|--|----|
| <b>Introducción</b> .....  | 5  |
| <b>Objetivos</b> .....   | 7  |
| <b>1. Ecuación y tipo de ecualizadores</b> .....   | 9  |
| <b>2. Ecualizadores lineales</b> .....   | 14 |
| 2.1. Forzador de ceros (ZF) .....  | 15 |
| 2.1.1. Estudio en el dominio del tiempo .....  | 15 |
| 2.1.2. Estudio en el dominio de la frecuencia .....  | 17 |
| 2.1.3. Inconvenientes del forzador de ceros .....  | 19 |
| 2.2. Ecualizador de mínimo error cuadrático medio (MMSE) .....                                 | 22 |
| 2.2.1. Caso 1: Respuesta impulsional global $c[n]$ conocida .....                              | 26 |
| 2.2.2. Caso 2: Respuesta impulsional global $c[n]$<br>desconocida .....                        | 27 |
| 2.3. Ecualizador adaptativo .....  | 28 |
| 2.3.1. Ecualizador LMS: algoritmo de gradiente .....   | 29 |
| 2.3.2. Secuencia de entrenamiento para controlar la<br>actualización de los coeficientes ..... | 33 |
| <b>3. Ecualizadores no lineales</b> .....  | 39 |
| 3.1. Estimador de máxima verosimilitud (MLSE) .....  | 39 |
| 3.2. Estimador MLSE implementado con el algoritmo de Viterbi .....                             | 40 |
| 3.2.1. Inconvenientes del algoritmo de Viterbi .....   | 49 |
| 3.3. Ecualizador no lineal. Ecualizador con realimentación de<br>decisiones (DFE) .....        | 51 |
| <b>Resumen</b> .....   | 53 |
| <b>Actividades</b> .....   | 55 |
| <b>Bibliografía</b> .....  | 57 |



## Introducción

Ya sabemos que por lo general el canal de comunicaciones no es un canal ideal. Tal como se ha visto, en general este puede tener un comportamiento selectivo en frecuencia y también puede variar en el tiempo. Cuando el canal es dispersivo en tiempo aparece el efecto de interferencia intersimbólica (ISI) que, si el canal varía de manera lo bastante lenta, puede ser mitigado mediante un ecualizador. El funcionamiento se basa en la estimación del efecto dispersivo del canal y el correspondiente diseño de un sistema (lineal o no lineal) que procese las muestras a la salida del filtro adaptado y reduzca el efecto de la ISI.

En este módulo, nos vamos a centrar específicamente en el bloque ecualizador. El primer apartado, aunque pueda parecer redundante con algunas ideas ya presentadas, volverá a poner el énfasis en el problema de la ISI para motivar la necesidad del bloque ecualizador. Este apartado, a modo de introducción al módulo, finalizará presentando una clasificación de los ecualizadores, que se pueden agrupar en dos grandes bloques: ecualizadores lineales y ecualizadores no lineales.

El segundo apartado presentará los ecualizadores lineales. El apartado presenta este tipo de ecualizadores de manera natural, desde el más sencillo conceptualmente hablando, hasta el más complejo. Para cada uno de los ecualizadores se presentará su objetivo, función de coste que se optimiza, la manera de diseñar los coeficientes, y finalmente una descripción de los principales problemas de aquel tipo de ecualizador, que motivará el paso al ecualizador siguiente. Así, se empezará por el ecualizador forzador de ceros para pasar a continuación a presentar el ecualizador de mínimo error cuadrático medio, y por último, los ecualizadores adaptativos (basados en el criterio de mínimo error cuadrático medio).

Finalmente, el tercer apartado presentará el ecualizador no lineal basado en el estimador de máxima verosimilitud. Si bien en algunos textos este ecualizador se presenta el primero por tratarse del ecualizador óptimo, y a partir de este, por razones de complejidad, se motiva la necesidad de recurrir a los ecualizadores lineales, nosotros hemos preferido presentar primero los ecualizadores lineales por tratarse de los más sencillos (y a menudo los más empleados) y dejar para el final la descripción de cómo tendría que ser el ecualizador óptimo.

Como ya podréis imaginar, los ecualizadores presentados en este módulo, si bien incluyen los esquemas más utilizados, no completan toda la gama de ecualizadores y técnicas de ecualización existentes. Sin embargo, el objetivo

### Ved también

Ved el módulo "Canal de comunicaciones" de esta asignatura.

es que este módulo proporcione una idea del problema de la ecualización y la manera habitual de abordarlo. Al final del módulo, vais a encontrar unas referencias bibliográficas por si queréis profundizar más en el tema.

Para terminar esta introducción, hacemos la misma reseña que ya hicimos en el módulo anterior respecto al hecho de trabajar en todo momento con el equivalente paso bajo cuando hablamos de la señal y de la respuesta impulsional del canal. Así, en ningún momento a lo largo del módulo haremos referencia a la banda de frecuencias o a la frecuencia concreta a la que se lleva a cabo la comunicación. La teoría presentada para el ecualizador es válida para cualquier frecuencia de trabajo.

En el módulo, cuando hablemos de *canal*, siempre nos vamos a referir en general a lo que se conoce como *equivalente paso bajo del canal*, que es el canal complejo que verá el *equivalente paso bajo de la señal*. En este aspecto, destacamos que, aunque en los ejemplos y la derivación de las ecuaciones consideramos en todo momento el canal real, este será casi siempre complejo en la práctica (porque se trata del equivalente paso bajo). El hecho de utilizar la notación real en primera instancia es para facilitar la comprensión y no complicar las ecuaciones más de lo necesario. Solo en los casos de comunicaciones en banda base tendremos un canal que no estará desplazado en banda, y por lo tanto, donde el canal será real tal como se presenta en el módulo (algunos ejemplos de canales en banda base son el canal de voz o el canal ADSL –ambos a través del bucle de abonado de la línea telefónica– o los canales con problemas de acústica).

#### Recordad

El equivalente paso bajo de la señal se presenta en el módulo didáctico “Comunicaciones analógicas: señales paso banda” de la asignatura *Sistemas de comunicaciones I*. Recordemos que el equivalente paso bajo es una definición matemática que pretende compactar la notación de la descomposición de una modulación paso banda en sus componentes en fase y cuadratura eliminando toda referencia a la frecuencia de la portadora. Si  $s(t)$  es una señal paso banda centrada en la frecuencia portadora  $f_0$ , esta se puede escribir como  $s(t) = \text{Re}\{b_s(t) \cdot \exp(j2\pi f_0 t)\}$ , donde  $b_s(t)$  es el equivalente paso bajo de la señal.

## Objetivos

Con el estudio de este módulo didáctico, alcanzaréis los objetivos siguientes:

1. Entender el efecto que un canal no ideal tiene sobre la señal (revisar el concepto de ISI) para poder asimilar la función del ecualizador.
2. Conocer los diferentes tipos de ecualizadores y su clasificación en lineales y no lineales.
3. Entender la problemática del ecualizador forzador de ceros y la alternativa del ecualizador de mínimo error cuadrático medio.
4. Ser conscientes de la necesidad de los ecualizadores adaptativos, en especial cuando el canal varía en el tiempo.
5. Conocer los dos modos de diseño de ecualizadores: los asistidos por datos (DA) y los asistidos por decisiones (DD).
6. Ser capaces de diseñar un ecualizador para problemas sencillos.
7. Poder implementar el ecualizador óptimo en términos de minimización de la probabilidad de error utilizando el algoritmo de Viterbi.



## 1. Ecuación y tipo de ecualizadores

Ya se ha puesto énfasis en el hecho de que el canal de comunicaciones no es ideal (canal AWGN) cuando la señal recibida no se corresponde con una réplica atenuada y retardada de la señal transmitida contaminada por un ruido aditivo.

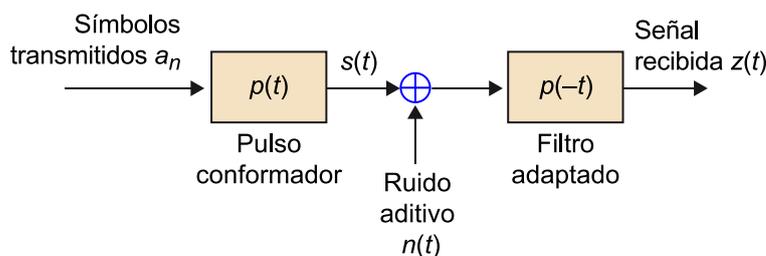
El efecto que esto tiene sobre la señal digital ya se ha presentado en otros materiales. Sobre la señal recibida aparece el fenómeno de interferencia intersimbólica (*ISI*, del inglés: *inter-symbol interference*) que se produce cuando, al muestrear la señal a la salida del filtro adaptado a una muestra por símbolo, la muestra obtenida, además del símbolo que se tiene que detectar, y de la componente de ruido, presenta una contribución superpuesta de símbolos precedentes y posteriores.

A continuación, revisamos la forma que toma la señal en el receptor y la funcionalidad de los diferentes bloques e introducimos la notación que seguiremos en este módulo. Haremos el estudio de la forma que toma la señal en cada punto en el dominio del tiempo y analizaremos su comportamiento cuando el transmisor envía solo un símbolo (la ampliación al problema real en el que el transmisor envía una secuencia de símbolos es inmediata, simplemente aplicando la superposición de todos los símbolos enviados, dado que el conjunto de pulso conformador más canal más filtro adaptado se comporta como un sistema lineal e invariante). Así pues, el resultado de enviar un solo símbolo (y por lo tanto, un solo pulso) corresponderá a la respuesta impulsional global, que denominaremos  $c(t)$  y nos será útil para el diseño de algunos de los ecualizadores que vamos a ver en los apartados siguientes.

Empecemos revisando el problema en presencia de un canal ideal.

Figura 1. Esquema de un sistema de comunicaciones digitales en presencia de un canal ideal (AWGN)

### Canal ideal



Si en el instante  $t = 0$  enviamos el símbolo  $a_0 = +1$ , siguiendo el esquema de la figura 1, la señal analógica que se enviará será  $s(t) = p(t)$ , donde  $p(t)$  será en general el pulso raíz de coseno realzado. En ausencia de canal que distorsione

### Ved también

Ved los canales AWGN en la asignatura *Sistemas de comunicaciones I* y también el módulo "Canal de comunicaciones" de esta asignatura.

### Notación

A lo largo de todo el módulo, vamos a distinguir entre la notación genérica  $z(t)$ , que se referirá a la señal a la salida del filtro adaptado para cualquier señal transmitida  $s(t)$ , y la notación particular  $c(t)$ , que corresponderá a la señal a la salida del filtro adaptado cuando se envía un solo pulso conformador  $s(t) = p(t)$  (o, de manera equivalente, un solo símbolo  $a_0 = +1$ ). En este segundo caso, la señal  $c(t)$  se podrá interpretar como la respuesta impulsional global del sistema equivalente que (sin considerar el término de ruido) agrupa pulso conformador, canal y filtro adaptado. Desde un punto de visto de ISI, este canal equivalente  $c(t)$  es el que nos interesará que idealmente genere una señal libre de ISI.

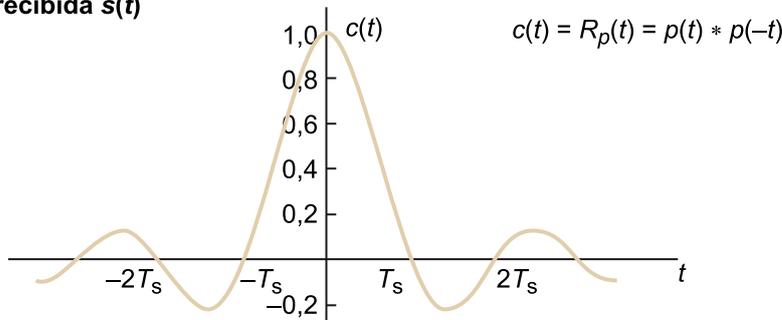
la señal, y sin considerar el término de ruido aditivo, la señal que se obtendrá a la salida del filtro adaptado al enviar un solo símbolo será la convolución del pulso transmitido con la respuesta impulsional del filtro adaptado, y corresponderá a lo que hemos denominado *respuesta impulsional global*,  $c(t)$ . Así pues:

$$c(t) = R_p(t) = p(t) * p(-t)$$

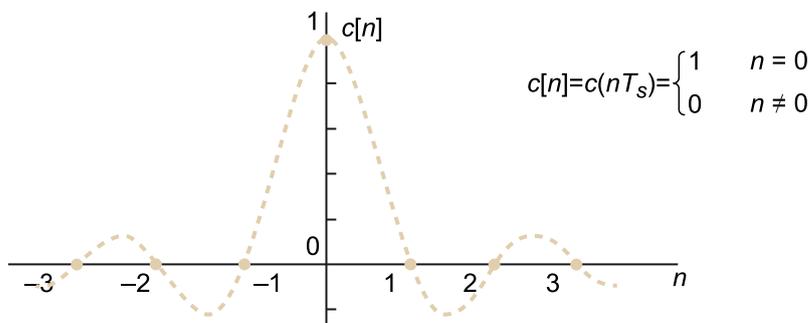
Tal como ya se ha indicado, si el pulso conformador y el filtro adaptado se han diseñado bien, la señal  $c(t)$  corresponde a un pulso de Nyquist que no genera ISI. Por lo tanto, al intentar recuperar los símbolos transmitidos, la señal discreta muestreada a tiempo de símbolo,  $c[n] = c(nT_s)$ , será la que se indica en la figura 2, que dará lugar a una señal libre de ISI, y por lo tanto no requiere de ningún tipo de ecualización.

Figura 2. Respuesta impulsional global a la salida del filtro adaptado antes y después de muestrear a una muestra por símbolo por un canal ideal

### Señal recibida $s(t)$



### Señal muestreada $c[n]$

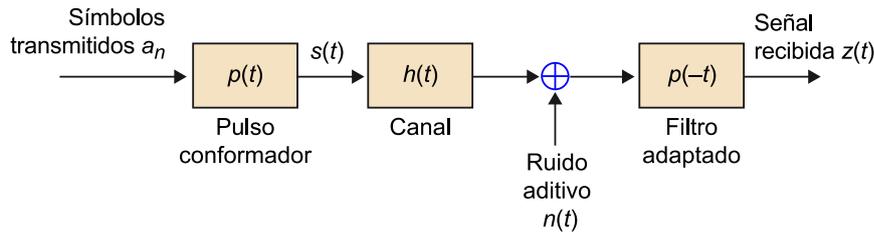


Ahora bien, en presencia de un canal que tenga por respuesta impulsional  $h(t)$ , tal como muestra la figura 3, la señal que se obtendrá a la salida del filtro adaptado será la convolución del pulso transmitido con la respuesta impulsional del canal y con la del filtro adaptado. Cuando se envía un solo símbolo, esta señal, que corresponde a la respuesta impulsional global  $c(t)$ , será:

$$c(t) = p(t) * h(t) * p(-t)$$

Figura 3. Esquema de un sistema de comunicaciones digitales en presencia de canal no ideal

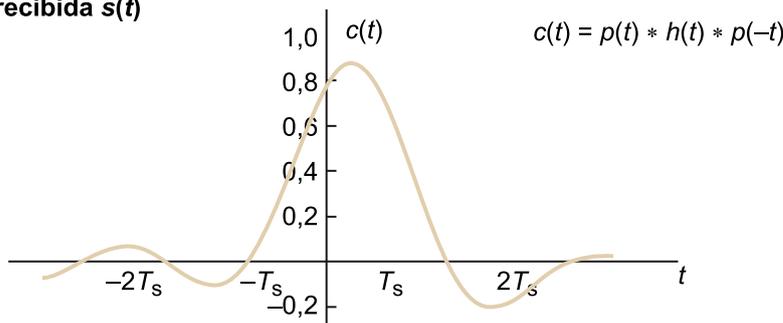
**Canal no ideal**



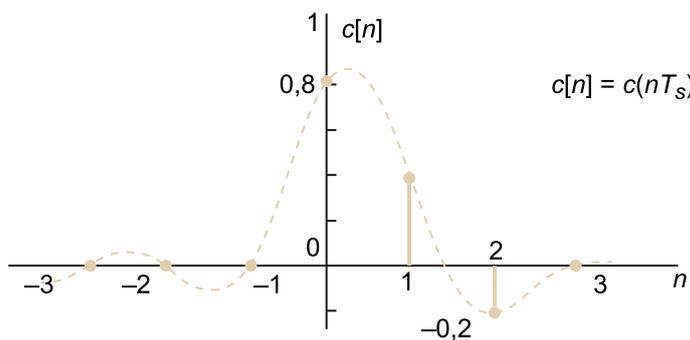
En este caso, si asumimos de nuevo que el pulso conformador y el filtro adaptado se han diseñado bien para que su convolución sea un pulso de Nyquist, la señal discreta muestreada a tiempo de símbolo,  $c[n]$ , será la que se indica en la figura 4. Tal como se ve con claridad, las muestras  $c[n]$  para  $n \neq 0$  no son nulas, y por lo tanto la señal no estará libre de ISI (cuando muestreemos la señal en el instante  $n = 1, 2, 3, \dots$  para detectar el símbolo  $a_1, a_2, a_3, \dots$  tendremos interferencia del símbolo  $a_0$ ).

Figura 4. Respuesta impulsional global a la salida del filtro adaptado antes y después de muestrear a una muestra por símbolo por un canal no ideal

**Señal recibida  $s(t)$**



**Señal muestreada  $c[n]$**

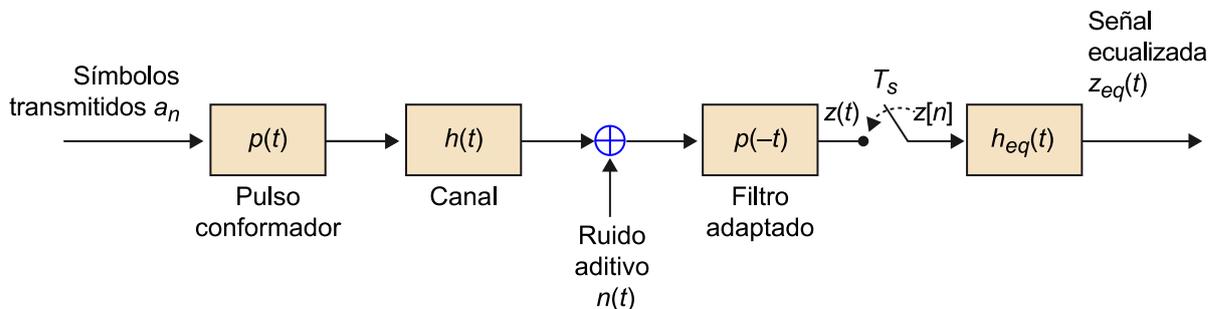


De este modo, el objetivo del bloque ecualizador será intentar obtener a la salida una señal libre de ISI (solución ideal) o intentar mitigar su efecto tanto como sea posible (solución práctica). La figura 5 muestra el esquema completo de un sistema de comunicaciones digitales en presencia de canal no ideal y ecualizador. Con referencia a este esquema, cabe mencionar que se consideran por separado el filtro adaptado y el filtro ecualizador. Algunos autores consideran ambos bloques juntos, y entonces el filtro conjunto cumple dos funciones: la de filtro adaptado, que filtra el ruido buscando maximizar la relación

señal a ruido –SNR: del inglés: *signal noise ratio*– y la de ecualizador, que mitiga el efecto del canal no ideal. Nada varía en este segundo planteamiento, si bien desde un punto de vista de diseño la complejidad aumenta innecesariamente, ya que es necesario incorporar la parte de filtro adaptado a la fase de diseño del ecualizador, que es determinista.

Figura 5. Esquema de un sistema de comunicaciones digitales en presencia de canal no ideal, incluido el bloque ecualizador

### Canal no ideal + Ecualizador



Para poder diseñar el ecualizador, habrá que tener un conocimiento de la respuesta impulsional del canal o bien tener conocimiento de los símbolos transmitidos y compararlos con la señal recibida con objeto de poder inferir qué ha hecho el canal. En el caso de requerir el conocimiento de la respuesta impulsional, en este módulo vamos a asumir que esta es conocida, sin entrar en detalles de cómo se ha obtenido (a partir de enviar una secuencia de entrenamiento y aplicar un algoritmo de estimación de canal).

Cuando haya que conocer los símbolos transmitidos, se puede trabajar en dos modos: un modo asistido por datos (DA), en el que los símbolos son conocidos a partir del envío de una secuencia de entrenamiento, y un segundo modo asistido por decisiones (DD), en el que los símbolos son conocidos una vez se ha decidido sobre ellos a la salida del ecualizador. El primero implica una pérdida de eficiencia al enviar datos que no son de información, mientras que el segundo requiere una mínima calidad en las decisiones, ya que estas deben ser lo bastante fiables como para considerarlas en el diseño mismo del ecualizador. Todavía hay una tercera manera de proceder, en entornos muy ruidosos, en que se trabaja con las probabilidades del símbolo en lugar de con sus valores decididos. Observad que estos modos DA y DD son conceptualmente iguales a los descritos en el apartado 2 del módulo de sincronización.

#### Ved también

Ved la tercera manera de proceder en entornos muy ruidosos en el subapartado 2.3 de este módulo didáctico.

El objetivo de este módulo será trabajar el bloque ecualizador y describir sus diferentes opciones de diseño. Una primera clasificación de los ecualizadores los divide en dos grandes bloques, los **ecualizadores lineales**, que se implementan siguiendo un esquema lineal, y los **ecualizadores no lineales**, que se implementan siguiendo un esquema no lineal. Podemos decir que los primeros obedecen a la estructura de un sistema lineal e invariado, y por lo tanto su descripción se podrá hacer a partir de la respuesta impulsional o la respuesta en frecuencia, tal como ya habéis visto en asignaturas de señales y sistemas.

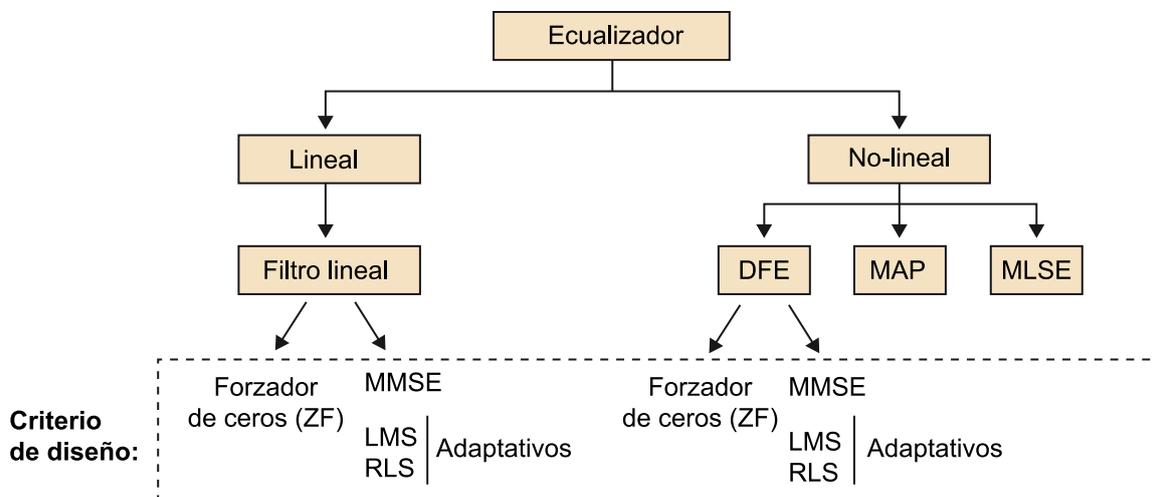
Por el otro lado, los ecualizadores no lineales, precisamente por su naturaleza no lineal, no se podrán caracterizar a partir de la respuesta impulsional y habrá que recurrir a otras formas de caracterizarlos y describirlos.

Como ya podéis intuir, los ecualizadores lineales serán mucho más sencillos a la hora de diseñarlos e implementarlos. A pesar de este hecho, en general, las prestaciones que ofrecerán serán peores que las que se pueden obtener con un ecualizador no lineal. Concretando más, podemos afirmar que el ecualizador óptimo en términos de probabilidad de error en los símbolos recibidos será el ecualizador basado en la estimación de máxima verosimilitud (MLSE), que corresponde a un ecualizador no lineal.

Dejamos la descripción de este ecualizador para la última parte del módulo. La estructura del módulo se ha planteado de modo que se presentan los ecualizadores más habituales, desde el más sencillo desde un punto de vista conceptual, hasta el más complejo. Sin embargo, este orden no se corresponde con el de las prestaciones, que podríamos decir que es el inverso: el primer ecualizador presentado es el que tiene peores prestaciones y el último corresponde al que ofrece las mejores prestaciones.

La figura 6 resume una clasificación de los ecualizadores.

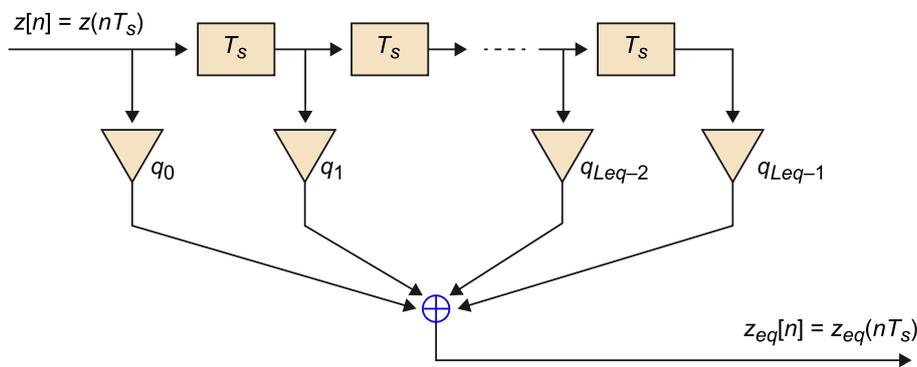
Figura 6. Clasificación de los ecualizadores



## 2. Ecuadores lineales

La estructura de los ecualizadores lineales siempre es la misma: la señal a la salida (señal ecualizada) se obtiene al hacer pasar la señal de entrada por un filtro con una línea de retardos, de modo que la suma de todas las muestras de la memoria de los filtros ponderadas por unos coeficientes genera la señal ecualizada. La figura 7 ilustra el esquema de cualquier ecualizador lineal discreto, donde  $q_0, q_1, \dots, q_{L_{eq}-1}$  son los coeficientes del ecualizador, y  $L_{eq}$  su longitud.

Figura 7. Esquema de un ecualizador lineal



La expresión matemática de la señal a la salida del ecualizador en función de la señal a la entrada (trabajando con señales analógicas) es la siguiente:

$$z_{eq}(t) = \sum_{k=0}^{L_{eq}-1} q_k z(t - kT_s) \quad (1)$$

De manera equivalente si, tal como se muestra en la figura 5, consideramos el ecualizador digital que trabaja con la señal a la entrada ya muestreada con un periodo de muestreo igual al tiempo de símbolo  $T_s$ , es decir,  $z[n] = z(nT_s)$ , la expresión discreta de las muestras a la salida  $z_{eq}[n]$  será la siguiente:

$$z_{eq}[n] = \sum_{k=0}^{L_{eq}-1} q_k z[n - k] \quad (2)$$

Tal como puede verse en la ecuación (2), los parámetros que definen cualquier ecualizador lineal son: el número de coeficientes del ecualizador, que hemos denotado con  $L_{eq}$ , y el valor de los coeficientes  $q_0, q_1, \dots, q_{L_{eq}-1}$ . La manera como se obtiene el valor de los coeficientes es lo que va a determinar la diferencia entre los diversos ecualizadores lineales que veremos en los subapartados siguientes. Asimismo, el valor de estos coeficientes puede ser fijo a lo largo del

### Ved también

Ved los ecualizadores adaptativos en el subapartado 2.3 de este módulo didáctico.

tiempo, o irse ajustando para adaptarse a las variaciones que la respuesta del canal puede experimentar a lo largo de la transmisión, como se verá cuando diseñemos los ecualizadores adaptativos.

### Reflexión

En el ecualizador de la figura 7, el retardo entre muestras (en la línea de retardos) coincide con el tiempo de símbolo  $T_s$ . Esto indica que el ecualizador trabaja a una muestra por símbolo. Si bien este esquema es habitual, se pueden diseñar ecualizadores que trabajen a más de una muestra por símbolo, es decir, que el valor de los retardos en la línea de retardos sea inferior a  $T_s$ . En este caso, se habla de **ecualizador fraccional**.

Conceptualmente, las ideas que vamos a exponer a continuación no varían, si bien se tienen que modificar las ecuaciones para calcular los valores de los coeficientes. Asimismo, si por ejemplo consideramos que el ecualizador trabaja a  $m$  muestras por símbolo (valor de los retardos  $T_s/m$ ), como al final solo queremos conocer el valor de los símbolos transmitidos, a la salida del ecualizador solo habrá que mirar la señal cada  $m$  muestras si queremos tener los valores de las muestras correspondientes a los diferentes símbolos transmitidos.

## 2.1. Forzador de ceros (ZF)

En un ecualizador forzador de ceros, comúnmente abreviado con el acrónimo ZF (del inglés: *zero-forcing*), el objetivo es que los coeficientes se diseñen de modo que se fuerce la condición de ISI nula a la salida del ecualizador.

Para analizar y diseñar este ecualizador, vamos a estudiar qué ocurre cuando el transmisor envía solo un símbolo, y por lo tanto tal como ya se ha hecho en la figura 4, vamos a trabajar con la respuesta impulsional global,  $c(t)$ . Por otro lado, como el forzador de ceros centra la atención en el término de ISI, tampoco vamos a considerar en el proceso de diseño el término de ruido (asumimos por un momento que no hay ruido AWGN en la comunicación).

### Ved también

Ved la figura 4 en el apartado 1 de este módulo didáctico.

### 2.1.1. Estudio en el dominio del tiempo

Como ya se ha visto, en presencia de canal no ideal, las muestras de la respuesta impulsional global  $c[n]$  para  $n \neq 0$  no son nulas (ved el ejemplo de la figura 4), y por lo tanto la señal no está libre de ISI (cuando muestreemos la señal en el instante  $n = 1, 2, 3, \dots$  para detectar el símbolo  $a_1, a_2, a_3, \dots$  tendremos una interferencia del símbolo  $a_0$ ). Idealmente, el objetivo del forzador de ceros será diseñar un ecualizador que elimine esta interferencia y garantice que la señal a la salida esté libre de ISI o, lo que es lo mismo, que cumpla la condición (3):

$$z_{eq}[n] = z_{eq}(nT_s) = \begin{cases} 1 & n=0 \\ 0 & n \neq 0 \end{cases} \quad (3)$$

En la práctica, esto no será posible, y por lo tanto se intentará diseñar un ecualizador que reduzca –sin llegar a eliminar– la ISI tanto como sea posible. El objetivo será determinar qué valor deben tomar los coeficientes del ecualizador ( $q_0, q_1, \dots, q_{L_{eq}-1}$ ) para forzar que cuando a la entrada se tiene la respuesta

impulsional global  $c(t)$ , la señal a la salida valga cero en los instantes  $n = 1, 2, 3, \dots, L_{eq} - 1$ , y aproximar así la señal  $z_{eq}[n]$  a una señal libre de ISI que intente aproximarse a la condición de la ecuación (3).

Diseñar los valores de los coeficientes requiere resolver un sistema de ecuaciones. A partir de la ecuación de la señal a la salida del ecualizador lineal (2), tenemos un problema de  $L_{eq}$  incógnitas (el valor de los  $L_{eq}$  coeficientes del ecualizador), y por lo tanto necesitaremos  $L_{eq}$  ecuaciones que generaremos dando  $L_{eq}$  valores al  $n \in [0, 1, \dots, L_{eq} - 1]$ . Para el valor  $n = 0$ , se tendrá que cumplir  $z_{eq}[0] = 1$ , y para el resto de valores de  $n$ , se tendrá que forzar  $z_{eq}[n] = 0$ .

Este problema de  $L_{eq}$  ecuaciones con  $L_{eq}$  incógnitas se puede escribir matricialmente de la manera siguiente:

$$\begin{pmatrix} d[0] & d[-1] & \dots & d[-(L_{eq}-1)] \\ d[1] & d[0] & \dots & d[-(L_{eq}-2)] \\ \vdots & \vdots & \ddots & \vdots \\ d[L_{eq}-1] & d[L_{eq}-2] & \dots & d[0] \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ q_{L_{eq}-1} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (4)$$

Observad que la manera de diseñar el ecualizador es muy sencilla. Conocida la respuesta impulsional global,  $c[n]$ , se construye la matriz  $\mathbf{Z}$  como (5), y los  $L_{eq}$  coeficientes del ecualizador se obtienen después de invertir esta matriz y multiplicarla por el vector todo ceros excepto el primer elemento a 1 (6):

$$\mathbf{Z} = \begin{pmatrix} d[0] & d[-1] & \dots & d[-(L_{eq}-1)] \\ d[1] & d[0] & \dots & d[-(L_{eq}-2)] \\ \vdots & \vdots & \ddots & \vdots \\ d[L_{eq}-1] & d[L_{eq}-2] & \dots & d[0] \end{pmatrix} \quad (5)$$

$$\begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ q_{L_{eq}-1} \end{pmatrix} = \mathbf{Z}^{-1} \cdot \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (6)$$

**Ejemplo**

¿Cuál es el forzador de ceros de ocho coeficientes para el canal utilizado en la figura 4, donde la respuesta impulsional global cumple la condición siguiente?

$$\begin{aligned} d[0] &= +0,8 \\ d[1] &= +0,4 \\ d[2] &= -0,2 \\ d[n] &= 0 \quad n \neq 0, 1, 2 \end{aligned}$$

A partir de esta información construimos la matriz  $\mathbf{Z}$ :

**Terminología**

Como ya habréis deducido, la denominación forzador de ceros proviene del hecho que estamos forzando a la salida del ecualizador una señal con ceros en determinados instantes de tiempo.

**Reflexión**

Podemos relajar la condición forzada en la ecuación (4) para permitir un cierto retardo de la señal a la salida del ecualizador. En ese caso, la posición del 1 en el vector de la ecuación (4), en lugar de estar en la primera posición, podría estar en cualquier otro lugar.

$$\mathbf{Z} = \begin{pmatrix} +0,8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ +0,4 & +0,8 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0,2 & +0,4 & +0,8 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0,2 & +0,4 & +0,8 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0,2 & +0,4 & +0,8 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0,2 & +0,4 & +0,8 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0,2 & +0,4 & +0,8 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0,2 & +0,4 & +0,8 \end{pmatrix}$$

Y diseñamos los coeficientes del ecualizador:

$$\begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ q_{Leq-1} \end{pmatrix} = \mathbf{Z}^{-1} \cdot \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} +1,250 \\ -0,625 \\ +0,625 \\ -0,469 \\ +0,391 \\ -0,312 \\ +0,254 \\ -0,205 \end{pmatrix}$$

### 2.1.2. Estudio en el dominio de la frecuencia

Un estudio en el dominio del tiempo del valor de las muestras de la señal a la salida del canal nos ha permitido obtener el diseño del ecualizador ZF, que pretende forzar una señal libre de ISI a la salida. Un estudio en el dominio de la frecuencia nos permitirá interpretar mejor cómo se comporta el forzador de ceros y llegar a la conclusión de que este tipo de ecualizador no es la mejor solución para mitigar el efecto del canal.

Si mantenemos los supuestos de que solo se ha transmitido un símbolo y que el estudio se hace en ausencia de ruido, la transformada de Fourier de la señal a la salida del canal será el producto de la transformada de Fourier de pulso conformador  $P(f)$ , multiplicado por la respuesta en frecuencia del canal  $H(f)$  y multiplicado por la respuesta en frecuencia del filtro adaptado  $P^*(f)$ . Así pues:

$$C(f) = P(f) \cdot H(f) \cdot P^*(f) \quad (7)$$

Recordad que el pulso conformador se ha diseñado para que, después de que la señal sea muestreada a una muestra por símbolo, se cumpla la condición  $C_{s\text{amp}}(f) = 1$ , donde  $C(f)$  en ausencia de canal vale  $C(f) = P(f) \cdot P^*(f)$ , y el subíndice *samp* denota el espectro de la señal después de muestrear a una muestra por símbolo. De este modo, en ausencia de canal, el muestreo del pulso conformador convolucionado con el filtro adaptado da lugar a una señal espectralmente plana (de manera equivalente genera una delta en el dominio del tiempo, tal como vemos en la figura 2).

Ahora bien, en presencia de canal, la respuesta impulsional del canal rompe esta propiedad, y por lo tanto se obtiene una señal que no es espectralmente plana, sino que está afectada por la respuesta en frecuencia del canal. Para compensar esta distorsión, se propone introducir un ecualizador lineal, de modo que la transformada de Fourier de la señal a la salida del ecualizador

sea (8), donde  $H_{eq}(f)$  será la respuesta en frecuencia del ecualizador y de nuevo el subíndice  $samp$  denota el espectro de la señal después de muestrear a una muestra por símbolo:

$$Z_{eq}(f) = [P(f) \cdot H(f) \cdot P^*(f)]_{samp} \cdot H_{eq}(f) = C_{samp}(f) \cdot H_{eq}(f) \quad (8)$$

En el caso del forzador de ceros, nos vamos a preguntar cuál tiene que ser la respuesta en frecuencia del ecualizador para que la señal  $Z_{eq}(f)$  esté libre de ISI. Para que esto se cumpla, será necesario que  $H_{eq}(f)$  garantice que  $Z_{eq}(f)$  sea espectralmente plana y por lo tanto:

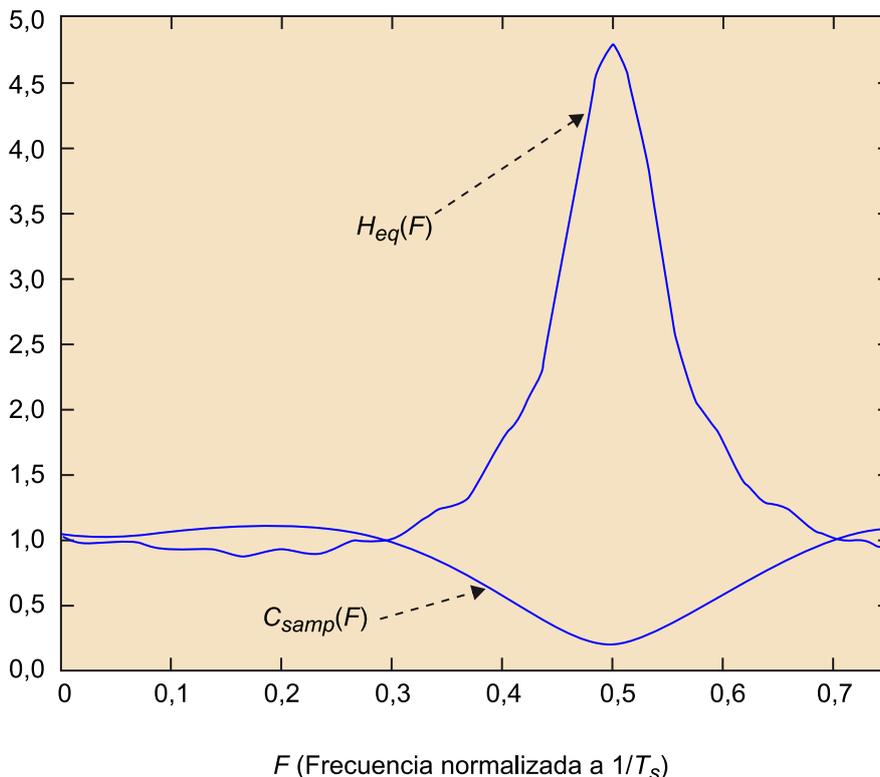
$$H_{eq}(f) = \frac{1}{C_{samp}(f)} \quad (9)$$

Recordemos que, formalmente, una atenuación de canal  $\alpha$  y un retardo de propagación  $t_0$  no afectan a la forma de onda de la señal. Por lo tanto, de manera estricta, la respuesta en frecuencia del ecualizador de la ecuación (9) podría ir multiplicada por una constante  $\alpha$  y por un término de fase lineal  $e^{-j2\pi f t_0}$  que mantuviera la señal a la salida del ecualizador libre de ISI.

#### Ved también

Ved la introducción del módulo "Canal de comunicaciones" de esta asignatura.

Figura 8. Ejemplo de respuesta en frecuencia de un canal y su ecualizador forzador de ceros



Si obviamos la constante y el término de fase, lo importante es ver que el papel del ecualizador forzador de ceros es invertir la respuesta en frecuencia del canal de modo que enfatice aquellas frecuencias atenuadas por el canal (valores de  $H(f)$  pequeños), y no lo haga para las frecuencias en las que el canal no atenúa. La figura 8 ilustra las respuestas en frecuencia del canal y del ecualizador para

el ejemplo del ecualizador antes diseñado (como las señales que se representan han sido muestreadas a una muestra por símbolo, el eje de frecuencia se representa en frecuencia normalizada a la frecuencia de muestreo).

### 2.1.3. Inconvenientes del forzador de ceros

Del análisis hecho sobre el comportamiento del forzador de ceros en el dominio de la frecuencia vemos dos problemas que nos harán cuestionar el forzador de ceros como ecualizador, el segundo de ellos más crítico que el primero:

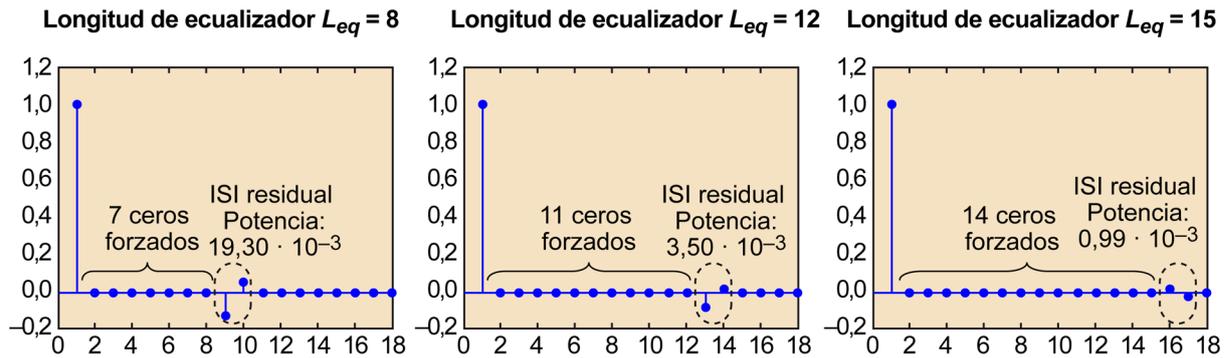
1) **El ecualizador forzador de ceros no puede eliminar completamente la ISI a la salida del ecualizador.** Para ver esta afirmación, basta con ver en la ecuación (9) cómo tiene que ser la respuesta en frecuencia del ecualizador y su respuesta impulsional asociada. Si recordamos lo que hemos estudiado en asignaturas de señales y sistemas y tenemos en cuenta que, excepto en aplicaciones muy concretas, el canal presenta una respuesta impulsional finita (FIR), el filtro lineal (ecualizador) que implemente (9) tendría que ser en general un filtro de respuesta impulsional infinita (IIR). Ahora bien, el ecualizador forzador de ceros se ha diseñado como un filtro con  $L_{eq}$  coeficientes; por lo tanto, como un filtro FIR. Dicho de otro modo, haría falta un número infinito de coeficientes  $L_{eq} \rightarrow \infty$  para eliminar por completo la ISI.

Este hecho ya se podía intuir a partir del diseño que se hacía en el dominio del tiempo. Una señal libre de ISI ha de cumplir  $z_{eq}[n]=0 \forall n \neq 0$ . En el diseño del ecualizador, se fuerzan  $L_{eq} - 1$  ceros a la salida del ecualizador  $n \in [1, \dots, L_{eq} - 1]$  sin tener ningún otro tipo de control sobre lo que ocurre fuera de este intervalo (figura 9).

En la práctica, si el valor de  $L_{eq}$  está muy ajustado, podremos asumir que la ISI residual fuera de este intervalo será lo bastante pequeña. En el caso de que no sea así, bastará con aumentar el orden del filtro ecualizador (aumentar el valor de  $L_{eq}$ ) hasta obtener un valor que cumpla con nuestra especificación. Observemos que el problema que nos podemos encontrar siguiendo esta política es que el número de coeficientes del ecualizador para un canal concreto sea excesivo y dé lugar a un ecualizador extremadamente complejo.

#### Ejemplo

La figura 9 muestra la ISI residual y su potencia para el ejemplo del ecualizador diseñado antes para diferentes longitudes del ecualizador  $L_{eq} = \{8, 12 \text{ y } 15\}$ . Podemos ver que la ISI residual varía en función del número de coeficientes.

Figura 9. ISI residual para diferentes longitudes de ecualizador  $L_{eq} = \{8, 12 \text{ y } 15\}$ 

2) **El filtro forzador de ceros enfatiza el ruido.** El principal problema del forzador de ceros es que en su diseño se ignora la presencia de ruido. Si nos centramos en exclusiva en obtener una señal libre de ISI, no tenemos en cuenta cuál es el efecto del ecualizador sobre el ruido, y cuál es su valor a la salida. Sin embargo, en la práctica, cualquier canal de comunicaciones presenta un término de ruido aditivo que se puede enfatizar al pasar por el ecualizador. En efecto, el ruido aditivo, que es espectralmente blanco, también es filtrado por el ecualizador, y por lo tanto multiplicado por la respuesta en frecuencia de este  $H_{eq}(f)$  (9).

El problema se presentará en aquellos casos en los que el canal presente una fuerte atenuación para alguna frecuencia,  $H(f) \rightarrow 0$ . Observemos que en este caso la respuesta en frecuencia del ecualizador tenderá a infinito. Esto no será un problema para la señal a la salida del canal, que habrá sido fuertemente atenuada a aquella frecuencia, pero sí lo será para el ruido que, como no ha sido afectado por la respuesta en frecuencia del canal, verá que se amplifica con fuerza su densidad espectral de potencia a aquella frecuencia. La figura 8 ilustra este problema. Tal como se puede ver, el canal presenta una fuerte atenuación a  $F = 0,5$ , y en esta frecuencia el ecualizador actúa amplificando la señal (y en consecuencia, también amplificará el ruido).

Las consecuencias de este problema pueden ser nefastas, en especial con canales que presentan fuertes atenuaciones a algunas frecuencias. En estos casos, solo tiene sentido utilizar el ecualizador ZF cuando no haya mucho ruido, es decir, cuando el escenario se acerque a las suposiciones hechas en el diseño (suponer que no hay ruido AWGN en la comunicación; en la práctica, esto quiere decir que el ruido es muy pequeño). En el resto de casos, será necesario considerar un ecualizador que tenga en cuenta el término de ISI y también el término de ruido en el diseño de los coeficientes.

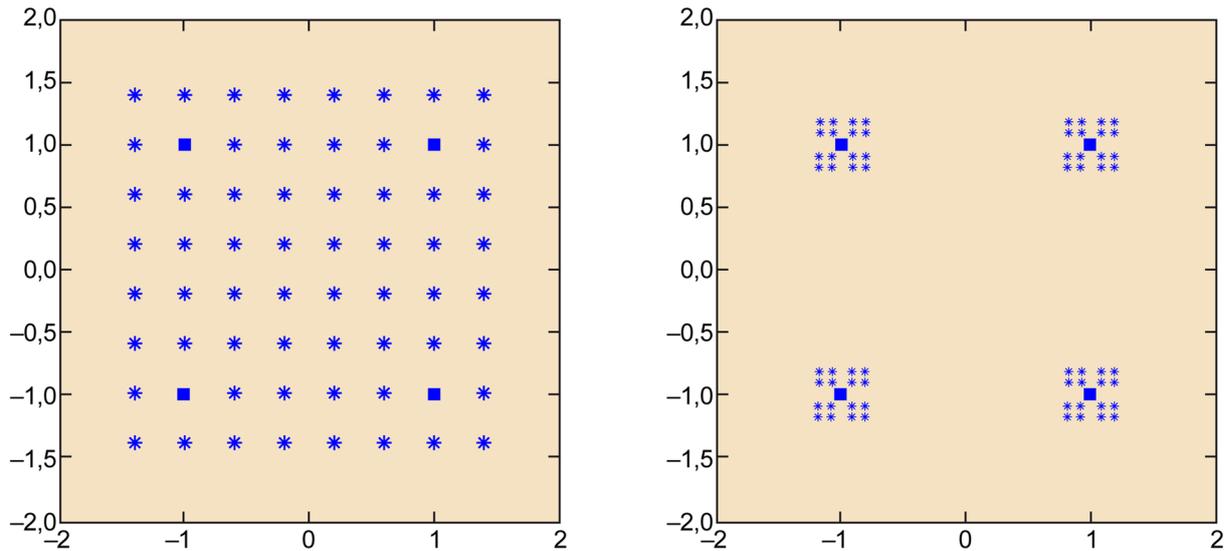
### Ejemplo

A continuación, ilustramos los dos problemas del ecualizador a partir del análisis, de la constelación de una señal modulada en QPSK en diferentes puntos de la cadena del receptor, transmitida a través del canal utilizado en el ejemplo de este apartado y ecualizada con el ecualizador ZF de ocho coeficientes diseñado con anterioridad.

Primero, en la figura 10 mostramos las constelaciones en ausencia de ruido para apreciar el efecto de la ISI residual:

- La gráfica izquierda de la figura muestra la constelación a la salida del canal. Como se puede ver, la constelación inicial (indicada con los cuatro cuadrados negros) se ve fuertemente distorsionada por la ISI.
- La gráfica derecha muestra la constelación a la salida del ecualizador forzador de ceros en ausencia de ruido. Tal como se ve, la constelación se acerca más a la de la QPSK (de nuevo, los cuatro cuadrados negros muestran la constelación original y las marcas con asterisco, las de la constelación ecualizada). Debido a la ISI residual comentada antes, no regeneramos la constelación de la QPSK, sino que se aprecia una interferencia que distorsiona los cuatro puntos de la QPSK.

Figura 10. Ejemplo de constelación de una señal QPSK en ausencia de ruido



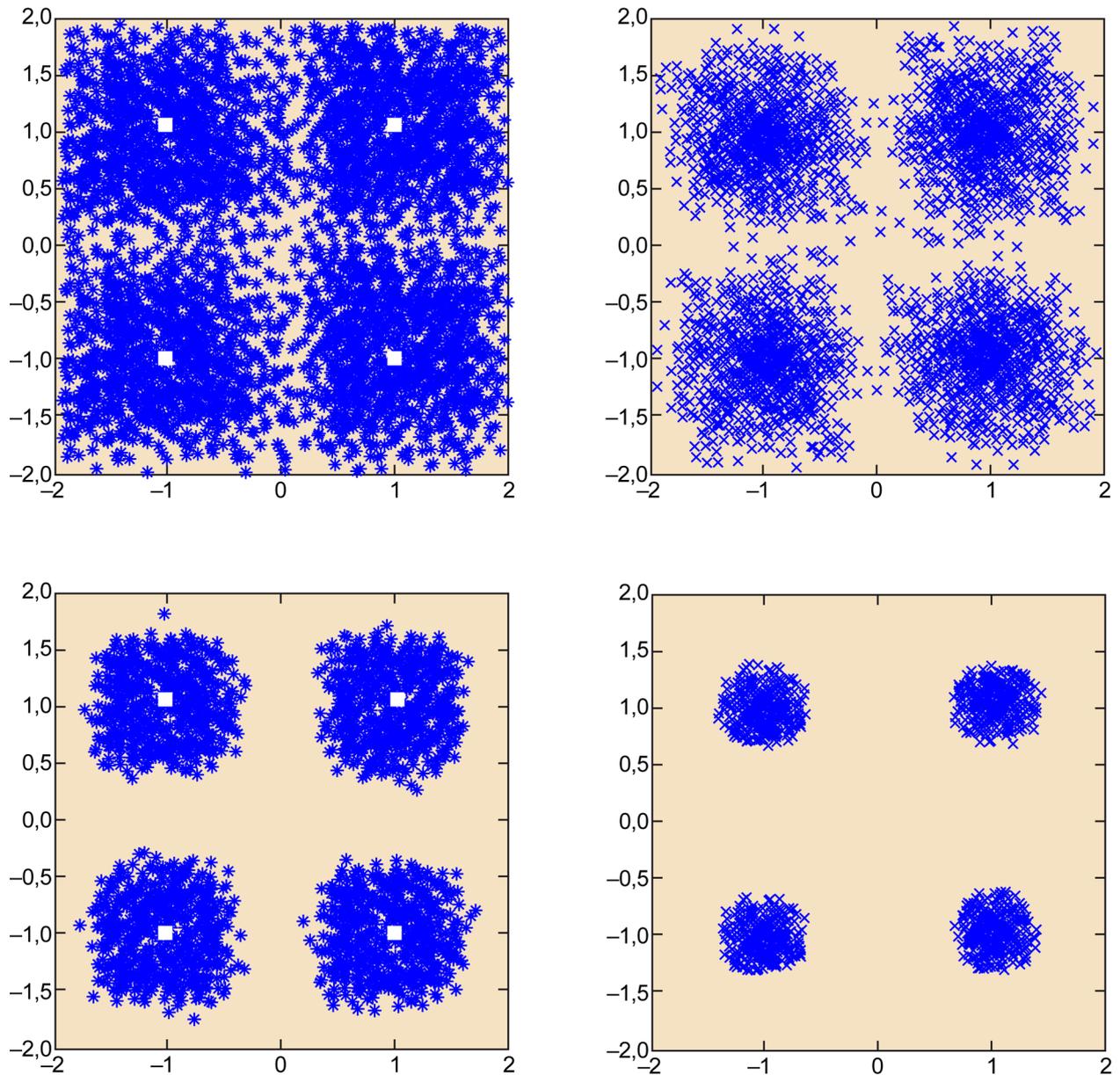
Ausencia de ruido a la salida del canal (izquierda) y a la salida del ecualizador forzador de ceros (derecha).

A continuación, en la figura 11 se muestra la constelación de la señal ecualizada, incluido el término de ruido AWGN, para apreciar el efecto de la enfatización del ruido. La constelación superior izquierda corresponde a la señal ecualizada con una SNR = 10 dB (valor razonable en un sistema de comunicaciones, a pesar de que actualmente la tendencia es que sea menor).

Como se aprecia, la señal está altamente degradada por el ruido (comparemos con la que se obtenía en el caso sin ruido). Para poder apreciar lo que se debe propiamente al término de ruido y lo que se debe a la enfatización del ruido provocada por el ecualizador ZE, la gráfica superior derecha muestra la constelación de una QPSK con ruido con una SNR = 10 dB, pero con canal ideal.

Así pues, la diferencia entre las constelaciones se debe al efecto de amplificación del ruido provocado por el forzador de ceros. De manera similar, y para ver que el efecto de enfatización es menor a medida que se reduce el ruido, la constelación inferior derecha corresponde a un canal con un término aditivo de ruido con una SNR = 20 dB (valor demasiado alto en un sistema real); se ve que, aunque mejora, todavía está degradada por el ruido.

Figura 11. Ejemplo de constelación de una señal QPSK con ruido



Ruido a la salida del ecualizador forzador de ceros (gráficas de la izquierda) para una  $SNR = 10$  dB (superior) y  $SNR = 20$  dB (inferior). A modo de referencia, las constelaciones de la derecha corresponden a una señal QPSK con ruido, pero con canal ideal.

## 2.2. Ecualizador de mínimo error cuadrático medio (MMSE)

El principal problema que se ha encontrado en el ecualizador forzador de ceros es que en el diseño no se tiene en cuenta el término de ruido, y por lo tanto no se controla el efecto que el ecualizador provoca sobre este. La alternativa será diseñar un ecualizador que sí tenga en cuenta el término de ruido y el efecto que tiene el ecualizador sobre él.

Como en el caso del forzador de ceros, este ecualizador seguirá la estructura de un ecualizador lineal y el objetivo será encontrar el valor de los coeficientes que consigan que la señal a la salida del ecualizador (ahora con presencia de ruido) sea tan parecida como sea posible a los símbolos transmitidos:

$$\text{Objetivo: } z_{eq}[n] \sim a_n \quad (10)$$

Para cuantificar la diferencia entre la señal a la salida del ecualizador y los datos transmitidos, definimos el error y el error cuadrático medio (*MSE*, del inglés *mean-squared error*) –o potencia del error–, respectivamente, como las expresiones (11) y (12):

$$e[n] = z_{eq}[n] - a_n \quad (11)$$

$$MSE = E\{|e[n]|^2\} = E\{|z_{eq}[n] - a_n|^2\} \quad (12)$$

donde el operador  $E\{\cdot\}$  corresponde a la esperanza matemática que calcula el promedio estadístico del error.

Dado que queremos hacer la señal tan parecida como sea posible, estableceremos como criterio reducir el error al máximo, y en consecuencia, la potencia del error. Por lo tanto, el diseño del ecualizador, como su nombre indica, buscará encontrar el valor de los coeficientes que minimice el error cuadrático medio (*MMSE*, del inglés *minimum mean-squared error*).

Una vez cuantificado el parámetro que se optimizará en el diseño de los coeficientes del ecualizador ( $q_k$ ) y definida la función de coste que depende de estos coeficientes, el diseño se reduce a resolver un problema de optimización clásico (derivar la función de coste, respecto a la variable de diseño e igualar su derivada a cero para encontrar el valor que optimiza la función).

Las ecuaciones (13) desarrollan la expresión de la función de coste para dejarla en función de los coeficientes del ecualizador ( $q_k$ ):

- En la primera línea, se sustituye el error  $e[n]$  por su expresión concreta.
- En la segunda línea, la señal a la salida del ecualizador se pone en función de la señal a la entrada y de los coeficientes del ecualizador (2).
- A continuación, el término de módulo al cuadrado de la suma se desarrolla en tres términos y el operador esperanza pasa dentro de los sumatorios, ya que la esperanza de la suma es la suma de esperanzas.
- Finalmente, en la última igualdad, la esperanza sobre los datos a la salida del canal se corresponde con la autocorrelación de los datos a la salida del canal  $R_{zz}[k]$ , y del mismo modo, la esperanza sobre el producto de los datos para la señal transmitida se corresponde con la correlación cruzada entre los datos a la salida del canal y la secuencia de símbolos deseados  $R_{za}[k]$ .

$$\begin{aligned}
 MSE &= E\{|e[n]|^2\} = E\{|z_{eq}[n] - a_n|^2\} \\
 &= E\left\{\left|\sum_{k=0}^{L_{eq}-1} q_k z[n-k] - a_n\right|^2\right\} \\
 &= \sum_{k=0}^{L_{eq}-1} \sum_{k'=0}^{L_{eq}-1} q_k q_{k'} E\{z[n-k]z[n-k']\} \\
 &\quad - 2 \sum_{k=0}^{L_{eq}-1} q_k E\{z[n-k]a_n\} + E\{a_n^2\} \\
 &= \sum_{k=0}^{L_{eq}-1} \sum_{k'=0}^{L_{eq}-1} q_k q_{k'} R_{zz}(k-k') \\
 &\quad - 2 \sum_{k=0}^{L_{eq}-1} q_k R_{za}(k) + E\{a_n^2\}
 \end{aligned} \tag{13}$$

Una vez desarrollada la expresión del error cuadrático medio, calculamos la derivada de la expresión e igualamos a cero:

$$\min_{q_i} MSE \rightarrow \frac{\partial MSE}{\partial q_i} = 0 \quad i = 0 \dots L_{eq} - 1 \tag{14}$$

Como debemos encontrar el valor óptimo para los  $L_{eq}$  coeficientes (tenemos  $L_{eq}$  incógnitas), hay que calcular  $L_{eq}$  derivadas, cada una con respecto a uno de los coeficientes. El resultado final es un sistema de ecuaciones con  $L_{eq}$  incógnitas (los coeficientes del ecualizador) y  $L_{eq}$  ecuaciones (una por cada una de las derivadas), como se indica a continuación:

### Reflexión

Como ya se ha comentado en la introducción, para que podáis seguir la derivación del diseño sin tener que incorporar notación adicional que complicaría los pasos, asumimos que los símbolos transmitidos, el canal y la señal a la salida son reales, y por lo tanto también lo serán los coeficientes del ecualizador. En un problema real, con una modulación paso banda y un diseño hecho sobre el equivalente paso bajo, todos los términos serán complejos. Aun asumiendo que la señal y el canal son reales, los diferentes pasos y el resultado final serán los mismos (modificando algún término por su conjugado). En la ecuación final de diseño del ecualizador MMSE, introducimos los términos conjugados para dar la expresión válida para el caso general complejo.

$$\sum_{k=0}^{L_{eq}-1} q_k R_{zz}(k-i) = R_{zd}(i) \quad i=0 \dots L_{eq}-1 \quad (15)$$

Este sistema de ecuaciones se puede escribir de manera matricial (en el que ya introducimos la notación compleja):

$$\begin{pmatrix} R_{zz}[0] & R_{zz}[1] & \dots & R_{zz}[-(L_{eq}-1)] \\ R_{zz}^*[1] & R_{zz}[0] & \dots & R_{zz}[-(L_{eq}-2)] \\ \vdots & \vdots & \ddots & \vdots \\ R_{zz}^*[L_{eq}-1] & R_{zz}^*[L_{eq}-2] & \dots & R_{zz}[0] \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ q_{L_{eq}-1} \end{pmatrix} = \begin{pmatrix} R_{zd}[0] \\ R_{zd}[1] \\ \vdots \\ R_{zd}[L_{eq}-1] \end{pmatrix} \quad (16)$$

En esta expresión, se ha considerado que la función de autocorrelación cumple  $R_{zz}^*[k] = R_{zz}[-k]$  —donde  $R_{zz}[k-i] = E\{z[n-k]z^*[n-i]\}$ — y la función de correlación cruzada  $R_{zd}[k] = E\{z[n-k]a_n^*\}$ .

Observemos, pues, que la manera de diseñar el ecualizador es muy sencilla. Conocida la autocorrelación de los datos a la salida del canal  $R_{zz}[k]$  y la correlación cruzada entre los datos a la salida del canal y la secuencia de símbolos deseados  $R_{za}[k]$ , se construyen la matriz de correlación  $\mathbf{R}_{zz}$  y el vector de correlaciones cruzadas  $\mathbf{R}_{za}$ :

$$\mathbf{R}_{zz} = \begin{pmatrix} R_{zz}[0] & R_{zz}[1] & \dots & R_{zz}[-(L_{eq}-1)] \\ R_{zz}^*[1] & R_{zz}[0] & \dots & R_{zz}[-(L_{eq}-2)] \\ \vdots & \vdots & \ddots & \vdots \\ R_{zz}^*[L_{eq}-1] & R_{zz}^*[L_{eq}-2] & \dots & R_{zz}[0] \end{pmatrix}$$

$$\mathbf{R}_{za} = \begin{pmatrix} R_{zd}[0] \\ R_{zd}[1] \\ \vdots \\ R_{zd}[L_{eq}-1] \end{pmatrix}$$

Y los  $L_{eq}$  coeficientes del ecualizador se obtienen después de invertir  $\mathbf{R}_{zz}$  y multiplicarla por el vector  $\mathbf{R}_{za}$ :

$$\begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ q_{L_{eq}-1} \end{pmatrix} = \mathbf{R}_{zz}^{-1} \cdot \mathbf{R}_{za} \quad (17)$$

El principal problema del ecualizador MMSE (que también lo es del ZF), en especial cuando el número de coeficientes del ecualizador es elevado, es que hay que invertir la matriz de correlación  $\mathbf{R}_{zz}$ , tarea que puede resultar computacionalmente compleja cuando las dimensiones de la matriz son grandes. En el subapartado siguiente se verán formas de diseñar el ecualizador que no necesitan invertir la matriz.

**Reflexión**

Igual que en el forzador de ceros, podemos relajar el diseño del ecualizador y permitir un cierto retardo  $d$  de la señal a la salida del ecualizador. En este caso, el cálculo de la función de correlación cruzada será:

$$R_{zd}[k] = E\{z[n-k]a_{n-d}^*\}$$

**Reflexión**

El ecualizador de mínimo error cuadrático también se denomina *filtro de Wiener*, propuesto por el matemático Norbert Wiener.

Tal como se puede ver, el diseño del ecualizador MMSE depende de la estadística de las recepciones, que se tiene que conocer *a priori*. Llegados a este punto, cuando ya tenemos la solución a los ecualizadores, nos planteamos la pregunta siguiente: ¿cómo se conocen los valores de la autocorrelación  $R_{zz}[k]$  y la correlación cruzada  $R_{za}[k]$ ? Dicho con otras palabras, en un problema real, cuando un transmisor concreto transmite unos datos por un canal concreto, ¿cómo puede el receptor diseñar el ecualizador de MMSE? Para determinar los valores de las correlaciones, se pueden dar dos situaciones: que la respuesta impulsional global sea conocida (tal como se asume en el caso del ecualizador ZF), o que no se conozca. Esto da lugar a dos casos con soluciones finales diferentes, que vamos a ver en los subapartados siguientes.

### 2.2.1. Caso 1: Respuesta impulsional global $c[n]$ conocida

Si se conoce la respuesta del canal, se conoce el valor de las funciones de correlación. En efecto, como la señal a la salida, incluido el término de ruido, vendrá dada por la expresión (18), donde  $L$  denota la duración de la respuesta impulsional global, la expresión de correlaciones podrá obtenerse de manera analítica, una vez conocidas las estadísticas de primer y segundo orden de los símbolos transmitidos y del ruido a la salida del filtro adaptado.

$$z[n] = \sum_{k=0}^{L-1} c[n] a_{n-k} + \tilde{w}_n \quad (18)$$

Asumimos, como habitualmente se hace, las estadísticas siguientes para los símbolos y el ruido:

- **Estadística de los símbolos transmitidos:** símbolos complejos, independientes e idénticamente distribuidos, de media cero  $E\{a_n\} = 0$  y varianza unidad  $E\{|a_n|^2\} = 1$ .
- **Estadística del ruido:** ruido complejo, blanco, independiente de los símbolos y con una estadística gaussiana de media cero  $E\{\tilde{w}_n\} = 0$  y varianza  $E\{|\tilde{w}_n|^2\} = \sigma_w^2$ .

Entonces, la matriz de correlación  $\mathbf{R}_{zz}$  es igual a (19), donde la matriz  $\mathbf{C}$  de dimensiones  $(L_c + L_{eq} - 1) \times L_{eq}$  contiene en sus columnas versiones desplazadas de la respuesta impulsional global, y el vector  $\mathbf{R}_{za}$ , que también contiene los elementos de la respuesta impulsional global, es igual al vector hermítico obtenido a partir de la fila  $d$  de la matriz  $\mathbf{C}$ , donde  $d$  es el retardo fijado por el ecualizador.

$$\mathbf{R}_{zz} = (\mathbf{C}^H \mathbf{C} + \sigma_w^2 \mathbf{I}) \quad (19)$$

$$\mathbf{C} = \begin{pmatrix} c[0] & 0 & \dots & 0 & 0 \\ c[1] & c[0] & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ c[L-1] & c[L-2] & \dots & c[0] & 0 \\ 0 & c[L-1] & \ddots & \vdots & c[0] \\ 0 & 0 & \dots & c[L-1] & \vdots \\ 0 & 0 & \dots & 0 & c[L-1] \end{pmatrix}$$

Así, si no hay retardo:  $\mathbf{R}_{za} = (c[0], 0, 0, \dots, 0)^H$ ; si se admite un retardo de una muestra:  $\mathbf{R}_{za} = (c[1], c[0], 0, 0, \dots, 0)^H$ ; y así, sucesivamente.

### Reflexión

Observad que en el diseño del ecualizador interviene de manera directa la varianza de ruido  $\sigma_w^2$  que aparece directamente en la matriz  $\mathbf{R}_{zz}$ . Resulta interesante analizar los dos casos asintóticos: cuando no hay ruido (alta SNR) y cuando hay mucho ruido (baja SNR):

a) En el primero de los casos, aunque no resulta trivial, se puede ver que el ecualizador MMSE tiende a la solución del ecualizador ZF, hecho que quiere decir que el ecualizador ZF minimiza el error cuadrático medio en ausencia de ruido.

b) En el segundo de los casos, a muy baja SNR,  $\mathbf{R}_{zz} \rightarrow \sigma_w^2 \mathbf{I}$ . En este caso, la solución de los coeficientes del ecualizador, excepto un factor de escala  $1/\sigma_w^2$ , coincide con el vector  $\mathbf{R}_{za}$ , lo que quiere decir que el ecualizador tiende a combinar de manera ponderada las muestras a la salida del filtro adaptado. Dicho de otro modo, podemos interpretar que la señal recibida es tan ruidosa que no compensa intentar reducir el efecto de la ISI, ya que el ruido predomina sobre la interferencia provocada por el canal (ISI).

### Actividad

Os proponemos como ejercicio que verifiquéis el cálculo de la matriz  $\mathbf{R}_{zz}$  y del vector  $\mathbf{R}_{za}$ . Os proponemos también el estudio asintótico de estos para un ejemplo sencillo.

## 2.2.2. Caso 2: Respuesta impulsional global $c[n]$ desconocida

La manera de resolver esta situación es recurrir a una estimación de las correlaciones necesarias para resolver la ecuación de diseño. Para hacerlo, el transmisor envía una secuencia de  $N$  datos conocida:  $a_n$ ,  $n = 1, \dots, N$  (llamada *secuencia de entrenamiento* o *secuencia de datos piloto*), y el receptor hace las operaciones siguientes con los datos a la salida del canal:

$$\widehat{R}_{zz}[k] = \frac{1}{N} \sum_{n=1}^N z[n-k]z^*[n] \quad ; \quad \widehat{R}_{za}[k] = \frac{1}{N} \sum_{n=1}^N z[n-k]a_n^* \quad (20)$$

Como podéis ver, es necesario que el receptor conozca los datos transmitidos para poder estimar la correlación cruzada entre los datos a la salida del canal y la secuencia de símbolos deseados  $\widehat{R}_{za}[k]$ .

A partir de las estimaciones de la correlación se pueden construir las estimaciones de las matrices  $\widehat{\mathbf{R}}_{zz}$  y  $\widehat{\mathbf{R}}_{za}$ , y a partir de las matrices estimadas se pueden encontrar los valores de los coeficientes:

$$\begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ q_{L_{eq}-1} \end{pmatrix} = \widehat{\mathbf{R}}_{zz}^{-1} \cdot \widehat{\mathbf{R}}_{za} \quad (21)$$

Si bien esta solución no es la original, para un número lo bastante largo de datos de entrenamiento,  $N$ , el resultado de la estimación de la correlación será lo suficientemente bueno (cerca de la correlación real), y por lo tanto el diseño será válido.

### 2.3. Ecuación adaptativa

El diseño de los ecualizadores antes propuestos, tanto del forzador de ceros como del ecualizador de mínimo error cuadrático medio, presenta dos problemas:

- Necesita invertir una matriz, hecho que puede resultar computacionalmente complejo cuando las dimensiones de esta sean grandes (como las dimensiones de la matriz son  $L_{eq} \times L_{eq}$ , hablar de las dimensiones de la matriz es equivalente a hablar del número de coeficientes del ecualizador).
- Requiere de un conocimiento o estimación del canal (equivalentemente, de las funciones de correlación  $R_{zz}[k]$  y  $R_{za}[k]$ ) como paso previo al diseño del ecualizador, que tiene que mantenerse constante desde el momento en el que se hace la estimación del mismo hasta el momento en el que se aplica el ecualizador.

El primer problema impone una restricción en la complejidad computacional. El segundo problema, quizás más complejo, introduce problemas cuando el canal es variante en el tiempo (ya hemos visto que en muchos casos el canal no es constante sino que va variando en el tiempo). Si el canal varía a lo largo del tiempo, habrá que enviar un bloque de datos de entrenamiento cada cierto tiempo para volver a hacer una estimación del canal y rediseñar el ecualizador. Estas secuencias de entrenamiento reducen la eficiencia de la comunicación, ya que mientras se envían símbolos de entrenamiento, no se envían datos de información. Si por otro lado, para no penalizar la eficiencia se intenta ahorrar reduciendo el envío de nuevas secuencias de entrenamiento, los coeficientes del ecualizador no serán válidos y el ecualizador no cumplirá correctamente su función.

Como ya podéis imaginar, hay que encontrar alguna solución que permita hacer un diseño de ecualizadores sin necesidad de invertir la matriz o de modo que sean eficientes en presencia de canales variantes en el tiempo. Esta solución pasa por el diseño de ecualizadores adaptativos. Un ecualizador adaptativo, como su nombre indica, es un ecualizador que va adaptando (corrigiendo) el valor de los coeficientes a lo largo del tiempo.

#### Ved también

Ved los canales variantes en el tiempo en el módulo "Canal de comunicaciones" de esta asignatura.

El ecualizador adaptativo más sencillo de entender e implementar es el ecualizador basado en el algoritmo *LMS* (del inglés *least-mean-square*). Es la solución idónea cuando el canal varía en el tiempo y también se puede plantear como solución por canales constantes si queremos implementar un ecualizador MMSE que evite invertir la matriz de correlación.

Hay otras soluciones con mejores prestaciones, como el ecualizador basado en el algoritmo *RLS* (del inglés *recursive least-squares*), pensado en especial para no tener que invertir la matriz, pero de mayor complejidad, que no vamos a tratar en este módulo. A continuación, detallamos el ecualizador basado en el algoritmo LMS.

### 2.3.1. Ecualizador LMS: algoritmo de gradiente

El objetivo del ecualizador basado en el algoritmo *LMS* (del inglés *least-mean-square*) es el mismo que el ecualizador MMSE: encontrar los valores de los coeficientes del ecualizador lineal  $q_k$  que minimicen el error cuadrático medio. La diferencia está en la manera de llegar a la solución óptima. Mientras que el ecualizador MMSE recurría a una solución analítica (derivar la función de coste con respecto a la variable de diseño e igualar la derivada a cero para encontrar el valor que optimiza la función), el ecualizador LMS llegará a la solución mediante un algoritmo de gradiente.

Destacamos, antes de seguir con la derivación del ecualizador adaptativo, que se toma como función de optimización el error cuadrático medio, dado el interés de esta métrica para evaluar el parecido entre la señal a la salida del ecualizador y la señal deseada, aunque podríamos diseñar otros criterios, como por ejemplo un ecualizador adaptativo basado en el forzador de ceros (que por razones ya discutidas, no nos interesará).

Comentamos a continuación, de manera bastante ilustrativa, en qué consiste un algoritmo de gradiente. Para hacerlo, ilustramos el problema concreto de encontrar los valores de los coeficientes del ecualizador que minimizan la función de error cuadrático medio.

Observamos que la función que se quiere optimizar (la MSE) tiene una dependencia cuadrática con la variable independiente (los coeficientes). Simplificando el problema para poder representarlo en una gráfica 2D, suponemos que hay solo un coeficiente  $q$ , y por lo tanto la función  $MSE(q)$  correspondería a una parábola, como se ilustra en la figura 12. Sabemos que en una parábola hay un único mínimo, condición indispensable para garantizar que el algoritmo de gradiente que vamos a describir a continuación termine por proporcionarnos la solución óptima. La ampliación al problema de  $L_{eq}$  coeficientes también cumplirá esta condición por el hecho de depender cuadráticamente de los coeficientes.

#### Ved también

Ved los primeros párrafos del subapartado 2.2 de este módulo didáctico.

La figura 12 ilustra el problema y los diferentes pasos del algoritmo:

- 0 - El objetivo será llegar al valor del coeficiente que minimiza la función  $MSE(q)$ ; esto es  $q_{opt}$ .
- 1 - El problema se inicializa en un valor cualquiera para  $q$ , que denotaremos con  $q^0$  (el superíndice denotará la iteración en la que nos encontramos dentro del algoritmo). Si tenemos alguna información de dónde está aproximadamente el mínimo, tomaremos un valor cercano para reducir el tiempo necesario para llegar al mínimo; en caso contrario, cualquier valor aleatorio es válido.
- 2 - Evaluamos la pendiente de la función  $MSE(q)$  en el punto  $q^0$  a partir del valor de la derivada de la función  $MSE$  en  $q^0$ . Observad que el signo de la pendiente (signo de la derivada) nos indicará si el mínimo se encuentra a la derecha o a la izquierda de  $q^0$ , y el valor de la pendiente (valor de la derivada) nos dirá si estamos lejos o cerca de este punto mínimo.
- 3 - En la iteración siguiente damos un pequeño salto de un cierto valor en la dirección del mínimo hasta llegar a  $q^1$ . Para obtener  $q^1$ , lo haremos a partir de  $q^0$  y el valor de la derivada, según la ecuación siguiente:

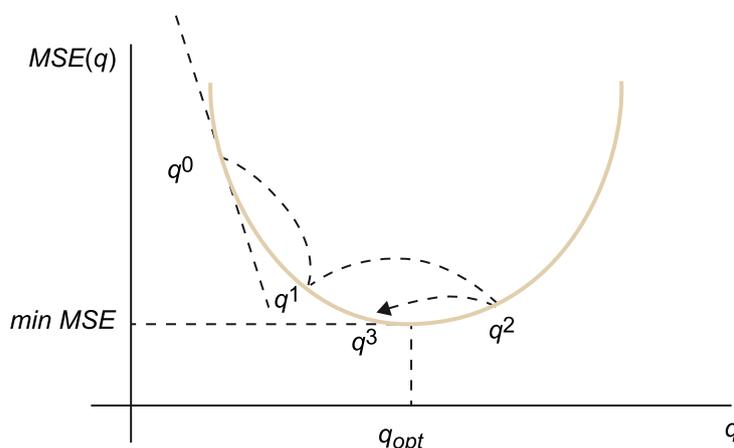
$$q^1 = q^0 - \mu \left. \frac{dMSE(q)}{dq} \right|_{q^0}$$

donde  $\mu$  es un escalar llamado paso de adaptación, del que hablaremos más tarde.

A partir de  $q^1$ , repetimos el mismo procedimiento que en 2 para determinar dónde se encuentra el mínimo  $q_{opt}$ . Si el paso de adaptación  $\mu$  se ha diseñado correctamente,  $q^1$  estará más cerca del mínimo en el sentido que  $MSE(q^1) < MSE(q^0)$ .

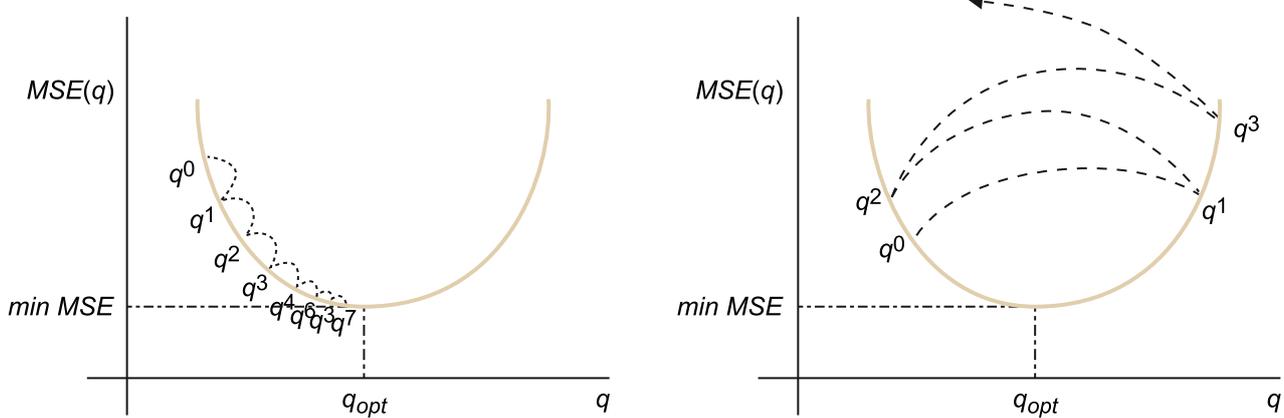
- 4 - Iteramos los pasos 2 y 3 para ir acercándonos al mínimo cada vez más. Fijémonos en que, si en algún momento llegáramos al mínimo exacto, como la derivada en aquel punto es cero, el algoritmo iterativo quedaría congelado en  $q_{opt}$ .

Figura 12. Esquema del algoritmo de gradiente



Observemos que, a diferencia de la solución analítica (consistente en igualar la derivada a cero y despejar), que permite obtener la solución directamente, el algoritmo de gradiente, al tratarse de un algoritmo iterativo, necesitará de un tiempo de convergencia hasta llegar a la solución. El valor del paso de adaptación  $\mu$  tendrá un papel importante en el tiempo de convergencia. Un valor de  $\mu$  pequeño garantizará que se llegue paso a paso al mínimo de la función, pero el tiempo necesario para alcanzar el punto óptimo será elevado. Por el contrario, un valor de  $\mu$  excesivamente grande puede tener consecuencias desastrosas si en cada iteración, en lugar de acercarnos al mínimo, nos alejamos cada vez más de él. En el primer caso, diremos que el algoritmo converge, mientras que en el segundo hablaremos de divergencia del algoritmo. Así pues, encontrar el valor de  $\mu$  que garantice la convergencia y a la vez no alargue en exceso el tiempo de convergencia será una tarea de difícil solución. La figura 13 ilustra ambas situaciones, que podemos encontrar en función de  $\mu$ .

Figura 13. Evolución de un algoritmo de gradiente para  $\mu$  pequeña (izquierda) y  $\mu$  grande (derecha)



Una vez descrito el algoritmo de gradiente, formalizamos los pasos necesarios hasta llegar a derivar el ecualizador basado en el algoritmo LMS. Recordemos que queremos encontrar el valor de los  $L_{eq}$  coeficientes que minimizan el error cuadrático medio a la salida del ecualizador. Para hacerlo, apliaremos la idea formulada en el ejemplo para el caso de un solo coeficiente en el caso de  $L_{eq}$  coeficientes. Así, si denotamos por  $\mathbf{q}$  el vector que contiene los  $L_{eq}$  coeficientes del ecualizador:

$$\mathbf{q} = \begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ q_{L_{eq}-1} \end{pmatrix}$$

y por  $\mathbf{q}^m$  el valor de este vector de coeficientes en la iteración  $m$ , el algoritmo de gradiente iterativo se puede escribir según la expresión (22):

$$\mathbf{q}^{m+1} = \mathbf{q}^m - \mu \cdot \nabla_{MSE}(\mathbf{q}^m) \quad (22)$$

**El gradiente de una función**

Recordad que el gradiente de una función  $f(\mathbf{r})$  en un punto indica la dirección de máxima pendiente y se define como el vector de derivadas parciales con respecto a cada una de las dimensiones (o elementos) de  $\mathbf{r}$ :

$$\nabla f(\mathbf{r}) = \left( \frac{\partial f(\mathbf{r})}{\partial r_1}, \dots, \frac{\partial f(\mathbf{r})}{\partial r_n} \right)^T$$

donde  $\nabla_{MSE}(\mathbf{q}^m)$  es el gradiente de la función evaluado en  $\mathbf{q}^m$ .

Recordemos que la función MSE está definida como sigue:

$$MSE = E\left\{e[n]^2\right\} = E\left\{z_{eq}[n] - a_n\right\}^2 = E\left\{\sum_{k=0}^{L_{eq}-1} q_k z[n-k] - a_n\right\}^2 \quad (23)$$

Entonces, su gradiente para el caso complejo se puede expresar como (24):

$$\nabla_{MSE}(\mathbf{q}) = E\{e[n] \cdot \mathbf{z}^*\} \quad (24)$$

En esta expresión,  $e[n]$ , que depende del vector de coeficientes, es el error entre la señal deseada y la señal a la salida del ecualizador, y el vector  $\mathbf{z}^*$  contiene las muestras conjugadas de la señal a la entrada del ecualizador (que coinciden con las muestras que hay en la línea de retardos del ecualizador):

$$\mathbf{z}^* = \begin{pmatrix} z[n]^* \\ z[n-1]^* \\ \vdots \\ z[n-(L_{eq}-1)]^* \end{pmatrix}$$

En la práctica, como sucede con el diseño del ecualizador MMSE, la solución requiere del cálculo de la esperanza matemática, que no puede obtenerse en tiempo real, ya que necesita conocer la estadística del canal. La solución alternativa que se implementa prescinde de esta esperanza matemática y reduce el promedio estadístico al valor instantáneo (es decir, sin la esperanza). Con esta simplificación, que es la que da lugar al algoritmo LMS, en cada nueva muestra se hace una nueva iteración del algoritmo de gradiente, y por lo tanto, podemos unificar la variable  $n$  y el índice  $m$  de modo que  $m = n$ .

Así pues, el algoritmo LMS se implementa según la expresión (25):

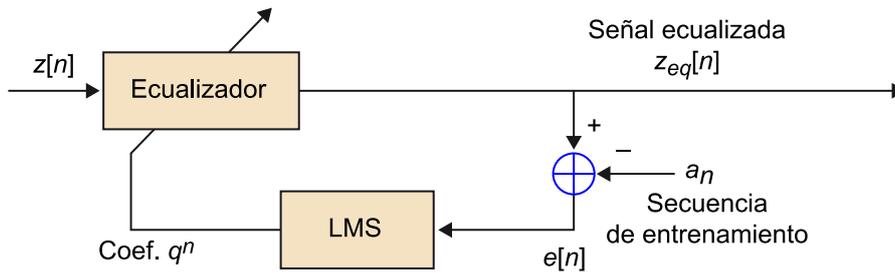
$$\mathbf{q}^{n+1} = \mathbf{q}^n - \mu e[n] \mathbf{z}^* \quad ; \quad \text{donde: } e[n] = \sum_{k=0}^{L_{eq}-1} q_k^n z[n-k] - a_n \quad (25)$$

donde la actualización depende del vector  $\mathbf{z}^*$  que contiene las muestras conjugadas de la señal a la entrada del ecualizador –como ya se ha definido antes–, del error cometido a la salida de la ecualización  $e[n]$  en la iteración  $m$ , y del paso de adaptación  $\mu$ .

Con respecto al valor que ha de tomar  $\mu$  hay que hacer un estudio de convergencia riguroso que obviaremos en este módulo. Para más detalles, os remitimos a la bibliografía, donde encontraréis detalles sobre este estudio.

La figura 14 ilustra la implementación del algoritmo adaptativo.

Figura 14. Esquema de implementación de un ecualizador adaptativo



Resumimos a continuación la idea del ecualizador LMS:

- 1) El transmisor envía una secuencia de datos, que por el momento asumiremos que son conocidos:  $a_0, a_1, a_2, \dots, a_{N-1}$ . El número de símbolos de entrenamiento  $N$  tiene que ser mayor o igual que el tiempo de convergencia del algoritmo.
- 2) El ecualizador se inicializa con un vector de coeficientes cualquiera  $\mathbf{q}^0$ . Cualquier conjunto de valores sirve. Algunos ejemplos de valores que se acostumbra a tomar pueden ser el primer coeficiente a 1 y el resto a 0, todos los coeficientes a 1, o una inicialización con valores aleatorios.
- 3) Con los coeficientes  $\mathbf{q}^0$  se obtiene el primer dato a la salida del ecualizador, que se compara con  $a_0$  para obtener el primer valor de error  $e[0]$ , que obviamente será muy grande al obtenerse a partir de un vector de coeficientes  $\mathbf{q}^0$  generado sin ningún tipo de criterio.
- 4) A partir de  $\mathbf{q}^0$  y del error  $e[0]$ , se itera el algoritmo LMS para obtener el vector de coeficientes  $\mathbf{q}^1$ . El valor de estos coeficientes se introducirá en el ecualizador y permitirá obtener el segundo dato en la salida  $z_{eq}[1]$ , que, comparada con  $a_1$ , nos proporcionará el valor del error  $e[1]$ .
- 5) El algoritmo continúa iterando en el paso 4. A cada paso, el error será menor y el vector de coeficientes irá convergiendo a la solución MMSE.

### 2.3.2. Secuencia de entrenamiento para controlar la actualización de los coeficientes

Observemos que para hacer la actualización de los coeficientes es necesario calcular el error  $e[n]$ , y por lo tanto, conocer los datos transmitidos. Esto quiere decir que una vez se deja de transmitir la secuencia de entrenamiento, la ecuación que regula las iteraciones y que por lo tanto adapta los coeficientes del ecualizador, ya no se puede calcular. Con respecto a este tema se pueden proponer diferentes políticas dependiendo de las características del canal.

## 1) Canal constante (invariante en el tiempo):

### Algoritmo asistido por datos (DA del inglés: *data aided*) con secuencia de entrenamiento

En este caso, se implementa estrictamente el algoritmo antes descrito con un número de iteraciones  $N$  que coincide con la duración de la secuencia de entrenamiento. Una vez pasadas las  $N$  iteraciones, se obtiene un vector de coeficientes del ecualizador  $\mathbf{q}$  que, considerando que ha convergido a la solución final, se congela y se aplica, sin ninguna nueva actualización, al bloque de datos de información –desconocidos–, que empiezan a transmitirse pasados los primeros  $N$  símbolos de entrenamiento.

Esta solución es válida cuando asumimos que el canal no variará a lo largo del tiempo, de manera que con los primeros  $N$  datos entrenamos el ecualizador, hacemos que aprenda a adaptarse al canal, y a continuación lo utilizamos para ecualizar. Esta solución nos tiene que proporcionar las mismas prestaciones que las obtenidas con el ecualizador MMSE y nos ha ahorrado la necesidad de invertir la matriz de correlación, que es uno de los problemas que hemos planteado al principio de este subapartado.

#### Reflexión

Aunque en teoría el ecualizador MMSE y el ecualizador LMS tienen las mismas prestaciones, ya que buscan optimizar la misma función de coste, en la práctica, el ecualizador LMS tendrá unas prestaciones ligeramente peores que el MMSE en el caso ideal en el que la respuesta del canal (o las matrices de correlación) sean conocidas. Al tratarse de un algoritmo adaptativo controlado por el error instantáneo  $e[n]$  (que siempre estará sujeto a la existencia del ruido AWGN), se obtendrá una solución cercana a la de mínimo error cuadrático medio, pero ruidosa, hecho que implicará una pequeña pérdida en prestaciones. Con respecto a la comparación del ecualizador LMS con la solución del ecualizador MMSE para el caso 2, en el que se estiman las matrices de correlación, se obtendrán resultados similares, puesto que en el proceso de estimación de las matrices de correlación también hay implícita una ligera degradación al quedar esta sujeta a la existencia del ruido AWGN.

## 2) Canal variante en el tiempo:

El segundo problema que hemos planteado al principio del subapartado es que el canal pueda ir variando a lo largo del tiempo. Como ya podéis intuir, en este caso la solución planteada en el punto anterior de congelar los coeficientes del ecualizador no resulta factible, puesto que a medida que el canal varía con el tiempo, el ecualizador va quedando obsoleto.

Ante este problema se pueden encontrar dos soluciones.

- **Algoritmo DA con secuencia de entrenamiento + símbolos piloto**

La primera solución consiste en enviar una secuencia de entrenamiento de  $N$  símbolos que permita entrenar al ecualizador, seguidamente empezar a enviar datos de información, que serán ecualizados con el valor del ecualizador calculado, y cada cierto tiempo intercalar algún símbolo conocido (que denomi-

haremos *símbolo piloto*) entre los datos de información que permita al ecualizador monitorizar el error de ecualización  $e[n]$  e ir adaptándose a las variaciones del canal.

Cada vez que llegue un símbolo piloto se podrá hacer una nueva iteración del algoritmo LMS para corregir el valor de los coeficientes. La distancia entre los símbolos piloto vendrá regulada por la velocidad de variación del canal. Así, si el canal varía muy lentamente, bastará con insertar un símbolo piloto cada cierto tiempo, mientras que si el canal varía con mayor rapidez, habrá que reducir la separación entre símbolos piloto con el objeto de aumentar la frecuencia de adaptación de los coeficientes del ecualizador.

- **Algoritmo asistido por decisiones (DD, del inglés: *decision directed*)**

El principal problema de los símbolos piloto es que reducen la eficiencia de la comunicación. Como ya se ha explicado, mientras se envían datos conocidos (secuencia de entrenamiento o símbolos piloto) no se envían datos de información, y por lo tanto se reduce la velocidad de transmisión efectiva.

Una solución para evitar la necesidad de insertar símbolos piloto es utilizar las propias decisiones para calcular el error cometido. Cuando el error a la salida del ecualizador es pequeño, la probabilidad de equivocarse en el símbolo a la salida del decisor es baja, lo que quiere decir que mirando la salida del decisor se puede saber (con una probabilidad muy baja de equivocarnos) cuál ha sido el símbolo transmitido. Así pues, si sabemos cuál ha sido el símbolo transmitido sin que este corresponda a un símbolo piloto, podemos utilizar las mismas decisiones de los símbolos de información como si fueran datos conocidos prescindiendo de la necesidad de enviar símbolos piloto.

La manera de proceder en este caso sería la siguiente (ved la figura 15): el transmisor envía una secuencia de entrenamiento de  $N$  símbolos que permite entrenar el ecualizador, y seguidamente empezar a enviar datos de información. Mientras se reciben los datos de la secuencia de entrenamiento, el cálculo del error para adaptar el valor de los coeficientes se hace a partir del conocimiento de los datos transmitidos. Una vez se deja de enviar la secuencia de entrenamiento, los coeficientes del ecualizador no se congelan, sino que el algoritmo LMS continúa iterando a razón de una iteración por símbolo transmitido. La diferencia es que ahora el cálculo del error  $e[n]$  se hará a partir de las decisiones que se tomen sobre los datos transmitidos a la salida misma del ecualizador.

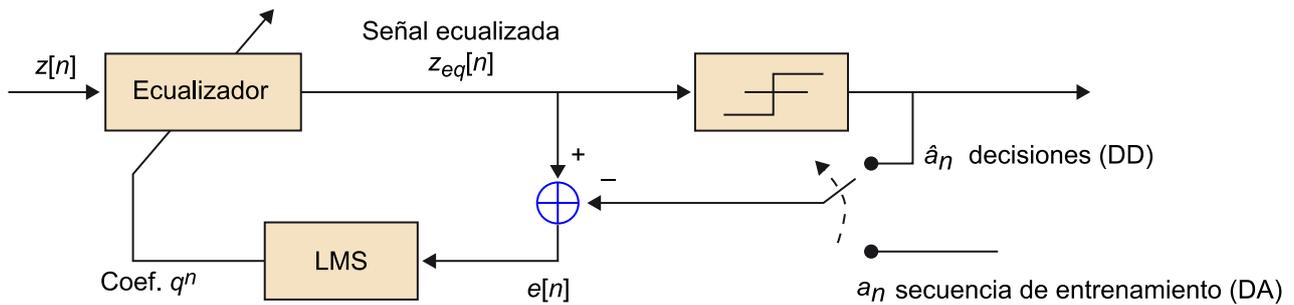
Observemos que esta estrategia solo podrá aplicarse cuando el error a la salida del ecualizador sea muy pequeño, o lo que es lo mismo, cuando la probabilidad de error a la salida del decisor sea baja. Para que esto suceda, será necesario que se cumplan dos condiciones: que los coeficientes del ecualizador estén lo suficientemente bien calculados, y que el nivel de ruido sea bajo. Para resolver

el primer problema, se envía una secuencia de entrenamiento de  $N$  símbolos antes de conmutar al modo DD, que garantizará que el algoritmo LMS haya convergido.

A partir de este momento, asumiremos que las variaciones del canal serán lo bastante lentas como para que en pocos símbolos el canal haya variado muy poco, de modo que el error de ecualización continúe siendo pequeño y la adaptación de los coeficientes del ecualizador pueda ir siguiendo las variaciones del canal. Con respecto al segundo problema, no tendremos mucha capacidad de actuación. Si el canal es poco ruidoso (trabajamos en alta SNR) podremos aplicar una estrategia basada en DD, mientras que si el canal es muy ruidoso (trabajamos a baja SNR) las decisiones no serán lo bastante fiables para aplicar una estrategia DD. En este caso, tenemos dos alternativas:

- La primera es transmitir símbolos piloto periódicamente para que el algoritmo LMS del receptor pueda actualizar los coeficientes del ecualizador trabajando en modo DA.
- La alternativa, que ahorra la transmisión de símbolos piloto, es utilizar un esquema denominado *turboecualizador*, en el que el ecualizador, seguido de un decodificador de canal, se diseña utilizando las decisiones blandas (en inglés: *soft decisions*) sobre los símbolos transmitidos, obtenidas del decodificador de canal. Cuando en una estrategia DD decidimos que el símbolo enviado transmitido sea  $-1$  o  $+1$ , estamos tomando decisiones duras o firmes (en inglés: *hard decisions*) sobre los símbolos enviados. A baja SNR, normalmente es demasiado arriesgado tomar estas decisiones, ya que la probabilidad de equivocarse es elevada. Por el contrario, si la estructura que el codificador de canal proporciona a los datos permite evaluar la probabilidad de que el símbolo enviado sea  $-1$  o  $+1$  y utilizamos esta probabilidad, evitaremos el riesgo de equivocarnos. Así, el decodificador calcula la métrica  $\ln(p(a_n = +1)/p(a_n = -1))$  de este símbolo, que nos dice cuánto más probable es un valor del símbolo que el otro. Si el valor absoluto de esta métrica es grande, quiere decir que estamos bastante seguros de que ha sido el símbolo transmitido, y vamos a utilizar este símbolo para diseñar el ecualizador; en cambio, si el valor absoluto de la métrica es pequeño debido al ruido, no nos fiaremos demasiado de este símbolo para diseñar el ecualizador y evitar que LMS diverja.

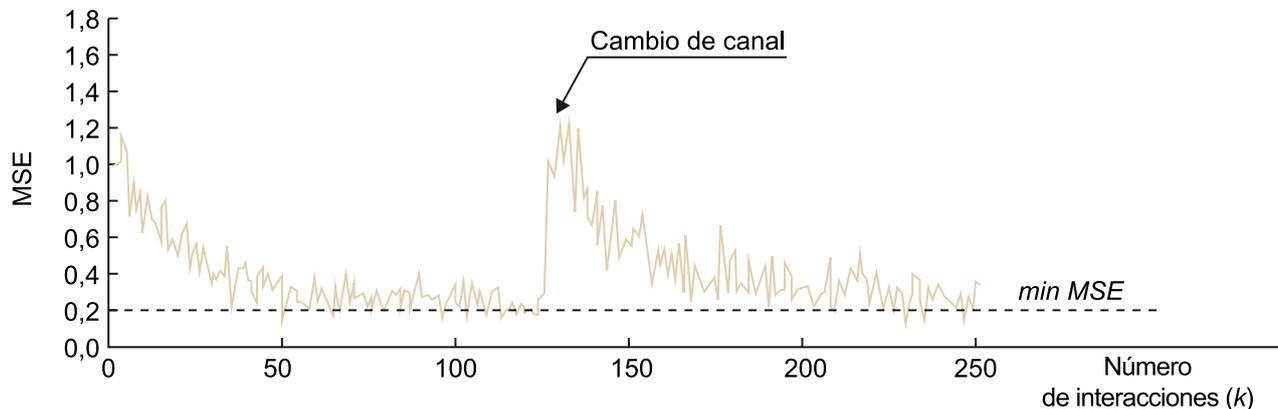
Figura 15. Esquema de implementación de un ecualizador adaptativo con conmutación DA a DD



En el caso de canales variantes en el tiempo, el valor del paso de adaptación  $\mu$  vuelve a tomar un papel importante. Cuando el canal varía en el tiempo, el algoritmo LMS tendrá que diseñarse para poder seguir las variaciones del canal, y por lo tanto el valor de la  $\mu$ , que ya hemos visto que es relevante para determinar el tiempo de convergencia (y de manera equivalente, la capacidad de seguimiento de las fluctuaciones del canal) tendrá que fijarse de modo adecuado. Así, habrá que fijar un valor grande para la  $\mu$  cuando el canal varíe con rapidez y una  $\mu$  pequeña cuando lo haga lentamente.

La figura 16 ilustra un ejemplo del valor del error cuadrático medio con un ecualizador adaptativo implementando el algoritmo LMS. En el eje vertical, está el error cuadrático medio a la salida del ecualizador, y en el eje horizontal el número de iteraciones del algoritmo. En la línea discontinua está el mínimo error cuadrático (valor que se lograría con un ecualizador MMSE).

Figura 16. Ejemplo de MSE medido a la salida de un ecualizador adaptativo basado en LMS



Tal como se puede ver en la figura 16, al principio el error es grande, y a medida que el algoritmo va iterando, el error converge hacia el valor mínimo, hecho que indica que los coeficientes del ecualizador han convergido hacia la solución óptima. El tiempo necesario para llegar a la solución del mínimo es el tiempo de convergencia. El algoritmo trabaja en modo DA hasta la iteración 50 y a partir de aquel momento conmuta a modo DD.

Para ilustrar la capacidad de adaptación del algoritmo a las variaciones del canal, en la iteración 125 se fuerza un cambio en la respuesta impulsional del canal. Como se puede ver, en aquel instante de tiempo el error vuelve a crecer,

ya que el ecualizador pasa a estar “desadaptado” con respecto a la solución óptima para el nuevo canal. A pesar de ello, se ve que el ecualizador es capaz de adaptarse a la nueva situación, y pasado un número de iteraciones, vuelve a converger hacia la nueva solución.

Hay que tener en cuenta que en general los cambios del canal no son bruscos como se ha forzado en el ejemplo (con objeto de ilustrar la capacidad de adaptación del ecualizador), sino que van variando lentamente a lo largo del tiempo. Por lo tanto, no es habitual encontrar este pico en el que el error crece bruscamente sino que los coeficientes del canal evolucionan a la misma velocidad que lo hacen las variaciones del canal.

El algoritmo LMS descrito en este módulo es el algoritmo adaptativo más sencillo y de los más utilizados en el diseño de ecualizadores adaptativos. No obstante, no es el único algoritmo que existe con estas características. Una variación del algoritmo LMS es el algoritmo *NLMS* (*normalized LMS*), que introduce un término de normalización junto con la  $\mu$ , con lo que pretende resolver algunos de los problemas del algoritmo LMS (donde el valor de  $\mu$  adecuado es sensible a la potencia de la señal recibida). Otro algoritmo iterativo que mejora la convergencia del algoritmo LMS es el algoritmo *RLS* (*recursive least-squares*). Este algoritmo, como el LMS, ahorra la inversión de la matriz de correlación, y si se diseña de manera adecuada, también permite adaptarse a las variaciones del canal. Por último, para terminar el listado de algunos ecualizadores adaptativos, merece la pena mencionar el ecualizador basado en el filtro de Kalman, por tratarse de la mejor implementación de un filtro lineal que minimiza el error cuadrático medio.

#### Lectura complementaria

Ninguno de estos otros ecualizadores adaptativos (NLMS, RLS, Kalman) se van a tratar en este curso. Se puede encontrar información detallada sobre los mismos en la obra siguiente:

**Simon O. Haykin** (2013). *Adaptive filter theory*. Upper Saddle River: Prentice Hall.

### 3. Ecuadores no lineales

Como ya hemos explicado en la introducción, un ecualizador lineal no es el mejor ecualizador que se puede implementar desde un punto de vista de querer minimizar la probabilidad de error en el proceso de detección de la secuencia de símbolos transmitida. En un sistema de comunicaciones digitales con presencia de canal con ISI, el algoritmo óptimo capaz de estimar los símbolos de la secuencia transmitida con la mínima probabilidad de error corresponde al estimador que hace la estimación de los símbolos más probables, dada una secuencia de muestras recibida. Este estimador, o detector, se denomina *estimador MLSE* (del inglés: *maximum likelihood sequence estimator*). En este apartado, trabajamos con detalle el estimador MLSE y presentamos brevemente el ecualizador DFE como otro ejemplo de ecualizador no lineal.

#### 3.1. Estimador de máxima verosimilitud (MLSE)

Formalizando matemáticamente el concepto, si dada una secuencia de datos recibidos  $\mathbf{z}$  (correspondiente a un vector de datos a la salida del filtro adaptado) denotamos con  $p(\mathbf{z}|\mathbf{a})$  la función de verosimilitud (en inglés: *likelihood function*) de los datos recibidos condicionada a haber transmitido una secuencia de símbolos concreta,  $\mathbf{a}$ , el objetivo será encontrar la secuencia de símbolos transmitidos que maximice esta función:

$$\hat{\mathbf{a}} = \arg \max_{\mathbf{a}} p(\mathbf{z}|\mathbf{a}) \quad (26)$$

Si asumimos que el ruido es gaussiano, con muestras de ruido incorreladas, y denotamos con  $K$  la longitud de la secuencia de símbolos transmitidos, la función de densidad de probabilidad condicionada será la expresión (27) (de nuevo asumimos canal, ruido y símbolos reales por simplicidad en la notación), donde  $\sigma_z^2$  es la varianza de las muestras del ruido blanco a la salida del filtro adaptado,  $L$  es la duración del canal en muestras (a tiempo de símbolo),  $a_{l-k}$  son los elementos del vector  $\mathbf{a}$  correspondientes a los símbolos transmitidos y los coeficientes  $c_k$  son las muestras (a tiempo de símbolo) de la respuesta impulsional global que, como se ha visto en el apartado 1, corresponde a la convolución del pulso conformador con la respuesta impulsional del canal y con el filtro adaptado.

$$p(\mathbf{z}|\mathbf{a}) = \frac{1}{(2\pi\sigma_z^2)^{K/2}} \exp \left\{ -\frac{1}{2\sigma_z^2} \sum_{l=1}^K \left| z[l] - \sum_{k=0}^{L-1} c_k a_{l-k} \right|^2 \right\} \quad (27)$$

#### Reflexión

Observemos que la ecuación (26) es idéntica a la utilizada para el diseño del sincronizador (ved el subapartado 2.1 del módulo de sincronización). En ambos casos se intenta determinar (estimar) un cierto valor desconocido siguiendo un criterio que maximice la función de verosimilitud.

Como es habitual en estos problemas de optimización, tomamos el logaritmo de la función de verosimilitud (maximizar la función  $p(\mathbf{z}|\mathbf{a})$  o su logaritmo es equivalente), y después de obviar una serie de constantes que no dependen de la secuencia de símbolos  $\mathbf{a}$ , obtenemos que maximizar la función de verosimilitud es equivalente a minimizar la suma (28):

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\operatorname{argmin}} \left( \sum_{l=1}^K \left| z[l] - \sum_{k=0}^{L-1} c_k a_{l-k} \right|^2 \right) \quad (28)$$

Observemos que la suma anterior equivale a la distancia euclídea entre una secuencia de datos recibidos  $\mathbf{z}$  y la secuencia de símbolos  $\mathbf{a}$  pasada por el canal, y por lo tanto, podemos afirmar que, en presencia de AWGN, maximizar la función de máxima verosimilitud es equivalente a minimizar esta distancia euclídea.

A partir de la ecuación (28), podemos concluir la siguiente manera de proceder: Dada una secuencia de datos recibidos,  $\mathbf{z}$ , se calcula la distancia euclídea para cada una de las secuencias posibles a la salida del canal dependientes de los símbolos  $\mathbf{a}$  y se acaba decidiendo como secuencia transmitida (y, por lo tanto, aquella que minimiza la probabilidad de error) la de distancia mínima. La pregunta es: ¿cuál es la complejidad de este algoritmo? Como ya podéis intuir, la respuesta es que es excesiva, y por lo tanto resulta inaceptable la implementación del algoritmo tal como se ha descrito. En efecto, si los símbolos transmitidos pertenecen a una constelación de dimensión  $M$  y la secuencia transmitida es de longitud  $K$ , el número de secuencias posibles con las que comparar es  $M^K$  (podéis hacer algún cálculo para una BPSK  $M = 2$  o QPSK  $M = 4$  y una longitud de secuencia de  $K = 1.000$  símbolos).

Sin embargo, hay una manera computacionalmente eficiente de resolver este problema. La solución es hacer uso del algoritmo de Viterbi, que como vamos a ver a continuación, permite resolver el problema siguiendo un algoritmo donde la complejidad solo depende exponencialmente de la memoria del canal  $L$  y donde la dimensión de la constelación  $M$  y la longitud de la secuencia transmitida  $K$  afectan linealmente.

### 3.2. Estimador MLSE implementado con el algoritmo de Viterbi

El objetivo del algoritmo de Viterbi será encontrar la secuencia de símbolos que maximiza la función de verosimilitud (en el caso gaussiano equivalentemente minimiza la distancia euclídea), dada una secuencia de datos recibidos. El algoritmo de Viterbi no tiene solo aplicación en la estimación de máxima verosimilitud de secuencias transmitidas en canales con ISI, sino que también se utiliza ampliamente en aplicaciones de otras áreas del procesamiento de señal, como por ejemplo para la decodificación de códigos convolucionales, en algunos métodos para el reconocimiento de voz y en varios esquemas para

la clasificación de patrones. En todos estos problemas, hay un denominador común, que es buscar un camino óptimo (o el camino más probable) dentro de un diagrama de Trellis.

El algoritmo de Viterbi tiene un enunciado formal bastante complejo y los detalles de las demostraciones escapan a los objetivos de este texto. Por eso, vamos a presentar el algoritmo mediante un ejemplo y luego intentaremos generalizar los resultados de modo que quede clara la esencia del algoritmo y el procedimiento que hay que seguir para descodificar una secuencia mediante este.

Supongamos que se transmite una señal binaria  $a_n \in \{-1, +1\}$  por un canal que tiene la misma respuesta impulsional global que la que hemos utilizado para diseñar el ecualizador ZF, esto es:

$$\begin{aligned} c[0] &= +0,8 \\ c[1] &= +0,4 \\ c[2] &= -0,2 \\ c[n] &= 0 \quad n \neq 0, 1, 2 \end{aligned}$$

Para hacer la estimación de la secuencia transmitida es necesario conocer la respuesta impulsional global  $c[n]$  y elaborar un diagrama, que denominaremos **diagrama de Trellis** y que contiene todos los posibles valores a la salida del canal (sin ruido) y las combinaciones de símbolos a la entrada que las generen, tal como detallamos a continuación. El valor de  $c[n]$ , que necesitamos conocer, se puede estimar en el periodo de entrenamiento a partir del envío de una secuencia de datos conocidos y de observar su valor a la salida del filtro adaptado.

Observemos que la respuesta global está formada por tres coeficientes,  $L = 3$ , lo que quiere decir que la memoria es  $L - 1 = 2$ . Esta información puede verse si calculamos la señal a la salida del filtro adaptado en función de los símbolos a la entrada y de la respuesta impulsional global:

$$z[n] = c[0] \cdot a_n + c[1] \cdot a_{n-1} + c[2] \cdot a_{n-2} \quad (29)$$

Como se puede ver, la salida en el instante  $n$  depende del símbolo a la entrada en aquel instante (multiplicado por  $c[0]$ ) más el valor de los dos símbolos anteriores. Por lo tanto, la memoria es de 2 símbolos. A esta memoria, en la notación del algoritmo de Viterbi, la llamaremos *estado* y, como se puede observar, este estado puede tomar en general  $M^{L-1}$  valores (donde  $M$  es la dimensión de la constelación). En nuestro caso,  $M = 2$  y la memoria es también 2; por lo tanto, el número de estados puede ser  $2^2 = 4$ . Observemos que, para un estado (memoria) concreto, la señal a la salida  $z[n]$  puede tomar tantos valores como opciones haya para el símbolo actual  $a_n$  (en nuestro caso 2:  $a_n \in \{-1, +1\}$ ). La tabla siguiente reúne todas las opciones para el caso de

la ecuación (29) (en nuestro caso, cuatro estados para dos posibles entradas da lugar a ocho opciones), y en la tercera columna se indica la notación que utilizaremos en el diagrama de Trellis:

| $a_{n-1}$ | $a_{n-2}$ | Notación en el diagrama de Trellis | $a_n$ | $z[n]$ |
|-----------|-----------|------------------------------------|-------|--------|
| -1        | -1        | --                                 | -1    | -1     |
|           |           |                                    | +1    | 0,6    |
| +1        | -1        | +-                                 | -1    | -0,2   |
|           |           |                                    | +1    | 1,4    |
| -1        | +1        | -+                                 | -1    | -1,4   |
|           |           |                                    | +1    | 0,2    |
| +1        | +1        | ++                                 | -1    | -0,6   |
|           |           |                                    | +1    | 1      |

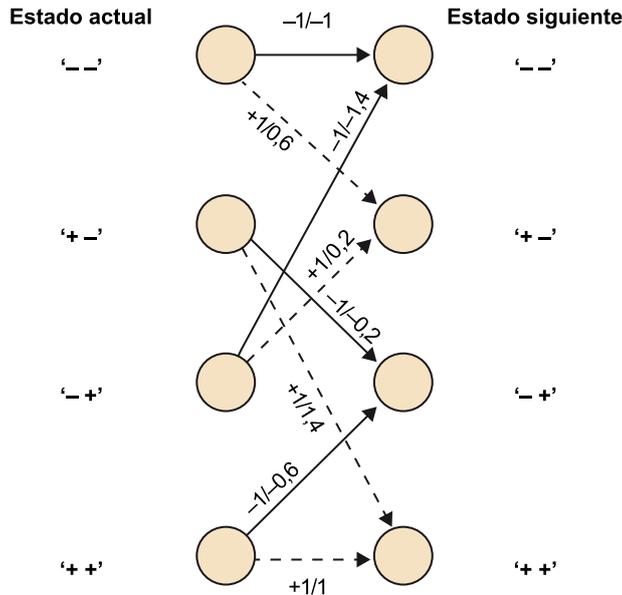
Toda la información de la tabla anterior se puede recoger en un diagrama, llamado **diagrama de Trellis**, que contenga los diferentes estados y para cada estado el número de opciones que se pueden dar de símbolo a la entrada con su salida  $z[n]$  asociada. La figura 17 corresponde a este diagrama de Trellis. A continuación, explicamos cómo se elabora:

- Si el estado del canal (memoria) es '- -' y el símbolo actual es también  $a_n = -1$ , la salida será  $z[n] = -0,8 - 0,4 + 0,2 = -1$  y en la iteración siguiente estaremos de nuevo en el estado '- -'. Toda esta información se resume en la primera flecha del diagrama de Trellis que, saliendo del estado '- -', va al estado '- -', y sobre la flecha se pone la información  $a_n/z[n]$ ; en nuestro caso, vemos -1/-1.
- Si el estado del canal (memoria) es '- -' y el símbolo actual es  $a_n = +1$ , la salida será  $z[n] = +0,8 - 0,4 + 0,2 = 0,6$  y en la iteración siguiente estaremos en el estado '+ -'. Toda esta información se resume en la flecha del diagrama de Trellis que, saliendo del estado '- -', va al estado '+ -', y sobre la flecha se pone la información  $a_n/z[n]$ ; en nuestro caso, vemos +1/0,6.
- Si el estado del canal (memoria) es '+ -' y el símbolo actual es  $a_n = -1$ , la salida será  $z[n] = -0,8 + 0,4 + 0,2 = -0,2$  y en la iteración siguiente estaremos en el estado '- +'. Toda esta información se resume en la flecha del diagrama de Trellis que, saliendo del estado '+ -', va al estado '- +', y sobre la flecha se pone la información  $a_n/z[n]$ ; en nuestro caso, vemos -1/-0,2.
- Si el estado del canal (memoria) es '+ -' y el símbolo actual es  $a_n = +1$ , la salida será  $z[n] = +0,8 + 0,4 + 0,2 = 1,4$  y en la iteración siguiente estaremos en el estado '+ +'. Toda esta información se resume en la flecha del

diagrama de Trellis que, saliendo del estado '+ -', va al estado '+ +', y sobre la flecha se pone la información  $a_n/z[n]$ ; en nuestro caso, vemos  $+1/1,4$ .

- El resto del diagrama se elabora del mismo modo ... (dejamos que lo reviséis vosotros para que comprobéis que habéis entendido el procedimiento).

Figura 17. Ejemplo de diagrama de Trellis



Símbolos  $a_n \in \{-1, +1\}$ , canal  $c_0 = +0,8$ ,  $c_1 = +0,4$ ,  $c_2 = -0,2$ . La línea continua denota que el símbolo transmitido es un +1 y la discontinua, que el símbolo transmitido es un -1.

Una vez tenemos el diagrama de Trellis, estamos en condiciones de utilizar el algoritmo de Viterbi para encontrar la secuencia de símbolos transmitidos a partir de una secuencia recibida. Una clave importante para la estimación es que conocemos el estado inicial del canal, es decir, el valor de los símbolos  $a_{-1}$  y  $a_{-2}$ . Para resolver este punto, se puede asumir que antes de iniciar la transmisión de símbolos de información el transmisor envía dos símbolos conocidos, por ejemplo  $a_{-1} = -1$  y  $a_{-2} = -1$  (cualquier otra opción sería igual de válida), de modo que sabemos que el estado inicial del canal es '- -'.

**Reflexión**

También podríamos considerar que la memoria del canal está vacía (no se ha enviado nada), lo que significaría que  $a_{-1} = 0$  y  $a_{-2} = 0$ . Sin embargo, este caso implicaría alguna pequeña modificación en los  $L - 1$  primeros pasos del Viterbi y no la consideramos para no complicar la explicación inicial.

Supongamos que la secuencia recibida es  $z[n] = \{+0,7, +1,2, -0,5, +0,1, -0,3, +0,6\}$ . Para poder seguir el algoritmo, damos como información adicional que esta secuencia corresponde a la secuencia de símbolos transmitida  $a_n = \{+1, +1, -1, +1, -1, +1\}$  corrompida por el ruido AWGN del canal (la secuencia sin ruido tendría que ser  $\{+0,6, +1,4, -0,6, +0,2, -0,2, +0,2\}$ ). Sin embargo, observemos que toda esta información adicional no está disponible en el receptor, que solo ve  $z[n]$ .

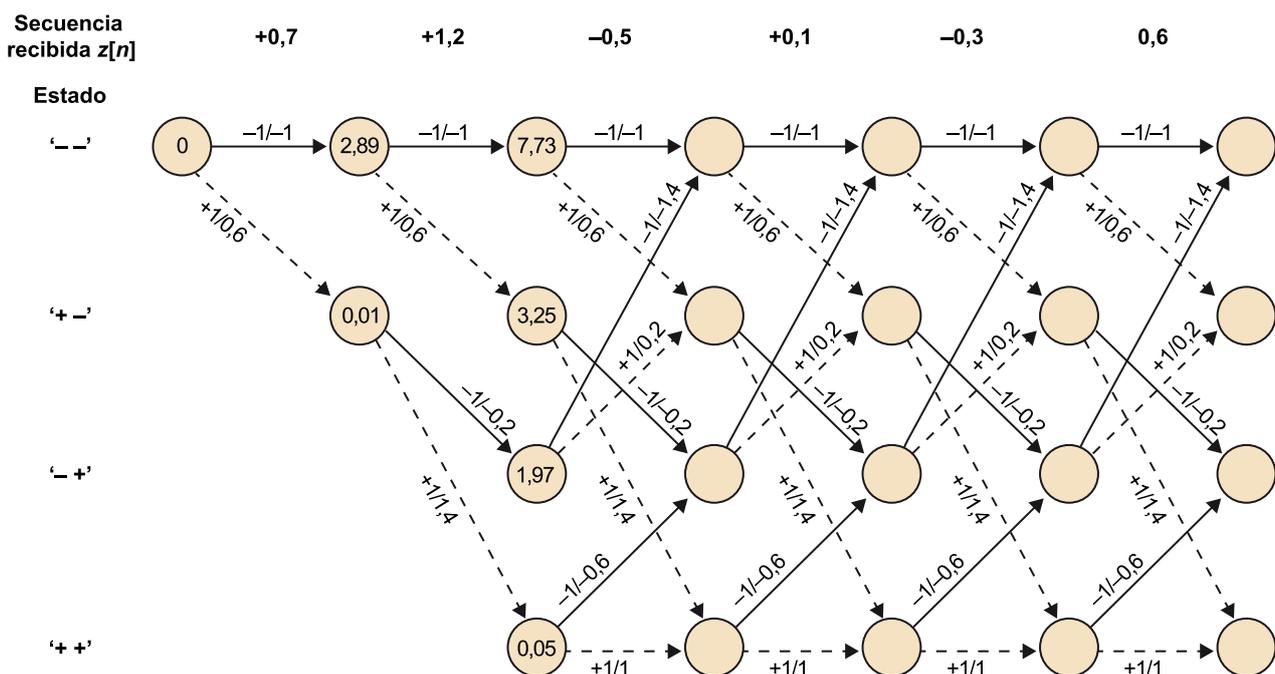
Las primeras etapas del algoritmo, explicadas a continuación, se muestran en la figura 18. Para empezar con el proceso de estimación de la secuencia, vemos que desde el estado inicial '- -' podemos llegar a los estados '- -' o '+ -' (dependiendo del valor que haya tomado  $a_0$ ). En el primer caso, la señal a la salida

del canal tendría que ser  $-1$ , y en el segundo caso, tendría que ser  $0,6$ . Como vemos, lo que hemos recibido  $z[0] = +0,7$  no coincide con lo que tendríamos que haber recibido, ni en un caso ni en el otro, debido al ruido. En ambos casos hay una cierta diferencia. Para continuar con el procedimiento de estimación, calculamos la distancia euclídea entre lo que hemos recibido y cada una de las dos opciones, tal como se indica en la ecuación (28), y anotamos los valores de ambas distancias en los espacios correspondientes (que denominaremos *nodos*). Así, observamos que  $d_{Euclídea}(0,7; -1) = 2,89$  y  $d_{Euclídea}(0,7; 0,6) = 0,01$ .

En la segunda etapa, dado que estamos intentando evaluar todas las posibles secuencias transmitidas, empezamos a concatenar hipótesis. Así, si partimos del estado ‘ $--$ ’, que sucede si  $a_0$  hubiera sido  $a_0 = -1$ , podemos llegar a los estados ‘ $--$ ’ o ‘ $+ -$ ’ (dependiendo del valor que hubiera tomado  $a_1$ ), y si partimos del estado ‘ $+ -$ ’, que sucede si  $a_0$  hubiera sido  $a_0 = +1$ , podemos llegar a los estados ‘ $- +$ ’ o ‘ $+ +$ ’ (de nuevo dependiendo del valor que hubiera tomado  $a_1$ ). Observemos que con estas cuatro hipótesis ya podemos haber alcanzado cualquiera de los cuatro estados del codificador, que corresponderían a las cuatro posibles maneras de iniciar la secuencia transmitida considerando todas las opciones para  $a_0$  y  $a_1$ .

Ahora procedemos a rotular cada nodo en función de la distancia euclídea. Primero, calculamos las cuatro distancias euclídeas entre lo que hemos recibido,  $z[1] = +1,2$ , y lo que tendríamos que haber recibido en cada caso. Una vez calculadas, el valor que anotamos en el nodo no es el valor de la distancia euclídea sino el acumulado de la distancia desde el estado inicial. Así, en el primer caso, si  $d_{Euclídea}(1,2; -1) = 4,84$ , lo que anotamos es  $7,73 = 4,84 + 2,89$ , donde  $2,89$  es la métrica acumulada hasta el nodo anterior.

Figura 18. Dos primeras etapas del algoritmo de Viterbi



De modo análogo, podemos calcular el resto de distancias. Esta manera de proceder es una implementación directa de la ecuación (28) que, como podemos ver, nos dice que tenemos que calcular el sumatorio sobre  $l$  de todas las distancias euclídeas (ahora estamos en el paso  $l = 2$ ) para todas las secuencias transmitidas posibles, hasta ahora cuatro posibles valores puesto que solo hemos hecho hipótesis sobre  $a_0$  y  $a_1$ . Así, los cuatro valores que hay en los cuatro nodos correspondientes en la última iteración se obtienen a partir de cada una de las cuatro opciones para  $a_0$  y  $a_1$  según la ecuación (30):

$$E = (\mathcal{Z}[0] - (c_0 a_0 + c_1 a_{-1} + c_2 a_{-2}))^2 + (\mathcal{Z}[1] - (c_0 a_1 + c_1 a_0 + c_2 a_{-1}))^2 \quad (30)$$

En este punto, cuando ya hemos alcanzado todos los posibles estados, es donde empieza el algoritmo de Viterbi propiamente dicho. Observemos que si seguimos procediendo como hasta ahora, en la iteración siguiente tendríamos ocho valores posibles de distancia acumulada correspondientes a las ocho opciones para  $a_0$ ,  $a_1$  y  $a_2$ ; a continuación tendríamos 16, 32, 64... Precisamente, este crecimiento exponencial es el que queremos evitar con el algoritmo de Viterbi. Este algoritmo propone que a partir de los cuatro estados presentes se calculen las ocho distancias euclídeas y que de entre las posibles opciones solo se tomen aquellas que tienen menor distancia acumulada, y se descarten las demás. Veámoslo en nuestro ejemplo y después volveremos a ello para interpretar lo que hemos hecho. La figura 19 ilustra esta tercera etapa.

Podemos alcanzar el estado ‘- -’ a partir de los estados ‘- -’ o ‘- +’ en la etapa previa:

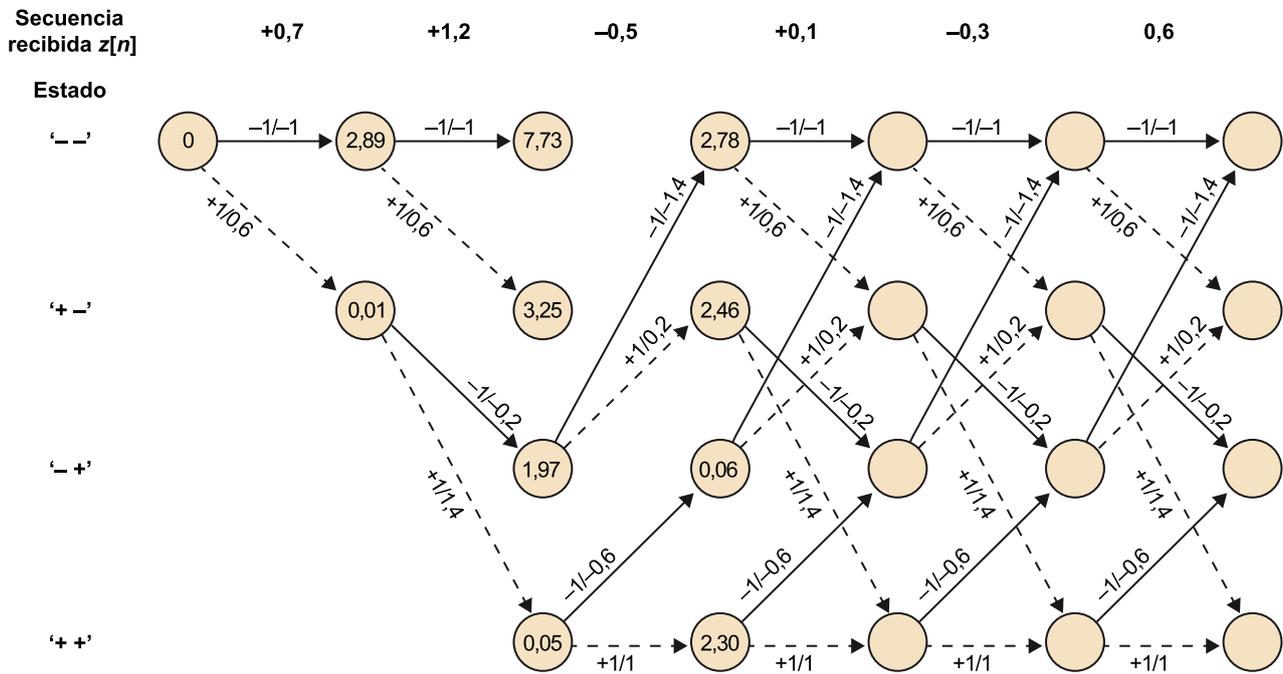
- Si llegamos desde el estado ‘- -’ la señal a la salida del canal tendría que ser -1; así, procedemos a calcular la distancia euclídea acumulada por este camino:  $7,73 + d_{Euclídea}(-0,5, -1) = 7,98$ .
- Análogamente, si llegamos desde el estado ‘- +’, la señal a la salida del canal tendría que ser -1,4; calculamos asimismo la distancia euclídea acumulada siguiendo este segundo camino:  $1,97 + d_{Euclídea}(-0,5, -1,4) = 2,78$ .

De este modo, observamos que si seguimos el primer camino (camino que supondría que  $a_0 = -1$ ,  $a_1 = -1$  y  $a_2 = -1$ ), tendríamos una distancia euclídea acumulada de 7,98; mientras que si seguimos el segundo camino (camino que supondría  $a_0 = +1$ ,  $a_1 = -1$  y  $a_2 = -1$ ), tendríamos una distancia euclídea acumulada mucho menor. Así pues, es el momento de descartar el primer camino y mantener la hipótesis de que la secuencia transmitida es más probable que sea la correspondiente al segundo camino. Como resultado final, nos quedamos con la rama procedente del estado ‘- +’ y rotulamos el nodo con la distancia euclídea acumulada por este camino, es decir, 2,78.

Hagamos la misma operación con el resto de estados de la tercera etapa con la misma idea en mente: de los dos posibles caminos de llegada a un estado, solamente tenemos que considerar aquel que presente una distancia acumulada

menor, y descartar el otro. El resultado final tras efectuar todas estas operaciones para la etapa tres se muestra en la figura 19, donde se han eliminado las flechas procedentes de los nodos que han sido descartados.

Figura 19. Tercera etapa del algoritmo de Viterbi



La idea básica del esquema de Viterbi es que para un estado en la etapa  $l$  al que se puede llegar desde dos estados (en un caso genérico serían  $M$ ) de la etapa anterior ( $l - 1$ ), tenemos que considerar únicamente el camino que presenta la menor distancia. La demostración de que este procedimiento es óptimo no es evidente, pero los detalles se pueden encontrar en un artículo clásico (Viterbi, 1967) y en algunos textos avanzados de comunicaciones.

**Actividad**

Os proponemos que verifiquéis los detalles de cómo se van obteniendo los estados en las diversas etapas del algoritmo de Viterbi.

Aunque la probabilidad de que suceda es muy baja, en algunos casos (en concreto, cuando se trabaja con datos cuantificados) podría ocurrir que dos ramas que llegan a un nodo lo hicieran con el mismo valor de distancia acumulada. En ese caso, el algoritmo de Viterbi permite elegir de manera arbitraria cualquiera de las dos ramas. En efecto, es posible demostrar que no hay manera de solucionar este tipo de ambigüedades, hecho por el que ambas soluciones son equivalentes (igual de buenas o malas).

En las figuras 20 a 22 se muestran las etapas sucesivas que se van obteniendo al aplicar el algoritmo en las etapas 4, 5 y 6, respectivamente. Se presentan de manera individual para que podáis comprobar que habéis entendido el procedimiento.

Figura 20. Cuarta etapa del algoritmo de Viterbi

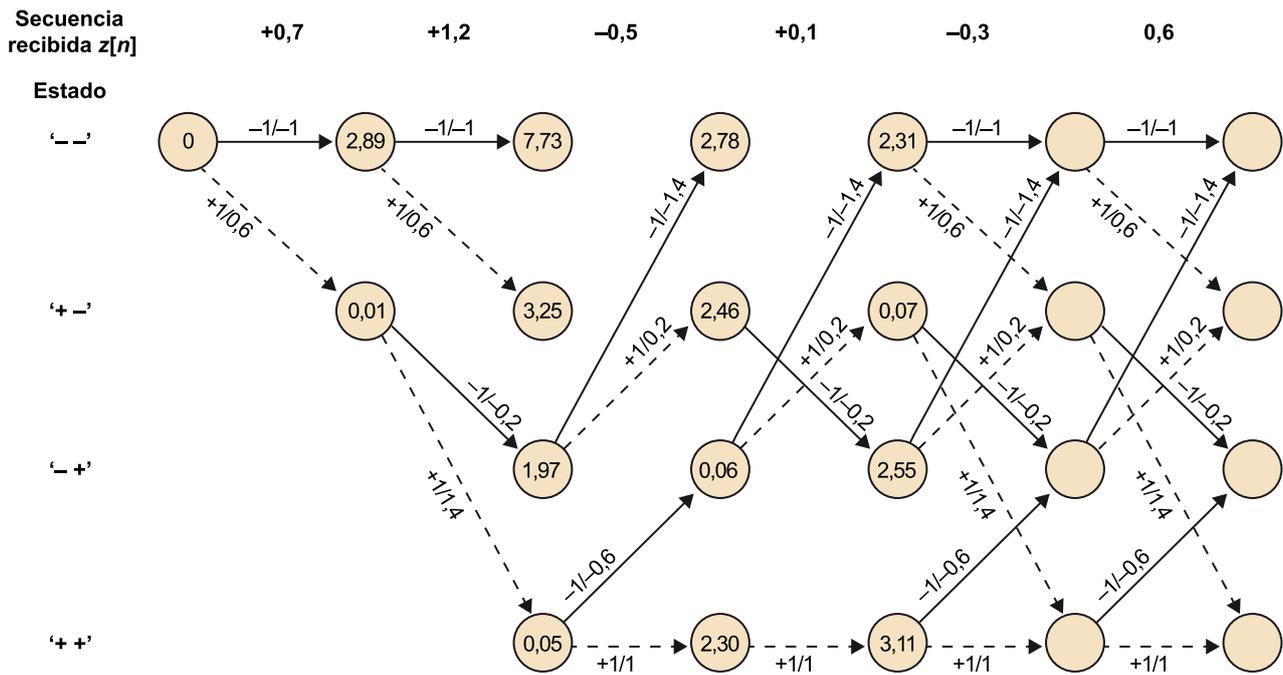


Figura 21. Quinta etapa del algoritmo de Viterbi

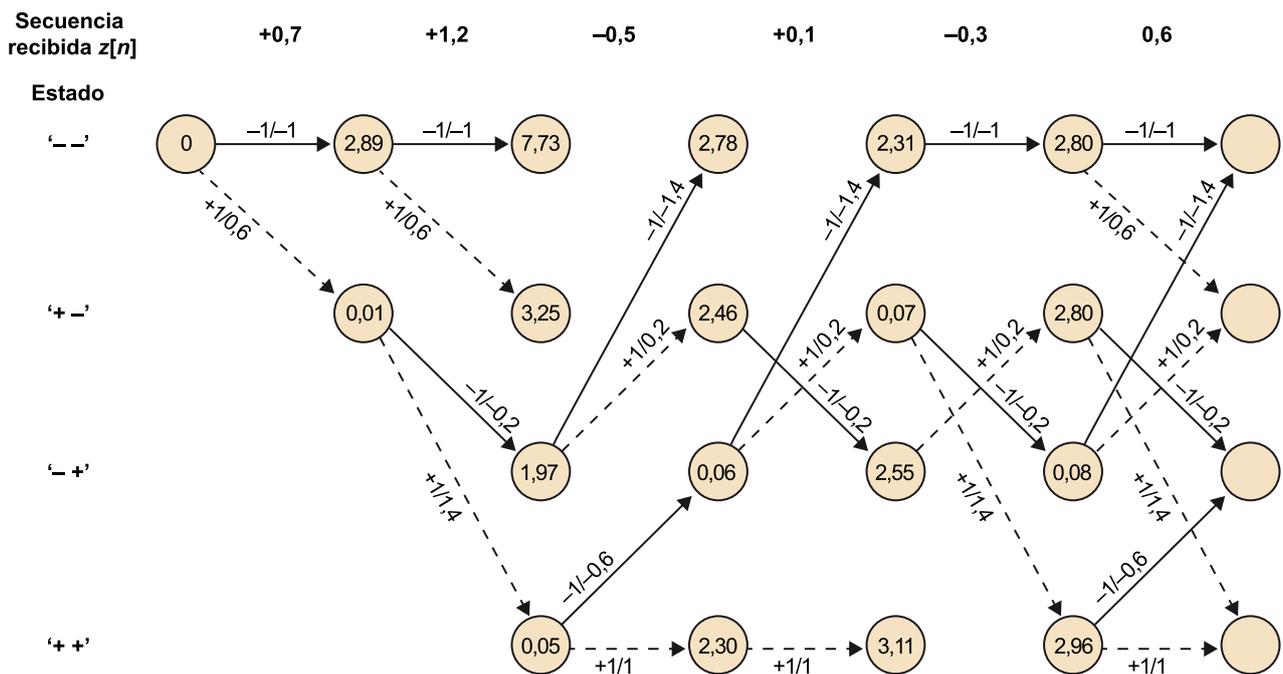
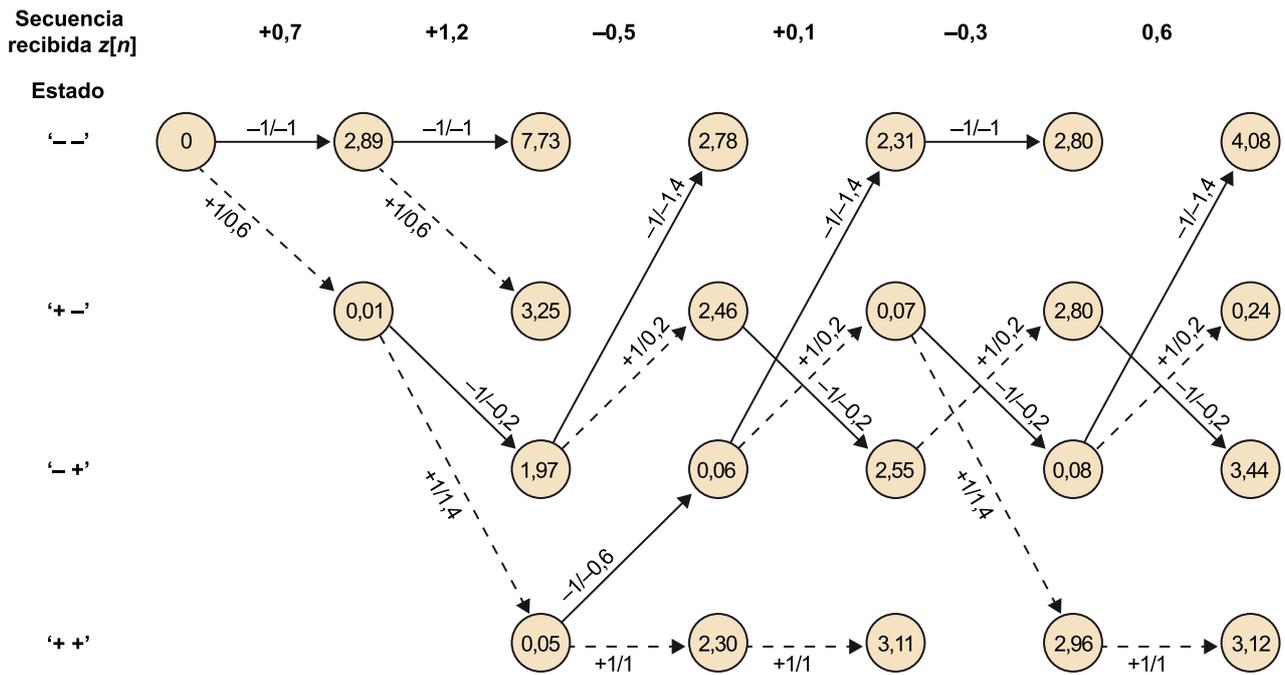
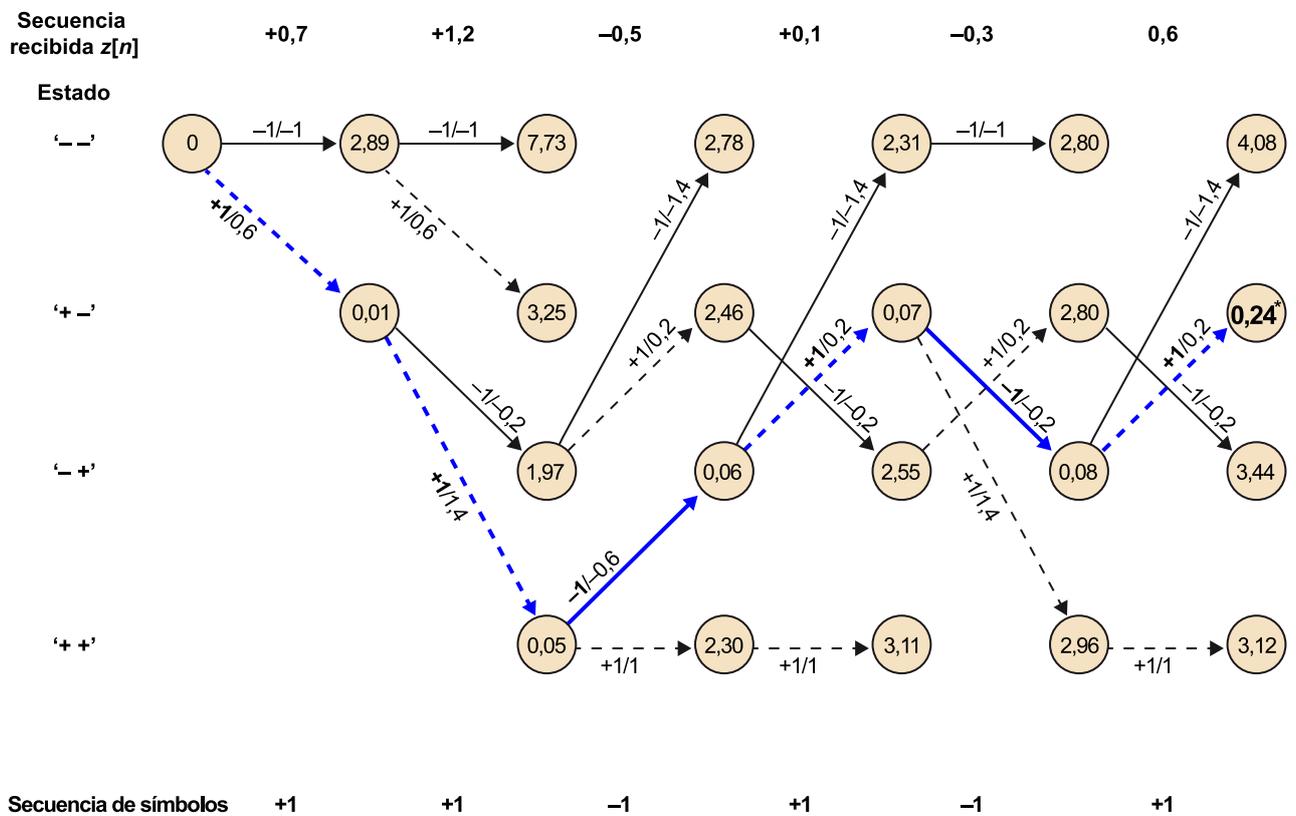


Figura 22. Sexta (y última) etapa del algoritmo de Viterbi



Una vez se llega al final de la secuencia de datos recibidos, es el momento de tomar decisiones sobre cuál es la secuencia de símbolos que se ha transmitido.

Figura 23. Decisión del camino y de la secuencia de símbolos transmitida con el algoritmo de Viterbi



\* Mínima distancia euclídea acumulada.

Fijaos en que, aunque en algún momento puede parecer evidente cuáles han sido los primeros símbolos transmitidos después de haber avanzado algunas etapas, hasta el final de la secuencia no se inicia el proceso de decisión de la secuencia transmitida. La manera de proceder es muy sencilla: se identifica el nodo de la etapa final con una distancia euclídea acumulada menor y se toma el camino que llega a aquel nodo como el camino “ganador”. Así, desde el final se va deshaciendo el camino hasta llegar al estado inicial, y una vez identificado el camino, se identifica la secuencia de símbolos asociada a este camino.

### Reflexión

En este ejemplo de implementación del algoritmo de Viterbi, siempre hemos considerado la minimización de la distancia euclídea como parámetro o métrica para hacer la optimización. Por esta razón, en cada etapa del algoritmo de Viterbi hemos calculado la distancia euclídea entre los datos recibidos y cada una de las opciones que se habrían podido recibir, y hemos anotado en los nodos los valores de las distancias euclídeas acumuladas. Pero, en general, en la implementación del algoritmo de Viterbi se puede trabajar con otras métricas. Observamos, por ejemplo, que si en la ecuación (27) no hubiéramos tomado el logaritmo de la función de verosimilitud que nos ha dado lugar al problema de optimización (28), también habríamos podido resolver el problema buscando la secuencia de símbolos transmitida que habría maximizado la función de verosimilitud de la ecuación (27). En este caso, la métrica que habríamos tenido que calcular en cada paso habría sido la (31), y la métrica acumulada se hubiera obtenido como el productorio de métricas, en lugar del sumatorio como hemos hecho en el ejemplo anterior (fórmula (32)):

$$\Lambda(z[l]|a_l \dots a_{l-(L+1)}) = \frac{1}{\sqrt{(2\pi\sigma_z^2)^L}} \exp\left\{-\frac{1}{2\sigma_z^2} \left|z[l] - \sum_{k=0}^{L-1} c_k a_{l-k}\right|^2\right\} \quad (31)$$

$$p(z[1] \dots z[L]|a_1 \dots a_1) = \prod_{l=1}^L \Lambda(z[l]|a_l \dots a_{l-(L+1)}) \quad (32)$$

Observemos que cuando llegamos a la última etapa  $l = K$ , la expresión acumulada anterior coincidirá con la función de verosimilitud de la ecuación (27). Un último aspecto que tendríamos que tener en cuenta con esta métrica es que en este caso buscaríamos la maximización (y no la minimización, como hemos hecho en el ejemplo), y por lo tanto en cada etapa tendríamos que mantener el camino con la métrica máxima, y al final no seleccionar como ganador el camino con distancia mínima, sino con métrica máxima.

Resumiendo, la algorítmica presentada con el ejemplo anterior describe la manera de proceder con el algoritmo de Viterbi, que básicamente se reduce a evaluar en cada etapa una métrica que puede variar en función de la aplicación (distancia euclídea, función de verosimilitud, distancia de Hamming, entre otros), una métrica acumulada y una selección de caminos que minimiza o maximiza (según el caso) esta métrica.

### 3.2.1. Inconvenientes del algoritmo de Viterbi

Aunque el algoritmo de Viterbi es ampliamente utilizado y en muchos casos no dispone de alternativa práctica para resolver determinados problemas, podemos encontrarle algunos inconvenientes, que enumeramos a continuación:

**1) Latencia elevada.** El primer inconveniente que presenta es la elevada latencia que el algoritmo introduce en la señal. Como hemos visto en el ejemplo, hasta que no se reciben las  $K$  muestras (que corresponden a la secuencia completa) no se inicia el proceso de decisión para seleccionar la secuencia que se ha transmitido. Eso introduce un retardo elevado en la comunicación, que en la mayoría de casos es inaceptable (imaginemos la transmisión continua de una conversación de voz), y cuando es aceptable, implica la necesidad de disponer de mucha memoria para poder almacenar toda la información de todas las métricas acumuladas en cada una de las etapas: en total  $K \cdot M^{L-1}$  (imaginemos la transmisión de una señal de voz donde  $K$  sería el número de muestras de la señal de voz).

Como alternativa a este problema, se propone no esperar a la última etapa para decidir sobre el primer símbolo enviado. Esta modificación sobre el algoritmo de Viterbi original fija una cierta longitud  $Q$  de modo que después de  $Q$  etapas se pasa a decidir sobre el primer símbolo transmitido del mismo modo que se describe en la figura 23. Es importante destacar que solo se decide sobre el primer símbolo, de modo que en la etapa  $Q + 1$  se decidirá sobre el segundo, en la  $Q + 2$  sobre el tercero y así, sucesivamente.

Como vemos, esta solución fija una ventana deslizante a lo largo del tiempo con la que la latencia queda reducida al tiempo necesario para recibir  $Q$  muestras y la memoria del algoritmo también queda fijada en  $Q \cdot M^{L-1}$ , con independencia del número total de símbolos que se transmiten. Los valores típicos de  $Q$  se determinan en función de  $L$  y acostumbran a ser del orden de  $Q = 6, 7$  o  $9$  veces  $L$ . Como podéis imaginar, después de este tiempo el valor del primer símbolo está casi determinado. Ahora bien, esta modificación del algoritmo de Viterbi hace que deje de ser óptimo, ya que no responde fielmente a la finalidad de encontrar la secuencia de máxima verosimilitud. A pesar de ello, para valores de  $Q$  grandes, la diferencia en términos de probabilidad de error respecto de la solución óptima es baja (menor, cuanto mayor sea la  $Q$ ) y aumenta el beneficio en cuanto a reducción de latencia y memoria.

## **2) Complejidad de implementación para canales con mucha memoria.**

Aunque el algoritmo de Viterbi reduce la complejidad, al resultar esta insensible a la longitud de la secuencia transmitida  $K$ , continúa existiendo el problema de que la complejidad del algoritmo crece exponencialmente con la memoria del canal  $L - 1$  (recordemos que el número de estados depende de  $L$ ) y del número de bits por símbolo. Por lo tanto, un ecualizador no lineal basado en este esquema solo será práctico cuando la constelación sea reducida y la memoria del canal sea pequeña ( $L$  pequeña). Cuando estas condiciones no se cumplan, habrá que recurrir o bien a los ecualizadores lineales o bien a otras opciones, como por ejemplo la que se presenta a continuación.

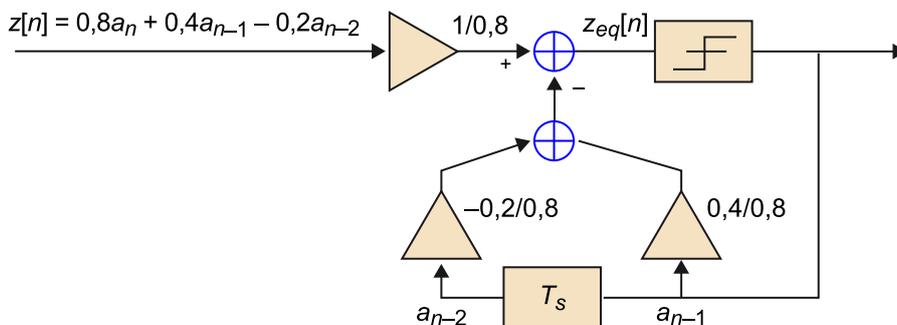
### 3.3. Ecuador no lineal. Ecuador con realimentación de decisiones (DFE)

Otro ecualizador adaptativo no lineal que se puede plantear como alternativa entre el ecualizador óptimo basado en MLSE (de gran complejidad) y el ecualizador lineal (de menores prestaciones) es el ecualizador con realimentación de decisiones (*DFE*, del inglés: *decision feedback equalizer*). No es el objetivo de este módulo trabajar este tipo de ecualizador, y por lo tanto solo dedicamos al mismo una breve reseña dentro del apartado de ecualizadores no lineales con objeto de poder dejarlo bien clasificado como solución entre el ecualizador lineal y el MLSE.

El ecualizador DFE explota el hecho de que los símbolos anteriores a un instante dado son los que principalmente contribuyen al término de ISI (solo en casos de canales no causales también los símbolos futuros provocan ISI). Por lo tanto, si a partir de las decisiones de los símbolos anteriores se tiene conocimiento de cuáles han sido estos símbolos, y a la vez se conoce la respuesta impulsional global,  $c(t)$ , se puede regenerar (parcialmente) el término de ISI y eliminarlo de la señal antes de tomar la nueva decisión (que idealmente ya estaría libre de ISI).

De manera muy intuitiva, intentamos resumir la idea en el ejemplo siguiente (figura 24), que parte del mismo canal (causal) empleado antes, donde para ilustrar la idea asumimos una vez más que no hay ruido.

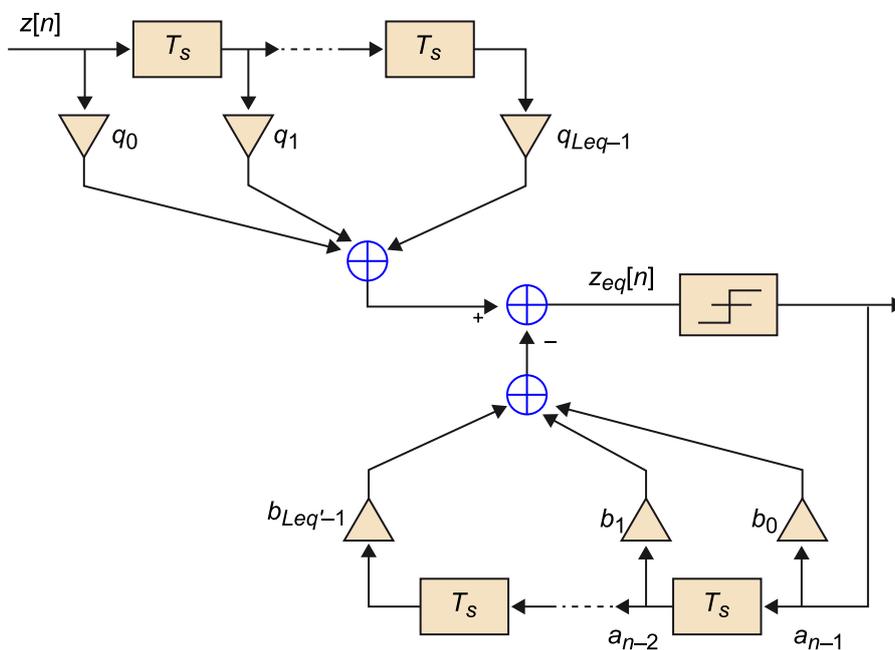
Figura 24. Ejemplo de un ecualizador DFE para el canal  $c_0 = +0,8$ ,  $c_1 = +0,4$ ,  $c_2 = -0,2$



El ecualizador es no lineal porque la función de decisión que decide sobre los símbolos anteriores (entre un conjunto de niveles discretos) es no lineal. Una vez se ha decidido el símbolo, una estructura de filtro lineal con realimentación puede calcular el efecto de la ISI tal como muestra la figura 24. Como en el caso del ecualizador dirigido por decisiones (DD), este ecualizador solo es factible cuando la probabilidad de error es baja, puesto que en caso contrario, si las decisiones son malas, en lugar de reducir el término de ISI al intentar eliminarlo, lo que haríamos sería introducir más error.

La figura 25 muestra un esquema general del ecualizador DFE donde se pueden ver dos estructuras lineales: la inferior, con coeficientes  $b_0, \dots, b_{L_{eq}-1}$ , tiene como entrada las decisiones anteriores, y por lo tanto trabaja en modo realimentado (en inglés se denomina *feedback filter*); y la superior, con coeficientes  $q_0, \dots, q_{L_{eq}-1}$ , tiene como entrada la salida del filtro adaptado, y por lo tanto actúa del mismo modo que los filtros lineales antes vistos (en inglés se denomina *feedforward filter*). En el caso de canales causales y sin tener en cuenta el ruido, solo sería necesario el filtro realimentado para eliminar la ISI, tal como se ha ilustrado en el ejemplo de la figura 24. Ahora bien, si el canal es no causal (no toda la ISI se debe a símbolos anteriores), o en presencia de ruido, el filtro lineal superior es necesario para reducir la enfatización de ruido (el mismo fenómeno que se había observado para el ecualizador forzador de ceros).

Figura 25. Esquema de un ecualizador DFE



Como en el caso del ecualizador lineal, según el criterio que se siga para encontrar los coeficientes del ecualizador, podríamos estar ante un criterio forzador de ceros o un criterio MMSE. Asimismo, también se podría implementar el ecualizador DFE adaptativo si el criterio de diseño de los coeficientes se hace mediante un algoritmo de este tipo. Como ya se ha indicado, no es el objetivo de este módulo entrar en más detalles sobre el diseño de los coeficientes de este ecualizador, y por lo tanto dejamos en este punto la presentación del ecualizador DFE.

## Resumen

En este módulo hemos estudiado el problema de la ecualización de canal proponiendo y comparando varios tipos de ecualizadores. Como ya se ha tratado en otro módulo y se ha destacado al inicio de este, la ecualización de canal se aplica a los casos en los que el canal es dispersivo en el tiempo y su respuesta impulsional varía de manera lo bastante lenta como para poder identificar el canal (o la correlación de la señal a la salida del filtro adaptado) en una primera fase y utilizar el resultado de esta identificación en una segunda fase para diseñar el ecualizador y procesar los datos recibidos. Si el canal varía excesivamente rápido, el diseño del ecualizador siempre estará desadaptado en el canal real, puesto que en el intervalo de tiempo entre la fase de identificación y la de aplicación del ecualizador, el canal habrá variado demasiado y el ecualizador no cumplirá su función.

Para los casos en los que el canal no varía, o lo hace de manera muy lenta como para considerarlo constante para un bloque de datos lo suficientemente largo, se han presentado en el módulo el ecualizador forzador de ceros (ZF), el ecualizador de mínimo error cuadrático medio (MMSE) y el estimador de máxima verosimilitud (MLSE). Para los casos en los que el canal varía a una velocidad lo bastante razonable como para poder seguir sus variaciones, se han presentado los ecualizadores adaptativos, y en concreto, el ecualizador LMS.

A partir de la clasificación de los ecualizadores que hemos hecho en el primer apartado, se ha visto que estos pueden dividirse en lineales y no lineales. La solución óptima corresponde a un ecualizador no lineal que tiene por objetivo encontrar la secuencia más verosímil a partir de una secuencia de datos recibidos. Este ecualizador, presentado en el último apartado, corresponde al ecualizador de máxima verosimilitud (MLSE). Por desgracia, este ecualizador óptimo presenta una complejidad que en algunos casos resulta inabordable pese a que utiliza el algoritmo de Viterbi para la estimación de la secuencia más verosímil. También hemos aprendido el funcionamiento del algoritmo de Viterbi a partir de un ejemplo sencillo.

Cuando el objetivo es diseñar ecualizadores que no sean excesivamente complejos, a menudo se recurre a ecualizadores lineales, mucho más sencillos de implementar aunque con peores prestaciones. El primer ecualizador que se ha presentado ha sido el ecualizador ZF. Este ecualizador resulta muy fácil de diseñar conceptualmente, aunque su uso queda restringido a escenarios de alta SNR, puesto que al no haber sido diseñado teniendo en cuenta el término de ruido, presenta el grave problema de amplificar el ruido a la salida. Como alternativa al ecualizador ZF, se ha introducido el ecualizador MMSE, que resuel-

ve los problemas del forzador de ceros al tener en cuenta el término de ruido en su diseño. Como se ha dicho (y se propone evaluar en un ejercicio), este ecualizador tiende a la solución del forzador de ceros en ausencia de ruido.

Como ejemplo de algoritmo adaptativo se ha presentado con detalle el ecualizador LMS, que aunque corresponde al más sencillo de los ecualizadores adaptativos, sus prestaciones son lo bastante buenas como para ser utilizado en muchos problemas reales. Este ecualizador capaz de seguir las variaciones del canal solo presenta el problema de tener un tiempo de convergencia largo.

Para completar el módulo de ecualización, se han introducido sin demasiado detalle otros ecualizadores que se utilizan de manera común. Así, hemos citado el ecualizador basado en el algoritmo RLS y el ecualizador basado en el filtro de Kalman como otros ejemplos de ecualizadores adaptativos; así como el ecualizador DFE como otro ejemplo de ecualizador no lineal. Tal como podéis intuir, la lista de ecualizadores no termina en los que se han presentado en el módulo, si bien se han intentado incluir los más habituales.

No se ha hecho ninguna mención a otras estrategias para luchar contra el canal, como puede ser la precodificación de los datos en transmisión, que puede llevarse a cabo cuando el transmisor tiene conocimiento de la respuesta del canal. Estos tipos de soluciones requieren que el receptor estime e informe al transmisor (a través de un canal de regreso) del canal que verán los datos. El resultado son soluciones que, bien diseñadas, hacen un uso más eficiente de la potencia, ya que permiten que el transmisor conforme la forma de onda de la señal para que esta llegue al receptor en condiciones óptimas.

Al contrario de lo que se podría pensar *a priori*, estos esquemas no refuerzan con más potencia las frecuencias en las que el canal presenta más atenuación, sino que las técnicas óptimas de precodificación trabajan en sentido contrario. La manera óptima de repartir la potencia es hacerlo de modo que no se inyecte señal en aquellas frecuencias en las que el canal presenta fuertes atenuaciones (sería malgastar el recurso potencia), y a cambio se destine esta potencia a reforzar la señal en aquellas frecuencias en las que el canal es más benigno para sacar mayor provecho de la potencia.

Para terminar el módulo, os recomendamos que volváis a revisar el listado de objetivos para ver si se han alcanzado y que hagáis las actividades que hay a continuación para practicar los conceptos presentados a lo largo del módulo.

## Actividades

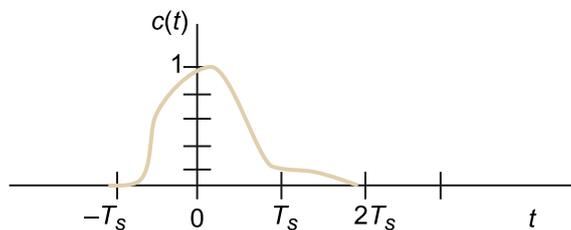
1. Sea un sistema de comunicaciones en el que el pulso conformador  $p(t)$  es un pulso rectangular de duración  $T_s$  segundos.

a) Indicad cuál será la respuesta impulsional global  $c(t)$  y su versión muestreada a tiempo de símbolo  $c[n]$  en el supuesto de que el canal sea ideal (asumid que el sistema empieza a muestrear en  $t = 0$ ).

b) Repetid el apartado anterior suponiendo que el canal tiene una respuesta impulsional  $h(t) = \delta(t) + 0,2 \cdot \delta(t - T_s)$ .

2. La señal de la figura 26 corresponde a la respuesta impulsional global  $c(t)$  de un sistema de comunicaciones en presencia de canal no ideal. Si el receptor empieza a muestrear en  $t = 0$ , encontrad la respuesta impulsional discreta  $c[n]$  y dibujad la constelación que habría a la entrada del ecualizador en ausencia de ruido si los símbolos transmitidos corresponden a una QPSK de potencia unidad.

Figura 26



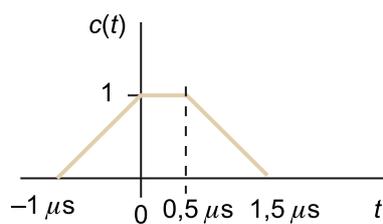
3. La figura 27 representa la respuesta impulsional global de un sistema digital a la salida del filtro adaptado. Si la velocidad de transmisión es de  $r_s = 10^6$  símbolos por segundo:

a) Calculad la potencia de ISI (utilizad la fórmula (33)) si la señal se empieza a muestrear en  $t = 0$ .

$$P_{ISI} = \sum_{n \neq 0} |c_n|^2 \quad (33)$$

b) Determinad cuál tendría que ser el instante en el que habría que iniciar el muestreo para minimizar la potencia de ISI (utilizad de nuevo la fórmula (33)).

Figura 27



4. En un sistema de comunicaciones con canal no ideal se mide la respuesta impulsional global  $c[n] = \delta(n) + \delta(n - 1)$ .

a) Dibujad el módulo de la respuesta en frecuencia de un filtro ecualizador ZF ideal si sabemos que la respuesta global del canal es  $C(F) = 2 \cdot \cos(\pi F) e^{-j\pi F}$ .

b) Diseñad el ecualizador ZF de tres coeficientes considerando que no se permite ningún retardo en la señal a la salida.

c) Para este ecualizador ZF, encontrad el valor de la ISI residual.

d) Para el mismo ecualizador ZF, si el ruido a la entrada tiene una varianza  $\sigma_w^2$ , encontrad el factor de amplificación del ruido a la salida del ecualizador.

5. Para un sistema con respuesta impulsional global  $c[n]$  definido por los coeficientes siguientes:

$$\begin{aligned}d[0] &= +0,8 \\d[1] &= +0,4 \\d[2] &= -0,2 \\d[n] &= 0 \quad n \neq 0, 1, 2\end{aligned}$$

a) Encontrad el ecualizador ZF de 4 coeficientes, si no se permite ningún retardo en la salida del ecualizador (es decir, forzamos que el 1 esté en la primera posición).

b) Repetid el diseño, si ahora se permiten retardos de  $d = 1$ ,  $d = 2$  y  $d = 3$  (diseñad en total tres ecualizadores ZF diferentes).

c) Para cada uno de los cuatro ecualizadores obtenidos entre los apartados a y b, encontrad la potencia de ISI residual (utilizad la fórmula (33)) y la potencia de ruido, si el ruido a la entrada tiene una varianza  $\sigma_w^2 = 1$ .

d) Teniendo en cuenta que la medida final que nos condicionará la probabilidad de error será el cociente entre potencia de señal y potencia de ruido más ISI:  $SINR = \frac{P_{Señal}}{P_{Ruido} + P_{ISI}}$ , indicad cuál de los cuatro ecualizadores es más apropiado.

6. Verificad el cálculo de la matriz  $\mathbf{R}_{zz}$  y del vector  $\mathbf{R}_{za}$  para el caso 1: “Respuesta impulsional global  $c[n]$  conocida” del ecualizador MMSE (podéis ver el caso 1: “Respuesta impulsional global  $c[n]$  conocida” en el subapartado 2.2.1 de este módulo didáctico).

7. Diseñad el filtro ecualizador MMSE de ocho coeficientes de un sistema que tiene la respuesta impulsional global  $c[n]$  que se indica en el ejercicio de autoevaluación 5 y confirmad que, en ausencia de ruido, la solución es la misma que la obtenida para el caso del ecualizador ZF obtenido en el ejemplo del subapartado 2.1. Utilizad algún tipo de soporte informático para encontrar la solución.

8. Dado un sistema de comunicaciones con canal no ideal, cuando se envía la secuencia conocida  $a_n = \{+1, +1, -1, -1, +1\}$ , se obtiene, después de muestrear la salida del filtro adaptado, la secuencia  $z[n] = \{+0,9; +1,1; -0,7; -1,1; +0,7; +0,2\}$ . Calculad las primeras iteraciones de un ecualizador adaptativo basado en LMS, si el paso de adaptación es  $\mu = 0,1$  y si el algoritmo se inicializa con el vector  $\mathbf{q}^0$  siguiente:

$$\mathbf{q}^0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

9. Verificad el algoritmo de Viterbi de las figuras 20 a 22.

10. Dado un sistema de comunicaciones con canal no ideal, cuando se envía la secuencia conocida  $a_n = \{+1, +1, -1, -1, +1\}$ , se obtiene, después de muestrear la salida del filtro adaptado, la secuencia  $z[n] = \{+0,59, +1,21, -0,71, -1,36, +0,79\}$ . Si sabemos que la respuesta impulsional global es  $c[n] = 0,9 \cdot \delta(n) + 0,3 \cdot \delta(n-1) - 0,1 \cdot \delta(n-2)$  y que el transmisor envía los símbolos  $a_{-1} = -1$  y  $a_{-2} = -1$  antes de iniciar la transmisión de símbolos de información con el fin de conocer el estado inicial del algoritmo de Viterbi, encontrad la secuencia transmitida aplicando el estimador MLSE.

## Bibliografía

### Bibliografía básica

**Proakis, J. G.; Salehi, M.** (2002). *Communication systems engineering* (2.<sup>a</sup> ed.). Nueva York: Prentice Hall.

### Bibliografía complementaria

**Proakis, J.** (2008). *Digital communications* (5.<sup>a</sup> ed.). Nueva York: McGraw Hill.

**Benedetto, S.; Biglieri, E.** (1999). *Principles of digital transmission*. Nueva York: Kluwer Academic Press / Plenum Publishers.

**Sklar, B.** (2001). *Digital communications: fundamentals and applications* (2.<sup>a</sup> ed.). Upper Saddle River: Prentice Hall.

**Haykin, S. O.** (2013). *Adaptive filter theory* (5.<sup>a</sup> ed.). Upper Saddle River: Prentice Hall. [También se pueden consultar versiones anteriores].

### Artículos de interés

**Falconer, D.** (2011, octubre). "History of equalization 1860-1980". *IEEE Communications magazine* (vol. 49, n.º 10).

