

Capítulo 2: Procesos Aleatorios

2.1 Resumen

En este capítulo se muestran los métodos para la generación de variables aleatorias y muestras de procesos aleatorios. Comenzamos con la descripción de un método para la generación de variables aleatorias con una función de distribución de probabilidad especificada. Entonces tenemos en cuenta los procesos de Gauss y Gauss-Markov e ilustrar un método para generar muestras de estos procesos. El tercer tema que consideramos es la caracterización de un proceso aleatorio estacionario por su autocorrelación en el dominio del tiempo y por su espectro de potencia en el dominio de la frecuencia. Como los filtros lineales juegan un papel muy importante en los sistemas de comunicación, también consideramos la función de autocorrelación y el espectro de potencia de un proceso aleatorio linealmente filtrada. La sección final de este capítulo trata de las características de los procesos aleatorios de paso bajo y de paso de banda.

2.2 Generación de variables aleatorias

Los generadores de números aleatorios se usan a menudo en la práctica para simular el efecto del ruido como señales y otros fenómenos aleatorios que se encuentran en el mundo físico. Tal ruido está presente en dispositivos y sistemas electrónicos y por lo general limita nuestra capacidad de comunicarse a través de grandes distancias y para detectar señales relativamente débiles. Mediante la generación de este tipo de ruido en un equipo, somos capaces de estudiar sus efectos a través de la simulación de sistemas de comunicación y para evaluar el funcionamiento de tales sistemas en presencia de ruido.

La mayoría de las bibliotecas de software de ordenador incluye un generador de números aleatorios uniforme. Tal generador de números aleatorios genera un número entre 0 y 1, con igual probabilidad. Llamamos a la salida del generador de números aleatorios una **variable aleatoria**. Si A denota una variable aleatoria tal que, su rango es el intervalo de $0 \leq A \leq 1$.

Sabemos que la salida numérica de una computadora digital ha limitado la precisión, y, como consecuencia, es imposible representar la continuidad de los números en el intervalo de $0 \leq A \leq 1$. Sin embargo, se puede suponer que nuestro ordenador representa cada una de las salidas por un gran número de bits en cualquier punto fijo o punto flotante. En consecuencia, para todos los fines prácticos, el número de salidas en el intervalo de $0 \leq A \leq 1$ es lo suficientemente grande como para que estamos justificados al suponer que cualquier valor en el intervalo de una posible salida del generador.



Figura 2.1: Función de densidad de probabilidad $f(A)$ y la función de distribución de probabilidad $F(A)$ de una variable aleatoria uniformemente distribuida.

La función de densidad de probabilidad uniforme para la variable aleatoria A , denotado como $f(A)$, se ilustra en la Figura 2.1 (a). Observamos que el valor promedio o valor medio de A , denotado como m_A , es $m_A = \frac{1}{2}$. La integral de la función de densidad de probabilidad, que representa el área bajo $f(A)$, se denomina función de distribución de probabilidad de la variable aleatoria A y se define como

$$F(A) = \int_{-\infty}^A f(x) dx \quad (2.2.1)$$

Para cualquier variable aleatoria, esta área debe ser siempre la unidad, que es el valor máximo que se puede lograr por una función de distribución. Por lo tanto, para la variable aleatoria uniforme A tenemos

$$F(1) = \int_{-\infty}^1 f(x) dx = 1 \quad (2.2.2)$$

Y el intervalo de $F(A)$ es $0 \leq F(A) \leq 1$ para $0 \leq A \leq 1$. La función de distribución de probabilidad se muestra en la Figura 2.1 (b).

Si deseamos generar ruido uniformemente distribuido en un intervalo $(b, b + 1)$, se puede llevar a cabo simplemente mediante el uso de la salida A del generador de números aleatorios y el cambio por una cantidad b . Así, una nueva variable aleatoria B se puede definir como

$$B = A + b \quad (2.2.3)$$

Que ahora tiene un valor medio $m_B = b + \frac{1}{2}$. Por ejemplo, si $b = -\frac{1}{2}$, la variable aleatoria B se distribuye uniformemente en el intervalo $(-\frac{1}{2}, \frac{1}{2})$, como se muestra en la Figura 2.2 (a).

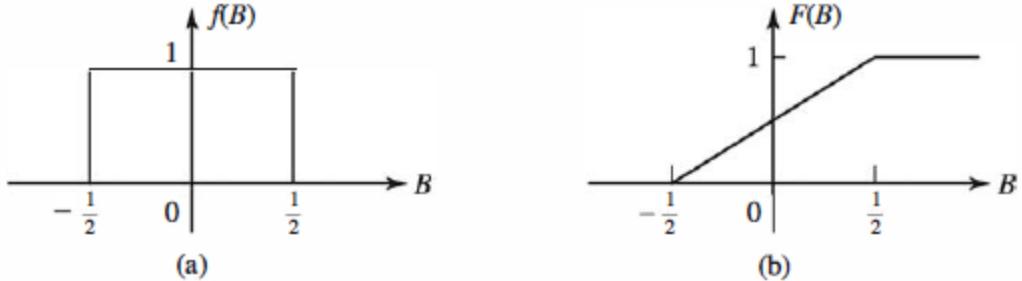


Figura 2.2: Función de densidad de probabilidad y la función de distribución de probabilidad de una variable aleatoria con media cero distribuidas uniformemente.

Su función de distribución de probabilidad $F(B)$ se muestra en la figura 2.2 (b). Una variable aleatoria uniformemente distribuida en el intervalo $(0, 1)$ se puede utilizar para generar variables aleatorias con otras funciones de distribución de probabilidad. Por ejemplo, supongamos que se desea para generar una variable aleatoria C con la función de distribución de probabilidad $F(C)$, como se ilustra en la Figura 2.3.

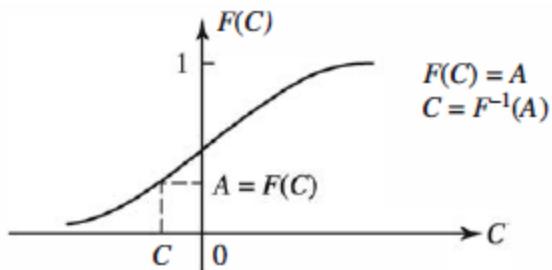


Figura 2.3: aplicación inversa de la variable aleatoria A uniformemente distribuida a la nueva variable aleatoria C

Debido a que el rango de $f(C)$ es el intervalo $(0,1)$, empezamos por la generación de una variable aleatoria uniformemente distribuida en el intervalo A $(0,1)$. Si establecemos

$$F(C) = A \quad (2.2.4)$$

Entonces

$$C = F^{-1}(A) \quad (2.2.5)$$

Por lo tanto resolvemos (2.2.4) para C , y la solución de (2.2.5) proporciona el valor de C para el que $F(C) = A$. Por este medio se obtiene una nueva variable aleatoria C con la función de distribución de probabilidad $F(C)$. Esta aplicación inversa de A a C se ilustra en la Figura 2.3.

Problema Ilustrativo 2.1 Generar una variable aleatoria C que tiene la función de densidad de probabilidad lineal que se muestra en la Figura 2.4 (a); es decir,

$$f(C) = \begin{cases} \frac{1}{2}C, & 0 \leq C \leq 2 \\ 0, & \text{otherwise} \end{cases}$$

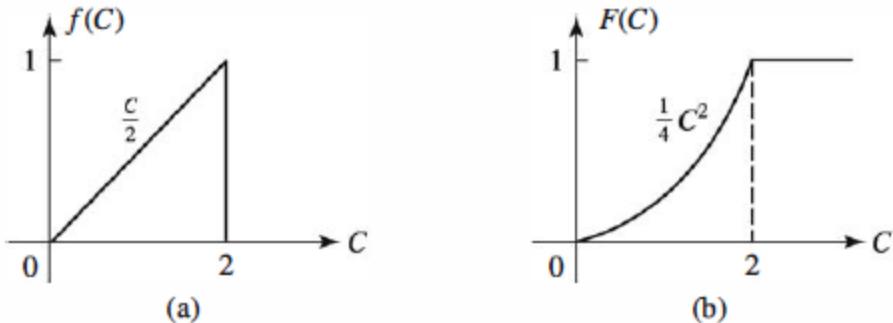


Figura 2.4: Función de densidad de probabilidad lineal y la función de distribución de probabilidad correspondiente.

Solución: Esta variable aleatoria tiene una función de distribución de probabilidad

$$F(C) = \begin{cases} 0, & C < 0 \\ \frac{1}{4}C^2, & 0 \leq C \leq 2 \\ 1, & C > 2 \end{cases}$$

Que se ilustra en la figura 2.4 (b). Generamos una variable aleatoria A uniformemente distribuida y conjunto $F(C) = A$. Por lo tanto

$$F(C) = \frac{1}{4}C^2 = A \quad (2.2.6)$$

Al resolver para C, obtenemos

$$C = 2\sqrt{A} \quad (2.2.7)$$

De este modo se genera una variable aleatoria C con la función de distribución de probabilidad $F(C)$, como se muestra en la figura 2.4 (b).

En Problema ilustrativo 2.1 la inversa cartografía $C = F^{-1}(A)$ era simple. En algunos casos no es así. Este problema surge en el intento de generar números aleatorios que tienen una función de distribución normal.

El ruido que se encuentran en los sistemas físicos se caracteriza a menudo por la distribución de probabilidad normal o Gaussiana, que se ilustra en la Figura 2.5. La función de densidad de probabilidad está dada por

$$f(C) = \frac{1}{\sqrt{2\pi}\sigma} e^{-C^2/2\sigma^2}, \quad -\infty < C < \infty \quad (2.2.8)$$

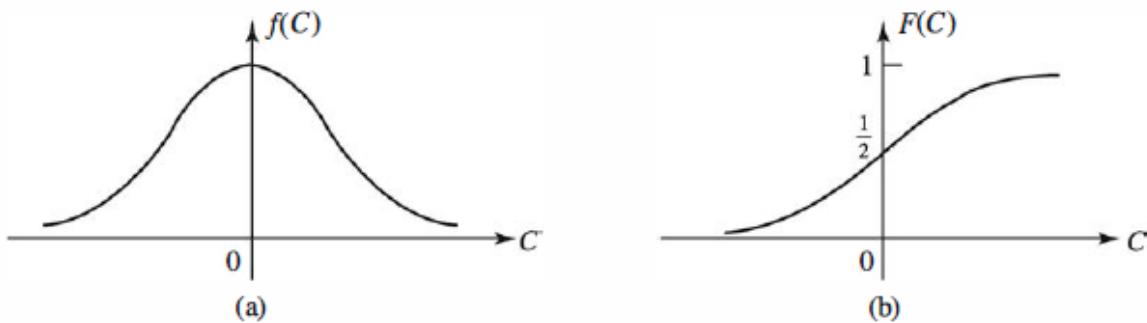


Figura 2.5: Función de densidad de probabilidad de Gauss y la función de distribución de probabilidad correspondiente.

Donde σ^2 es la varianza de C , que es una medida de la extensión de la función de densidad de probabilidad $f(C)$. La función de distribución de probabilidad $F(C)$ es el área bajo $f(C)$ en el rango $(-\infty, C)$. Entonces

$$F(C) = \int_{-\infty}^C f(x) dx \quad (2.2.9)$$

Por desgracia, la integral en (2.2.9) no se puede expresar en términos de funciones simples. En consecuencia, la aplicación inversa es difícil de lograr. Una forma se ha encontrado para evitar este problema. De la teoría de probabilidad se sabe que una distribución de Rayleigh de variable aleatoria R , con la función de distribución de probabilidad

$$F(R) = \begin{cases} 0, & R < 0 \\ 1 - e^{-R^2/2\sigma^2}, & R \geq 0 \end{cases} \quad (2.2.10)$$

Se relaciona con un par de variables Gaussianas C y D al azar a través de la transformación

$$C = R \cos \Theta \quad (2.2.11)$$

$$D = R \sin \Theta \quad (2.2.12)$$

Donde Θ es una variable uniformemente distribuida en el intervalo $(0, 2\pi)$. El parámetro σ^2 es la varianza de C y D . Debido a que (2.2.10) se invierte fácilmente, tenemos

$$F(R) = 1 - e^{-R^2/2\sigma^2} = A \quad (2.2.13)$$

Y por lo tanto

$$R = \sqrt{2\sigma^2 \ln \left(\frac{1}{1-A} \right)} \quad (2.2.14)$$

Donde A es una variable aleatoria uniformemente distribuida en el intervalo (0,1). Ahora, si generamos una segunda variable aleatoria B distribuida uniformemente y definimos

$$\Theta = 2\pi B \quad (2.2.15)$$

A continuación, a partir de (2.2.11) y (2.2.12) obtenemos estadísticamente dos variables aleatorias independientes C y D de distribución de Gauss.

El método descrito anteriormente se utiliza a menudo en la práctica para generar variables aleatorias con distribución de Gauss. Como se muestra en la Figura 2.5, estas variables aleatorias tienen un valor medio de cero y una varianza σ^2 . Si se desea una variable aleatoria gaussiana no de media cero, a continuación, C y D pueden ser traducidas por la adición del valor medio. La secuencia de comandos MATLAB que implementa el método anterior para la generación de variables aleatorias de distribución Gaussiana se da a continuación.

M-FILE

```
function [gsrv1 ,gsrv2]=gngauss (m,sgma)
% [gsrv1,gsrv2]=gngauss (m,sgma)
% [gsrv1,gsrv2]=gngauss (sgma)
% [gsrv1,gsrv2]=gngauss
% GNGAUSS genera dos variables aleatorias gaussianas independientes con
% media m y la desviación estándar sigma. Si uno de los argumentos de
% entrada no se encuentra, se toma la media como 0. Si no se da ni la media
% ni la varianza, genera dos variables aleatorias gaussianas.
if nargin == 0,
m=0; sgma=1;
elseif nargin == 1 ,
sgma=m; m=0;
end;
u=rand; % Una variable aleatoria uniforme en (0,1)
z=sgma*(sqrt(2*log(1/(1-u))));% Una variable aleatoria con distribución de
% Rayleigh
u=rand; % Otra variable aleatoria uniforme en (0,1)
gsrv1=m+z*cos(2*pi*u);
gsrv2=m+z*sin(2*pi*u);
```

2.2.1 Estimación de la media de una variable aleatoria:

Supongamos que tenemos N observaciones estadísticamente independientes x_1, x_2, \dots, x_N de una variable aleatoria X. Se desea estimar el valor medio de X a partir de las N observaciones. La estimación del valor medio es

$$\hat{m} = \frac{1}{N} \sum_{k=1}^N x_k \quad (2.2.16)$$

Debido a que \hat{m} es una suma de variables aleatorias, además también es una variable aleatoria. Observamos que el **valor esperado** de la estimación \hat{m} es

$$E[\hat{m}] = \frac{1}{N} \sum_{k=1}^N E[x_k] = \frac{1}{N} \cdot mN = m \quad (2.2.17)$$

Donde m es la media real de X. Por lo tanto, se dice que la estimación \hat{m} debe ser imparcial.

La varianza de la estimación \hat{m} es una medida de la difusión o dispersión de \hat{m} respecto a su valor medio. La varianza de \hat{m} se define como

$$\begin{aligned} E[(\hat{m} - m)^2] &= E[\hat{m}^2] - 2E[\hat{m}]m + m^2 \\ &= E[\hat{m}^2] - m^2 \end{aligned}$$

Pero $E[\hat{m}^2]$ es

$$\begin{aligned} E[\hat{m}^2] &= \frac{1}{N^2} \sum_{k=1}^N \sum_{n=1}^N E[x_k x_n] \\ &= \frac{\sigma^2}{N} + m^2 \end{aligned}$$

Donde σ^2 es la varianza real de X. Por lo tanto la varianza de \hat{m} es

$$E[(\hat{m} - m)^2] = \frac{\sigma^2}{N} \quad (2.2.18)$$

Note que cuando $N \rightarrow \infty$, la varianza de la estimación tiende a cero.

Problema ilustrativo 2.2 [Estimación de la media de una variable aleatoria] Generar 10 muestras de una variable aleatoria X que se distribuye de manera uniforme en el intervalo $[0, 2]$. Calcular la estimación de la media \bar{m} y su varianza y comparar el resultado con el valor medio verdadero de X. Repita este experimento 10 veces y calcular y trazar las estimaciones $\bar{m}_1, \bar{m}_2, \dots, \bar{m}_{10}$. También calcular el valor medio de las estimaciones; es decir,

$$\hat{m} = \frac{1}{10} \sum_{n=1}^{10} \hat{m}_n$$

Y comparar el resultado con el valor medio verdadero de X.

Solución: La figura 2.6 ilustra las 100 estimaciones de los valores medios de X. Nota que el clúster estimaciones en torno al valor verdadero de media $E(X) = 1$. También se muestra en la figura que es el valor medio \bar{m} de los 100 estimados. Este es el valor que habría obtenido si simplemente habíamos promedio de las 100 observaciones de X.

La secuencia de comandos de MATLAB para el problema se da la siguiente página.

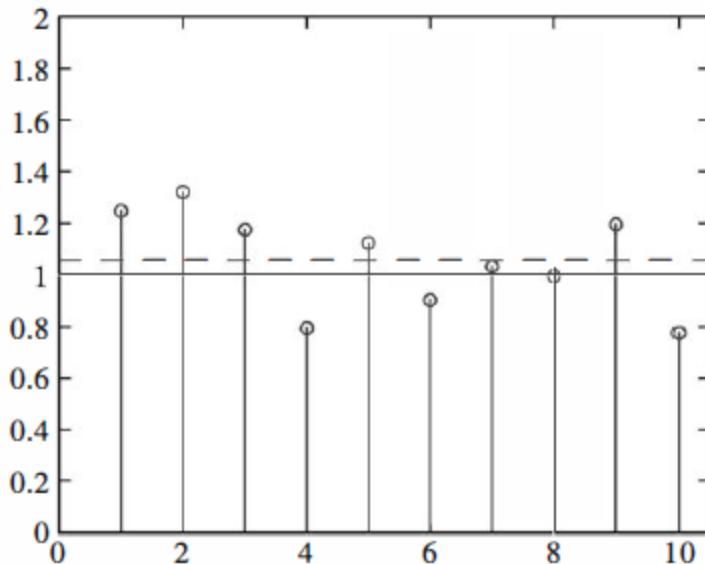


Figura 2.6: Las medias estimadas en el Problema ilustrativo 2.2 se indican mediante pequeños círculos. La línea de trazos horizontal representa el valor de \bar{m} y la línea continua horizontal es el valor esperado $E(X)$ de la variable aleatoria uniforme.

2.3 Procesos Gaussianos y de Gauss-Markov

Procesos de Gauss juegan un papel importante en sistemas de comunicación. La razón fundamental de su importancia es que el ruido térmico en dispositivos electrónicos, que se produce por el movimiento aleatorio de electrones debido a la agitación térmica, puede estar estrechamente modelado por un proceso de Gauss. La razón para el comportamiento gaussiana de ruido térmico es que la corriente introducida por el movimiento de los electrones en un circuito eléctrico puede considerarse como la suma de pequeñas corrientes de un muy grande número de fuentes, a saber, los electrones individuales. **Se puede suponer que al menos una mayoría de estas fuentes se comportan de forma independiente y, por lo tanto, la corriente total es la suma de un gran número de variables aleatorias independientes e idénticamente distribuidos (i.i.d.).** Cuando se aplica el **teorema del límite central**, esta corriente total tiene una distribución gaussiana. Aparte de ruido térmico, procesos Gaussianos proporcionan más bien buenos modelos para algunas fuentes de información también. Algunas propiedades interesantes de los procesos de Gauss, que se dan a continuación, hacen que estos procesos matemáticamente manejables y fáciles de tratar. Empezamos con una definición formal de un proceso de Gauss.

Definición: un proceso aleatorio $X(t)$ es un proceso Gaussiano si para todo n y todo (t_1, t_2, \dots, t_n) , las variables aleatorias $\{X(t_i)\}_{i=1}^n$ tienen una función de densidad conjunta de Gauss, que puede expresarse como

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}[\det(\mathbf{C})]^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \mathbf{m})^t \mathbf{C}^{-1} (\mathbf{x} - \mathbf{m})\right] \quad (2.3.1)$$

En el que el vector $\mathbf{X} = (x_1, x_2, \dots, x_n)^t$ denota las n variables aleatorias $x_i = X(t_i)$, \mathbf{m} es el vector de valor medio (es decir, $\mathbf{m} = E(\mathbf{X})$) y \mathbf{C} es la matriz covarianza $n \times n$ de las variables aleatorias (x_1, x_2, \dots, x_n) con elementos

$$c_{ij} = E[(x_i - m_i)(x_j - m_j)] \quad (2.3.2)$$

El superíndice t indica la transposición de un vector o una matriz y \mathbf{C}^{-1} es la inversa de la matriz de covarianza \mathbf{C} .

De la definición anterior se ve, en particular, **que en cualquier instante de tiempo de la variable aleatoria $X(t_0)$ es gaussiana, y en cualquier de los dos puntos (t_1, t_2) las variables aleatorias $(X(t_1), X(t_2))$ son distribuidos de acuerdo con una variable aleatoria gaussiana bidimensional.** Por otra parte, debido a una descripción estadística completa de $\{X(t_i)\}_{i=1}^n$ depende solamente de la media del vector \mathbf{m} y la matriz de covarianza \mathbf{C} , tenemos la siguiente propiedad.

Propiedad 1: Para los procesos de Gauss, el conocimiento de la media \mathbf{m} y covarianza \mathbf{C} proporciona una descripción estadística completa del proceso.

Otra propiedad muy importante de un proceso de Gauss se refiere a sus características cuando se pasa a través de un sistema lineal invariante en el tiempo. Esta propiedad puede expresarse de la siguiente manera.

Propiedad 2: Si el proceso de Gaussiano $X(t)$ se pasa a través de un sistema lineal invariante en el tiempo (LTI), la salida del sistema es también un proceso de Gauss. El efecto del sistema de $X(t)$ se refleja simplemente por un cambio en el valor medio y la covarianza de $X(t)$.

Problema ilustrativo 2.3 [Generación de muestras de un proceso multivariable de Gauss]
Generar muestras de un proceso aleatorio gaussiana multivariante $X(t)$ que tiene un valor medio \mathbf{m}_x especificado y una covarianza C_x .

Solución: En primer lugar, se genera una secuencia de n V.A estadísticamente independiente, de media cero y de varianza unitaria con variables aleatorias gaussianas mediante el método descrito en la sección 2.2. Denotemos esta secuencia de n muestras por el vector $\mathbf{Y} = (y_1, y_2, \dots, y_n)^t$. En segundo lugar, nosotros tomamos la matriz de covarianza $n \times n$ deseada como C_x

$$C_x = C_x^{1/2} (C_x^{1/2})^t \quad (2.3.3)$$

A continuación se define la transformación lineal ($n \times 1$) como vector X

$$X = C_x^{1/2} Y + \mathbf{m}_x \quad (2.3.4)$$

Así, la covarianza de X es

$$\begin{aligned} C_x &= E[(X - \mathbf{m}_x)(X - \mathbf{m}_x)^t] \\ &= E[C_x^{1/2} Y Y^t (C_x^{1/2})^t] \\ &= C_x^{1/2} E(Y Y^t) (C_x^{1/2})^t \\ &= C_x^{1/2} (C_x^{1/2})^t \end{aligned} \quad (2.3.5)$$

El paso más difícil en este proceso es la factorización de la matriz de covarianza C_x . Veamos un ejemplo de este procedimiento por medio de un ejemplo que emplea la distribución gaussiana bivariante. Supongamos que empezamos con un par de variables aleatorias gaussianas estadísticamente independientes Y_1 e Y_2 , que tiene media cero y varianza unitaria. Deseamos transformar éstos en un par de variables aleatorias Gaussianas x_1 y x_2 de media $m = 0$ y matriz de covarianza

$$\begin{aligned}\mathbf{C} &= \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix} \\ &= \begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & 1 \end{bmatrix}\end{aligned}\tag{2.3.6}$$

Donde σ_1^2 y σ_2^2 son las varianzas de x_1 y x_2 , respectivamente, y ρ es el covarianza normalizada, definida como

$$\rho = \frac{E[(X_1 - \mathbf{m}_1)(X_2 - \mathbf{m}_2)]}{\sigma_1\sigma_2} = \frac{c_{12}}{\sigma_1\sigma_2}\tag{2.3.7}$$

La matriz de covarianza \mathbf{C} se puede factorizar como

$$\mathbf{C} = \mathbf{C}^{1/2}(\mathbf{C}^{1/2})^t$$

Donde

$$\mathbf{C}^{1/2} = \frac{1}{2\sqrt{2}} \begin{bmatrix} \sqrt{3} + 1 & \sqrt{3} - 1 \\ \sqrt{3} - 1 & \sqrt{3} + 1 \end{bmatrix}\tag{2.3.8}$$

Por lo tanto

$$\begin{aligned}\mathbf{X} &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= \mathbf{C}^{1/2} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \\ &= \frac{1}{2\sqrt{2}} \begin{bmatrix} \sqrt{3} + 1 & \sqrt{3} - 1 \\ \sqrt{3} - 1 & \sqrt{3} + 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \\ &= \frac{1}{2\sqrt{2}} \begin{bmatrix} (\sqrt{3} + 1)y_1 + (\sqrt{3} - 1)y_2 \\ (\sqrt{3} - 1)y_1 + (\sqrt{3} + 1)y_2 \end{bmatrix}\end{aligned}\tag{2.3.9}$$

Las secuencias de comandos de MATLAB para este cálculo se dan a continuación.

M-FILE

```
% Script de MATLAB para problema ilustrativo 2.3
clear
echo on % Muestra las siguientes lineas por el command Window hasta que aparezca "echo off"
mx=[0 0]';
Cx=[1 1/2;1/2 1];
x=multi_gp(mx,Cx);
% Cálculo de la función de densidad fdp de (x1, x2) como sigue.
delta=0.1;
x1=-4:delta:4;% rango de valores de la V.A gaussiana X1
x2=-4:delta:4;% rango de valores de la V.A gaussiana X2
% Como X1 y X2 en cualquier instante de tiempo t son gaussianas las mismas
% podran representarse por una funcion densidad bidimensional (para este caso de 2 V.A)
% y que es equivalente a la distribucion gausiana bidimensional. Se contruye de la
% siguiente forma:
for i=1 :length(x1),
for j=1 :length(x2),
f(i,j)=(1/((2*pi)*det(Cx)^1/2))*exp((-1/2)*(([x1(i) x2(j)]-mx')*inv(Cx)*([x1(i);x2(j)]-mx)));
echo off ;
end;
end;
echo on
% Grafico de la funcion densidad de probabilidad pdf resultante.
mesh(x1,x2,f);
```

M-FILE

```
function [x] = multi_gp(m,C)
% [x]=multi_gp(m,C)
% MULTLGP genera una multivariante aleatoria gaussiana
% proceso con media de vector m (vector columna) y la matriz de covarianza C
N=length(m);
for i=1:N,
y(i)=gngauss;
end;
y=y.';
x=(C^(1/2))*y+m;
%x=sqrtn(C)*y+m;
```

La Figura 2.7 ilustra la función de densidad $f(x_1, x_2)$ para la matriz de covarianza C dada por (2.3.6).

Como se ha indicado, el paso más difícil en el cálculo es determinar $C^{1/2}$. Dada la matriz de covarianza deseada, es posible determinar los valores propios $\{\lambda_k, 1 \leq k \leq n\}$ y los vectores propios correspondientes $\{v_k, 1 \leq k \leq n\}$. A continuación, la matriz de covarianza C se puede expresar como

$$C = \sum_{k=1}^n \lambda_k v_k v_k^t \quad (2.3.10)$$

Y debido a que $C = C^{1/2} * (C^{\frac{1}{2}})^t$, se deduce que

$$C^{1/2} = \sum_{k=1}^n \lambda_k^{1/2} v_k v_k^t \quad (2.3.11)$$

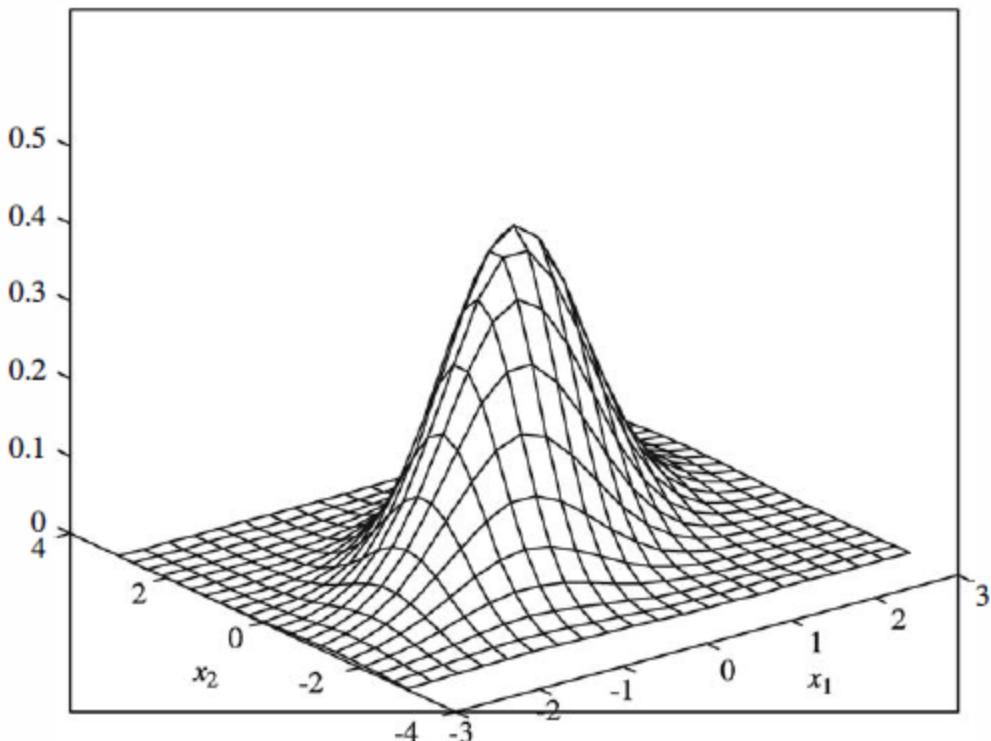


Figura 2.7: Función de densidad de probabilidad conjunta de X1 y X2

2.4 Espectro de potencia de procesos aleatorios y procesos blanco:

Un proceso aleatorio estacionario $X(t)$ se caracteriza en el dominio de la frecuencia por su espectro de potencia $S_x(f)$, que es la transformada de Fourier de la función de autocorrelación $R_x(\tau)$ del proceso aleatorio; es decir,

$$S_x(f) = \int_{-\infty}^{\infty} R_x(\tau) e^{-j2\pi f\tau} dt \quad (2.4.1)$$

Por el contrario, la función de autocorrelación $R_x(\tau)$ de un proceso aleatorio estacionario $X(t)$ se obtiene a partir del espectro de potencia $S_x(f)$ por medio de la transformada inversa de Fourier; es decir,

$$R_x(\tau) = \int_{-\infty}^{\infty} S_x(f) e^{j2\pi f\tau} df \quad (2.4.2)$$

En el modelado de ruido térmico que se genera en dispositivos electrónicos utilizados en la implementación de sistemas de comunicación, a menudo asumimos que tal ruido es un proceso aleatorio blanco. Tal proceso se define como sigue.

Definición: Un proceso aleatorio $X(t)$ se llama un **proceso blanco** si tiene un espectro de potencia plano, es decir, si $S_x(f)$ es una constante para todo f .

Como se ha indicado anteriormente, la importancia de los procesos blanco se deriva del hecho de que el ruido térmico puede estar estrechamente modelado como un espectro constante en una amplia gama de frecuencias. Además, un número de procesos que se utilizan para describir una variedad de fuentes de información se modela como la salida de sistemas LTI conducidos por un proceso blanco. Observamos, sin embargo, que si $S_x(f) = C$ para todos f , entonces

$$\int_{-\infty}^{\infty} S_x(f) df = \int_{-\infty}^{\infty} C df = \infty \quad (2.4.3)$$

De manera que la potencia total es infinito. Obviamente, no hay un proceso físico real que puede tener poder infinito y, por lo tanto, un proceso blanco no puede ser un proceso físico significativo. Sin embargo, el análisis de la mecánica cuántica del ruido térmico muestra que tiene una densidad espectral de potencia dada por

$$S_n(f) = \frac{hf}{2(e^{hf/kT} - 1)} \quad (2.4.4)$$

En la que \hbar denota la constante de Planck (igual a 6.6×10^{-34} J·s) y k es la constante de Boltzmann (igual a 1.38×10^{-23} J/K). T indica la temperatura en grados Kelvin. Este espectro de potencia se muestra en la Figura 2.10.

Este espectro alcanza su máximo a $f = 0$, y el valor de este máximo es $kT/2$. El espectro va a 0 cuando f tiende a infinito, pero la tasa de convergencia a 0 es muy baja. Por ejemplo, a temperatura ambiente ($T = 300$ K), $S_n(f)$ se reduce a 90% de su máximo a aproximadamente $f \approx 2 \times 10^{12}$ Hz, lo que está más allá de las frecuencias empleadas en sistemas de comunicación convencionales. **De esto concluimos que el ruido térmico, aunque no precisamente blanco, puede ser modelado a efectos prácticos como un proceso blanco con el espectro de potencia igual a $kT/2$.** El valor kT por lo general se denota por N_0 ; Por lo tanto, la densidad espectral de potencia de ruido térmico se da generalmente como $S_n(f) = N_0/2$ y se refiere a veces como la densidad espectral de potencia de dos caras, haciendo hincapié en que este espectro se extiende a ambas frecuencias positivas y negativas. Vamos a evitar esta terminología a lo largo y simplemente usar el espectro de potencia o densidad de potencia espectral. Para un proceso aleatorio blanco $X(t)$ con espectro de potencia $S_x(f) = N_0/2$, la función de autocorrelación $R_x(\tau)$ es

$$R_x(\tau) = \int_{-\infty}^{\infty} S_x(f) e^{j2\pi f\tau} df = \frac{N_0}{2} \int_{-\infty}^{\infty} e^{j2\pi f\tau} df = \frac{N_0}{2} \delta(\tau) \quad (2.4.5)$$

Donde $\delta(\tau)$ es el impulso unitario. En consecuencia, para todos $\tau \neq 0$, tenemos $R_x(\tau) = 0$; es decir, si nos muestra un proceso blanco en dos puntos t_1 y t_2 ($t_1 \neq t_2$), las variables aleatorias resultantes serán no correlacionadas. Si, además de ser blanco, el proceso aleatorio es gaussiano, las variables aleatorias de la muestra serán variables aleatorias gaussianas estadísticamente independientes.

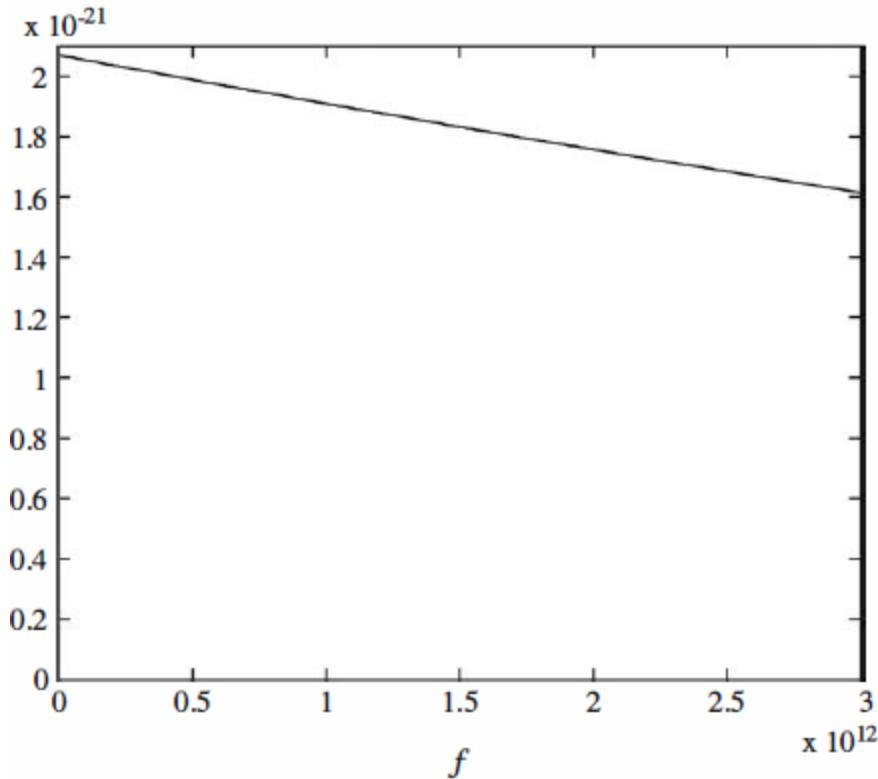


Figure 2.10: Plot of $S_n(f)$ in (2.4.4)

Problema ilustrativo 2.5 [Autocorrelación y Espectro de Potencia] Generar una secuencia de tiempo discreto de $N = 1000$ números aleatorios distribuidos uniformemente i.i.d. en el intervalo $(-\frac{1}{2}, \frac{1}{2})$, y calcular la estimación de la autocorrelación de la secuencia $\{X_n\}$,

$$\begin{aligned}\hat{R}_x(m) &= \frac{1}{N-m} \sum_{n=1}^{N-m} X_n X_{n+m}, \quad m = 0, 1, \dots, M \\ &= \frac{1}{N-|m|} \sum_{n=|m|}^N X_n X_{n+m}, \quad m = -1, -2, \dots, -M\end{aligned}\quad (2.4.6)$$

Además, calcular el espectro de potencia de la secuencia $\{X_n\}$ mediante la evaluación de la transformada de Fourier discreta (DFT) de $R_x(m)$. La DFT, que se calcula de manera eficiente mediante el uso del algoritmo de FFT, se define como

$$S_x(f) = \sum_{m=-M}^M R_x(m) e^{-j2\pi fm/(2M+1)} \quad (2.4.7)$$

Solución: La secuencia de comandos de MATLAB que implementa la generación de la secuencia $\{X_n\}$, el cálculo de la autocorrelación, y el cálculo del espectro de potencia $S_x(f)$ se da a continuación. Debemos tener en cuenta que las estimaciones de la función de

autocorrelación y el espectro de potencia presentan una variabilidad significativa. Por lo tanto, es necesario promediar la autocorrelación de la muestra durante varias realizaciones. Las figuras 2.11 y 2.12 ilustran $\hat{R}_X(m)$ y $\hat{S}_X(f)$ obtenido mediante la ejecución de este programa utilizando la autocorrelación promedio de más de diez realizaciones del proceso aleatorio.

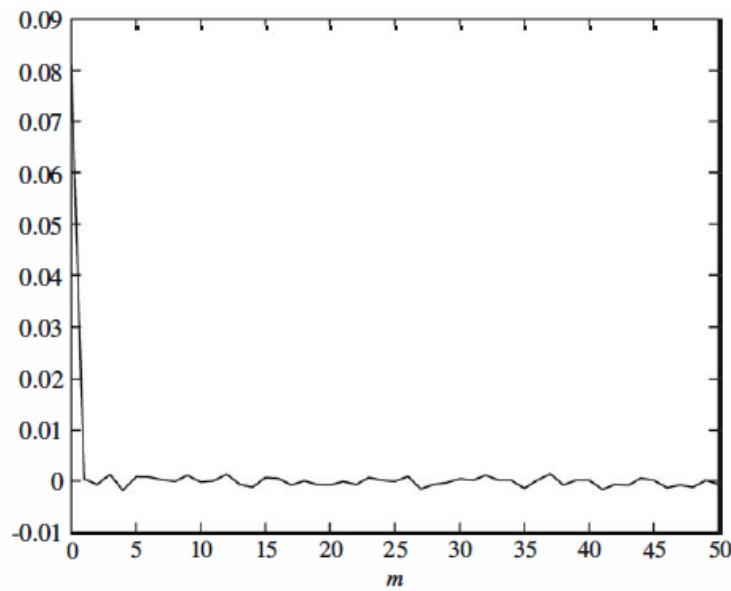


Figure 2.11: The autocorrelation function in Illustrative Problem 2.5

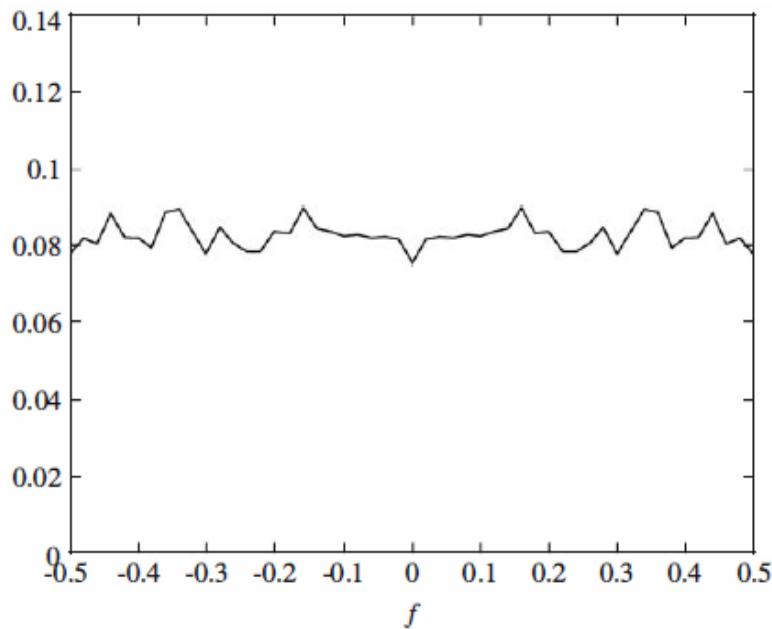


Figure 2.12: The power spectrum in Illustrive Problem 2.5

M-FILE

```
% Problema ilustrativo 2.5: Script de MATLAB
clear;
echo on
N=1000;
M=50;
delta = 1/50;
f= (-0.5:delta:0.5);
Rx_av=zeros(1,M+ 1);
Sx_av=zeros(1,M+ 1);
for j=1:10, % Tomar la media de conjunto más de diez realizaciones
    X=rand(1,N)-1/2; % N variables aleatorias distribuidas uniformemente i.i.d.
    % entre -1/2 y 1/2.
    Rx=Rx_est(X,M); % autocorrelation de la realizacion
    Sx=fftshift(abs(fft(Rx))); % espectro de potencia de la realizacion
    Rx_av=Rx_av+Rx;% suma de las autocorrelaciones
    Sx_av=Sx_av+Sx;% suma de los espectros
    echo off;
end;

echo on;
Rx_av=Rx_av/10;% autocorrelacion media del conjunto
Sx_av=Sx_av/10;% espectro medio del conjunto

figure(1)
plot(Rx_av)
figure(2)
plot(f,Sx_av)
```

Un proceso aleatorio de banda limitada $X(t)$ tiene el espectro de potencia

$$S_x(f) = \begin{cases} \frac{N_0}{2}, & |f| \leq B \\ 0, & |f| > B \end{cases} \quad (2.4.8)$$

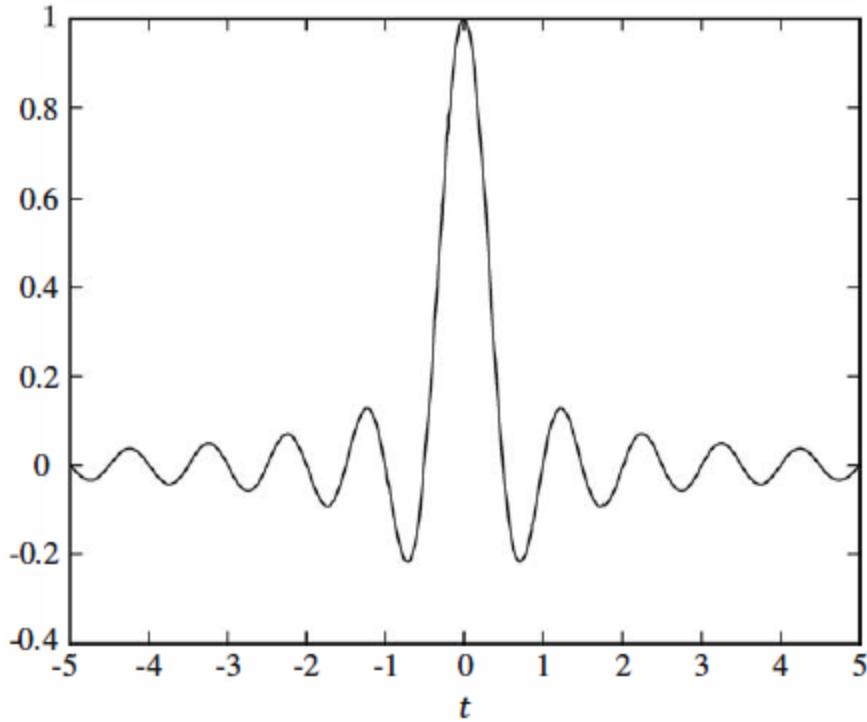


Figura 2.13: Grafico de función de autocorrelación $R_x(\tau)$ dada por (2.4.9) para $B = N_0 = 1$

Determinemos su función de autocorrelación. A partir de (2.4.1) tenemos

$$\begin{aligned} R_x(\tau) &= \int_{-B}^B \frac{N_0}{2} e^{j2\pi f\tau} df \\ &= N_0 B \left(\frac{\sin 2\pi B\tau}{2\pi B\tau} \right) \end{aligned} \quad (2.4.9)$$

La Figura 2.13 ilustra $R_x(\tau)$, con $R_x(0)$ normalizada a la unidad.

En MATLAB se puede usar para calcular $R_x(\tau)$ de $S_x(f)$ y viceversa. La transformada rápida de Fourier (FFT) es un algoritmo que puede ser utilizado para este tipo de cálculo.

Problema ilustrativo 2.6 [Autocorrelación y Espectro de Potencia] Calcular la autocorrelación $R_x(T)$ para el proceso aleatorio cuyo espectro de potencia está dada por (2.4.8).

Solución: Para realizar el cálculo, representamos $S_x(f)$ por N muestras en el rango de frecuencias $|F| \leq B$, con cada muestra normalizada a la unidad. El resultado del cálculo de la FFT inversa con $N = 32$ se ilustra en la Figura 2.14. Observamos que obtenemos sólo una representación gruesa de la función de autocorrelación $R_x(\tau)$, ya que probamos $S_x(f)$ sólo en el rango de frecuencia $|F| \leq B$. La separación de frecuencia en este ejemplo es $\Delta f = \frac{2B}{N}$. Si mantenemos Δf fijo y aumentar el número de muestras mediante la inclusión de

muestras para $F > B$, obtenemos valores intermedios de $R_x(\tau)$. La figura 2.15 ilustra el resultado de calcular la inversa FFT con $N_1 = 256$ muestras, de las cuales $N=32$ son unitarias.

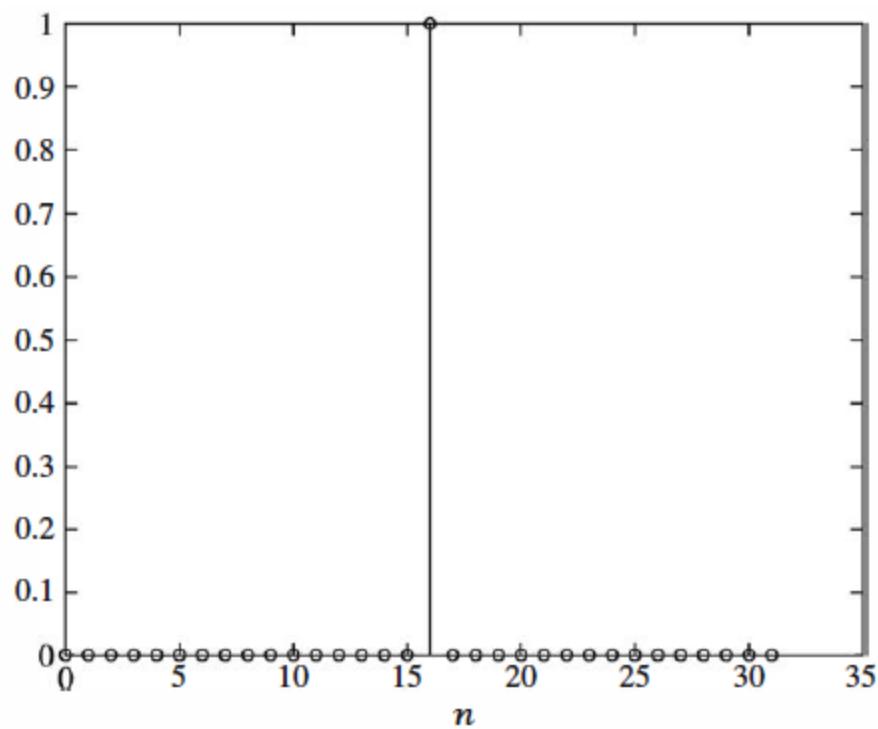


Figura 2.14: FFT inversa del espectro de potencia del proceso aleatorio de banda limitada del Problema ilustrativo 2.6 con 32 muestras.

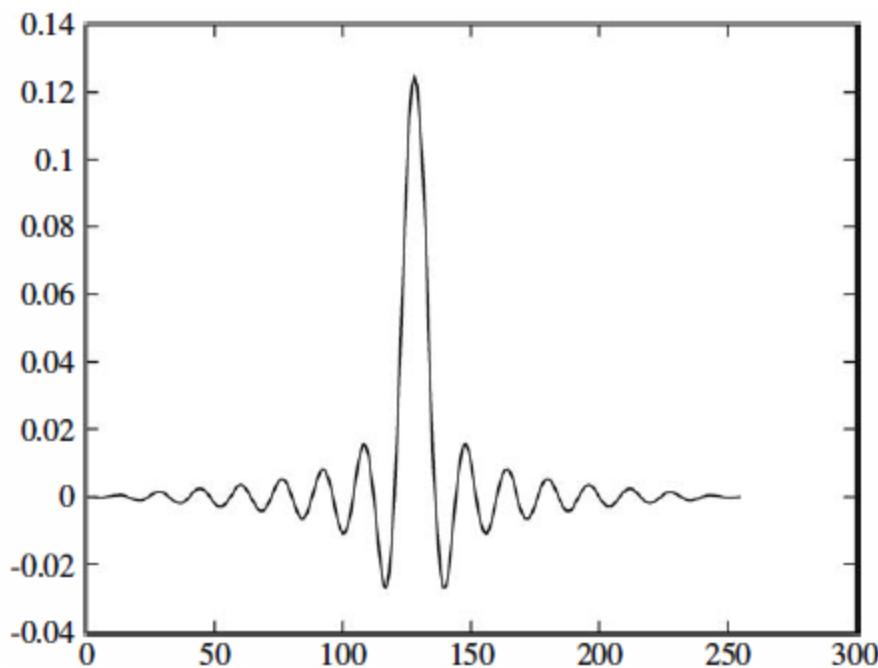


Figura 2.15: FFT inversa del espectro de potencia del proceso aleatorio de banda limitada del Problema ilustrativo 2.6 con 256 muestras.

2.5 Filtro Lineal de Procesos Aleatorios

Supongamos que un proceso aleatorio estacionario $X(t)$ se pasa por un filtro lineal invariante en el tiempo, que se caracteriza en el dominio del tiempo por su respuesta de impulso $h(t)$ y en el dominio de la frecuencia por su respuesta de frecuencia.

$$H(f) = \int_{-\infty}^{\infty} h(t) e^{-j2\pi ft} dt \quad (2.5.1)$$

De la expresión anterior se deduce que la salida del filtro lineal es un proceso aleatorio.

$$Y(t) = \int_{-\infty}^{\infty} X(\tau) h(t - \tau) d\tau \quad (2.5.2)$$

El valor medio de $Y(t)$ es

$$\begin{aligned} m_y &\equiv E[Y(t)] \\ &= \int_{-\infty}^{\infty} E[X(\tau)] h(t - \tau) d\tau \\ &= m_x \int_{-\infty}^{\infty} h(t - \tau) d\tau \\ &= m_x \int_{-\infty}^{\infty} h(\tau) d\tau \\ &= m_x H(0) \end{aligned} \quad (2.5.3)$$

Donde $H(0)$ es la respuesta de frecuencia $H(f)$ del filtro evaluada en $f = 0$.

La función de auto correlación de $Y(t)$ es

$$\begin{aligned} R_y(\tau) &= E[Y(t)Y(t + \tau)] \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} E[X(\beta)X(\alpha)] h(t - \beta) h(t + \tau - \alpha) d\beta d\alpha \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} R_x(\beta - \alpha) h(t - \beta) h(t + \tau - \alpha) d\beta d\alpha \end{aligned} \quad (2.5.4)$$

En el dominio de la frecuencia, el espectro de potencia del proceso de salida $Y(t)$ está relacionada con el espectro de potencia del proceso de entrada $X(t)$ y la respuesta de frecuencia del filtro lineal por la expresión

$$S_y(f) = S_x(f)|H(f)|^2 \quad (2.5.5)$$

Esto se muestra fácilmente tomando la transformada de Fourier de (2.5.4).

Problema ilustrativo 2.7 [ruido filtrado] Supongamos que un proceso aleatorio blanca $X(t)$ con espectro de potencia $S_x(f) = 1$ para todos f excita un filtro lineal con respuesta de impulso

$$h(t) = \begin{cases} e^{-t}, & t \geq 0 \\ 0, & t < 0 \end{cases} \quad (2.5.6)$$

Determinar el espectro de potencia $S_y(f)$ de la salida del filtro.

La respuesta de frecuencia del filtro se muestra fácilmente

$$H(f) = \frac{1}{1 + j2\pi f} \quad (2.5.7)$$

Por lo tanto,

$$\begin{aligned} S_y(f) &= |H(f)|^2 \\ &= \frac{1}{1 + (2\pi f)^2} \end{aligned} \quad (2.5.8)$$

La gráfica de $S_y(j)$ se ilustra en la Figura 2.16. La secuencia de comandos de MATLAB para calcular $S_y(f)$ para un $S_x(f)$ y $H(j)$ específico se da a continuación.

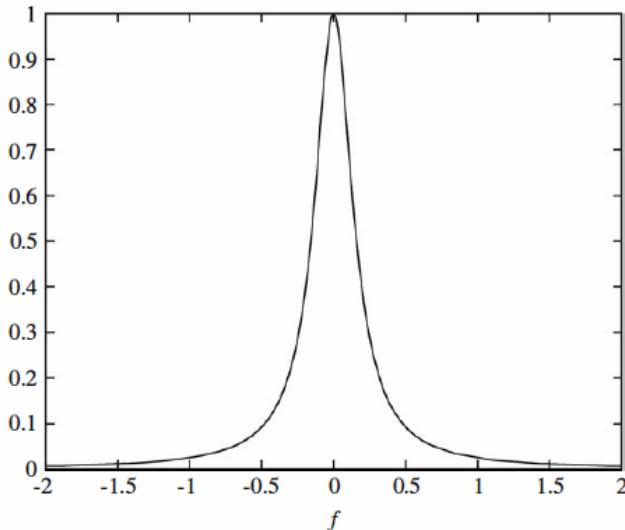


Figure 2.16: Plot of $S_y(f)$ given by (2.5.8)

```
% MATLAB script for Illustrative Problem 2.7.
echo on
delta=0.01;
F_min=-2;
F_max=2;
f=F_min:delta:F_max;
Sx=ones(1,length(f));
H=1./(1+(2*pi*f).^2);
Sy=Sx.*H.^2;
```

Problema ilustrativo 2.8 [Autocorrelación y Espectro de Potencia] Calcular la autocorrelación función de $R_y(t)$ correspondiente a $S_y(f)$ en Ilustrativo Problema 2.7 para la $S_x(f) = 1$ especificada.

En este caso, podemos utilizar el algoritmo FFT inversa sobre muestras de $S_y(f)$ dadas por (2.5.8). La figura 2.17 ilustra este cálculo con $N = 256$ muestras de frecuencia y una separación de frecuencia $\Delta f = 0.1$. Se da la secuencia de comandos de MATLAB para este cómputo siguiente.

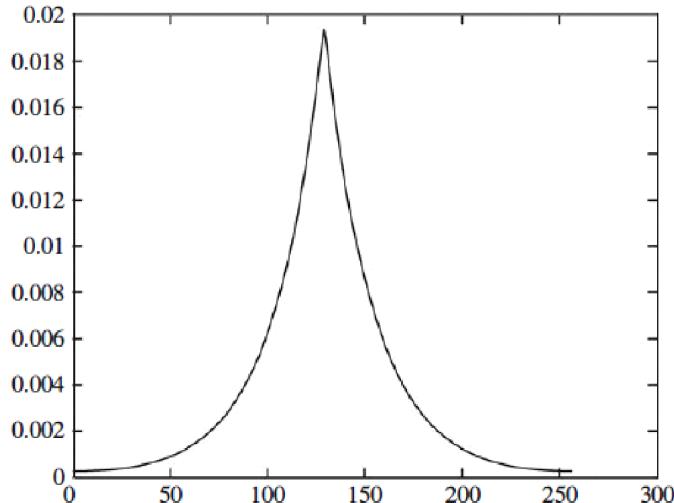


Figure 2.17: Plot of $R_Y(\tau)$ in Illustrative Problem 2.8

```
% MATLAB script for Illustrative Problem 2.8.
echo on
N=256; % number of samples
deltaf=0.1; % frequency separation
f=[0:deltaf:(N/2)*deltaf, -(N/2-1)*deltaf:deltaf:-deltaf];
% Swap the first half.
Sy=1./(1+(2*pi*f).^2); % sampled spectrum
Ry=ifft(Sy); % autocorrelation of Y
% Plotting command follows.
plot(fftshift(real(Ry)));
```

Consideremos ahora el problema equivalente en tiempo discreto. Supongamos que un proceso aleatorio estacionario $X(t)$ es muestreado y las muestras se pasan por un filtro lineal de tiempo discreto con respuesta de impulso $h(n)$. La salida del filtro lineal está dada por la siguiente formula

$$Y(n) = \sum_{k=0}^{\infty} h(k)X(n-k) \quad (2.5.9)$$

Donde $X(n)$ es equivalente a $X(t_n)$ son los valores de tiempo discreto del proceso aleatorio de entrada e $Y(n)$ es la salida del filtro de tiempo discreto. El valor medio del proceso de salida es

$$\begin{aligned}
m_y &= E[Y(n)] \\
&= \sum_{k=0}^{\infty} h(k)E[X(n-k)] \\
&= m_x \sum_{k=0}^{\infty} h(k) \\
&= m_x H(0)
\end{aligned} \tag{2.5.10}$$

Donde $H(0)$ es la respuesta de frecuencia $H(f)$ del filtro evaluada en $f = 0$ y

$$H(f) = \sum_{n=0}^{\infty} h(n)e^{-j2\pi f n} \tag{2.5.11}$$

La función de autocorrelación del proceso de salida es

$$\begin{aligned}
R_y(m) &= E[Y(n)Y(n+m)] \\
&= \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} h(k)h(l)E[X(n-k)X(n+m-l)] \\
&= \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} h(k)h(l)R_x(m-l+k)
\end{aligned} \tag{2.5.12}$$

La expresión correspondiente en el dominio de la frecuencia es

$$\mathcal{S}_y(f) = \mathcal{S}_x(f)|H(f)|^2 \tag{2.5.13}$$

Donde los espectros de potencia se definen como

$$\mathcal{S}_x(f) = \sum_{m=-\infty}^{\infty} R_x(m)e^{-j2\pi fm} \tag{2.5.14}$$

Y

$$S_y(f) = \sum_{m=-\infty}^{\infty} R_y(m) e^{-j2\pi f m} \quad (2.5.15)$$

Problema ilustrativo 2.9 [Filtrado de ruido blanco] Supongamos que un proceso aleatorio blanco con muestras $\{X(n)\}$ se hace pasar a través de un filtro lineal con respuesta de impulso

$$h(n) = \begin{cases} (0.95)^n, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

Determinar el espectro de potencia del proceso de salida de $\{Y(n)\}$.

Fácilmente se ve que

$$\begin{aligned} H(f) &= \sum_{n=0}^{\infty} h(n) e^{-j2\pi f n} \\ &= \sum_{n=0}^{\infty} (0.95 e^{-j2\pi f})^n \\ &= \frac{1}{1 - 0.95 e^{-j2\pi f}} \end{aligned} \quad (2.5.16)$$

Y

$$\begin{aligned} |H(f)|^2 &= \frac{1}{|1 - 0.95 e^{-j2\pi f}|^2} \\ &= \frac{1}{1.9025 - 1.9 \cos(2\pi f)} \end{aligned} \quad (2.5.17)$$

Por lo tanto, el espectro de potencia del proceso de salida es

$$S_y(f) = |H(f)|^2 S_x(f) \quad (2.5.18)$$

$$= \frac{1}{1.9025 - 1.9 \cos(2\pi f)} \quad (2.5.19)$$

Donde se asume que $S_y(f)$ es normalizada a la unidad. La figura 2.18 ilustra $S_y(f)$. Se nota que $S_y(f)$ es periódica con periodo 2π . La secuencia de comandos de MATLAB para este cálculo es el siguiente.

```
% MATLAB script for Illustrative Problem 2.9.
delta_w=2*pi/100;
w=-pi:delta_w:pi; % one period of S_y
S_y=1./(1.9025-1.9*cos(w));
% Plotting command follows.
plot(w,S_y);
```

La autocorrelación del proceso de salida de $\{Y(n)\}$ se puede determinar tomando la FFT inversa de $S_y(f)$. El estudiante podrá comparar esta autocorrelación con la obtenida en ilustrativa Problema 2.4.

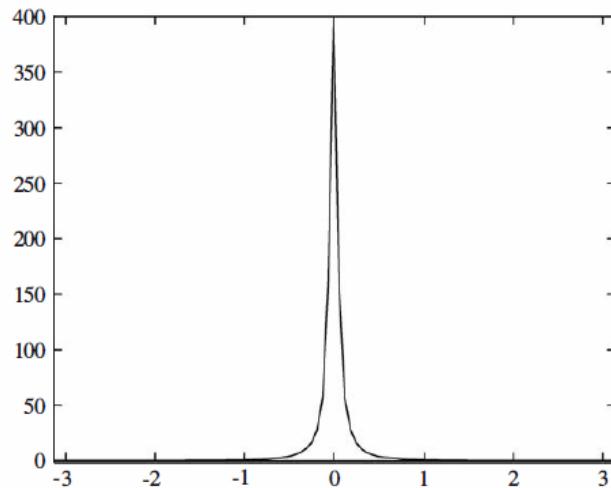


Figure 2.18: Plot of $S_y(f)$ in Illustrative Problem 2.9

Problemas:

2.1 Generar un conjunto de 1000 números aleatorios uniformes en el intervalo [0, 1] utilizando la función rand (L, N) de MATLAB. Trazar el histograma y la función de distribución de probabilidad de la secuencia. El histograma se puede determinar cuantificando el intervalo en diez subintervalos de igual anchura que cubren el rango [0, 1] y contando los números en cada subintervalo.

2.2 Generar un conjunto de 1000 números aleatorios uniformes en el intervalo $[-\frac{1}{2}, \frac{1}{2}]$. Utilizando la función rand(L, N) de MATLAB. Trazar el histograma y la función de distribución de probabilidad de la secuencia.

2.3 Generar un conjunto de 1000 números aleatorios uniformes en el intervalo [-1, 1] mediante la función rand(1, N) de MATLAB. Trazar el histograma y la función de distribución de probabilidad de la secuencia.

2.5 Generar un conjunto de 1000 números aleatorios gaussianos que tienen media cero y varianza unitaria utilizando el método descrito en la sección 2.2. Trazar el histograma y la función de distribución de probabilidad de la secuencia. En la determinación del histograma, el rango de los números aleatorios se puede subdividir en subintervalos de anchura $\sigma^2/5$, comenzando con el primer intervalo de escala de $-\sigma^2/10 < x < \sigma^2/10$, donde σ^2 es la varianza.

2.6 Generar un conjunto de 1000 números aleatorios gaussianos tienen media cero y varianza unitaria mediante la función randn(1,N) de MATLAB. Trazar el histograma y la función de distribución de probabilidad de la secuencia. Comparar estos resultados con los resultados obtenidos en el problema 2.5.

2.7 El teorema del límite central establece que si las variables aleatorias X_i , $1 \leq i \leq n$ son i.i.d. (Independientes e idénticamente distribuidas), con media y varianza finita, y n es grande, entonces su promedio (es decir, $Y = \frac{1}{n} \sum_{i=1}^n X_i$) tiene aproximadamente una distribución de

Gauss o normal. Este teorema explica por qué el ruido térmico generado en los circuitos eléctricos tiene una distribución de Gauss. En este problema se verifica este teorema utilizando MATLAB.

a- Usando MATLAB, generar un vector x de la longitud 1.000.000 cuyos componentes son variables aleatorias uniformes distribuidos entre 0 y 1. Los componentes de este vector son las X_i en el teorema del límite central. (Utilice MATLAB comando "rand" para generar la secuencia.)

b- Encuentra el promedio de cada 100 elementos consecutivos de x y generar la secuencia y de 10.000 elementos de longitud. Así, Y_1 es el promedio de X_1 a X_{100} , Y_2 es el promedio de X_{101} a X_{200} , y así sucesivamente.

c- Trazar el histograma de la secuencia y usando el comando "hist". Utilice 40 contenedores para generar su trama. Incluir la lista de su programa y la trama. Tenga en cuenta que el histograma está muy cerca de una distribución de Gauss.

2.8 Generar 1000 pares de números aleatorios gaussianos (X_1, X_2) que tienen media del vector

$$\mathbf{m} = E [x_1 \ x_2] = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

Y la matriz de covarianza C

$$\mathbf{C} = \begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & 1 \end{bmatrix}$$

a. Determinar las medias de las muestras (X_{1i}, X_{2i}), para $i = 1, 2, \dots, 1000$, definido como

$$\hat{\mathbf{m}}_1 = \frac{1}{1000} \sum_{i=1}^{1000} x_{1i}$$

$$\hat{\mathbf{m}}_2 = \frac{1}{1000} \sum_{i=1}^{1000} x_{2i}$$

También determinar sus varianzas

$$\hat{\sigma}_1^2 = \frac{1}{1000} \sum_{i=1}^{1000} (x_{1i} - \hat{m}_1)^2$$

$$\hat{\sigma}_2^2 = \frac{1}{1000} \sum_{i=1}^{1000} (x_{2i} - \hat{m}_2)^2$$

Y sus covarianzas (donde $i \neq j$)

$$\hat{c}_{ij} = \frac{1}{1000} \sum_{i=1}^{1000} (x_{1i} - \hat{m}_1)(x_{2i} - \hat{m}_2)$$

b. Comparación de los valores obtenidos de las muestras con los valores teóricos.

2.11 Repita el Problema ilustrativo 2.5 con una secuencia i.i.d. de media cero, varianza unitaria, de variables aleatorias gaussianas.

2.12 Repita el Problema ilustrativo 2.6 cuando el espectro de potencia de un proceso aleatorio de banda limitada es

$$S_x(f) = \begin{cases} 1 - \frac{|f|}{B}, & |f| \leq B \\ 0, & \text{otherwise} \end{cases}$$

Capítulo 5: Transmisión de banda base digital

5.1 Resumen

En este capítulo consideramos varias técnicas de banda base digital de modulación y demodulación para transmitir información digital a través de un canal de ruido blanco gaussiano aditivo. Comenzamos con la modulación de impulsos binarios y luego introducimos varios métodos de modulación no binarios. Se describen los receptores óptimos para estas diferentes señales y considerar la evaluación de su rendimiento en términos de la probabilidad media de error.

5.2 Transmisión de señal binaria

En un sistema de comunicación binario, los datos binarios que consisten en una secuencia de ceros y unos, se transmiten por medio de dos formas de onda como señales, digamos, $s_0(t)$ y $s_1(t)$. Supongamos que la velocidad de datos se especifica como R bits por segundo. A continuación, cada bit se asigna a una forma de onda de la señal correspondiente de acuerdo con la regla

$$\begin{aligned} 0 &\rightarrow s_0(t), \quad 0 \leq t \leq T_b \\ 1 &\rightarrow s_1(t), \quad 0 \leq t \leq T_b \end{aligned}$$

Donde $T_b = 1/R$, se define como el intervalo de tiempo del bit. Suponemos que los datos de los bits 0 y 1 son igualmente probables, es decir, cada uno ocurre con probabilidad $\frac{1}{2}$ y son mutuamente estadísticamente independientes.

El canal a través del cual se transmite la señal se supone que corrompe la señal por la adición de ruido, denotado como $n(t)$, que es una función de muestras de un proceso Gaussiano blanco con espectro de potencia $N_0/2$ vatios/hertzios. Tal canal se llama un **canal de ruido gaussiano blanco aditivo (AWGN)**. En consecuencia, la forma de onda de la señal recibida se expresa como

$$r(t) = s_i(t) + n(t), \quad i = 0, 1, \quad 0 \leq t \leq T_b \quad (5.2.1)$$

La tarea del receptor es determinar si un 0 o un 1 se remitió después de observar la señal recibida $r(t)$ en el intervalo de $0 \leq t \leq T_b$. El receptor está diseñado para minimizar la probabilidad de error. Tal receptor se denomina el receptor óptimo.

5.2.1 Receptor óptimo para el canal AWGN

En los textos de comunicación digital casi todos básicos, se muestra que el receptor óptimo para el canal AWGN consta de dos bloques de construcción. Una de ellos es o bien un correlador de señal o un filtro adaptado. El otro es un detector.

Correlador de señal

El correlador de señal correlador-cruzado recibe la señal $r(t)$ con las dos posibles señales transmitidas $s_0(t)$ y $s_1(t)$, como se ilustra en la figura 5.1. Es decir, el correlador de señal calcula las dos salidas

$$\begin{aligned} r_0(t) &= \int_0^t r(\tau) s_0(\tau) d\tau \\ r_1(t) &= \int_0^t r(\tau) s_1(\tau) d\tau \end{aligned} \quad (5.2.2)$$

En el intervalo de $0 \leq t \leq T_b$, las muestras de las dos salidas de $t = T_b$, y alimenta las salidas de la muestra al detector.

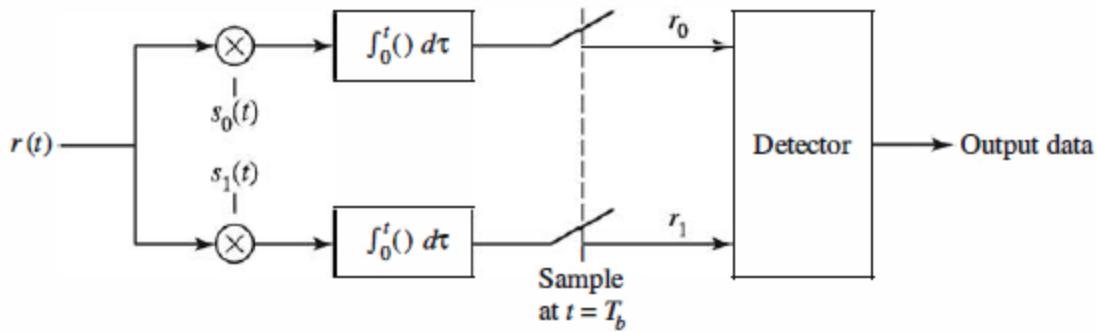


Figura 5.1: correlación cruzada de la señal recibida $r(t)$ con las dos señales de transmisión

Problema ilustrativo 5.1 [Correladores de señal] Supongamos que las formas de onda de señal de forma $s_0(t)$ y $s_1(t)$ son como se muestra en la Figura 5.2, y sea $s_0(t)$ la señal transmitida. A continuación, la señal recibida es

$$r(t) = s_0(t) + n(t), \quad 0 \leq t \leq T_b \quad (5.2.3)$$

Determinar las salidas de correlación en los instantes de muestreo.

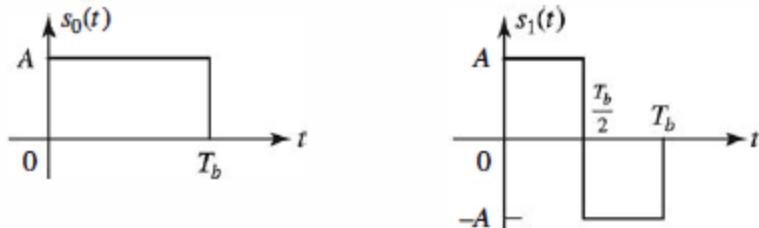


Figura 5.2: Formas de onda de señal $s_0(t)$ y $s_1(t)$ para un sistema de comunicación binario

Solución

Cuando la señal $r(t)$ es procesada por los dos correladores de señales que se muestran en la Figura 5.1, las salidas r_0 y r_1 en el instante de muestreo $t = T_b$ son

$$\begin{aligned} r_0 &= \int_0^{T_b} r(t)s_0(t) dt \\ &= \int_0^{T_b} s_0^2(t) dt + \int_0^{T_b} n(t)s_0(t) dt \\ &= E + n_0 \end{aligned} \quad (5.2.4)$$

$$\begin{aligned} r_1 &= \int_0^{T_b} r(t)s_1(t) dt \\ &= \int_0^{T_b} s_0(t)s_1(t) dt + \int_0^{T_b} n(t)s_1(t) dt \\ &= n_1 \end{aligned} \quad (5.2.5)$$

Donde n_0 y n_1 son los componentes de ruido en la salida de las correlaciones de señal; es decir,

$$\begin{aligned} n_0 &= \int_0^{T_b} n(t)s_0(t) dt \\ n_1 &= \int_0^{T_b} n(t)s_1(t) dt \end{aligned} \quad (5.2.6)$$

Y $E = (A^2 * T_b)$ es la energía de las señales de $s_0(t)$ y $s_1(t)$. También observamos que las dos formas de onda de las señales son ortogonales; es decir,

$$\int_0^{T_b} s_0(t)s_1(t) dt = 0 \quad (5.2.7)$$

Por otro lado, cuando $s_1(t)$ es la señal transmitida, la señal recibida es

$$r(t) = s_1(t) + n(t), \quad 0 \leq t \leq T_b$$

Es fácil mostrar que, en este caso, las salidas de señal del correlador son

$$\begin{aligned} r_0 &= n_0 \\ r_1 &= E + n_1 \end{aligned} \quad (5.2.8)$$

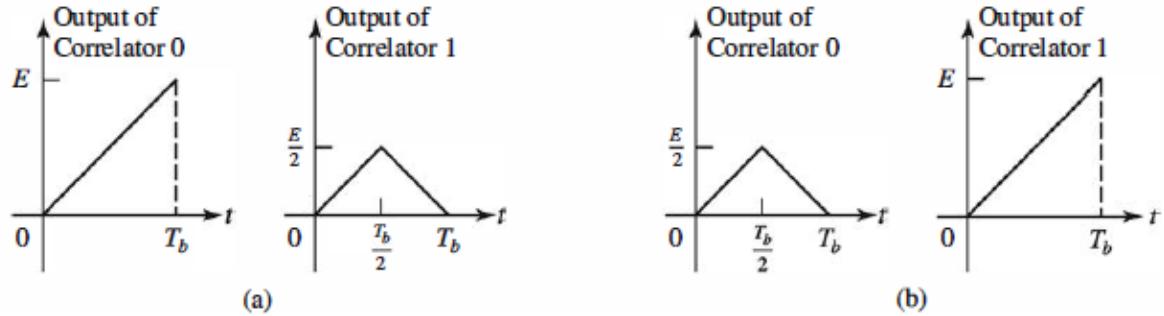


Figura 5.3: salidas del correlador insonorizadas. (a) si $s_0(t)$ fue transmitida. (b) si $s_1(t)$ fue transmitida

La Figura 5.3 ilustra las dos salidas del correlador libres de ruido en el intervalo de $0 \leq t \leq T_b$ para cada uno de los dos casos, es decir, cuando $s_0(t)$ se transmite y cuando $s_1(t)$ se transmite. Debido a que $n(t)$ es una función de ejemplo de un proceso Gaussiano blanco con espectro de potencia $N_0/2$, significa las componentes de ruido n_0 y n_1 son gaussiana con media cero, es decir,

$$\begin{aligned} E(n_0) &= \int_0^{T_b} s_0(t) E[n(t)] dt = 0 \\ E(n_1) &= \int_0^{T_b} s_1(t) E[n(t)] dt = 0 \end{aligned} \quad (5.2.9)$$

Y varianzas σ_i^2 , para $i=1,2$ en donde:

$$\begin{aligned} \sigma_i^2 &= E(n_i^2) \\ &= \int_0^{T_b} \int_0^{T_b} s_i(t) s_i(\tau) E[n(t)n(\tau)] dt d\tau \\ &= \frac{N_0}{2} \int_0^{T_b} s_i(t) s_i(\tau) \delta(t - \tau) dt d\tau \\ &= \frac{N_0}{2} \int_0^{T_b} s_i^2(t) dt \\ &= \frac{EN_0}{2}, \quad i = 0, 1 \end{aligned} \quad (5.2.10)$$

$$(5.2.11)$$

Por lo tanto, cuando $s_0(t)$ se transmite, las funciones de densidad de probabilidad de r_0 y r_1 son

$$\begin{aligned} p(r_0 | s_0(t) \text{ was transmitted}) &= \frac{1}{\sqrt{2\pi}\sigma} e^{-(r_0-E)^2/2\sigma^2} \\ p(r_1 | s_0(t) \text{ was transmitted}) &= \frac{1}{\sqrt{2\pi}\sigma} e^{-r_1^2/2\sigma^2} \end{aligned} \quad (5.2.12)$$

Estas funciones de densidad de probabilidad, denotadas como $p(r_0 | s_0)$ y $p(r_1 | s_0)$, se ilustran en la Figura 5.4. Del mismo modo, cuando $s_1(t)$ se transmite, r_0 es de media cero gaussiana con varianza σ^2 y r_1 es gaussiana con valor medio E y varianza σ^2 .

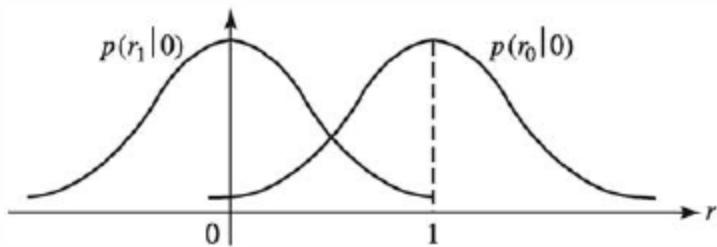


Figura 5.4: funciones de densidad de probabilidad $p(r_0 | 0)$ y $p(r_1 | 0)$ cuando $s_0(t)$ se transmite

Problema ilustrativo 5.2 [La correlación de formas de onda de señal] Muestrear las formas de onda de señal en el Problema ilustrativo 5.1 con una velocidad de $F_s = 20/T_b$ (con un intervalo de muestreo de $T_s = Tb/20$) y realizar la correlación de $r(t)$, con $s_0(t)$ y $s_1(t)$ numéricamente; es decir, a través de la computadora y graficar

$$r_0(kT_s) = \sum_{n=1}^k r(nT_s)s_0(nT_s), \quad k = 1, 2, \dots, 20$$

$$r_1(kT_s) = \sum_{n=1}^k r(nT_s)s_1(nT_s), \quad k = 1, 2, \dots, 20$$

Cuando (a) $s_0(t)$ es la señal transmitida y (b) $s_1(t)$ es la señal transmitida.

Repetir los cálculos y los gráficos cuando las muestras de la señal $r(kT_s)$ están degradados por las muestras del ruido aditivo gaussiano blanco $n(kT_s)$, para $1 \leq k \leq 20$, que tienen media cero y varianza $\sigma^2 = 0,1$ y $\sigma^2 = 1$ respectivamente.

Solución

La Figura 5.5 ilustra las salidas de correladores. Observamos el efecto del ruido aditivo sobre las salidas del correlador, especialmente cuando se da $\sigma^2 = 1$. La secuencia de comandos de MATLAB para estos cálculos se da a continuación.

M-FILE

```
% Secuencia de comandos de Problema ilustrativo 5.2
% Inicializacion
K=20; % Numero de muestras
A=1; % Amplitud de Señal
l= 0:K; % vector con valores de 0 a K (paso por defecto 1 [0,1,...,19,20])
% Definicion de la forma de onda de la señal:
s_0 = A*ones(1,K);
figure(1)
plot(s_0,'linewidth',3)
s_1 =[A *ones(1,K) ];
s_1(11:K) =[s_1((K/2)+1:K) - 2*A *ones(1,K/2)];
figure(2)
plot(s_1,'linewidth',3)

% Inicializacion de señales de salida:
r_0 = zeros(1,K);
r_1 = zeros(1,K);

% Caso 1: ruido~N(0,0)--> de distribucion normal con media 0
noise=random('Normal',0,0,1,K);% ruido
% Sub-caso s = s_0:
s=s_0;
r=s+noise; % señal recibida
for n=1:K
r_0(n)= sum(r(1:n).*s_0(1:n));
r_1(n)= sum(r(1:n).*s_1(1:n));
end
% Graficando resultados:
figure(3)
subplot(3,2,1)
plot(l,[0 r_0],'-',l,[0 r_1], '--')
set(gca, 'XTickLabel', {'0','5Ts','10Ts','15Ts', '20Ts'})
axis([0 20 -5 30])
xlabel('(a) \sigma^2 = 0 & S_0 es transmitido','fontsize',10)
% Sub-caso s= s_1:
s=s_1;
r=s+noise; % señal recibida
for n=1:K
r_0(n)= sum(r(1:n).*s_0(1:n));
r_1(n)= sum(r(1:n).*s_1(1:n));
end
```

```

% Graficando resultados:
subplot(3,2,2)
plot(l,[0 r_0],'-',l,[0 r_1], '--')
set(gca, 'XTickLabel' ,{'0','5Ts','10Ts','15Ts', '20Ts'})
axis([0 20 -5 30])
xlabel('(b) \sigma^2 = 0 & S_{\{1\}} es transmitido','fontsize',10)

% Case 2: ruido-N(0,0.1)
noise=random('Normal',0,0.1, 1,K); % ruido
% Sub-caso s = s_0:
s=s_0;
r=s+noise; % received signal
for n=1:K
r_0(n)= sum(r(1:n).*s_0(1:n));
r_1(n)= sum(r(1:n).*s_1(1:n));
end
% Graficando resultados:
subplot(3,2,3)
plot(l,[0 r_0],'-',l,[0 r_1], '--')
set(gca, 'XTickLabel' ,{'0','5Ts','10Ts','15Ts', '20Ts'})
axis([0 20 -5 30])
xlabel('(c) \sigma^2 = 0.1 & S_{\{0\}} es transmitido','fontsize',10)

% Sub-caso s= s_1:
s=s_1;
r=s+noise; % señal recibida
for n=1:K
r_0(n)= sum(r(1:n).*s_0(1:n));
r_1(n)= sum(r(1:n).*s_1(1:n));
end
% Graficando resultados:
subplot(3,2,4)
plot(l,[0 r_0],'-',l,[0 r_1], '--')
set(gca, 'XTickLabel' ,{'0','5Ts','10Ts','15Ts', '20Ts'})
axis([0 20 -5 30])
xlabel('(d) \sigma^2 = 0.1 & S_{\{1\}} es transmitido','fontsize',10)

% Case 3: ruido~N(0,1)
noise=random('Normal',0,1,1,K); % ruido
% Sub-caso s = s_0:
s=s_0;
r=s+noise; % received signal
for n=1:K
r_0(n)= sum(r(1:n).*s_0(1:n));
r_1(n)= sum(r(1:n).*s_1(1:n));
end

```

```

% Graficando resultados:
subplot(3,2,5)
plot(l,[0 r_0],'-',l,[0 r_1], '--')
set(gca, 'XTickLabel' ,{'0','5Ts','10Ts','15Ts', '20Ts'})
axis([0 20 -5 30])
xlabel('(e) \sigma^2 = 1 & S_{\{0\}} es transmitido','fontsize',10)

% Sub-caso s= s_1:
s=s_1;
r=s+noise; % señal recibida
for n=1:K
    r_0(n)= sum(r(1:n).*s_0(1:n));
    r_1(n)= sum(r(1:n).*s_1(1:n));
end
% Graficando resultados:
subplot(3,2,6)
plot(l,[0 r_0],'-',l,[0 r_1], '--')
set(gca, 'XTickLabel' ,{'0','5Ts','10Ts','15Ts', '20Ts'})
axis([0 20 -5 30])
xlabel('(f) \sigma^2 = 1 & S_{\{1\}} es transmitido','fontsize',10)

```

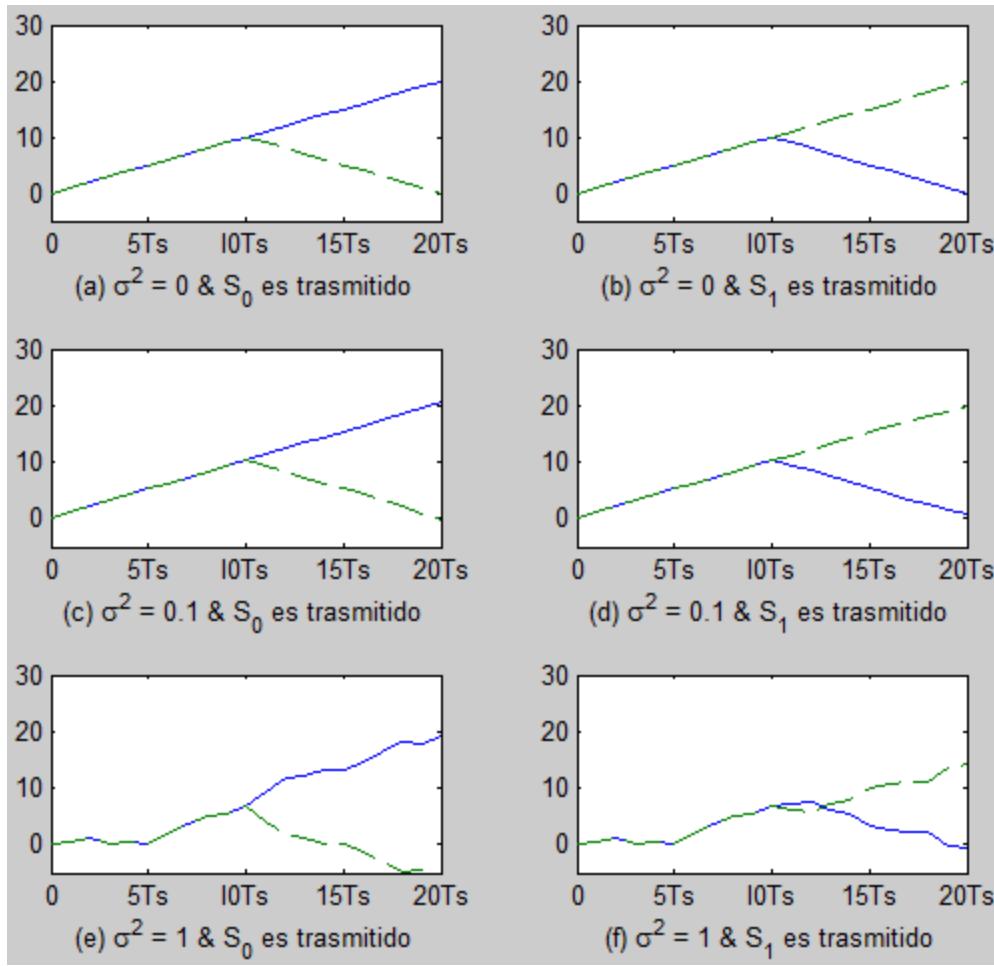


Figura 5.5: salidas del correlador del Problema ilustrativo 5.2 las líneas continuas y discontinuas representan la salida de correlación para $s_0(t)$ y $s_1(t)$, respectivamente.

Filtro Adaptado

El filtro adaptado proporciona una alternativa al correlador de señal para demodular la señal recibida $r(t)$. Un filtro que se adapta a la forma de onda de la señal $s(t)$, donde $0 \leq t \leq T_b$, tiene una respuesta de impulsos

$$h(t) = s(T_b - t), \quad 0 \leq t \leq T_b \quad (5.2.13)$$

En consecuencia, la señal de forma de onda, es decir, $y(t)$ a la salida del filtro adaptado cuando la forma de onda de entrada es $s(t)$ viene dada por la integral de convolución

$$y(t) = \int_0^t s(\tau)h(t - \tau) d\tau \quad (5.2.14)$$

Si sustituimos en (5.2.14) por $h(t - T)$ a partir de (5.2.13), obtenemos

$$y(t) = \int_0^t s(\tau) s(T_b - t + \tau) d\tau \quad (5.2.15)$$

Y si la muestra $y(t)$ en $t = T_b$, obtenemos

$$y(T_b) = \int_0^{T_b} s^2(t) dt = E \quad (5.2.16)$$

Donde E es la energía de la señal $s(t)$. Por lo tanto, la salida de filtro adaptado en el instante de muestreo $t = T_b$ es idéntica a la salida del correlador de señal.

Problema ilustrativo 5.3 [Filtro adaptado] Considere el uso de filtros adaptados para la demodulación de las dos formas de onda de señal que se muestran en la Figura 5.2 y determinar las salidas.

Solución

Las respuestas de impulso de los dos filtros adaptados son

$$\begin{aligned} h_0(t) &= s_0(T_b - t) \\ h_1(t) &= s_1(T_b - t) \end{aligned} \quad (5.2.17)$$

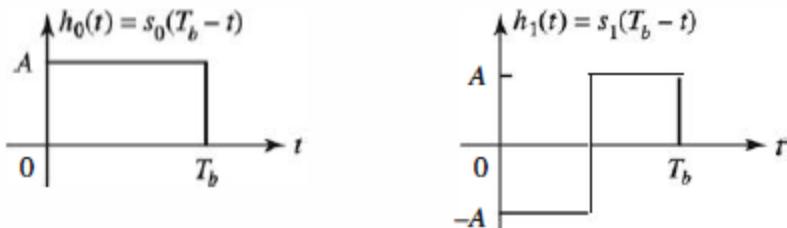


Figura 5.6: respuestas de impulso de filtros adaptados para señales $s_0(t)$ y $s_1(t)$

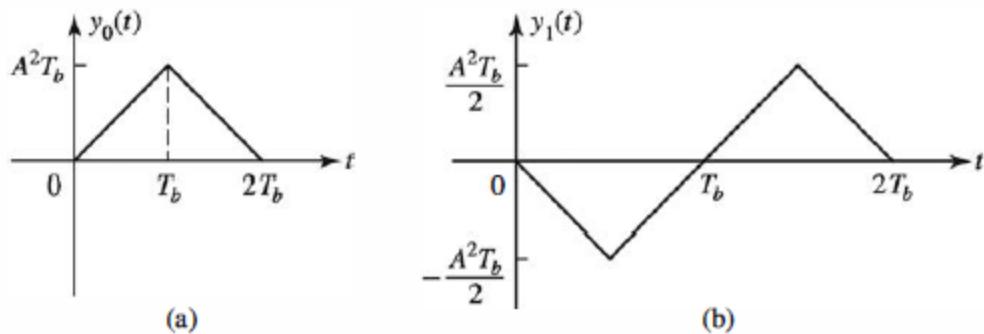


Figura 5.7: Las salidas de señales de filtros adaptados cuando $s_0(t)$ se transmite

Como se ilustran en la Figura 5.6. Tenga en cuenta que cada respuesta de impulso se obtiene por plegado de la señal $s(t)$ para obtener $s(-t)$ y luego el retraso de la señal plegada $s(-t)$ por T_b para obtener $s(T_b - t)$.

Supongamos ahora que la forma de onda de señal $s_0(t)$ se transmite. A continuación, la señal recibida $r(t) = s_0(t) + n(t)$ se pasa a través de los dos filtros adaptados. La respuesta del filtro con respuesta al impulso $h_0(t)$ a la componente de señal $s_0(t)$ se ilustra en la Figura 5.7 (a). Además, la respuesta del filtro con respuesta al impulso $h_1(t)$ a la componente de señal de $s_0(t)$ se ilustra en la figura 5.7 (b). Por lo tanto, en el instante de muestreo $t = T_b$, las salidas de los dos filtros adaptados con respuestas de impulso $h_0(t)$ y $h_1(t)$ son

$$\begin{aligned} r_0 &= E + n_0 \\ r_1 &= n_1 \end{aligned} \quad (5.2.18)$$

respectivamente. Tenga en cuenta que estas salidas son idénticas a las salidas obtenidas del muestreo de las salidas de señal de los correladores en $t = T_b$.

Problema ilustrativo 5.4 [Filtro adaptado de ondas de señal] Muestrear la forma de onda de la señal en el Problema ilustrativo 5.3 a una velocidad de $F_s = 20/T_b$ y realizar el filtrado adaptado de la señal recibida $r(t)$ con $s_0(t)$ y $s_1(t)$ numéricamente; es decir, a través de la computadora y graficar

$$y_0(kT_s) = \sum_{n=1}^k r(nT_s)s_0(kT_s - nT_s), \quad k = 1, 2, \dots, 20$$

$$y_1(kT_s) = \sum_{n=1}^k r(nT_s)s_1(kT_s - nT_s), \quad k = 1, 2, \dots, 20$$

Cuando (a) $s_0(t)$ es la señal transmitida y (b) $s_1(t)$ es la señal transmitida.

Repetir los cálculos y los gráficos cuando las muestras de la señal $r(KT_s)$ están degradados por las muestras del ruido aditivo gaussiano blanco $n(KT_s)$, para $1 \leq K \leq 20$, que tienen media cero y varianza $\sigma^2 = 0,1$ y $\sigma^2 = 1$ respectivamente.

Solución

La Figura 5.8 ilustra las salidas de filtro adaptado. Observamos el efecto del ruido aditivo sobre las salidas de filtro adaptado para las dos varianzas de ruido. Claramente, cuando $\sigma^2 = 1$, el efecto del ruido en las salidas de filtro adaptado es significativo. La secuencia de comandos de MATLAB para el cálculo a continuación

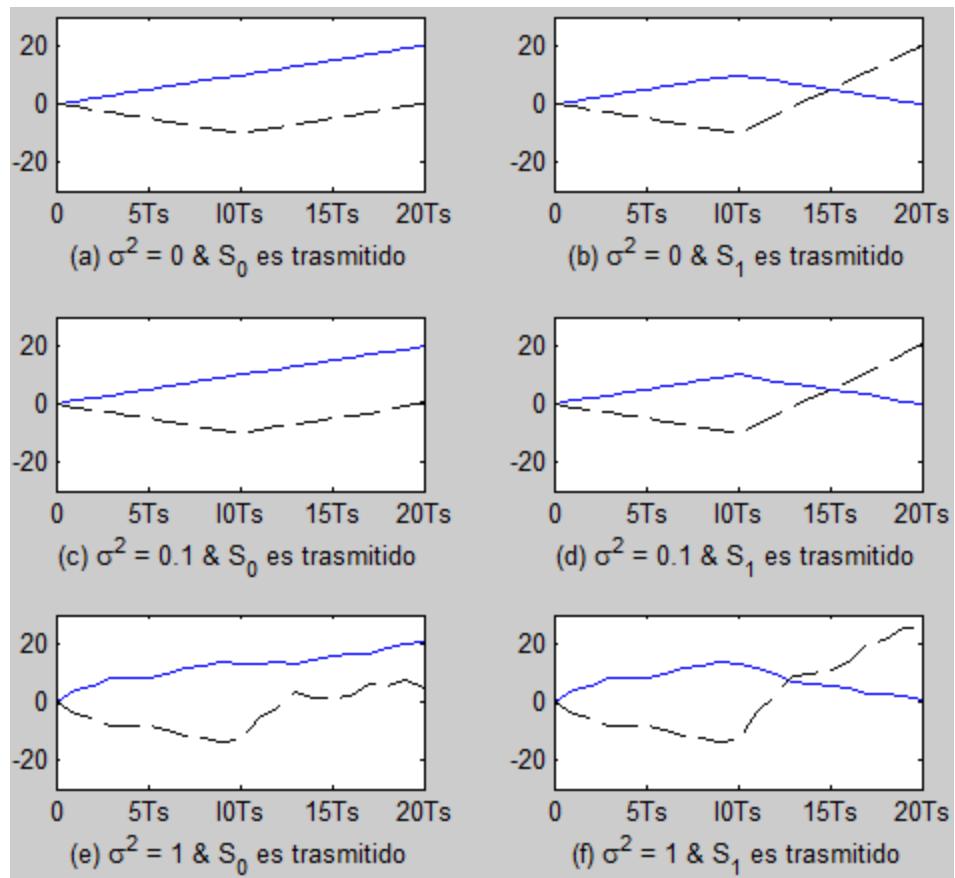


Figura 5.8: combinado de salida de los filtros del Problema ilustrativo 5.4. Las líneas continuas y discontinuas corresponden a las salidas de los filtros adaptados para $s_0(t)$ y $s_1(t)$, respectivamente.

M-FILE

```
% Secuencia de comandos de Problema ilustrativo 5.4
% Inicializacion
K=20; % numero de muestras
A=1; % Amplitud de Señal
l= 0:K; % vector con valores de 0 a K (paso por defecto 1 [0,1,...,19,20])
% Definicion de la forma de onda de la señal:
s_0 = A*ones(1,K);
figure(1)
plot(s_0,'linewidth',3)
s_1 =[A *ones(1,K) ];
s_1(11:K) =[s_1((K/2)+1:K) - 2*A *ones(1,K/2)];
figure(2)
plot(s_1,'linewidth',3)

% Inicializacion de señales de salida:
y_0 = zeros(1,K);
y_1 = zeros(1,K);

% Caso 1: ruido~N(0,0)--> de distribucion normal con media 0
noise=random('Normal',0,0,1,K);% ruido
% Sub-caso s = s_0:
s=s_0;
y=s+noise; % señal recibida
y_0= conv(y,wrev(s_0));
y_1= conv(y,wrev(s_1));

% Graficando resultados:
figure(3)
subplot(3,2,1)
plot(l,[0 y_0(1:K)],'--',l,[0 y_1(1:K)],'--k')
set(gca, 'XTickLabel', {'0','5Ts','10Ts','15Ts', '20Ts'})
axis([0 20 -30 30])
xlabel('(a) \sigma^2 = 0 & S_0 es transmitido','fontsize',10)
% Sub-caso s= s_1:
s=s_1;
y=s+noise; % señal recibida
y_0= conv(y,wrev(s_0));
y_1= conv(y,wrev(s_1));
```

```

% Graficando resultados:
subplot(3,2,2)
plot(l,[0 y_0(1:K)],'-',l,[0 y_1(1:K)],'--k')
set(gca, 'XTickLabel' ,{'0','5Ts','10Ts','15Ts', '20Ts'})
axis([0 20 -30 30])
xlabel('(b) \sigma^2 = 0 & S_{\{1\}} es transmitido','fontsize',10)

% Case 2: ruido-N(0,0.1)
noise=random('Normal',0,0.1,1,K); % ruido
% Sub-caso s = s_0:
s=s_0;
y=s+noise; % señal recibida
y_0= conv(y,wrev(s_0));
y_1= conv(y,wrev(s_1));

% Graficando resultados:
subplot(3,2,3)
plot(l,[0 y_0(1:K)],'-',l,[0 y_1(1:K)],'--k')
set(gca, 'XTickLabel' ,{'0','5Ts','10Ts','15Ts', '20Ts'})
axis([0 20 -30 30])
xlabel('(c) \sigma^2 = 0.1 & S_{\{0\}} es transmitido','fontsize',10)

% Sub-caso s= s_1:
s=s_1;
y=s+noise; % señal recibida
y_0= conv(y,wrev(s_0));
y_1= conv(y,wrev(s_1));

% Graficando resultados:
subplot(3,2,4)
plot(l,[0 y_0(1:K)],'-',l,[0 y_1(1:K)],'--k')
set(gca, 'XTickLabel' ,{'0','5Ts','10Ts','15Ts', '20Ts'})
axis([0 20 -30 30])
xlabel('(d) \sigma^2 = 0.1 & S_{\{1\}} es transmitido','fontsize',10)

% Case 3: ruido~N(0,1)
noise=random('Normal',0,1,1,K); % ruido
% Sub-caso s = s_0:
s=s_0;
y=s+noise; % señal recibida
y_0= conv(y,wrev(s_0));
y_1= conv(y,wrev(s_1));

```

```

% Graficando resultados:
subplot(3,2,5)
plot(1,[0 y_0(1:K)],'-',1,[0 y_1(1:K)],'--k')
set(gca, 'XTickLabel' ,{'0','5Ts','10Ts','15Ts', '20Ts'})
axis([0 20 -30 30])
xlabel('(e) \sigma^2 = 1 & S_0 es transmitido','fontsize',10)

% Sub-caso s= s_1:
s=s_1;
y=s+noise; % señal recibida
y_0= conv(y,wrev(s_0));
y_1= conv(y,wrev(s_1));

% Graficando resultados:
subplot(3,2,6)
plot(1,[0 y_0(1:K)],'-',1,[0 y_1(1:K)],'--k')
set(gca, 'XTickLabel' ,{'0','5Ts','10Ts','15Ts', '20Ts'})
axis([0 20 -30 30])
xlabel('(f) \sigma^2 = 1 & S_1 es transmitido','fontsize',10)

```

Detector

El detector observa las salidas de los correladores o del filtro adaptado r_0 y r_1 y decide sobre si la forma de onda de señal transmitida es $s_0(t)$ o $s_1(t)$, que corresponden a la transmisión de un 0 o un 1, respectivamente. El detector óptimo se define como el detector que minimiza la probabilidad de error.

Problema ilustrativo 5.5 [Detección binaria] Consideremos el detector de las señales que se muestran en la Figura 5.2, que son igualmente probables y tienen energías iguales. El detector óptimo para estas señales compara r_0 y r_1 y decide que un 0 se transmitió cuando $r_0 > r_1$ y que un 1 se remitió cuando $r_1 > r_0$. Determine la probabilidad de error.

Solución

Cuando $s_0(t)$ es la forma de onda de señal de transmisión, la probabilidad de error es

$$P_e = P(r_1 > r_0) = P(n_1 > E + n_0) = P(n_1 - n_0 > E) \quad (5.2.19)$$

Debido a que n_1 y n_0 son variables aleatorias gaussianas de media cero, su diferencia $n_1 - n_0$ también es de media cero Gaussiana. La varianza de la variable aleatoria x es

$$E(x^2) = E[(n_1 - n_0)^2] = E(n_1^2) + E(n_0^2) - 2E(n_1 n_0) \quad (5.2.20)$$

Pero $E(\mathbf{n}_1 \mathbf{n}_0) = 0$, porque las formas de onda de señales son ortogonales; es decir,

$$\begin{aligned}
E(\mathbf{n}_1 \mathbf{n}_0) &= E \int_0^{T_b} \int_0^{T_b} s_0(t) s_1(\tau) \mathbf{n}(t) \mathbf{n}(\tau) dt d\tau \\
&= \frac{N_0}{2} \int_0^{T_b} \int_0^{T_b} s_0(t) s_1(\tau) \delta(t - \tau) dt d\tau \\
&= \frac{N_0}{2} \int_0^{T_b} s_0(t) s_1(t) dt \\
&= 0
\end{aligned} \tag{5.2.21}$$

Por lo tanto

$$E(x^2) = 2 \left(\frac{EN_0}{2} \right) = EN_0 \equiv \sigma_x^2 \tag{5.2.22}$$

Por lo tanto, la probabilidad de error es

$$\begin{aligned}
P_e &= \frac{1}{\sqrt{2\pi} \sigma_x} \int_E^\infty e^{-x^2/2\sigma_x^2} dx \\
&= \frac{1}{\sqrt{2\pi}} \int_{\sqrt{E/N_0}}^\infty e^{-x^2/2} dx \\
&= Q \left(\sqrt{\frac{E}{N_0}} \right)
\end{aligned} \tag{5.2.23}$$

La relación de la E/N_0 se conoce como la relación señal-ruido (SNR).

La derivación de la actuación del detector dado en el ejemplo se basa en la transmisión de la forma de onda de señal $s_0(t)$. El lector puede verificar que la probabilidad de error que se obtiene cuando se transmite $s_1(t)$ es idéntica a la obtenida cuando $s_0(t)$ se transmite. Debido a que el 0 y el 1 están en la secuencia de datos, por lo cual, son igualmente probables, la probabilidad media de error es la dada por (5.2.23). Esta expresión de la probabilidad de error se evalúa usando la secuencia de comandos MATLAB dada a continuación y se representa en la figura 5.9 como una función de la SNR, donde se muestra la SNR en una escala logarítmica ($10 \log_{10} (E/N_0)$). Como era de esperar, la probabilidad de error disminuye exponencialmente a medida que aumenta la SNR.

M-FILE

```
% Secuencia de comandos de MATLAB que genera la probabilidad de error
% frente a la relación de señal a ruido..
initial_snr=0;
final_snr=15;
snr_step=0.25;
snr_in_dB=initial_snr:snr_step:final_snr;
]for i=1 :length(snr_in_dB),
    snr=10^(snr_in_dB(i)/10);
    Pe(i)=Qfunct(sqrt(snr));
    echo off;
end;
echo on;
semilogy(snr_in_dB,Pe);
```

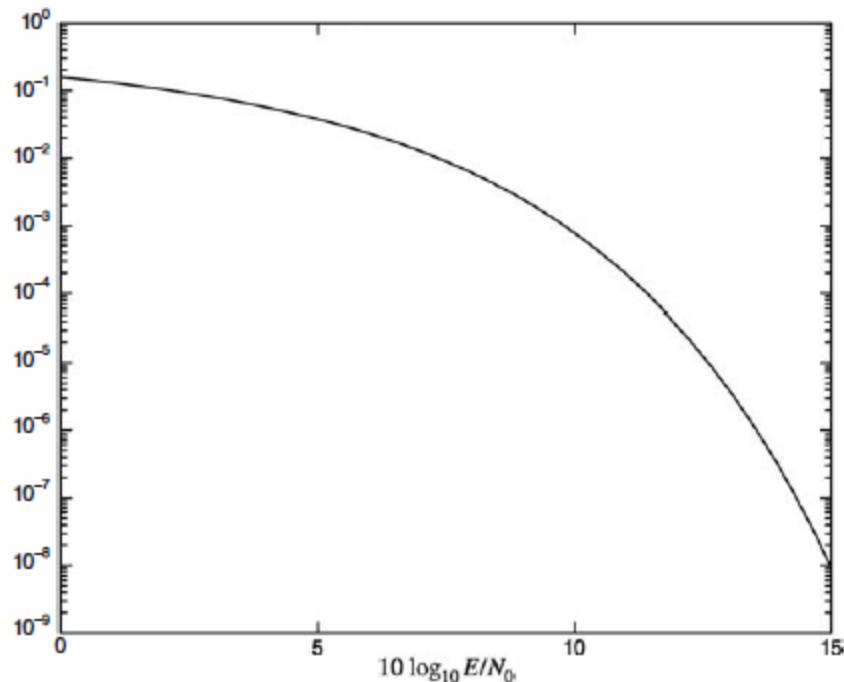


Figura 5.9: Probabilidad de error para señales ortogonales

Simulación Monte Carlo de Sistema de Comunicación Binario

Simulaciones por ordenador Monte Carlo se realizan generalmente en la práctica para estimar la probabilidad de error de un sistema de comunicación digital, especialmente en los casos en que el análisis de la actuación detector es difícil de realizar. Se demuestra el

método para la estimación de la probabilidad de error para el sistema de comunicación binario descrito anteriormente.

Problema ilustrativo 5.6 [Simulación de Montecarlo] Uso de Monte Carlo para estimar y graficar la trama Pe frente SNR para un sistema de comunicación que emplea correladores o filtros adaptados binario. El modelo del sistema se ilustra en la Figura 5.10.

Solución

Estamos simulando la generación de las variables aleatorias r_0 y r_1 , que constituyen la entrada al detector. Comenzamos generando una secuencia binaria de ceros y unos que se producen con igual probabilidad y son mutuamente estadísticamente independientes. Para llevar a cabo esta tarea, se utiliza un generador de números aleatorios que genera un número aleatorio uniforme con un intervalo $(0, 1)$. Si el número generado está en el intervalo $(0, 0.5)$, la salida de la fuente binaria es un 0. De lo contrario, es un 1. Si se genera un 0, entonces $r_0 = E + n_0$ y $r_1 = n_1$. Si se genera un 1, $r_0 = n_0$ y $r_1 = E + n_1$.

Los componentes de ruido aditivo n_0 y n_1 se generan por medio de dos generadores de ruido de Gauss. Sus medias son cero, y sus varianzas son un $\sigma^2 = (EN_0)/2$. Para mayor comodidad, podemos normalizar la energía de la señal a la unidad E ($E = 1$) y variando σ^2 .

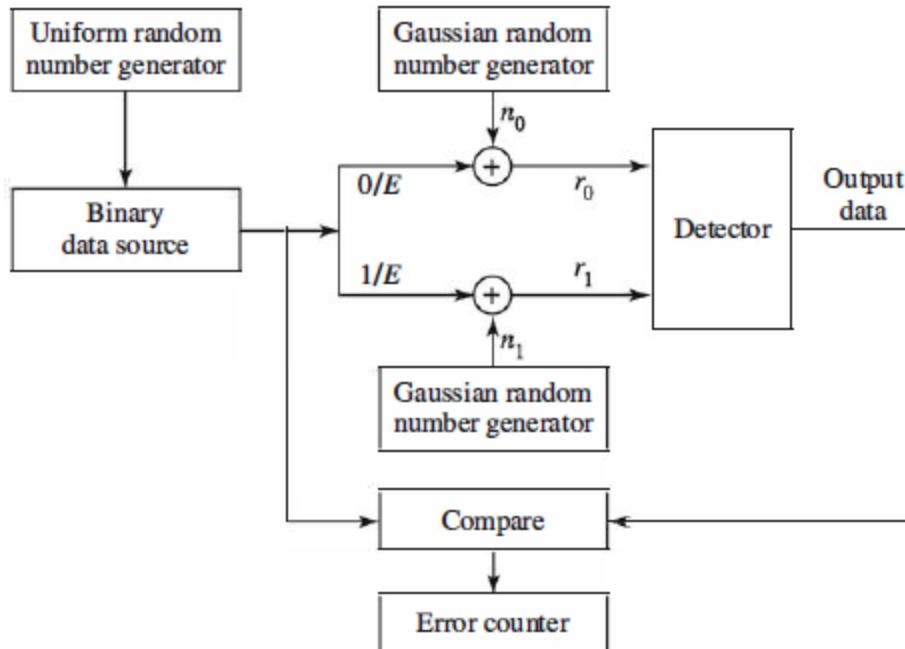


Figura 5.10: Modelo de simulación del Problema ilustrativo 5.6

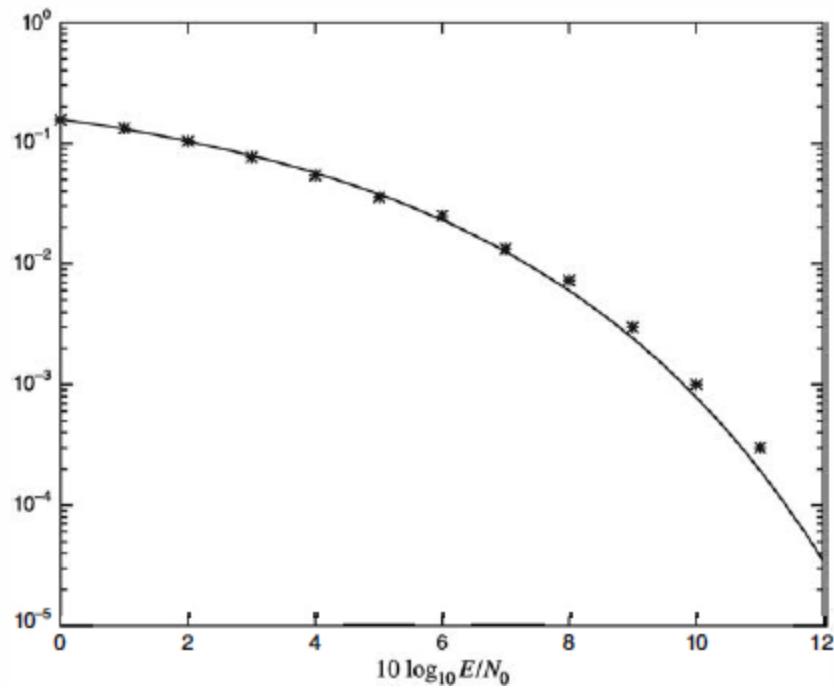


Figura 5 .11: la probabilidad de error de la simulación Monte Carlo en comparación con una probabilidad de error teórico para la señalización ortogonal

Tenga en cuenta que la SNR, que se define como E / N_0 , es entonces igual a $1/2\sigma^2$. La salida del detector se compara con la secuencia binaria de transmisión, y un contador de errores se utiliza para contar el número de errores de bit.

La Figura 5.11 ilustra los resultados de esta simulación para la transmisión de $N = 10.000$ bits a varios valores diferentes de SNR. Tenga en cuenta el acuerdo entre los resultados de la simulación y el valor teórico de P_e dada por (5.2.23). También hay que señalar que una simulación de $N = 10.000$ bits de datos nos permite estimar la probabilidad de error de forma fiable alrededor de $P_e = 10^{-3}$. En otras palabras, con los $N = 10.000$ bits de datos, deberíamos tener al menos diez errores para una estimación fiable de P_e . Las secuencias de comandos de MATLAB para este problema se dan a continuación.

M-FILE

```
% Secuencia de comandos en MATLAB del problema ilustrativo 5.6.
echo on
SNRindB1=0:1:12;
SNRindB2=0:0.1:12;
for i=1:length(SNRindB1),
    % tasa de error simulada
    smld_err_prb(i)=smldPe54(SNRindB1(i));
    echo off ;
end;
echo on ;
for i=1 :length(SNRindB2),
    SNR=exp(SNRindB2(i)*log(10)/10);
    % tasa de error teorica
    theo_err_prb(i)=Qfunct(sqrt(SNR));
    echo off ;
end;
echo on;
% Graficando los siguientes comandos
semilogy(SNRindB1,smld_err_prb,'*');
hold
semilogy(SNRindB2,theo_err_prb);
```

M-FILE

```
|function [p]=smldPe54(snr_in_dB)
|% [p]=smldPe54(snr_in....dB)
% SMWPE54 encuentra la probabilidad de error para la SNR dada
% En....dB, relacion señal ruido en dB.
E=1;
SNR=exp(snr_in_dB*log(10)/10); % relacion señal ruido
sgma=E/sqrt(2*SNR); % sigma: desviacion estandar del ruido
N=10000;
% Generacion de la fuente de datos binarios
for i=1:N,
    temp=rand; % una variable aleatoria uniforme entre (0,1)
    if (temp<0.5),
        dsOURCE(i)=0; % Con una probabilidad de 1/2, para la salida de 0s
    else
        dsOURCE(i)=1;% Con una probabilidad de 1/2, para la salida de 1s
    end
end;

% Deteccion, y calculo del error de probabilidad
numoferr=0;
for i=1:N,
    % Salidas de filtro adaptado
    if (dsOURCE(i)==0),
        r0=E+gngauss(sgma);
        rl=gngauss(sgma); % si la salida generada es "0"
    else
        r0=gngauss(sgma);
        rl=E+gngauss(sgma);% si la salida generada es "1"
    end;
    % Detector follows.
    if (r0>rl),
        decis=0; % la desicion es "0"
    else
        decis=1; % la desicion es "1"
    end;
    if (decis~=dsOURCE(i)), % Si sa trata de un error, aumentamos
        numoferr=numoferr+ 1 ; % el contador de errores
    end;
end;
p=numoferr/N; % Estimacion de la probabilidad de error
```

Pregunta

En la Figura 5.11, la simulación y los resultados teóricos coinciden por completo a baja relación señal-ruido, mientras que en las SNR más altas coinciden menos. ¿Puedes explicar porque? ¿Cómo debemos cambiar el proceso de simulación para dar lugar a una mejor coincidencia en la alta relación señal ruido?

5.2.2 Otros modos de transmisión de señales digitales binarias

El método de transmisión de señal digital descrito anteriormente se basa en el uso de señales ortogonales. A continuación, se describen otros dos métodos para la transmisión de información binaria a través de un canal de comunicación. Un método emplea señales antípodas. El otro método emplea una señal de encendido y apagado.

Señales antípodas para la transmisión de la señal binaria

Dos formas de onda de señal se dice que son antípodas si una forma de onda de la señal es el negativo de la otra. Por ejemplo, un par de señales de antípodas se ilustra en la Figura 5.12 (a). Un segundo par se ilustra en la Figura 5.12 (b).

Supongamos que utilizamos formas de onda de señal antípoda $s_0(t) = s(t)$ y $s_1(t) = -s(t)$ para transmitir información binaria, donde $s(t)$ es una forma de onda que tiene la energía arbitraria E. La forma de onda recibida señal de una canal AWGN puede expresarse como

$$r(t) = \pm s(t) + n(t), \quad 0 \leq t \leq T_b \quad (5.2.24)$$

El receptor óptimo para la recuperación de la información binaria emplea un único correlador o un solo filtro adaptado a $s(t)$, seguido de un detector, como se ilustra en la Figura 5.13. Supongamos que $s(t)$ se transmite, de modo que la señal recibida es

$$r(t) = s(t) + n(t) \quad (5.2.25)$$

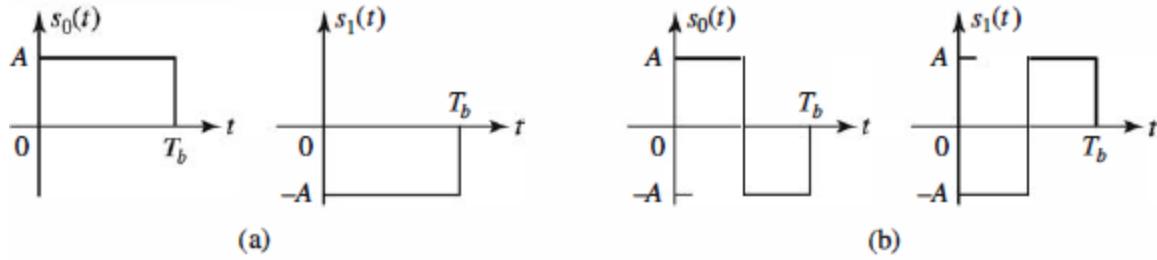


Figura 5.12: Ejemplos de señales antípodas. (a) un par de señales antípodas. (b) Otro par de señales antípodas

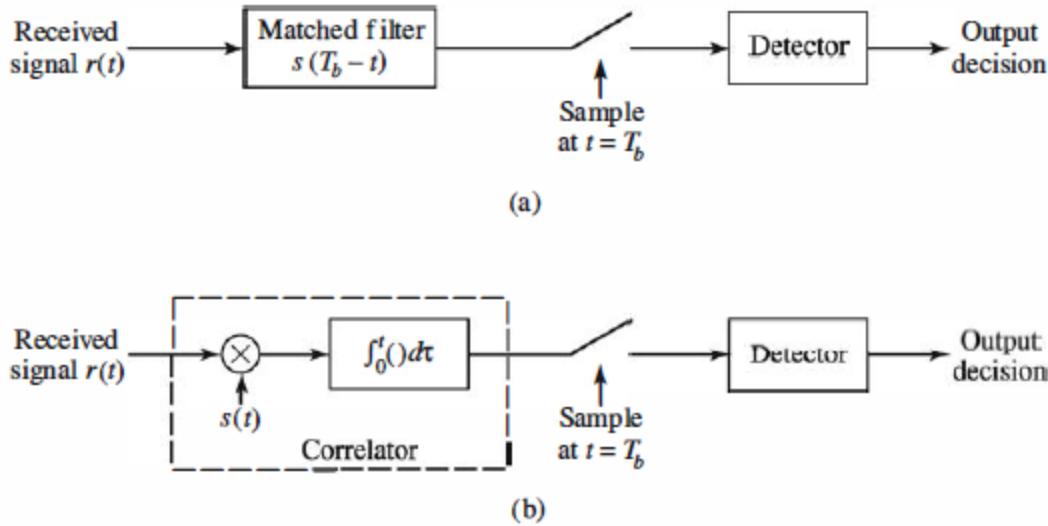


Figura 5.13: óptimos del receptor para señales antípodas. (a) demodulador de filtro acoplado. (a) demodulador de Correladores.

La salida del correlador y filtro adaptado en el instante de muestreo $t = T_b$ es

$$r = E + n \quad (5.2.26)$$

Donde E es la energía de la señal y n es la componente de ruido aditivo, que se expresa como

$$n = \int_0^{T_b} n(t)s(t) dt \quad (5.2.27)$$

Debido a que el proceso de ruido aditivo $n(t)$ es una media de cero, se deduce que $E(n) = 0$. La varianza de la componente de ruido n es

$$\begin{aligned}
\sigma^2 &= E(n^2) \\
&= \int_0^{T_b} \int_0^{T_b} E[n(t)n(\tau)] s(t)s(\tau) dt d\tau \\
&= \frac{N_0}{2} \int_0^{T_b} \int_0^{T_b} \delta(t - \tau) s(t)s(\tau) dt d\tau \\
&= \frac{N_0}{2} \int_0^{T_b} s^2(t) dt = \frac{N_0 E}{2}
\end{aligned} \tag{5.2.28}$$

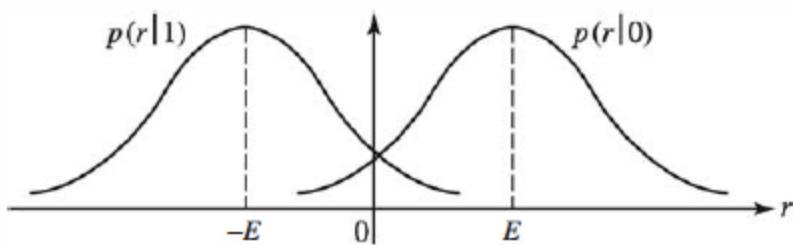


Figura 5.14: Función de densidad de probabilidad para la salida del detector.

En consecuencia, la función de densidad de probabilidad de r cuando $s(t)$ se transmite es

$$p(r | s(t) \text{ was transmitted}) \equiv p(r | 0) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(r-E)^2/2\sigma^2} \tag{5.2.29}$$

Del mismo modo, cuando la forma de onda de la señal $-s(t)$ se transmite, la entrada al detector es

$$r = -E + n \tag{5.2.30}$$

Y la función densidad de probabilidad de r es

$$p(r | -s(t) \text{ was transmitted}) \equiv p(r | 1) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(r+E)^2/2\sigma^2} \tag{5.2.31}$$

Estas dos funciones de densidad de probabilidad se ilustran en la Figura 5.14.

Para las formas de onda de señales igualmente probables, el detector óptimo compara r con el umbral cero. Si $r > 0$, se toma la decisión de que $s(t)$ se transmite. Si $r < 0$, se toma la decisión de que $-s(t)$ se transmitió. El ruido que corrompe la señal provoca errores en el detector. La probabilidad de un error detector se calcula fácilmente. Supongamos que $s(t)$ se transmitió. Entonces, la probabilidad de error es igual a la probabilidad de que $r < 0$; es decir,

$$\begin{aligned}
P_e &= P(r < 0) \\
&= \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^0 e^{-(r-E)^2/2\sigma^2} dr \\
&= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{-E/\sigma} e^{-r^2/2} dr \\
&= Q\left(\frac{E}{\sigma}\right) \\
&= Q\left(\sqrt{\frac{2E}{N_0}}\right)
\end{aligned} \tag{5.2.32}$$

Un resultado similar se obtiene cuando se transmite $-s(t)$. En consecuencia, cuando las dos formas de onda de señales son igualmente probables, la probabilidad media de error viene dado por (5.2.32).

Cuando comparamos la probabilidad de error para señales antípodas con la usada para señales ortogonales dadas por (5.2.23), se observa que, para el mismo transmite energía de la señal E, por lo que las señales antípodas dan como resultado un mejor rendimiento. Alternativamente, podemos decir que las señales antípodas dan el mismo rendimiento (la misma probabilidad de error) como señales ortogonales mediante el uso de la mitad de la energía transmitida de señales ortogonales. Por lo tanto, las señales son antípodas de 3dB son más eficiente que las señales ortogonales.

Problema ilustrativo 5.7 [La correlación de ondas de señal antípodas] En las antípodas de señalización, la forma de onda de señal de $s(t)$ se define como

$$s_0(t) = A, \quad 0 \leq t \leq T_b$$

y cero en caso contrario. La señal recibida es $r(t) = \pm s_0(t) + n(t)$. Muestrear la señal recibida $r(t)$ y $s_0(t)$ con una la tasa de $F_s = 20/T_b$ y llevar a cabo numéricamente la correlación de $r(t)$ con $s_0(t)$; es decir, a través de la computadora y la graficar

$$r_0(kT_s) = \sum_{n=1}^k r(nT_s)s_0(nT_s), \quad k = 1, 2, \dots, 20$$

Cuando (a) $s_0(t)$ es la señal transmitida y (b) $-s_0(t)$ es la señal transmitida. Realizar los cálculos cuando el ruido gaussiano aditivo tiene media cero y varianzas $\sigma^2 = 0$, $\sigma^2 = 0,1$ y $\sigma^2 = 1$.

Solución

La Figura 5.15 ilustra las salidas de correlación de las variaciones diferentes de ruido. La secuencia de comandos de MATLAB para el cálculo se a continuación

M-FILE

```
% MATLAB script for Illustrative Problem 5.7.

% Initialization:
K=20; % Number of samples
A=1; % Signal amplitude
l=0:K;
s_0=A*ones(1,K); % Signal waveform
r_0=zeros(1,K); % Output signal

% Case 1: noise~N(0,0)
noise=random('Normal',0,0,1,K);
% Sub-case s = s_0:
s=s_0;
r=s+noise; % received signal
for n=1:K
    r_0(n)=sum(r(1:n).*s_0(1:n));
end
% Plotting the results:
subplot(3,2,1)
plot(l,[0 r_0])
set(gca,'XTickLabel',{'0','5Tb','10Tb','15Tb','20Tb'})

axis([0 20 0 25])
xlabel('(a) \sigma^2 = 0 & S_0 is transmitted ','fontsize',10)
% text(15,3,'fontsize{10} r_0: - & r_1: -','hor','left')
% Sub-case s = s_1:
s=-s_0;
r=s+noise; % received signal
for n=1:K
    r_0(n)=sum(r(1:n).*s_0(1:n));
end
% Plotting the results:
subplot(3,2,2)
plot(l,[0 r_0])
set(gca,'XTickLabel',{'0','5Tb','10Tb','15Tb','20Tb'})
axis([0 20 -25 0])
xlabel('(b) \sigma^2 = 0 & S_1 is transmitted ','fontsize',10)

% Case 2: noise~N(0,0.1)
noise=random('Normal',0,0.1,1,K);
% Sub-case s = s_0:
s=s_0;
r=s+noise; % received signal
for n=1:K
    r_0(n)=sum(r(1:n).*s_0(1:n));
end
% Plotting the results:
```

```

% Case 3: noise~N(0,1)
noise=random('Normal',0,1,1,K);
% Sub-case s = s_0:
s=s_0;
r=s+noise; % received signal
for n=1:K
    r_0(n)=sum(r(1:n).*s_0(1:n));
end
% Plotting the results:
subplot(3,2,5)
plot(l,[0 r_0])
set(gca,'XTickLabel',{'0','5Tb','10Tb','15Tb','20Tb'})
axis([0 20 -5 25])
xlabel('(e) \sigma^2= 1 & S_{0} is transmitted ','fontsize',10)
% Sub-case s = s_1:
s=-s_0;
r=s+noise; % received signal
for n=1:K
    r_0(n)=sum(r(1:n).*s_0(1:n));
end
% Plotting the results:
subplot(3,2,6)
plot(l,[0 r_0])
set(gca,'XTickLabel',{'0','5Tb','10Tb','15Tb','20Tb'})
axis([0 20 -25 5])
xlabel('(f) \sigma^2= 1 & S_{1} is transmitted ','fontsize',10)

```

```

% Case 3: noise~N(0,1)
noise=random('Normal',0,1,1,K);
% Sub-case s = s_0:
s=s_0;
r=s+noise; % received signal
for n=1:K
    r_0(n)=sum(r(1:n).*s_0(1:n));
end
% Plotting the results:
subplot(3,2,5)
plot(l,[0 r_0])
set(gca,'XTickLabel',{'0','5Tb','10Tb','15Tb','20Tb'})
axis([0 20 -5 25])
xlabel('(e) \sigma^2= 1 & S_{0} is transmitted ','fontsize',10)
% Sub-case s = s_1:
s=-s_0;
r=s+noise; % received signal
for n=1:K
    r_0(n)=sum(r(1:n).*s_0(1:n));
end
% Plotting the results:
subplot(3,2,6)
plot(l,[0 r_0])
set(gca,'XTickLabel',{'0','5Tb','10Tb','15Tb','20Tb'})
axis([0 20 -25 5])
xlabel('(f) \sigma^2= 1 & S_{1} is transmitted ','fontsize',10)

```

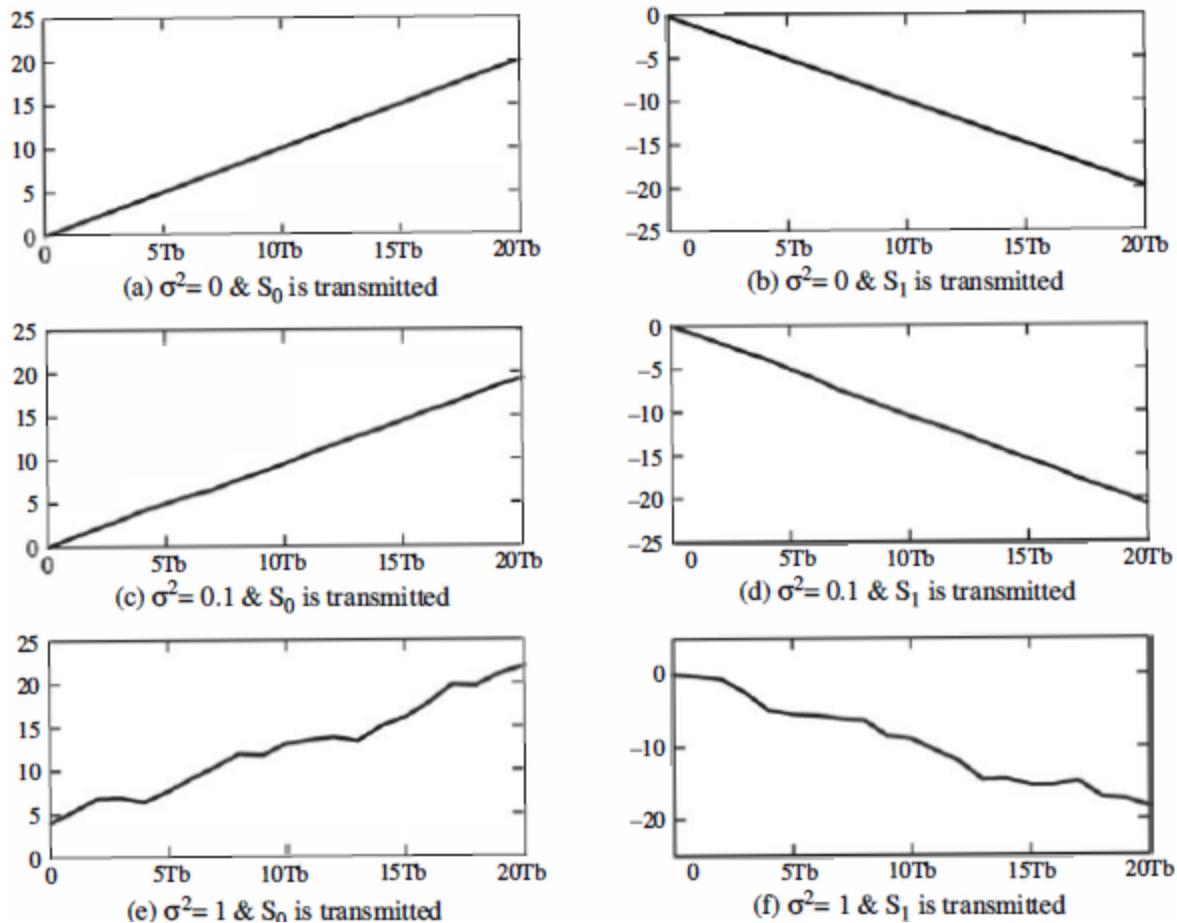


Figura 5.15: salidas del correlador en el Problema ilustrativo 5.7

Problema ilustrativo 5.8 [Simulación de señal antípoda binaria] Utilizar la simulación de Monte Carlo para estimar el rendimiento y la grafico de la probabilidad de error de un sistema de comunicación antípoda binario. El modelo del sistema se ilustra en la Figura 5.16.

Solución

Como se muestra, se simula la generación de la variable aleatoria r , que es la entrada al detector. Un generador de números aleatorio uniforme se utiliza para generar la secuencia de información binaria procedente de la fuente de datos binarios. La secuencia de ceros y unos se hace corresponder a una secuencia de $\pm E$, donde E representa la energía de la señal. E puede ser normalizada a la unidad. Un generador de ruido Gaussiano se utiliza para generar la secuencia de números aleatorios gaussianos con media cero y con varianza σ^2 . El detector compara la variable aleatoria r con el umbral de 0. Si $r > 0$, se toma la decisión de que el bit transmitido es un 0. Si $r < 0$, se toma la decisión de que el bit transmitido es un 1. La salida del detector se compara con la secuencia de transmisión de

bits de información, y se cuentan los errores de bit. La Figura 5.17 ilustra los resultados de esta simulación para la transmisión de 10000 bits a varios valores diferentes de SNR. El valor teórico de P_e dada por (5.2.32) también se representa en la figura 5.17 para la comparación. Tenga en cuenta que para 10.000 bits transmitidos, las estimaciones de Monte Carlo de P_e siguientes con 10^{-3} no son muy precisas.

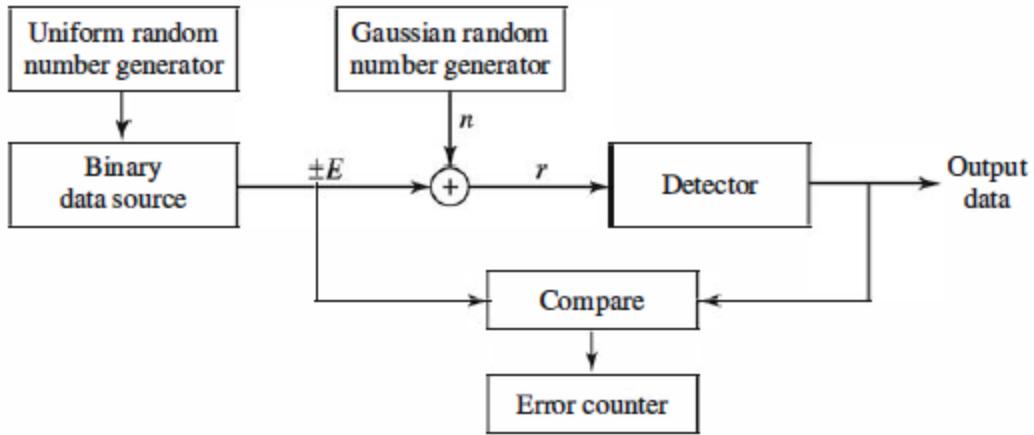


Figura 5.16: Modelo del sistema de comunicación que emplea señales binarias antípodas

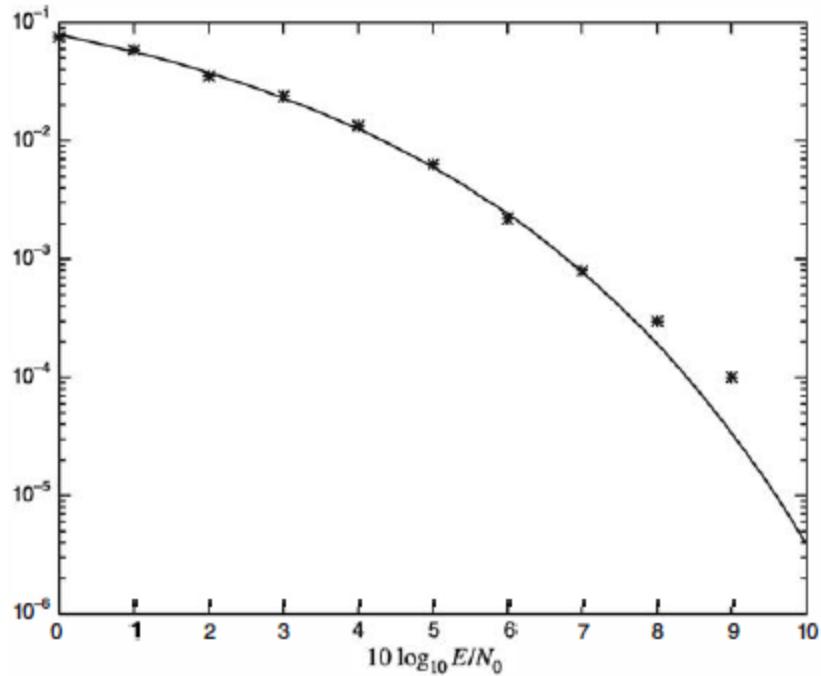


Figura 5.17: la probabilidad de error de la Simulación de Monte Carlo en comparación con una probabilidad de error teórica para señales antípodas

La secuencia de comandos de MATLAB del problema son las siguientes.

M-FILE

```
% MATLAB script for Illustrative Problem 5.8.  
echo on  
SNRindB1=0:1:10;  
SNRindB2=0:0.1:10;  
for i=1:length(SNRindB1),  
    % simulated error rate  
    smld_err_prb(i)=smldPe55(SNRindB1(i));  
    echo off;  
end;  
echo on;  
for i=1:length(SNRindB2),  
    SNR=exp(SNRindB2(i)*log(10)/10);  
    % theoretical error rate  
    theo_err_prb(i)=Qfunct(sqrt(2*SNR));  
    echo off;  
end;  
echo on;  
% Plotting commands follow.  
semilogy(SNRindB1,smld_err_prb, ' * ' );  
hold  
semilogy(SNRindB2,theo_err_prb);
```

M-FILE

```

function [p]=smldPe55(snr_in_dB)
% [p]=smldPe55(snr_in_dB)
% SMLDPE55 simulates the probability of error for the particular
% value of snr_in_dB, signal-to-noise ratio in dB.
E=1;
SNR=exp(snr_in_dB*log(10)/10); % signal-to-noise ratio
sgma=E/sqrt(2*SNR); % sigma, standard deviation of noise
N=10000;
% Generation of the binary data source follows.
for i=1:N,
    temp=rand; % a uniform random variable over (0,1)
    if (temp<0.5),
        dsoure(i)=0; % With probability 1/2, source output is 0.
    else
        dsoure(i)=1; % With probability 1/2, source output is 1.
    end
end;
% The detection, and probability of error calculation follows.
numoferr=0;
for i=1:N,
    % the matched filter outputs
    if (dsoure(i)==0),
        r=-E+gngauss(sgma); % if the source output is "0"
    else
        r=E+gngauss(sgma); % if the source output is "1"
    end;
    % Detector follows.
    if (r<0),
        decis=0; % Decision is "0".
    else
        decis=1; % Decision is "1".
    end;
    if (decis~=dsoure(i)),
        numoferr=numoferr+1; % If it is an error, increase the error counter.
    end;
end;
p=numoferr/N; % probability of error estimate

```

Señales de encendido y apagado (on-off) para la transmisión de la señal binaria

Una secuencia de información binaria también puede transmitirse mediante el uso de señales de encendido y apagado. Para transmitir un 0, no hay señal se transmite en el intervalo de tiempo de duración T_b . Para transmitir un 1, una forma de onda de la señal $s(t)$ se transmite. En consecuencia, la forma de onda de la señal recibida se puede representar como

$$r(t) = \begin{cases} n(t), & \text{if a 0 is transmitted} \\ s(t) + n(t), & \text{if a 1 is transmitted} \end{cases} \quad (5.2.33)$$

Donde $n(t)$ representa el ruido gaussiano blanco aditivo.

Como en el caso de señales antípodas, el receptor óptimo consiste en un correlador o un filtro adaptado a $s(t)$, cuya salida es muestreada a $t = T_b$, y seguido por un detector que compara la salida muestreada al umbral, que se denota como α . Si $r > \alpha$, se determina que un 1 ha sido transmitido; de lo contrario, se determina que un 0 ha sido transmitido.

La entrada al detector se puede expresar como

$$r = \begin{cases} n, & \text{if a 0 is transmitted} \\ E + n, & \text{if a 1 is transmitted} \end{cases} \quad (5.2.34)$$

Donde n es una variable aleatoria gaussiana de media cero con varianza $\sigma^2 = EN_0/2$. Por lo tanto, las funciones de densidad de probabilidad condicional de la variable aleatoria r son

$$\begin{aligned} p(r | 0) &= \frac{1}{\sqrt{2\pi}\sigma} e^{-r^2/2\sigma^2}, && \text{if a 0 is transmitted} \\ p(r | 1) &= \frac{1}{\sqrt{2\pi}\sigma} e^{-(r-E)^2/2\sigma^2}. && \text{if a 1 is transmitted} \end{aligned}$$

Estas funciones de densidad de probabilidad se ilustran en la figura 5.18.

Cuando un 0 se transmite, la probabilidad de error es

$$P_{e0}(\alpha) = P(r > \alpha) = \frac{1}{\sqrt{2\pi}\sigma} \int_{\alpha}^{\infty} e^{-r^2/2\sigma^2} dr \quad (5.2.35)$$

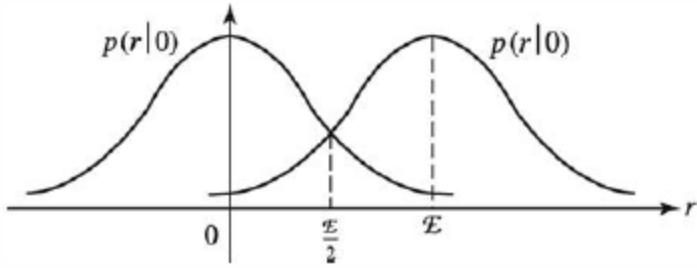


Figura 5.18: Las funciones de densidad de probabilidad de la señal recibida a la salida del correlador para las señales de encendido y apagado

Donde α es el umbral. Por otro lado, cuando se transmite un 1, la probabilidad de error es

$$P_{e1}(\alpha) = P(r < \alpha) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\alpha} e^{-(r-E)^2/2\sigma^2} dr \quad (5.2.36)$$

Suponiendo que los bits de información binaria son igualmente probables, tenemos para la probabilidad media de error:

$$P_e(\alpha) = \frac{1}{2}P_{e0}(\alpha) + \frac{1}{2}P_{e1}(\alpha) \quad (5.2.37)$$

El valor de la α umbral que minimiza la probabilidad media de error se encuentra diferenciando $P_e(\alpha)$ y resolviendo para el umbral óptimo. Se muestra fácilmente que el umbral óptimo es

$$\alpha_{\text{opt}} = \frac{E}{2} \quad (5.2.38)$$

La sustitución de este valor óptimo en (5.2.35), (5.2.36) y (5.2.37) da la probabilidad de error

$$P_e(\alpha_{\text{opt}}) = Q\left(\sqrt{\frac{E}{2N_0}}\right) \quad (5.2.39)$$

Se observa que el rendimiento de tasa de error con las señales de encendido-apagado no es tan buena como con las señales de antípodas. Parece ser 6 dB peor que con las señales de antípodas y 3 dB peor que con las señales ortogonales. Sin embargo, el promedio de energía transmitida para las señales de encendido y apagado es 3 dB menor que para las señales de antípodas y ortogonales. En consecuencia, esta diferencia debe tenerse en cuenta en la hora de hacer comparaciones de la actuación con otros tipos de señales.

Problema ilustrativo 5.9 [Correlación de señales de ondas encendido-apagado] La señal recibida en la transmisión de la señal de encendido y apagado se da como

$$r(t) = \begin{cases} n(t), & \text{if a 0 is transmitted} \\ s(t) + n(t), & \text{if a 1 is transmitted} \end{cases}$$

Donde $n(t)$ representa el ruido aditivo blanco gaussiano y $s(t)$ se define como

$$s(t) = A, \quad 0 \leq t \leq T_b$$

y cero en caso contrario. Supongamos que la señal recibida $r(t)$ se muestrea a la tasa de $F_s = 30/T_b$ y la correlación en el receptor se realiza numéricamente; es decir,

$$y(kT_s) = \sum_{n=1}^k r(nT_s)s(nT_s), \quad k = 1, 2, \dots, 30$$

Repetir los cálculos y los gráficos cuando las muestras de la señal $y(kT_s)$ están degradados por las muestras del ruido aditivo gaussiano blanco $n(kT_s)$, para $1 \leq k \leq 30$, que tienen varianza $\sigma^2 = 0$, $\sigma^2 = 0,1$ y $\sigma^2 = 1$ respectivamente.

Solución

La Figura 5.19 ilustra las salidas de correlación para las diferentes variaciones de ruido. La secuencia de comandos MATLAB para el cálculo es la siguiente

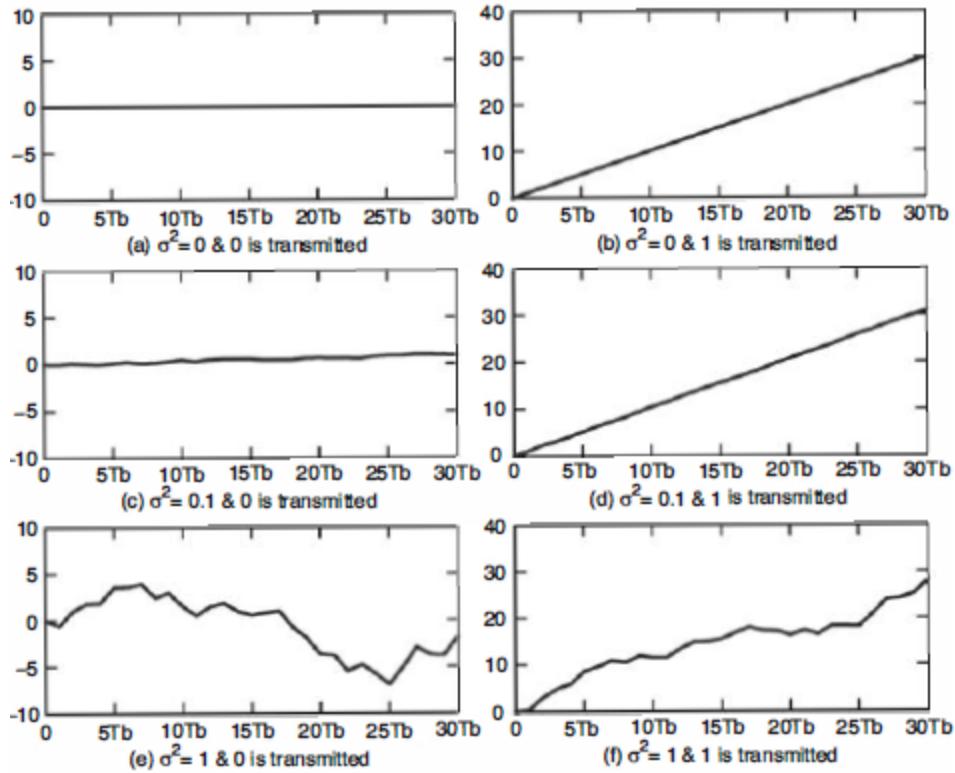


Figura 5.19: Salidas del correlador en el Problema ilustrativo 5.9

M-FILE

```
% MATLAB script for Illustrative Problem 5.9.
```

```
% Initialization:  
K=30; % Number of samples  
A=1; % Signal amplitude  
l=0:K;  
s=A*ones(1,K); % Signal waveform  
y=zeros(1,K); % Output signal
```

```

% Case1: noise~N(0,0)
noise=random('Normal',0,0,1,K);
% Sub-case: 0 is transmitted
r=noise; % received signal
for n=1:K
    y(n)=sum(r(1:n).*s(1:n));
end
% Plotting the results:
subplot(3,2,1)
plot(l,[0 y])
set(gca,'XTickLabel',{'0','5Tb','10Tb','15Tb','20Tb','25Tb','30Tb'})
axis([0 30 -10 10])
xlabel('(a) \sigma^2= 0 & 0 is transmitted','fontsize',10)
% Sub-case: 1 is transmitted
r=s+noise; % received signal
for n=1:K
    y(n)=sum(r(1:n).*s(1:n));
end

% Plotting the results:
subplot(3,2,2)
plot(l,[0 y])
set(gca,'XTickLabel',{'0','5Tb','10Tb','15Tb','20Tb','25Tb','30Tb'})
axis([0 30 0 40])
xlabel('(b) \sigma^2= 0 & 1 is transmitted','fontsize',10)
% Case2: noise~N(0,0.1)
noise=random('Normal',0,0,1,1,K);
% Sub-case: 0 is transmitted
r=noise; % received signal
for n=1:K
    y(n)=sum(r(1:n).*s(1:n));
end
% Plotting the results:
subplot(3,2,3)
plot(l,[0 y])
set(gca,'XTickLabel',{'0','5Tb','10Tb','15Tb','20Tb','25Tb','30Tb'})
axis([0 30 -10 10])
xlabel('(c) \sigma^2= 0.1 & 0 is transmitted','fontsize',10)
% Sub-case: 1 is transmitted
r=s+noise; % received signal
for n=1:K
    y(n)=sum(r(1:n).*s(1:n));
end

```

```

% Plotting the results:
subplot(3,2,4)
plot(l,[0 y])
set(gca,'XTickLabel',{'0','5Tb','10Tb','15Tb','20Tb','25Tb','30Tb'})
axis([0 30 0 40])
xlabel('(d) \sigma^2= 0.1 & 1 is transmitted', 'fontsize',10)
% Case3: noise~N(0,1)
noise=random('Normal',0,1,1,K);
% Sub-case: 0 is transmitted
r=noise; % received signal
for n=1:K
    y(n)=sum(r(1:n).*s(1:n));
end

% Plotting the results:
subplot(3,2,5)
plot(l,[0 y])
set(gca,'XTickLabel',{'0','5Tb','10Tb','15Tb','20Tb','25Tb','30Tb'})
axis([0 30 -10 10])
xlabel('(e) \sigma^2= 1 & 0 is transmitted', 'fontsize',10)
% Sub-case: 1 is transmitted
r=s+noise; % received signal
for n=1:K
    y(n)=sum(r(1:n).*s(1:n));
end

% Plotting the results:
subplot(3,2,6)
plot(l,[0 y])
set(gca,'XTickLabel',{'0','5Tb','10Tb','15Tb','20Tb','25Tb','30Tb'})
axis([0 30 0 40])
xlabel('(f) \sigma^2= 1 & 1 is transmitted', 'fontsize',10)

```

Problema ilustrativo 5.10 [Simulación de señales On-Off] Uso de simulación Monte Carlo para estimar y representar gráficamente el rendimiento de un sistema de comunicaciones binario con el empleo de la señalización de encendido y apagado.

Solución

El modelo para el sistema a simular es similar a la mostrada en la figura 5.16, excepto que una de las señales es 0. Por lo tanto, generar una secuencia de variables de $\{r_i\}$ como la dada por (2.5.34). El detector compara las variables aleatorias $\{r_i\}$ al umbral óptimo $E/2$ y toma las decisiones apropiadas. La Figura 5.20 ilustra la probabilidad de error estimado sobre la base de 10.000 dígitos binarios. La tasa de error teórico dado por (5.2.39) también se ilustra en esta figura.

Las secuencias de comandos de MATLAB para este problema se dan a continuación.

M-FILE

```
% MATLAB script for Illustrative Problem 5.10.  
echo on  
SNRindB1=0:1:15;  
SNRindB2=0:0.1:15;  
for i=1:length(SNRindB1),  
    smld_err_prb(i)=smldPe56(SNRindB1(i));           % simulated error rate  
    echo off;  
end;  
echo on;  
for i=1:length(SNRindB2),  
    SNR=exp(SNRindB2(i)*log(10)/10);                % signal-to-noise ratio  
    theo_err_prb(i)=Qfunct(sqrt(SNR/2));            % theoretical error rate  
    echo off;  
end;  
echo on;  
  
% Plotting commands follow.  
semilogy(SNRindB1,smld_err_prb, ' *');  
hold  
semilogy(SNRindB2,theo_err_prb);
```

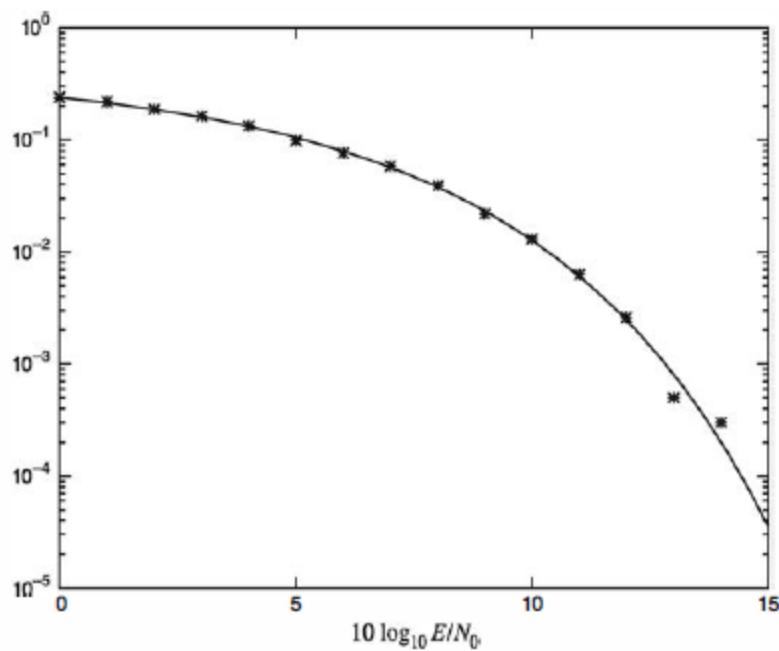


Figura 5 .20: la probabilidad de error de simulación de Monte Carlo en comparación con una probabilidad de error teórica para las señales de encendido y apagado

M-FILE

```

function [p]=smldPe56(snr_in_dB)
% [p]=smldPe56(snr_in_dB)
%
% SMLDPE56 simulates the probability of error for a given
% snr_in_dB, signal-to-noise ratio in dB.

E=1;
alpha_opt=1/2;
SNR=exp(snr_in_dB*log(10)/10); % signal-to-noise ratio
sgma=E/sqrt(2*SNR); % sigma, standard deviation of noise
N=10000;
% Generation of the binary data source follows.
for i=1:N,
    temp=rand; % a uniform random variable over (0,1)
    if (temp<0.5), % With probability 1/2, source output is 0.
        dsource(i)=0;
    else % With probability 1/2, source output is 1.
        dsource(i)=1;
    end
end;
% detection, and probability of error calculation
numoferr=0;
for i=1:N,
    % the matched filter outputs
    if (dsource(i)==0), % if the source output is "0"
        r=gngauss(sgma);
    else % if the source output is "1"
        r=E+gngauss(sgma);
    end;
    % Detector follows.
    if (r<alpha_opt), % Decision is "0".
        decis=0;
    else % Decision is "1".
        decis=1;
    end;
    if (decis~=dsource(i)), % If it is an error, increase the error counter.
        numoferr=numoferr+1;
    end;
end;
p=numoferr/N; % probability of error estimate

```

5.2.2 Diagramas de constelación de señales para señales binarias

Los tres tipos de señales binarias a saber, antípoda, de encendido y apagado, y ortogonal se pueden caracterizar como puntos geométricamente en "espacio de señal." En el caso de señales de antípodas, donde las señales son $s(t)$ y $-s(t)$, teniendo cada uno de energía E , los dos puntos de señal se dividen en la línea real en $\pm \sqrt{E}$, como se muestra en la Figura 5.21 (a). La representación geométrica de una sola dimensión de señales antípodas sigue del hecho de que sólo una forma de onda de señal o una función de base, a saber, $s(t)$ –basta para representar las señales antípodas en el espacio de señal. Las señales On-off también son de una sola dimensión. Por lo tanto, los dos puntos de señal también entran en la línea real en 0 y \sqrt{E} , como se muestra en la Figura 5.21 (b).

Por otra parte, las señales ortogonales binarios requieren una representación geométrica de dos dimensiones, porque hay dos funciones linealmente independientes, $s_0(t)$ y $s_1(t)$, por lo que constituyen las dos formas de onda de señal. En consecuencia, los puntos de señal correspondientes a estos puntos son $(\sqrt{E}, 0)$ y $(0, \sqrt{E})$, como se muestra en la Figura 5.21 (c).

Las representaciones geométricas de las señales binarias que se muestran en la Figura 5.21 se llaman constelaciones de señales.

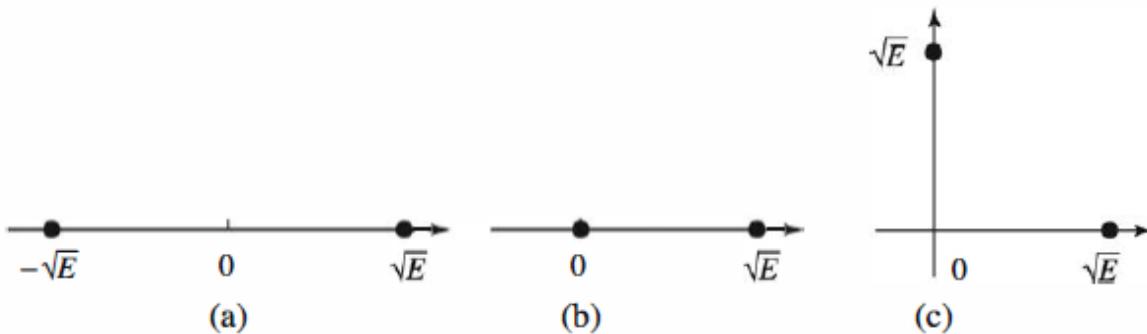


Figura 5.21: constelaciones de señales para las señales binarias. (a) Las señales antípodas. (b) señales on-off. (c) las señales ortogonales.

Problema ilustrativo 5.11 [Efecto de ruido en la constelación] El efecto del ruido en el rendimiento de un sistema de comunicación binario se puede observar desde la señal recibida más el ruido en la entrada al detector. Por ejemplo, consideremos las señales binarias ortogonales, para lo cual la entrada al detector consiste en el par de variables aleatorias (r_0, r_1) , donde cualquiera

$$(r_0, r_1) = (\sqrt{E} + n_0, n_1)$$

or

$$(r_0, r_1) = (n_0, \sqrt{E} + n_1)$$

Las variables aleatorias de ruido no y n1 son de media cero, variables aleatorias gaussianas independientes con varianza σ^2 . Como en Problema ilustrativo 5.6, utilizar la simulación Monte Carlo para generar 100 muestras de (r_0, r_1) , para cada valor de $\sigma^2 = 0,1$, $\sigma^2 = 0,3$, y $\sigma^2 = 0,5$, y graficar estas 100 muestras para cada σ^2 en diferentes gráficos bidimensionales. La energía E de la señal puede ser normalizada a la unidad.

Solución

Los resultados de la simulación de Monte Carlo se muestran en la Figura 5.22. Tenga en cuenta que en un nivel de potencia de ruido bajo (σ pequeño) el efecto del ruido sobre el rendimiento (tasa de error) del sistema de comunicación es pequeño. Como el ruido aumenta de nivel de potencia, los componentes de ruido aumentan de tamaño y causan más errores.

La secuencia de comandos de MATLAB para este problema para $\sigma = 0.5$ se da a continuación.

M-FILE

```
% MATLAB script for Illustrative Problem 5.11.  
echo on  
n0=.5*randn(100,1);  
n1=.5*randn(100,1);  
n2=.5*randn(100,1);  
n3=.5*randn(100,1);  
x1=1.+n0;  
y1=n1;  
x2=n2;  
y2=1.+n3;  
plot(x1,y1,'o',x2,y2,'*')  
axis('square')
```

5.3 Transmisión de señal de amplitud múltiple

En la sección anterior, se trataron la transmisión de información digital mediante el uso de formas de onda de señales digitales. Por lo tanto, cada forma de onda de la señal transmite 1 bit de información.

En esta sección, se utilizan formas de onda de señal que tienen en múltiples niveles de amplitud. Por lo tanto, podemos transmitir múltiples bits por onda de la señal.

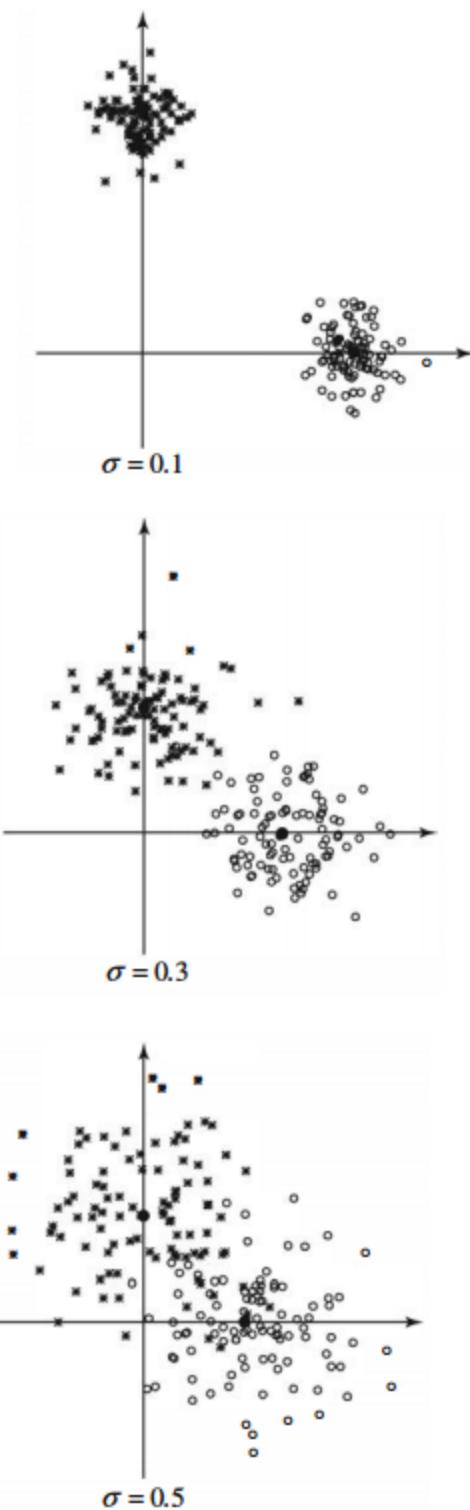


Figura 5.22: Puntos de señal recibidos en la entrada al detector de señales ortogonales (Simulación Monte Carlo)

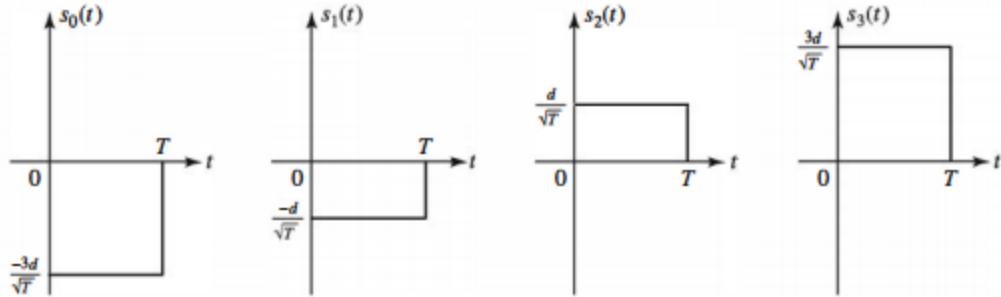


Figura 5.23: Forma de onda de señal de amplitud múltiple.

5.3.1 Las formas de onda de señal con cuatro niveles de amplitud

Consideremos un conjunto de formas de onda de señal de la forma

$$s_m(t) = A_m g(t), \quad 0 \leq t \leq T \quad (5.3.1)$$

Donde A_m es la amplitud de la forma de onda de orden m y $g(t)$ es un pulso rectangular definida como

$$g(t) = \begin{cases} \sqrt{1/T}, & 0 \leq t \leq T \\ 0, & \text{otherwise} \end{cases} \quad (5.3.2)$$

Donde la energía en el pulso $g(t)$ es normalizada a la unidad. En particular, consideramos el caso en el que la amplitud de la señal tarda en uno de los cuatro posibles valores-a saber, igualmente espaciados, $\{A_m\} = \{-3d, -d, d, 3d\}$ o, equivalentemente,

$$A_m = (2m - 3)d, \quad m = 0, 1, 2, 3 \quad (5.3.3)$$

Dónde $2d$ es la distancia euclíadiana entre dos niveles de amplitud adyacentes. Las cuatro formas de onda de señal se ilustran en la Figura 5.23. Llamamos a este conjunto de formas de onda como amplitud de pulso modulado (PAM) de las señales.

Las cuatro formas de onda de señal PAM mostradas en la Figura 5.23 se pueden usar para transmitir 2 bits de información por cada forma de onda. Por lo tanto, asignamos los siguientes pares de bits de información a las cuatro formas de onda de señal:

- $00 \rightarrow s_0(t)$
- $01 \rightarrow s_1(t)$
- $11 \rightarrow s_2(t)$
- $10 \rightarrow s_3(t)$

Cada par de bits de información {00, 01, 10, 11} se denomina un símbolo, y la duración T de tiempo se denomina el intervalo de símbolo. Tenga en cuenta que si la velocidad es de $R = 1/T_b$, el intervalo de símbolos es $T = 2T_b$. Debido a que todas las formas de onda de señal son versiones de la función de base de la señal $g(t)$ escaladas, estas formas de onda de señal pueden estar representados geométricamente como puntos en la línea real. Por lo tanto, la representación geométrica de las cuatro señales PAM es el diagrama de constelación de la señal se muestra en la Figura 5.24.

Como en el caso de señales binarias, se supone que las formas de onda de la señal PAM se transmiten a través de un canal AWGN. En consecuencia, la señal recibida se representa como en la que $n(t)$ es una función de ejemplo de un proceso de ruido blanco gaussiano con espectro de potencia $N_0/2$ vatios/hertzios. La tarea del receptor es determinar cuál de las cuatro formas de onda de señal se transmite después de la observación de la señal recibida $r(t)$ en el intervalo de $0 \leq t \leq T$. El receptor óptimo está diseñado para minimizar la probabilidad de error de un símbolo.

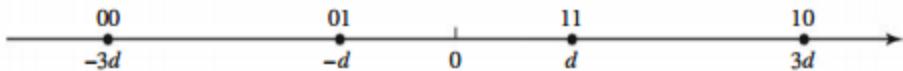


Figura 5.24: Constelación de señales para la forma de onda de señales con cuatro PAM.

5.3.2 Receptor óptimo para el canal AWGN

El receptor que minimiza la probabilidad de error se implementa haciendo pasar la señal a través de un correlador de señal o filtro adaptado seguido de un detector de amplitud. Debido a que el correlador de señal y el filtro adaptado producen la misma salida en el instante de muestreo, se considera sólo el correlador de señal en nuestro tratamiento.

Correlador de señal

El correlador de señal cruzada correlaciona la señal recibida $r(t)$ con el pulso de la señal $g(t)$ y su salida es muestreada a $t = T$. Por lo tanto, la salida de señal de correlación es

$$\begin{aligned}
\mathbf{r} &= \int_0^T \mathbf{r}(t)g(t) dt \\
&= \int_0^T A_i g^2(t) dt + \int_0^T g(t)\mathbf{n}(t) dt \\
&= A_i + \mathbf{n}
\end{aligned} \tag{5.3.5}$$

Donde \mathbf{n} representa el componente de ruido, definida como

$$\mathbf{n} = \int_0^T g(t)\mathbf{n}(t) dt \tag{5.3.6}$$

Observamos que \mathbf{n} es una variable aleatoria gaussiana con media

$$E(\mathbf{n}) = \int_0^T g(t)E[\mathbf{n}(t)] dt = 0 \tag{5.3.7}$$

y varianza

$$\begin{aligned}
\sigma^2 &= E(\mathbf{n}^2) \\
&= \int_0^T \int_0^T g(t)g(\tau)E[\mathbf{n}(t)\mathbf{n}(\tau)] dt d\tau \\
&= \frac{N_0}{2} \int_0^T \int_0^T g(t)g(\tau)\delta(t - \tau) dt d\tau \\
&= \frac{N_0}{2} \int_0^T g^2(t) dt \\
&= \frac{N_0}{2}
\end{aligned} \tag{5.3.8}$$

Por lo tanto, la función de densidad de probabilidad para la salida del correlador de señal es

$$p(\mathbf{r} | s_i(t) \text{ was transmitted}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(r-A_i)^2/2\sigma^2} \tag{5.3.9}$$

Donde A_i es una de los cuatro posibles valores de amplitud.

Detector

El detector observa la salida r del correlador y decide cuál de las cuatro señales PAM fue transmitida en el intervalo de la señal. En el siguiente desarrollo del funcionamiento del

detector óptimo, se supone que los cuatro posibles niveles de amplitud son igualmente probables. Debido a que la amplitud de la señal recibida A_i puede tomar los valores $\pm d$, $\pm 3d$, como se ilustra en la constelación de señales en la figura 5.24, el detector de amplitud óptima compara la salida r del correlador con las cuatro posibles niveles de amplitud de transmisión y selecciona el nivel de amplitud que está más cerca de la distancia euclídea a r . Por lo tanto, el detector de amplitud óptima calcula las distancias

$$D_i = |r - A_i|, \quad i = 0, 1, 2, 3 \quad (5.3.10)$$

Y selecciona la amplitud correspondiente a la distancia más pequeña. Observamos que se produce un error de decisión cuando la variable de ruido n excede en magnitud la mitad de la distancia entre los niveles de amplitud, es decir, cuando $|n| > d$. Sin embargo, cuando se transmite el nivel de amplitud $+3d$ o $-3d$, un error puede ocurrir en una sola dirección. Debido a que los cuatro niveles de amplitud son igualmente probables, la probabilidad media de un error de símbolo es

$$\begin{aligned} P_4 &= \frac{3}{4}P(|r - A_m| > d) \\ &= \frac{3}{2} \int_d^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2} dx \\ &= \frac{3}{2} \int_{d/\sigma}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx \\ &= \frac{3}{2}Q\left(\sqrt{\frac{d^2}{\sigma^2}}\right) \\ &= \frac{3}{2}Q\left(\sqrt{\frac{2d^2}{N_0}}\right) \end{aligned} \quad (5.3.11)$$

Observamos que la distancia al cuadrado entre los niveles de amplitud sucesivas es $(2d)^2 = \sigma^2$. Por lo tanto, la probabilidad media de error puede expresarse como

$$P_4 = \frac{3}{2}Q\left(\sqrt{\frac{\delta^2}{4N_0}}\right) \quad (5.3.12)$$

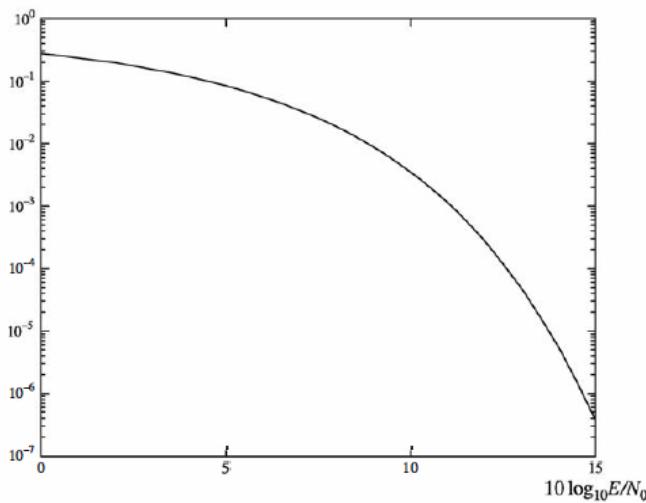


Figure 5.25: Probability of symbol error for four-level PAM

Alternativamente, la probabilidad media de error puede expresarse en términos de la energía de señal. Debido a los cuatro niveles de amplitud son igualmente probables, la energía media de la señal transmitida por símbolo es

$$E_{av} = \frac{1}{4} \sum_{k=1}^4 \int_0^T s_k^2(t) dt = 5d^2 \quad (5.3.13)$$

En consecuencia, $d^2 = \frac{E_{av}}{5}$ y, por lo tanto,

$$P_4 = \frac{3}{2} Q \left(\sqrt{\frac{2E_{av}}{5N_0}} \right) \quad (5.3.14)$$

Debido a que cada símbolo transmitido consta de 2 bits de información, la energía media transmitida por bit es $\frac{E_{av}}{2} \equiv E_{avb}$. La probabilidad del error promedio de P4 se representa gráficamente en la Figura 5.25 como una función de la SNR se define como $10\log_{10}(\frac{E_{avb}}{N_0})$.

Problema Ilustrativo 5.12 [Simulación de múltiples amplitudes de la señal] Realizar una simulación Monte Carlo del sistema de comunicación PAM de cuatro niveles (cuaternario) que emplea un correlador de señal, como se describe anteriormente, seguido de un detector de amplitud. El modelo para el sistema a simular se muestra en la Figura 5.26.

Como se muestra, se simula la generación de la variable aleatoria r , que es la salida del correlador de señal y la entrada del detector. Comenzamos generando una secuencia

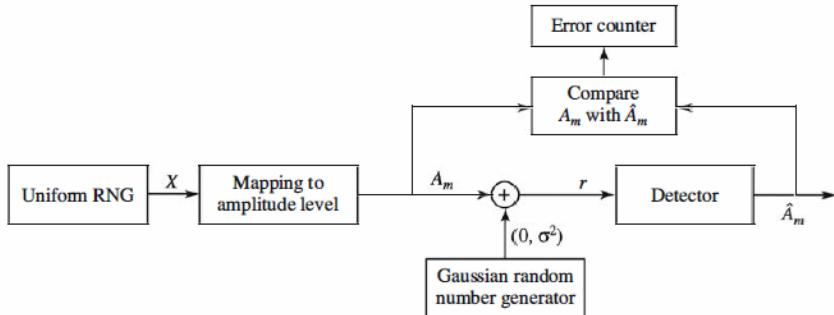


Figure 5.26: Block diagram of four-level PAM for Monte Carlo simulation

De símbolos cuaternarios que se asignan a los niveles de amplitud correspondientes {Am}. Para llevar a cabo esta tarea, se utiliza un generador de números aleatorios que genera un número aleatorio uniforme en el intervalo (0, 1). Esta gama se subdivide en cuatro intervalos iguales, (0,0.25), (0.25, 0.5), (0.5, 0.75), (0.75, 1.0), donde los subintervalos corresponden a los símbolos (pares de bits de información) 00, 01, 11, 10, respectivamente. Por lo tanto, la salida del generador de números aleatorio uniforme se hace corresponder a los niveles de amplitud de señal correspondientes -3d, -d, d, 3d, respectivamente. El componente de ruido aditivo que tiene media 0 y varianza σ^2 se genera por medio de un generador de números aleatorios de Gauss. Por conveniencia, es posible normalizar el parámetro de distancia $d = 1$ y variar σ^2 . El detector observa $r = Am + n$ y calcula la distancia entre r y las cuatro posibles amplitudes de señal de transmisión. Su salida, Am , es el nivel de amplitud de la señal correspondiente a la distancia más pequeña. Am se compara con la amplitud real de la señal transmitida, y un contador de errores se utiliza para contar los errores cometidos por el detector. Figura 5.0.27 ilustra los resultados de la simulación para las transmisiones de $N = 10.000$ símbolos en los diferentes valores de la SNR media de bits, que se define como

$$\frac{E_{avb}}{N_0} = \frac{5}{4} \left(\frac{d^2}{\sigma^2} \right) \quad (5.3.15)$$

Tenga en cuenta el acuerdo entre los resultados de la simulación y los valores teóricos de la computadora P4 de (5.3.14). Las secuencias de comandos de MATLAB para este problema se dan a continuación.

M-FILE

```
% MATLAB script for Illustrated Problem 5.12.

echo on
SNRindB1=0:1:12;
SNRindB2=0:0.1:12;
for i=1:length(SNRindB1),
    % simulated error rate

        smld_err_prb(i)=smldPe58(SNRindB1(i));
        echo off;
end;
echo on;
for i=1:length(SNRindB2),
    % signal-to-noise ratio
    SNR_per_bit=exp(SNRindB2(i)*log(10)/10);
    % theoretical error rate
    theo_err_prb(i)=(3/2)*Qfunct(sqrt((4/5)*SNR_per_bit));
    echo off;
end;
echo on;
% Plotting commands follow.
semilogy(SNRindB1,smld_err_prb,'*');
hold
semilogy(SNRindB2,theo_err_prb);
```

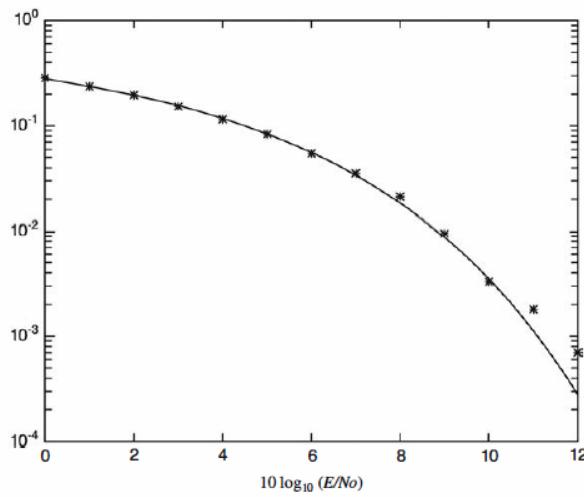


Figure 5.27: Error probability for Monte Carlo simulation compared with theoretical error probability for $M = 4$ PAM

M-FILE

```
function [p]=smldPe58(snr_in_dB)
% [p]=smldPe58(snr_in_dB)
%
% SMLDPE58 simulates the probability of error for the given
% snr_in_dB, signal to noise ratio in dB.
d=1;
SNR=exp(snr_in_dB*log(10)/10); % signal to noise ratio per bit
sgma=sqrt((5*d^2)/(4*SNR)); % sigma, standard deviation of noise

N=10000; % number of symbols being simulated
% Generation of the quaternary data source follows.
for i=1:N,
    temp=rand; % a uniform random variable over (0,1)
    if (temp<0.25),
        dsource(i)=0; % With probability 1/4, source output is "0"
    elseif (temp<0.5),
        dsource(i)=1; % With probability 1/4, source output is "0"
    elseif (temp<0.75),
        dsource(i)=2; % With probability 1/4, source output is "1"
    else
        dsource(i)=3; % With probability 1/4, source output is "1"
    end
end;
% detection, and probability of error calculation
numoferr=0;
for i=1:N,
    % the matched filter outputs
    if (dsource(i)==0),
        r=-3*d+gngauss(sgma); % if the source output is "00"
    elseif (dsource(i)==1),
        r=-d+gngauss(sgma); % if the source output is "01"
    elseif (dsource(i)==2)
        r=d+gngauss(sgma); % if the source output is "10"
    else
        r=3*d+gngauss(sgma); % if the source output is "11"
    end;
    % Detector follows.
    if (r<-2*d),
        decis=0; % Decision is "00."
    elseif (r<0),
        decis=1; % Decision is "01."
    elseif (r<2*d),
        decis=2; % Decision is "10."
    end;
```

```

else
    decis=3;                                % Decision is "11."
end;
if (decis~=dsource(i)),
    numoferr=numoferr+1;
end;
end;
p=numoferr/N;                            % probability of error estimate

```

5.3.3 Formas de onda de señales con niveles de amplitud Múltiples

Es relativamente sencillo de construir señales con múltiples amplitudes con más de cuatro niveles. En general, un conjunto de $M = 2k$ formas de onda de una señal con múltiples amplitudes se representa como

$$s_m(t) = A_m g(t), \quad 0 \leq t \leq T, \quad m = 0, 1, 2, \dots, M - 1$$

Donde los valores de amplitud M están igualmente espaciados y dan

$$A_m = (2m - M + 1)d, \quad m = 0, 1, \dots, M - 1 \quad (5.3.16)$$

Y $g(t)$ es un pulso rectangular, que se ha definido en (5.3.2). Cada forma de onda de la señal transmite $k = \log_2 M$ bits de información. Cuando la velocidad de bits es $R = 1 / Tb$, la velocidad de símbolos correspondiente es $1 / T = 1 / kTb$. Como en el caso de cuatro niveles PAM, el receptor óptimo consiste en un correlador de señal (o filtro adaptado) seguido de un detector de amplitud que calcula las distancias euclidianas dadas por forma (5.3.10) = 0, 1, ..., $M - 1$. Para niveles de amplitud igualmente probables, se toma la decisión a favor del nivel de amplitud que corresponde a la distancia más pequeña. La probabilidad de error para el detector óptimo en un sistema de nivel M PAM puede demostrarse que es

$$P_M = \frac{2(M - 1)}{M} Q \left(\sqrt{\frac{6(\log_2 M) E_{avb}}{(M^2 - 1) N_0}} \right) \quad (5.3.17)$$

Donde E_{avb} es la energía media de un bit de información. La Figura 5.28 ilustra la probabilidad de error de símbolo para $M = 2, 4, 8, 16$.

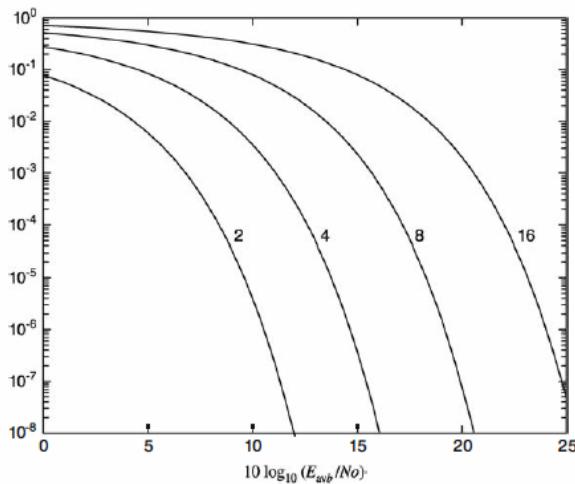


Figure 5.28: Symbol error probability for M -level PAM for $M = 2, 4, 16$

Problema ilustrativo 5.13 [Simulación PAM] Realizar una simulación de Monte Carlo de un sistema de comunicación digital PAM de 16 niveles y medir su rendimiento de tasa de error.

El diagrama de bloques básico mostrado en la Figura 5.26 aplica en general, un generador de números aleatorio uniforme que se utiliza para generar la secuencia de símbolos de información, que son vistos como bloques de 4 bits de información. Los símbolos 16 se pueden generar directamente.

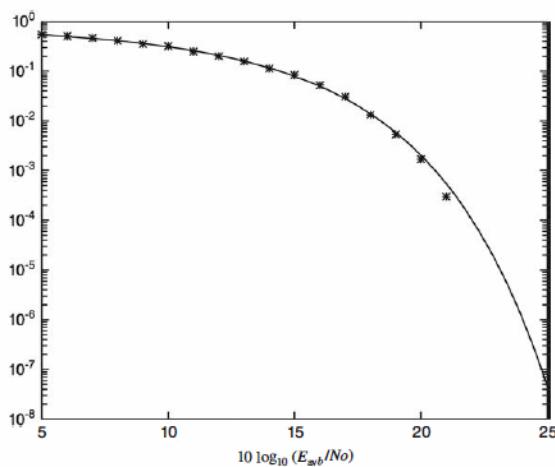


Figure 5.29: Error rate from Monte Carlo simulation compared with the theoretical error probability for $M = 16$ PAM

Subdividiendo el intervalo $(0, 1)$ en 16 subintervalos de igual anchura y la cartografía de los símbolos 16 en las amplitudes de señal 16. Una secuencia de ruido gaussiano blanco se añade a la secuencia de símbolo de información 16, y la señal resultante más el ruido se

alimenta al detector. El detector calcula las métricas de distancia dadas por (5.3.10) y selecciona la amplitud correspondiente a la menor métrica. La salida del detector se compara con la secuencia de símbolo de la información transmitida y errores se cuentan. La figura 5.29 ilustra la tasa de error de símbolo medido para 10000 símbolos transmitidos y la tasa de error de símbolo teórico dado por (5.3.17) con $M = 16$. Las secuencias de comandos de MATLAB para este problema se dan a continuación.

M-FILE

```
% MATLAB script for Illustrative Problem 5.13.  
echo on  
SNRindB1=5:1:25;  
SNRindB2=5:0.1:25;  
M=16;  
for i=1:length(SNRindB1),  
    % simulated error rate  
    smld_err_prb(i)=smldPe59(SNRindB1(i));  
    echo off;  
end;  
echo on ;  
for i=1:length(SNRindB2),  
    SNR_per_bit=exp(SNRindB2(i)*log(10)/10);
```

M-FILE

```

% theoretical error rate
theo_err_prb(i)=(2*(M-1)/M)*Qfunct(sqrt((6*log2(M)/(M^2-1))*SNR_per_bit));
echo off;
end;
echo on;
% Plotting commands follow.
semilogy(SNRindB1,smld_err_prb,'*');
hold
semilogy(SNRindB2,theo_err_prb);

function [p]=smldPe59(snr_in_dB)
% [p]=smldPe59(snr_in_dB)
% SMLDPE59 simulates the error probability for the given
% snr_in_dB, signal-to-noise ratio in dB.
M=16; % 16-ary PAM
d=1;
SNR=exp(snr_in_dB*log(10)/10); % signal-to-noise ratio per bit
sgma=sqrt((85*d^2)/(8*SNR)); % sigma, standard deviation of noise
N=10000; % number of symbols being simulated
% generation of the data source
for i=1:N,
    temp=rand; % a uniform random variable over (0,1)
    index=floor(M*temp); % The index is an integer from 0 to M-1, where
    % all the possible values are equally likely.
    dsoure(i)=index;
end;
% detection, and probability of error calculation
numoferr=0;
for i=1:N,
    % matched filter outputs

```

```
% (2*dsource(i)-M+1)*d is the mapping to the 16-ary constellation.  
r=(2*dsource(i)-M+1)*d+gngauss(sgma);  
% the detector  
if (r>(M-2)*d),  
    decis=15;  
elseif (r>(M-4)*d),  
    decis=14;  
elseif (r>(M-6)*d),  
    decis=13;  
elseif (r>(M-8)*d),  
    decis=12;  
elseif (r>(M-10)*d),  
    decis=11;  
elseif (r>(M-12)*d),  
    decis=10;  
elseif (r>(M-14)*d),  
    decis=9;  
elseif (r>(M-16)*d),  
    decis=8;  
elseif (r>(M-18)*d),  
    decis=7;
```

Problemas:

_M-FILE_____

Capítulo 6:

Transmisión Digital a través de canales de banda limitada

6.1 Introducción

En este capítulo tratamos varios aspectos de la transmisión digital a través de canales de ancho de banda limitado. Comenzamos con la descripción de las características espetrales de señales PAM. En segundo lugar, consideramos que la caracterización de los canales de banda limitada y el problema de diseñar formas de onda de señal para tales canales. A continuación, tratamos el problema de diseñar ecualizadores de canal que compensar la distorsión causada por los canales de banda limitada. Se demuestra que los resultados de distorsión del canal de interferencia entre símbolos (ISI), lo que provoca errores en la demodulación de la señal. Un ecualizador de canal es un dispositivo que reduce la interferencia entre símbolos y por lo tanto reduce la tasa de error en la secuencia de datos demodulados.

6.2 Espectro de Potencia de una señal Digital PAM

En el capítulo anterior hemos considerado la transmisión de información digital mediante modulación de amplitud de pulso (PAM). En esta sección se estudian las características espetrales de dichas señales. Una señal PAM digital en la entrada a un canal de comunicación se representa generalmente como

$$v(t) = \sum_{n=-\infty}^{\infty} a_n g(t - nT) \quad (6.2.1)$$

Donde $\{a_n\}$ es la secuencia de amplitudes correspondientes a los símbolos de información desde la fuente, $g(t)$ es una forma de onda de pulso, y T es el recíproco de la velocidad de símbolos. T también se llama el intervalo de símbolo. Cada elemento de la secuencia $\{a_n\}$ se selecciona de uno de los posibles valores de amplitud, que son

$$A_m = (2m - M + 1)d, \quad m = 0, 1, \dots, M - 1 \quad (6.2.2)$$

Donde d es un factor de escala que determina la distancia euclídea entre cualquier par de amplitudes de señal ($2d$ es la distancia euclídea entre los niveles de amplitud de señal adyacentes). Debido a que la secuencia de información es una secuencia aleatoria, la

secuencia $\{a_n\}$ de las amplitudes correspondientes a los símbolos de información de la fuente también es aleatoria. En consecuencia, la señal PAM $v(t)$ es una función de ejemplo de un proceso aleatorio $V(t)$. Para determinar las características espectrales del proceso aleatorio $V(t)$, debemos evaluar el espectro de potencia. En primer lugar, observamos que el valor medio de $V(t)$ es

$$E[V(t)] = \sum_{n=-\infty}^{\infty} E(a_n)g(t - nT) \quad (6.2.3)$$

Mediante la selección de las amplitudes de señal para ser simétrica alrededor de cero, como se indica en (6.2.2), e igualmente probables, tenemos $E(a) = 0$ y por lo tanto $E[V(t)] = 0$.

La función de autocorrelación de $V(t)$ es

$$R_v(t + \tau; t) = E[V(t)V(t + \tau)] \quad (6.2.4)$$

Se muestra en muchos textos estándar de comunicaciones digitales que la función de autocorrelación es una función periódica en la variable t período T . Los procesos aleatorios tienen un valor medio periódico y una función de autocorrelación periódica y son llamados periódicamente estacionario, o ciclo estacional. La variable tiempo t puede ser eliminado por un promedio de $Rv(t + \tau; t)$ durante un único período; es decir,

$$\bar{R}_v(\tau) = \frac{1}{T} \int_{-T/2}^{T/2} R_v(t + \tau; t) dt \quad (6.2.5)$$

Esta función promedio de autocorrelación para la señal PAM se puede expresar como

$$\bar{R}_v(\tau) = \frac{1}{T} \sum_{m=-\infty}^{\infty} R_a(m) R_g(\tau - mT) \quad (6.2.6)$$

Donde $R_a(m) = E(a_n a_{n+m})$ es la autocorrelación de la sucesión $\{a_n\}$ y $R_g(\tau)$ se define como

$$R_g(\tau) = \int_{-\infty}^{\infty} g(t)g(t + \tau) dt \quad (6.2.7)$$

El espectro de potencia de $V(t)$ es simplemente la transformada de Fourier de la función de autocorrelación promedio $Rv(\tau)$; es decir,

$$\begin{aligned}\mathcal{S}_v(f) &= \int_{-\infty}^{\infty} \bar{R}_v(\tau) e^{-j2\pi f\tau} dt \\ &= \frac{1}{T} \mathcal{S}_a(f) |G(f)|^2\end{aligned}\quad (6.2.8)$$

Donde $Sa(f)$ es el espectro de potencia de la secuencia de amplitud $\{a_n\}$ y $G(f)$ es la transformada de Fourier del pulso de $g(t)$. $Sa(f)$ se define como

$$\mathcal{S}_a(f) = \sum_{m=-\infty}^{\infty} R_a(m) e^{-j2\pi fmT} \quad (6.2.9)$$

A partir de (6.2.8) se observa que el espectro de potencia de la señal de PAM es una función del espectro de potencia de los símbolos de información $\{a_n\}$ y el espectro de impulso $g(t)$. En el caso especial en que la sucesión $\{a_n\}$ no está correlacionado, es decir,

$$R_a(m) = \begin{cases} \sigma_a^2, & m = 0 \\ 0, & m \neq 0 \end{cases} \quad (6.2.10)$$

Donde $\sigma_a^2 = E(a_n^2)$ se deduce que $Sa(f) = \sigma_a^2$ para todo f y

$$\mathcal{S}_v(f) = \frac{\sigma_a^2}{T} |G(f)|^2 \quad (6.2.11)$$

En este caso, el espectro de potencia de $V(t)$ es totalmente dependiente de las características espetrales del impulso $g(t)$.

Problema Ilustrativo 6.1 [PAM Espectro de Potencia] Determinar el espectro de potencia de $V(t)$ cuando $\{a_n\}$ es una secuencia no correlacionado y $g(t)$ es el pulso rectangular que se muestra en la Figura 6.1.

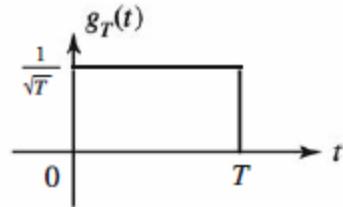


Figura 6.1: Transmisor de impulsos

Solución La transformada de Fourier de $g(t)$ es

$$\begin{aligned}
 G(f) &= \int_{-\infty}^{\infty} g(t) e^{-j2\pi f t} dt \\
 &= \sqrt{T} \frac{\sin \pi f T}{\pi f T} e^{-j\pi f T}
 \end{aligned} \tag{6.2.12}$$

y

$$S_v(f) = \sigma_a^2 \left(\frac{\sin \pi f T}{\pi f T} \right)^2 \tag{6.2.13}$$

Este espectro de potencia se ilustra en la Figura 6.2.

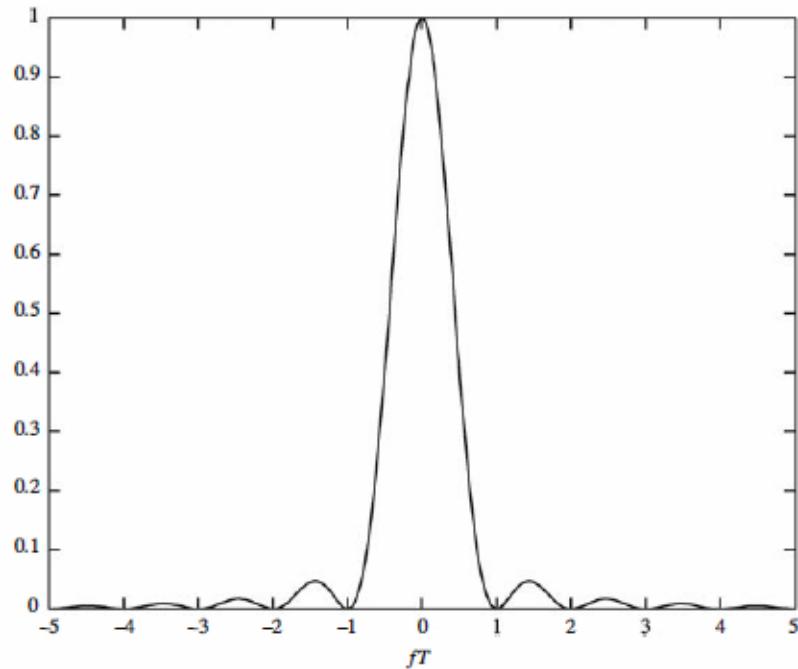


Figura 6.2: Espectro de potencia de la señal transmitida en Problema Ilustrativo 6.1 (para $\sigma_a^2 = 1$)

La secuencia de comandos MATLAB para este cálculo se da a continuación.

M-FILE

```
% MATLAB script for Illustrative Problem 6.1.  
echo on  
T=1;  
delta_f=1/(100*T);  
f=-5/T:delta_f:5/T;  
sigma_a=1;  
Sv=sigma_a^2*sinc(f*T).^2;  
% Plotting command follows.  
plot(f,Sv);
```

Problema Ilustrativo 6.2 Supongamos que la función de autocorrelación de la secuencia $\{a_n\}$ es

$$R_a(m) = \begin{cases} 1, & m = 0 \\ \frac{1}{2}, & m = 1, -1 \\ 0, & \text{otherwise} \end{cases} \quad (6.2.14)$$

y $g(t)$ es el pulso rectangular que se muestra en la Figura 6.1. Evaluar $S_v(f)$ en este caso.

Solución El espectro de potencia de la señal PAM $V(t)$ viene dada por (6.2.8). El espectro de potencia

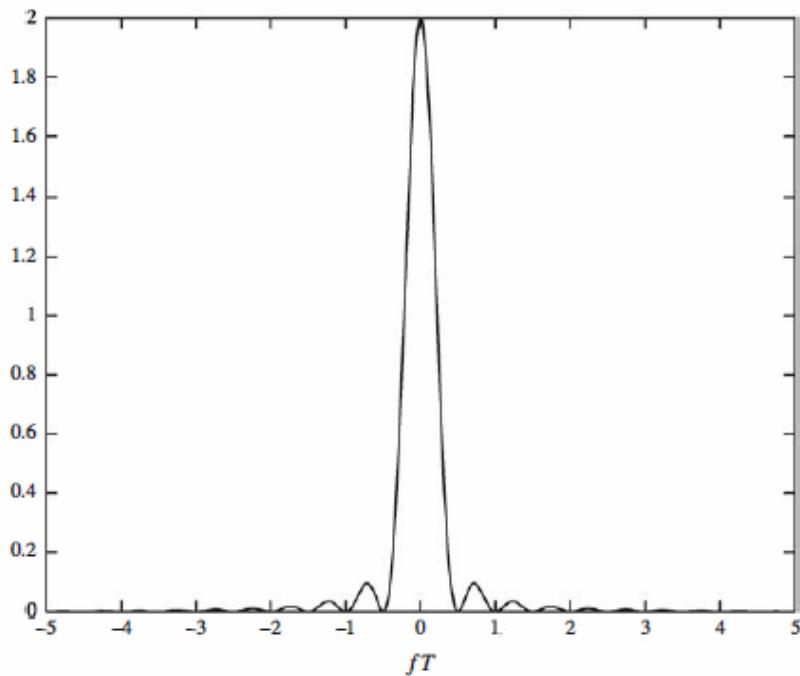


Figura 6.3: El espectro de potencia de la señal transmitida en Problema ilustrativa 6.2

$$(\text{para } \sigma_a^2 = 1)$$

De la sucesión $\{a_n\}$ es, a partir de (6.2.9) y (6.2.14),

$$\begin{aligned} S_a(f) &= 1 + \cos 2\pi fT \\ &= 2 \cos^2 \pi fT \end{aligned} \quad (6.2.15)$$

Por consiguiente,

$$S_v(f) = 2 \cos^2 \pi fT \left(\frac{\sin \pi fT}{\pi fT} \right)^2 \quad (6.2.16)$$

La gráfica de este espectro de potencia se muestra en la Figura 6.3. (6.2.15) (6.2.16) La secuencia de comandos de MATLAB para realizar este cálculo es la siguiente. En este caso, el espectro de potencia global de la señal transmitida $V(t)$ es significativamente más estrecho que el espectro en la Figura 6.2.

M-FILE

```
% MATLAB script for Illustrative Problem 6.2.
echo on
T=1;
delta_f=1/(100*T);
f=-5/T:delta_f:5/T;
Sv=2*(cos(pi*f*T).*sinc(f*T)).^2;
% Plotting command follows.
plot(f,Sv);
```

6.3 Caracterización de canales de banda limitada y la distorsión del canal

Muchos canales de comunicación, incluyendo canales telefónicos y algunos canales de radio, se pueden caracterizar en general como filtros lineales de banda limitada. En consecuencia, estos canales se describen por su respuesta de frecuencia $C(f)$, que puede expresarse como

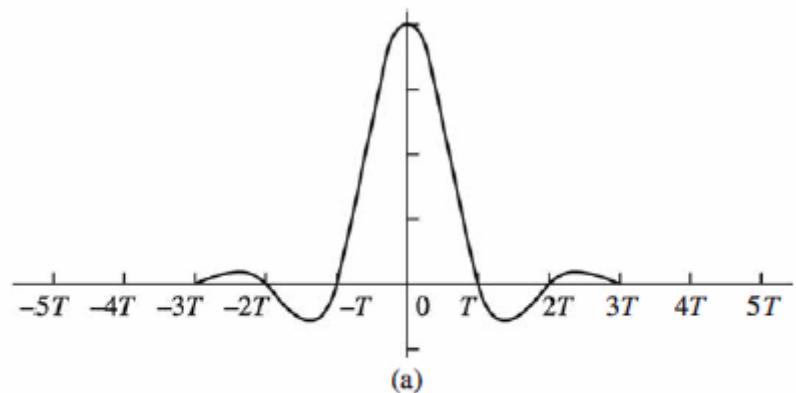
$$C(f) = A(f)e^{j\theta(f)} \quad (6.3.1)$$

donde $A(f)$ se llama la respuesta de amplitud y $\theta(f)$ se llama la respuesta de fase. Otra de las características que a veces se utiliza en lugar de la respuesta de fase es retardo envolvente, o retardo de grupo, que se define como

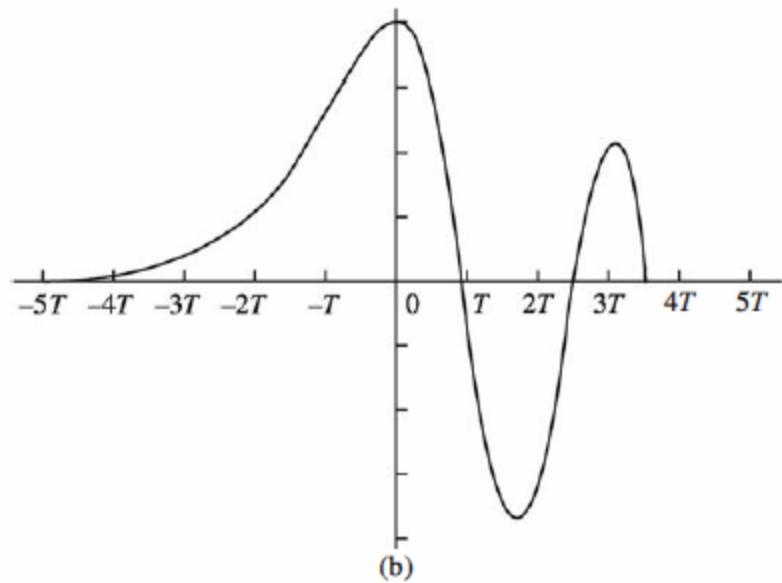
$$\tau(f) = -\frac{1}{2\pi} \frac{d\theta(f)}{df} \quad (6.3.2)$$

Un canal se dice que es no distorsivo, o ideal, si dentro del ancho de banda W ocupada por la señal transmitida, $A(f) = \text{constante}$ y $\theta(f)$ función lineal de la frecuencia (o retardo envolvente $\tau(f) = \text{constante}$). Por otro lado, si $A(f)$ y $\tau(f)$ no son constantes en el ancho de banda ocupado por la señal transmitida, el canal distorsiona la señal. Si $A(f)$ no es constante, la distorsión se llama distorsión de amplitud, y si $\tau(f)$ no es constante, la distorsión en la señal transmitida se denomina distorsión de retardo. Como resultado de la amplitud y de la distorsión de retardo causada por la respuesta de frecuencia del canal característico $C(f)$ no ideal, una sucesión de impulsos transmitidos a través del canal a un ritmo comparable al ancho de banda W se juntan hasta el punto de que ya no son distinguibles, como pulsos bien definidos en el terminal receptor. En su lugar, se superponen, por lo que tenemos la interferencia entre símbolos. Como un ejemplo del efecto de la distorsión de retardo de un impulso transmitido, la Figura 6.4 (a) ilustra un impulso de banda limitada que tiene ceros espaciados periódicamente en el tiempo en los puntos marcados $\pm T$, $\pm 2T$, y así sucesivamente. Cuando la información es transportada por la amplitud del pulso, como la PAM, entonces, se puede transmitir una secuencia de impulsos, cada uno de los cuales tiene un pico de ceros periódicos de los otros impulsos. Sin embargo, la transmisión del pulso a través de un modelado de canal tiene un retardo envolvente lineal característico $\tau(f)$ [fase cuadrática $\theta(f)$] resultados en el pulso recibido se muestra en la figura 6.4 (b) que tiene los cruces por cero que ya no son periódicamente espaciados. En consecuencia, una secuencia de impulsos sucesivos se juntan una en la otra, y los picos de los impulsos ya no sería distinguible. Así, la distorsión de retardo de canal resulta en la interferencia entre símbolos. Como será discutido en este capítulo, es posible compensar la característica de respuesta de frecuencia ideal no del canal mediante el uso de un filtro o de un ecualizador en el demodulador. La Figura 6.4 (c) ilustra la salida de un ecualizador lineal que compensa la distorsión lineal en el canal. A modo de ejemplo, consideremos la interferencia entre símbolos en un canal telefónico. La Figura 6.5 ilustra la amplitud media medida y el retardo como una función de la frecuencia para un canal telefónico de la red de telecomunicaciones comutada. Observamos que la banda utilizable del canal se extiende desde aproximadamente 300 Hz a aproximadamente 3200

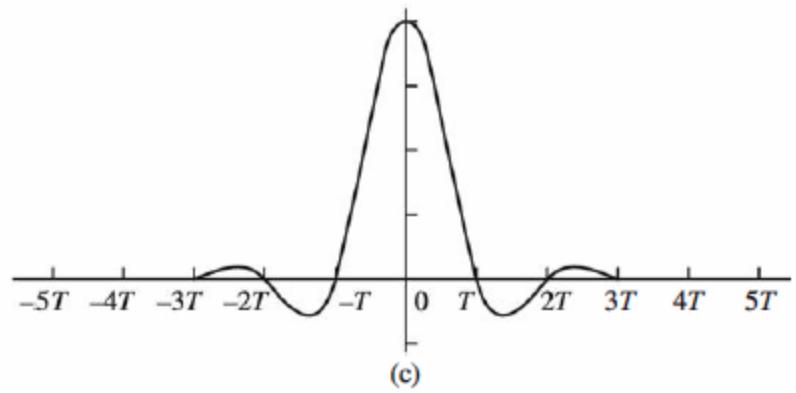
Hz. La respuesta de impulso promedio correspondiente al canal se muestra en la Figura 6.6.



(a)



(b)



(c)

Figura 6.4: Efecto de la distorsión del canal. (a) entrada del canal, (b) Salida del canal, (c) Salida del ecualizador

Su duración es de unos 10 ms. En comparación, las velocidades de símbolos de transmisión en un canal de este tipo puede ser del orden de 2500 pulsos o símbolos por segundo. Por lo tanto, la interferencia entre símbolos podría extenderse sobre 20 a 30 símbolos.

Además de canales telefónicos, hay otros canales físicos que exhiben alguna forma de dispersión en el tiempo y por lo tanto introducen interferencia entre símbolos. canales de radio de onda corta, como la propagación ionosférica (alta frecuencia, HF), la dispersión troposférica,

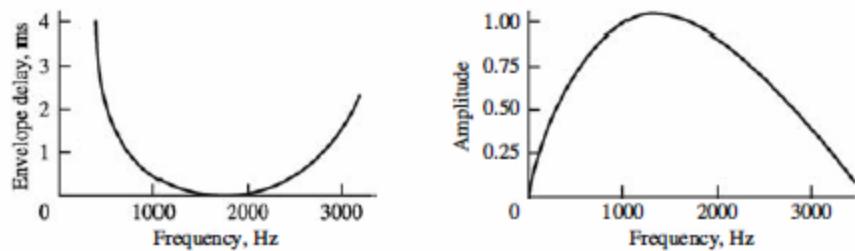


Figura 6.5: caracterización de la amplitud y retardo promedio de un canal telefónico de medio alcance

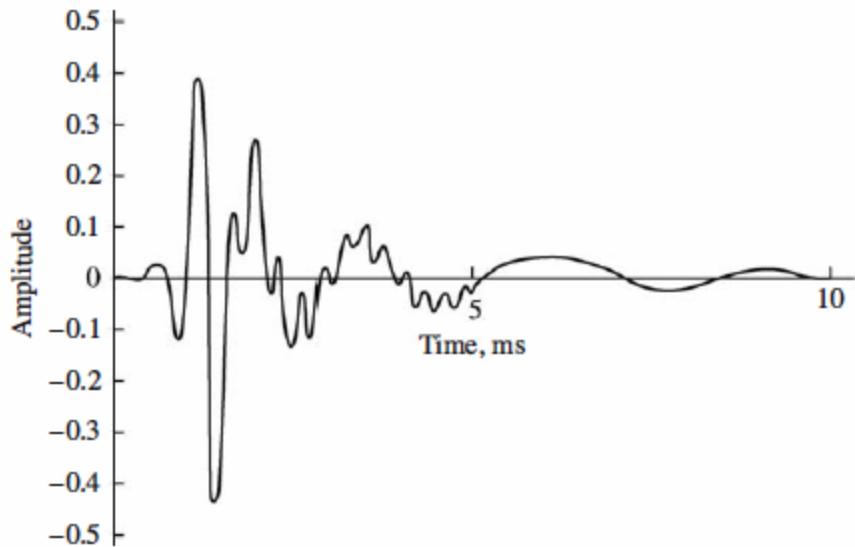


Figura 6.6: respuesta de impulso promedio del canal con la amplitud y el retardo mostrado en la Figura 6.5

y de radio celular son tres ejemplos de canales inalámbricos en tiempo dispersivo. En estos canales, la dispersión y el tiempo, por lo tanto, la interferencia entre símbolos es el resultado de múltiples trayectorias de propagación con diferentes retardos de trayectoria. El número de rutas y los retardos relativos entre los caminos varían con el tiempo; por esta razón, estos canales de radio son generalmente llamados canales de trayectoria múltiples variantes en el tiempo. Las condiciones de trayectoria múltiples variantes en el tiempo dan lugar a una amplia variedad de características de respuesta de frecuencia. En consecuencia, la caracterización de la respuesta de frecuencia que se utiliza para canales telefónicos es inapropiada para los canales de trayectoria múltiples variantes en el tiempo. En cambio, estos canales de radio se caracterizan estadísticamente en términos de la función de dispersión, que, en resumen, es una representación de dos dimensiones la de potencia media recibida de la señal como una función de retardo de tiempo relativo y la de frecuencia de dispersión Doppler. Con fines ilustrativos, una función de dispersión medida en un medio de rango (150- millas) del canal de dispersión troposférica se muestra en la Figura 6.7. La duración total de tiempo (por la dispersión de múltiples trayectos) de la respuesta del canal es de aproximadamente 0.7 microsegundos en promedio, y la diferencia entre los puntos de potencia de la mitad de la frecuencia Doppler es un poco menos de 1 Hz en el camino más fuerte y un poco más grande en otros caminos. Típicamente, si la transmisión se produce a una velocidad de 10^7 símbolos/segundo por un canal de este tipo, la propagación por trayectos múltiples de 0,7 microsegundos dará lugar a interferencia entre símbolos que se extiende alrededor de siete símbolos.

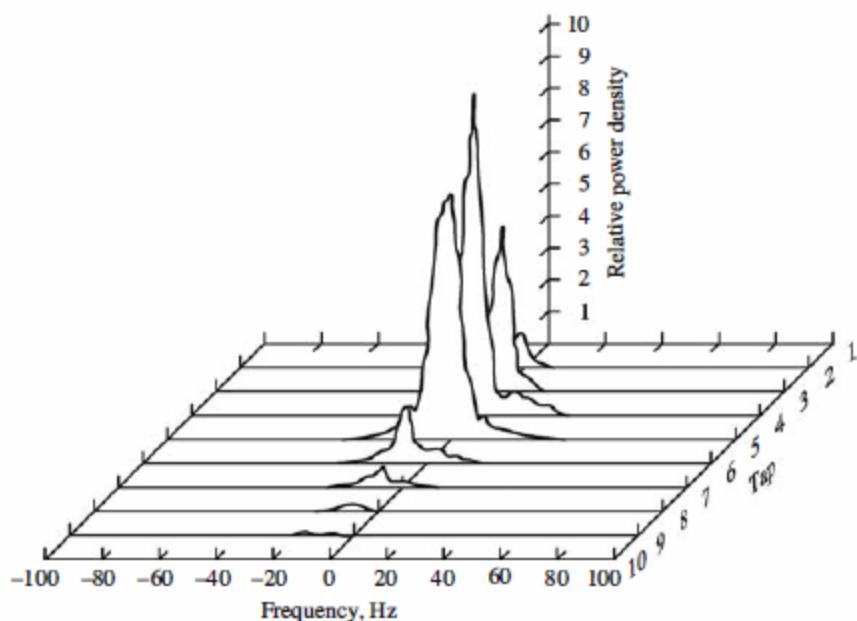


Figura 6.7: Función de Dispersión de un canal de dispersión troposférica de mediano alcance

Problema ilustrativo 6.3 Como se indicó anteriormente, un canal de comunicación de banda limitada puede ser modelado como un filtro lineal, cuyas características de respuesta en frecuencia coincide con las características de respuesta en frecuencia del canal. MATLAB puede usarse para diseñar respuesta digital de impulso finito (FIR) o los filtros de respuesta de impulso infinito (IIR) que es la aproximación de las características de respuesta en frecuencia de los canales de comunicación analógicos.

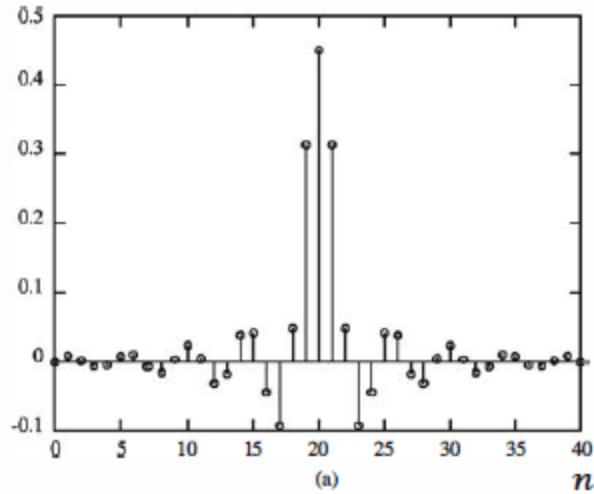
Supongamos que deseamos modelar un canal ideal que tiene una respuesta de amplitud $A(f) = 1$ para $|f| \leq 2000 \text{ Hz}$ y $A(f) = 0$ para $|f| > 2000 \text{ Hz}$ y retardo constante (fase lineal) para todo f . La velocidad de muestreo para el filtro digital se selecciona como $F_s = 10000 \text{ Hz}$. Debido a que la respuesta de fase deseada es lineal, solamente un filtro FIR podría satisfacer esta condición. Sin embargo, no es posible conseguir una respuesta cero en la banda de detención. En lugar de ello, seleccionamos la respuesta de banda de detención para ser -40 dB y la frecuencia de la banda de parada a ser 2500 Hz. Además, permitimos por una pequeña cantidad, 0,5 dB, de ripple en la banda de paso.

Solución

La respuesta al impulso y la respuesta en frecuencia de una longitud $N = 41$ de un filtro FIR que cumpla con estas especificaciones se ilustra en la Figura 6.8. Debido a que N es impar, el retardo a través del filtro es $(N + 1)/2$ pulsos, lo que corresponde a un retardo de tiempo de $(N + 1)/20 \text{ ms}$ a una frecuencia de muestreo de $F_s = 10 \text{ KHz}$. En este ejemplo, el filtro FIR fue diseñado en MATLAB mediante el método de aproximación de Chebyshev (algoritmo de Remez).

M-FILE

```
% MATLAB script for Illustrative Problem 6.3.  
echo on  
f_cutoff=2000; % the desired cutoff frequency  
f_stopband=2500; % the actual stopband frequency  
fs=10000; % the sampling frequency  
f1=2*f_cutoff/fs; % the normalized passband frequency  
f2=2*f_stopband/fs; % the normalized stopband frequency  
N=41; % This number is found by experiment.  
F=[0 f1 f2 1];  
M=[1 1 0 0]; % describes the lowpass filter  
B=remez(N-1,F,M); % returns the FIR tap coefficients  
% Plotting command follows.  
figure(1);  
[H,W]=freqz(B);  
H_in_dB=20*log10(abs(H));  
plot(W/(2*pi),H_in_dB);  
figure(2);  
plot(W/(2*pi),(180/pi)*unwrap(angle(H)));  
% Plot of the impulse response follows.  
figure(3);  
plot(zeros(size([0:N-1])));  
hold;  
stem([0:N-1],B);
```



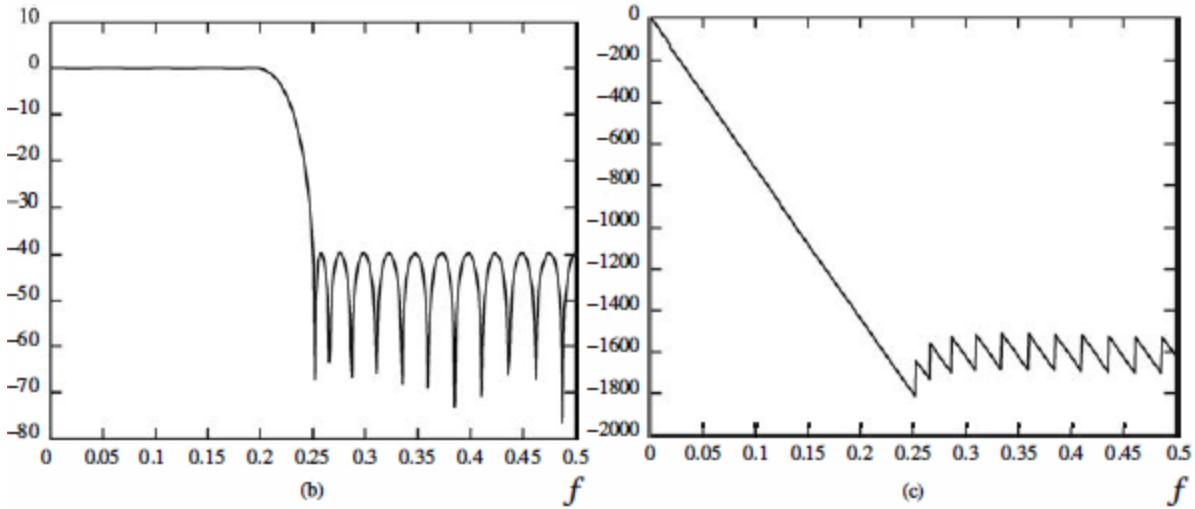


Figura 6.8: (a) Respuesta de impulsos, (b), (c) la respuesta de frecuencia del filtro FIR de fase lineal del problema ilustrativo 6.3

Problema Ilustrativo 6.4 Un método alternativo para el diseño de un filtro FIR que se aproxima a las características del canal deseado se basa en el método de ventana. Para ser más específicos, si la respuesta del canal deseado es $C(f)$ para $|f| \leq W$ y $C(f) = 0$ para $|f| > W$, la respuesta de impulso del canal es

$$h(t) = \int_{-W}^W C(f) e^{j2\pi f t} df \quad (6.3.3)$$

Por ejemplo, si el canal es ideal, entonces $C(f) = 1$, $|f| \leq W$, y por lo tanto

$$h(t) = \frac{\sin 2\pi W t}{\pi t} \quad (6.3.4)$$

Un filtro digital equivalente puede ser implementado por el muestreador $h(t)$ en $t = nT_s$, donde T_s es el intervalo de muestreo y $n = 0, \pm 1, \pm 2, \dots$. Veamos ahora diseñar un filtro FIR con $W = 2000$ Hz y $F_s = 1/T_s = 10$ KHz.

Solución

La versión muestreada de $h(t)$ - es decir, $h_n = h(nT_s)$ se ilustra en la Figura 6.9. Debido a que $\{h_n\}$ tiene una longitud infinita, podemos truncar con cierto detalle N . Este truncamiento es equivalente a multiplicar $\{h_n\}$ por una secuencia de ventana rectangular $w_n = 1$

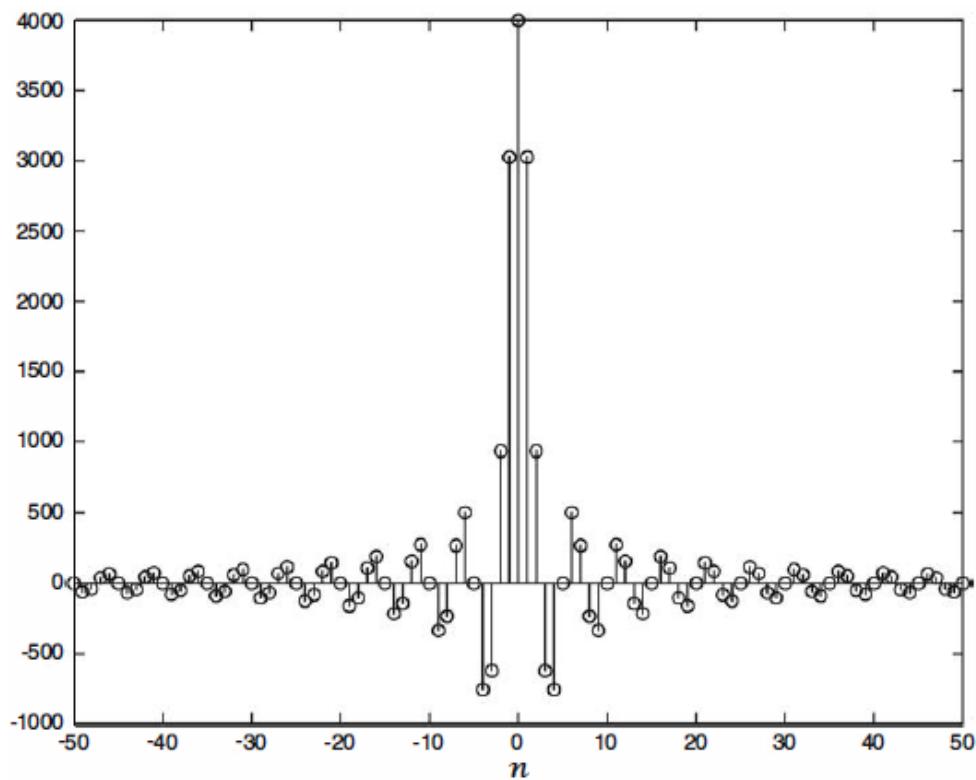


Figura 6.9: Las muestras de $h(n)$ del Problema ilustrativo 6.4

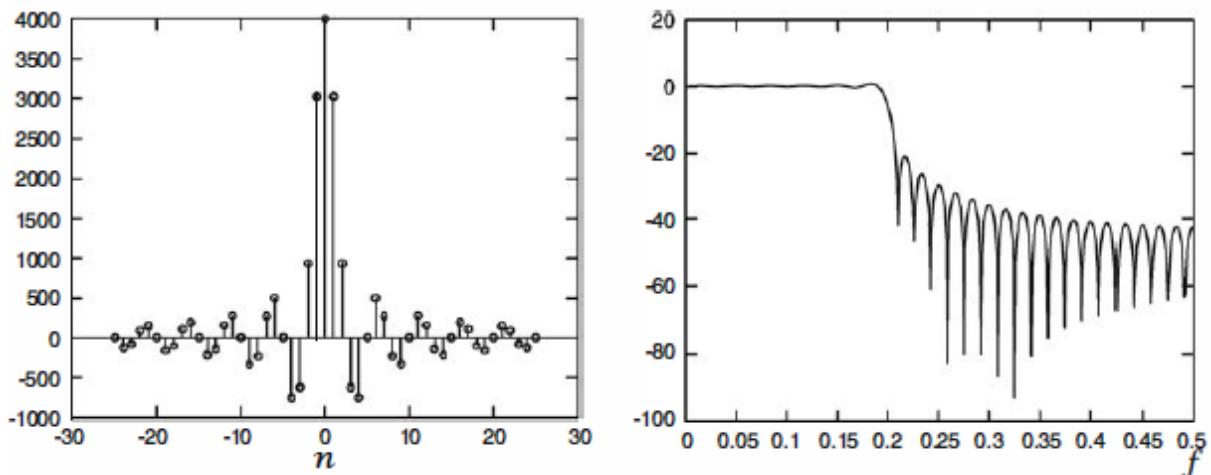


Figura 6.10: respuesta en frecuencia del filtro truncado con una ventana rectangular del Problema ilustrativo 6.4

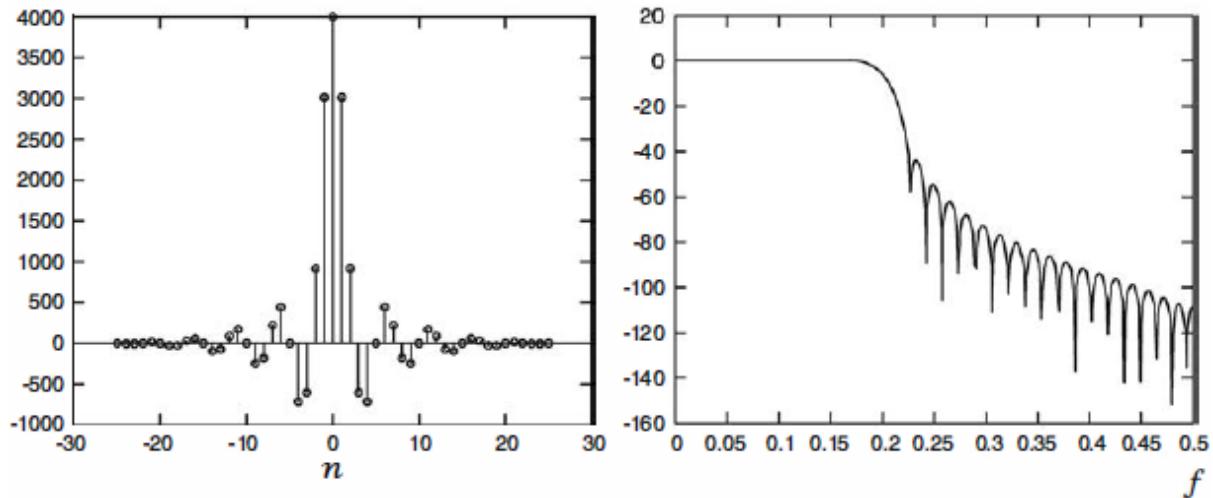


Figura 6.11: Respuesta en frecuencia del filtro truncado con una ventana Hanning del Problema ilustrativo 6.4

para $|n| \leq (N + 1)/2$ y $w_n = 0$ para $|n| \geq (N + 1)/2$. La respuesta de impulso $\{h_n^1 = w_n h_n\}$ y la respuesta de frecuencia correspondiente del filtro FIR (truncado) se ilustran en la Figura 6.10 para $N = 51$. Nota que el filtro truncado tiene grandes lóbulos laterales de la banda de detención. Por lo tanto, este filtro FIR es una pobre aproximación a las características del canal deseados. El tamaño de los lóbulos laterales se puede reducir significativamente mediante el empleo de una función de ventana más suave, tal como una ventana de Hamming o Hanning, para truncar la respuesta del canal ideal. La Figura 6.11 ilustra la respuesta de impulso y la respuesta en frecuencia de $\{h_n^1 = w_n h_n\}$ cuando la función de ventana es una ventana de Hanning de longitud $N = 51$.

MATLAB proporciona rutinas para implementar varios tipos de funciones diferentes de la ventana. La secuencia de comandos de MATLAB para este problema para implementar dos tipos de funciones de la ventana se da a continuación.

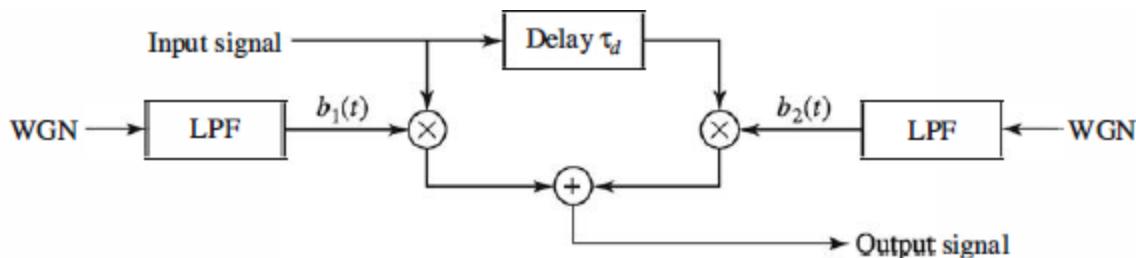


Figura 6.12: de dos trayectos modelo de canal de radio

M-FILE

```
% MATLAB script for Illustrative Problem 6.4.  
echo on  
Length=101;  
Fs=10000;  
W=2000;  
Ts=1/Fs;  
n=-(Length-1)/2:(Length-1)/2;  
t=Ts*n;  
h=2*W*sinc(2*W*t);  
% The rectangular windowed version follows.  
N=61;  
rec_windowed_h=h((Length-N)/2+1:(Length+N)/2);  
% Frequency response of rec_windowed_h follows.  
[rec_windowed_H,W1]=freqz(rec_windowed_h,1);  
% to normalize the magnitude  
rec_windowed_H_in_dB=20*log10(abs(rec_windowed_H)/abs(rec_windowed_H(1)));  
% The Hanning windowed version follows.  
hanning_window=hanning(N);  
hanning_windowed_h=h((Length-N)/2+1:(Length+N)/2).*hanning_window.';  
[hanning_windowed_H,W2]=freqz(hanning_windowed_h,1);  
hanning_windowed_H_in_dB=20*log10(abs(hanning_windowed_H)/abs(hanning_windowed_H(1)));  
% Plotting commands follow.
```

Problema Ilustrativo 6.5 un canal de radio de dos trayectos (multipath) puede ser modelada en el dominio del tiempo, como se ilustra en la Figura 6.12. Su respuesta de impulso se puede expresar como

$$c(t, \tau) = b_1(t)\delta(\tau) + b_2(t)\delta(\tau - \tau_d) \quad (6.3.5)$$

donde $b_1(t)$ y $b_2(t)$ son procesos aleatorios que representan el comportamiento de la propagación variable en el tiempo del canal y τ_d es el retardo entre las dos componentes de trayectos múltiples. El problema es para simular un canal de este tipo en la computadora.

Solución

Nuestro modelo $b_1(t)$ y $b_2(t)$ como procesos aleatorios gaussianos generados por procesos de ruido gaussiano blanco a través de filtros de paso bajo. En tiempo discreto, podemos utilizar filtros IIR digitales relativamente simples excitados por secuencias de ruido blanco gaussiano (WGN). Por ejemplo, un filtro de paso bajo digital simple que tiene dos polos idénticos se describe mediante la transformada z

$$H(z) = \frac{(1-p)^2}{(1-pz^{-1})^2} = \frac{(1-p)^2}{1 - 2pz^{-1} + p^2z^{-2}} \quad (6.3.6)$$

o la ecuación diferencial correspondiente

$$b_n = 2pb_{n-1} - p^2b_{n-2} + (1-p)^2w_n \quad (6.3.7)$$

donde $\{w_n\}$ es la secuencia de entrada WGN, $\{b_n\}$ es la secuencia de salida, y $0 < p < 1$ es la posición del polo. La posición del polo controla el ancho de banda del filtro y, por lo tanto, la tasa de variación de $\{b_n\}$. Cuando p es cercano a la unidad (cerca del círculo unidad), el ancho de banda del filtro es estrecha, mientras que cuando p es cercana a cero, el ancho de banda es amplia. Por lo tanto, cuando p es cerca del círculo unidad en el plano z, la secuencia de salida del filtro cambia más lentamente en comparación con el caso en que el polo está cerca del origen.

La figura 6.13 muestra las secuencias de salida $\{b_{1n}\}$ y $\{b_{2n}\}$ generados al pasar secuencias WGN estadísticamente independientes a través de un filtro que tiene $p = 0,99$. La respuesta al impulso del canal - de tiempo discreto

$$c_n = b_{1,n} + b_{2,n-d} \quad (6.3.8)$$

También se muestra, con $d= 5$ muestras de retraso. La figura 6.14 muestra las secuencias $\{b_{1,n}\}$, $\{b_{2,n}\}$ y $\{c_n\}$ cuando $p = 0,9$.

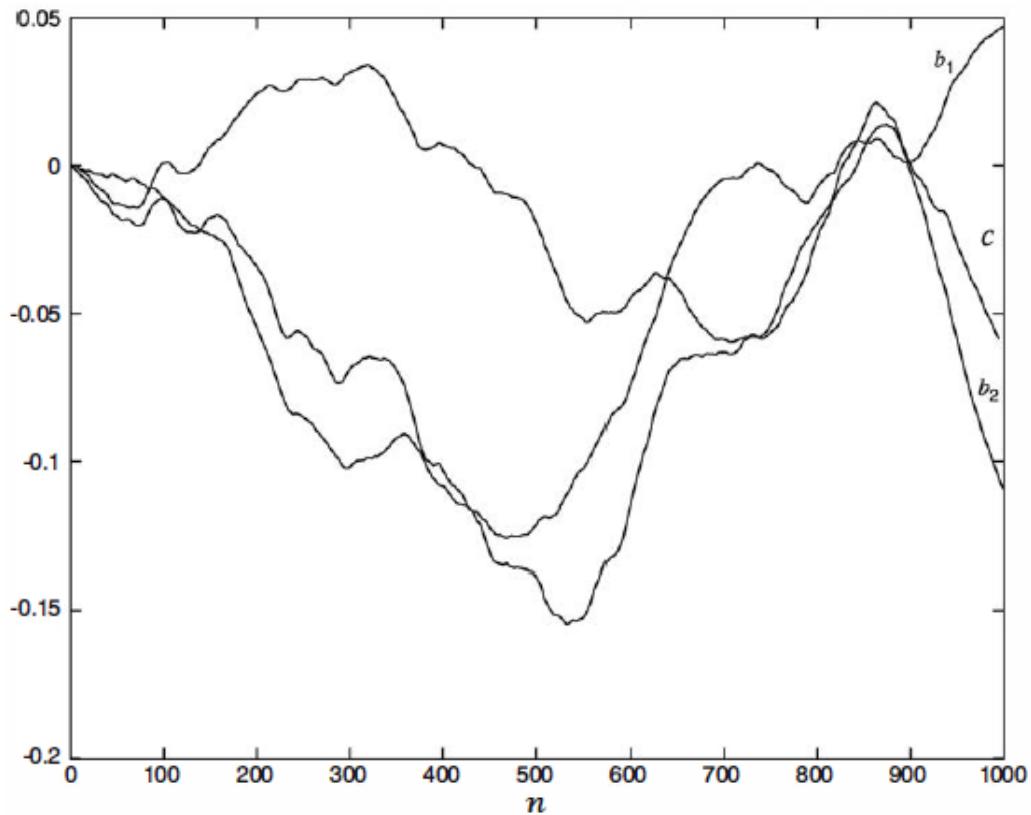


Figura 6.13: Salidas b_{1n} y b_{2n} de un filtro de paso bajo y de la c_n resultante para $p = 0,99$

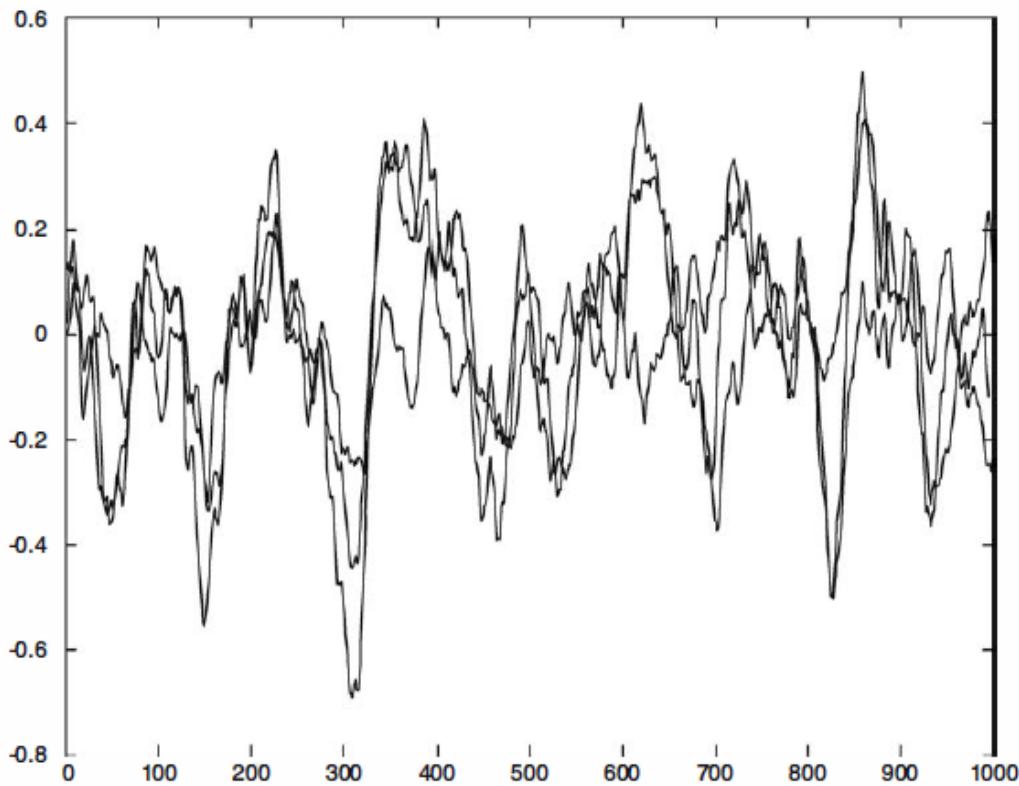


Figura 6.14: Salidas b_{1n} , b_{2n} , y c_n con el polo en $p = 0,9$ (de arriba a abajo en $n = 1000$, b_{1n} , c_n , y b_{2n})

Problema ilustrativo 6.6 [Estimación del espectro Autocorrelación y Espectro de Potencia de los coeficientes de los dos caminos del canal de Radio]

Para $p=0,99$ en el Problema ilustrativo 6.5, calcular y representar gráficamente las estimaciones de la función de autocorrelación de la secuencia $\{b_1\}$, $\{b_2\}$ y $\{c_n\}$ y sus espectros de potencia. La comparación de los espectros de potencia con la magnitud al cuadrado de la respuesta de frecuencia del filtro $H(f)$, es decir, $|H(f)|^2$, en donde $H(f)$ se obtiene a partir de (6.3.6) mediante la evaluación de $H(z)$ del círculo unitario.

Solución

La Figura 6.15 (a) ilustra la función de autocorrelación de las tres secuencias y en la figura 6.15 (b) se ilustra los espectros de potencia. Observamos que las funciones de autocorrelación son muy similares y los espectros de potencia correspondiente también son similares. También observamos que los espectros de potencia son una buena aproximación a la teoría verdadera del espectro de potencia $|H(f)|^2$ cuando la entrada al filtro es ruido blanco.

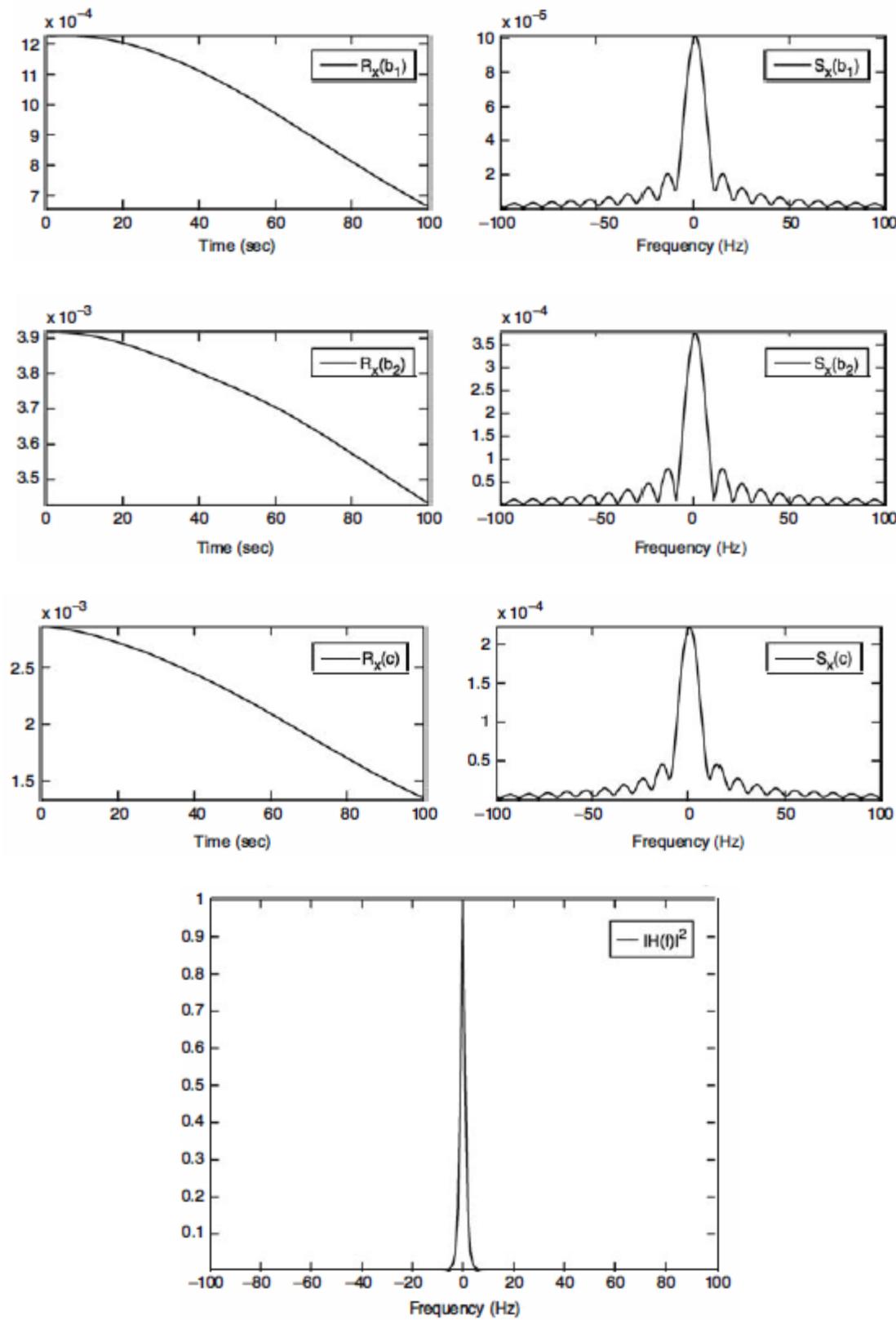


Figura 6.15: funciones de autocorrelación y los espectros del Problema Ilustrativo 6.6

La secuencia de comandos de MATLAB para el problema se da a continuación.

M-FILE

```
% MATLAB script for Illustrative Problem 6.6

N=1000; % Length of sequence
M=100; % Autocorrelation function length
Fs=N; % Sampling frequency
NFFT = 2^nextpow2(N); % Next power of 2 from length of y
f=F_s/2*linspace(-0.5,0.5,NFFT/2+1);
F=1/2*linspace(-0.5,0.5,NFFT/2+1);
p=0.99;
d=5; % Time delay between the two paths
% Preallocation for speed:
b1=zeros(1,N); b2=zeros(1,N); c=zeros(1,N);
% Input WGN sequence
w=randn(2,N);
% Output sequences:
b1(1)=(1-p)^2*w(1,1);
b1(2)=2*p*b1(1)+(1-p)^2*w(1,2);
b2(1)=(1-p)^2*w(2,1);
b2(2)=2*p*b2(1)+(1-p)^2*w(2,2);
u=1:M+1;
for n=3:N
    b1(n)=2*p*b1(n-1)-p^2*b1(n-2)+(1-p)^2*w(1,n);
    b2(n)=2*p*b2(n-1)-p^2*b2(n-2)+(1-p)^2*w(2,n);
end
% Channel impulse response:
for n=1:5
    c(n)=b1(n);
end
for n=6:N
    c(n)=b1(n)+b2(n-d);
end
% Autocorrelation calculations:
Rx_b1=Rx_est(b1,M);
Rx_b2=Rx_est(b2,M);
Rx_c =Rx_est(c,M);
% Power spectra calculations:
Sx_b1=fftshift(abs(fft(Rx_b1,NFFT)/N));
Sx_b2=fftshift(abs(fft(Rx_b2,NFFT)/N));
Sx_c =fftshift(abs(fft(Rx_c,NFFT)/N));
% Calculation of H(f):
z=exp(1j*2*pi*F);
num=(1-p)^2;
denom=(1-p*z.^-1).^2;
H=num./denom;
```

```

% Plot the results:
subplot(3,2,1)
plot(Rx_b1)
axis([0 M min(Rx_b1) max(Rx_b1)])
xlabel('Time (sec)')
legend('R_x(b_1)')
subplot(3,2,2)
plot(f,Sx_b1(NFFT/4:3*NFFT/4))
xlabel('Frequency (Hz)')
axis([-100 100 min(Sx_b1) max(Sx_b1)])
legend('S_x(b_1)')
subplot(3,2,3)
plot(Rx_b2)
axis([0 M min(Rx_b2) max(Rx_b2)])
xlabel('Time (sec)')
legend('R_x(b_2)')
subplot(3,2,4)
plot(f,Sx_b2((NFFT/4:3*NFFT/4)))
xlabel('Frequency (Hz)')
axis([-100 100 min(Sx_b2) max(Sx_b2)])
legend('S_x(b_2)')
subplot(3,2,5)
plot(Rx_c)
axis([0 M min(Rx_c) max(Rx_c)])
xlabel('Time (sec)')
legend('R_x(c)')
subplot(3,2,6)
plot(f,Sx_c((NFFT/4:3*NFFT/4)))
axis([-100 100 min(Sx_c) max(Sx_c)])
xlabel('Frequency (Hz)')
legend('S_x(c)')
figure
plot(f,abs(H).^2)
axis([-100 100 min(abs(H).^2) max(abs(H).^2)])
xlabel('Frequency (Hz)')
legend('|H(f)|^2')

```

6.4 Caracterización de interferencia intersímbolo:

En un sistema de comunicación digital, la distorsión de un canal provoca interferencia intersímbolos (ISI). En esta sección, se presenta un modelo que caracteriza a la ISI. Por simplicidad, suponemos que la señal transmitida es una señal de banda base PAM. Sin embargo, este tratamiento se extiende fácilmente para el portador (linealmente) de señales moduladas que se discuten en el siguiente capítulo.

La señal transmitida PAM se expresa como

$$s(t) = \sum_{n=0}^{\infty} a_n g(t - nT) \quad (6.4.1)$$

Donde $g(t)$ es la forma básica de pulso que se selecciona para controlar las características espectrales de la señal transmitida, $\{a_n\}$ es la secuencia de símbolos de información de transmisión seleccionados a partir de una constelación de señal que consta de puntos M , y T es el intervalo de la señal ($1/T$ es la tasa de símbolo).

La señal $s(t)$ se transmite por un canal de banda de base, que puede estar caracterizado por una respuesta de frecuencia $C(f)$. En consecuencia, la señal recibida se puede representar como

$$r(t) = \sum_{n=0}^{\infty} a_n h(t - nT) + w(t) \quad (6.4.2)$$

Donde $h(t) = g(t) * c(t)$, $c(t)$ es la respuesta al impulso del canal, $*$ denota la convolución, y $w(t)$ representa el ruido aditivo en el canal. Para caracterizar ISI, supongamos que la señal recibida se pasa a través de un filtro de recepción y luego se muestrea a la tasa de $1/T$ muestras/segundo. En general, el filtro óptimo en el receptor está adaptado a la señal recibida de impulsos $h(t)$. Por lo tanto, la respuesta en frecuencia de este filtro es $H^*(f)$. Denotamos su salida como

$$y(t) = \sum_{n=0}^{\infty} a_n x(t - nT) + v(t) \quad (6.4.3)$$

Donde $x(t)$ es la respuesta de impulso de la señal del filtro de recepción que es, $X(f) = |H(f)|^2$, y $v(t)$ es la respuesta del filtro de recepción para el ruido $w(t)$. Ahora, si $y(t)$ se muestrea en los tiempos $t = kT$, para $k = 0, 1, 2, \dots$, Tenemos

$$y(kT) = \sum_{n=0}^{\infty} a_n x(kT - nT) + v(kT)$$

$$y_k = \sum_{n=0}^{\infty} a_n x_{k-n} + v_k, \quad k = 0, 1, \dots \quad (6.4.4)$$

Los valores de la muestra $\{y_k\}$ se pueden expresar como

$$y_k = x_0 \left(a_k + \frac{1}{x_0} \sum_{\substack{n=0 \\ n \neq k}}^{\infty} a_n x_{k-n} \right) + v_k, \quad k = 0, 1, \dots \quad (6.4.5)$$

El término x_0 es un factor de escala arbitraria, lo proponemos igual a la unidad para mayor comodidad. Entonces

$$y_k = a_k + \sum_{\substack{n=0 \\ n \neq k}}^{\infty} a_n x_{k-n} + v_k \quad (6.4.6)$$

El término a_k representa el símbolo de la información deseada en el instante de muestreo k -ésimo, y el término

$$\sum_{\substack{n=0 \\ n \neq k}}^{\infty} a_n x_{k-n} \quad (6.4.7)$$

Representa el ISI, y v_k es el ruido aditivo en el instante de muestreo k -ésimo.

La cantidad de ISI y ruido en un sistema digital de comunicaciones se pueden ver en un osciloscopio. Para señales PAM, podemos visualizar la señal recibida $y(t)$ en la entrada vertical con el conjunto velocidad de barrido horizontal en $1/T$. La pantalla del osciloscopio resultante se llama diagrama de ojo debido a su parecido con el ojo humano. Por ejemplo, la Figura 6.16 ilustra los diagramas de ojo para dos y cuatro niveles de modulación PAM. El efecto de la ISI es causar que el ojo se cierre, lo que reduce el margen de ruido aditivo para causar errores. La Figura 6.17 ilustra gráficamente el efecto de la ISI en la reducción de la apertura de un ojo para 2 niveles de modulación. Tenga en cuenta que la ISI distorsiona la posición de los cruces por cero y causa una reducción en la apertura de los ojos. Por lo tanto, hace que el sistema sea más sensible a un error de sincronización.

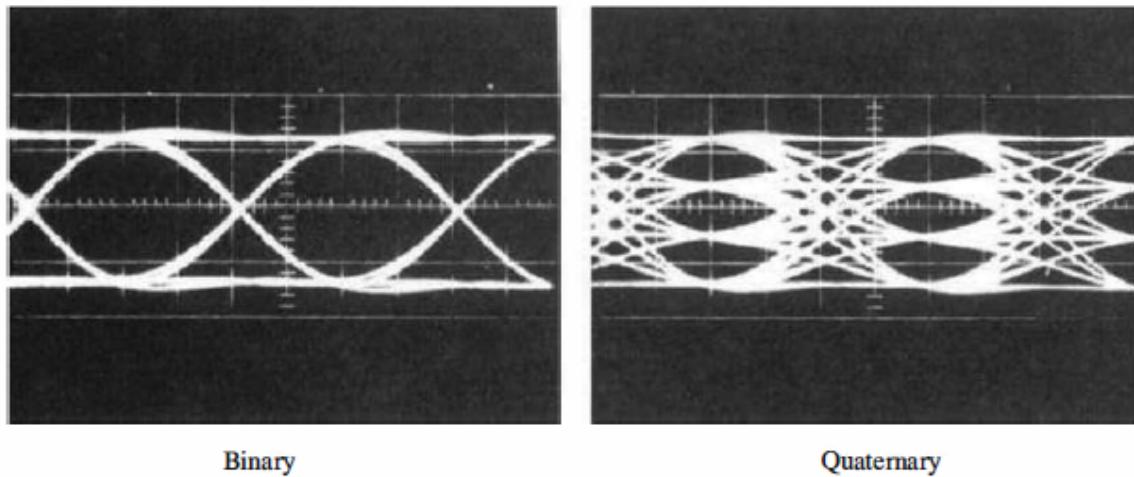


Figura 6.16: Ejemplos de patrones de ojo para modulación por desplazamiento de amplitud binaria y cuaternaria (o PAM)

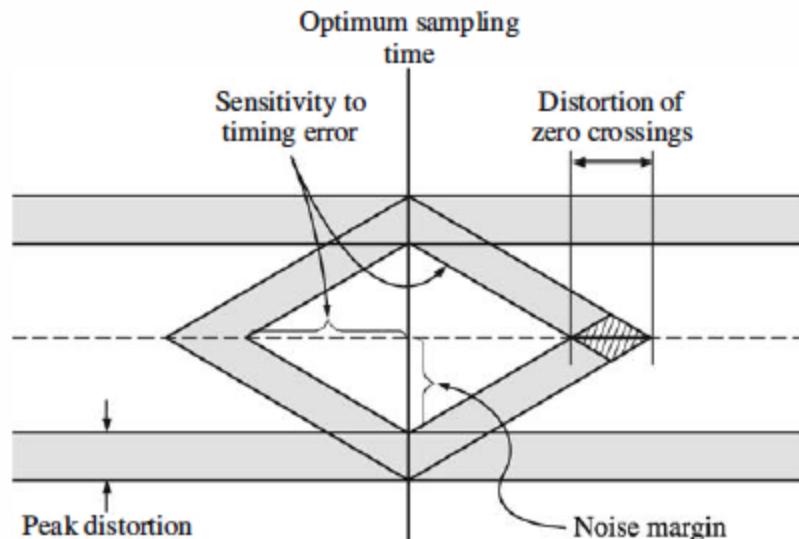


Figura 6.17: Efecto de la ISI en la apertura de los ojos

Problema Ilustrativo 6.7 En este problema tenemos en cuenta el efecto de la interferencia intersímbolos (ISI) en la secuencia de señal recibida $\{y_k\}$ para dos canales que dan lugar a las secuencias $\{y_k\}$ de la siguiente manera:

Channel 1

$$x_n = \begin{cases} 1, & n = 0 \\ -0.25, & n = \pm 1 \\ 0.1, & n = \pm 2 \\ 0, & \text{otherwise} \end{cases}$$

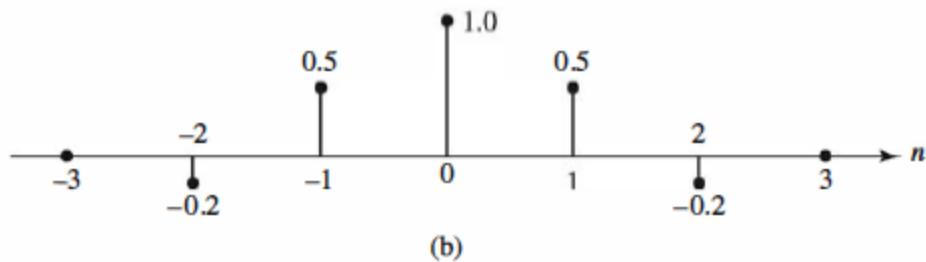
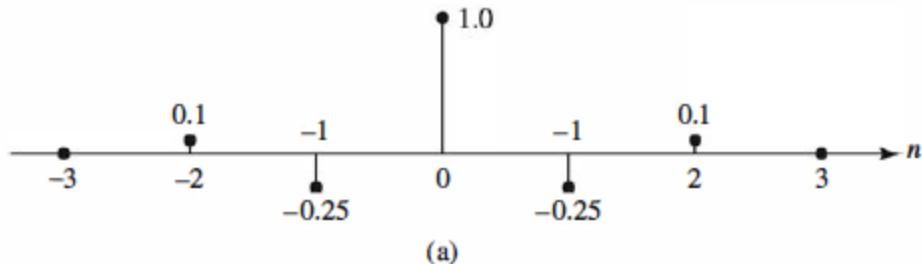


Figura 6.18: Modelos de canales FIR con la ISI. (a)Canal 1. (b) Canal 2

Channel 2

$$x_n = \begin{cases} 1, & n = 0 \\ 0.5, & n = \pm 1 \\ -0.2, & n = \pm 2 \\ 0, & \text{otherwise} \end{cases}$$

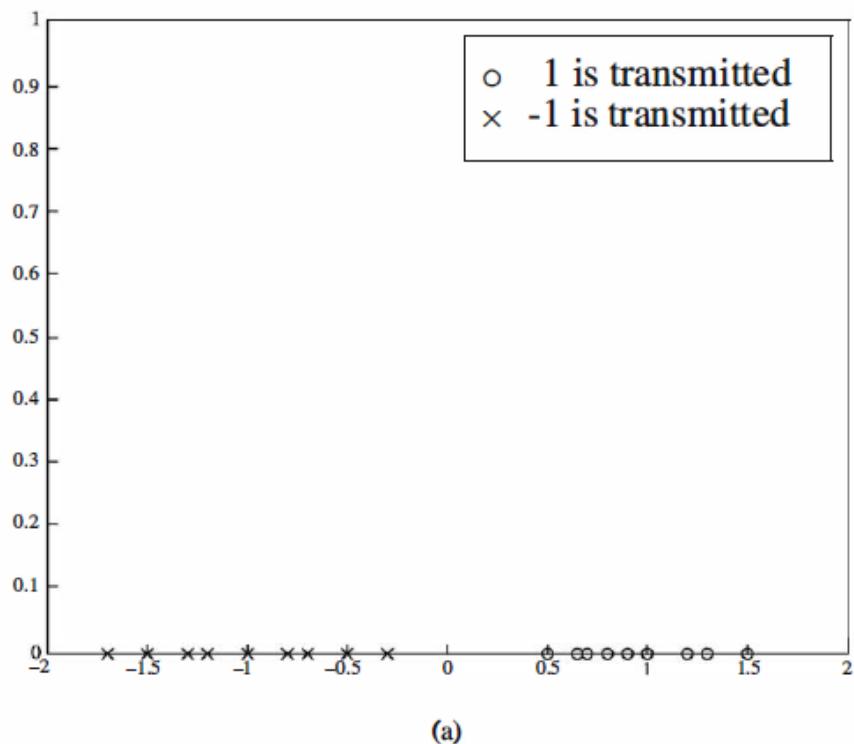
Tenga en cuenta que en estos canales, el ISI se limita a dos símbolos a cada lado de la señal transmitida deseada. Por lo tanto, la cascada de los filtros del transmisor y receptor y el canal en los instantes de muestreo están representados por el equivalente filtro FIR del canal en tiempo discreto que se muestra en la Figura 6.18. Supongamos ahora que la secuencia señal transmitida es binaria, es decir, $\{a_n = \pm 1\}$. Entonces, para el canal 1, la secuencia de señal recibida $\{y_n\}$ en la ausencia de ruido se muestra en la Figura 6.19 (a), y con ruido gaussiano blanco aditivo que tiene una varianza de $\sigma = 0.1$, la secuencia de señal recibida se muestra en la Figura 6.19 (b). Tomamos nota de que, en ausencia de ruido, la ISI sola no causa errores en el detector que compara la secuencia señal recibida $\{y_n\}$ con el umbral se establece en cero. Por lo tanto, el diagrama de ojo está abierto en la ausencia de

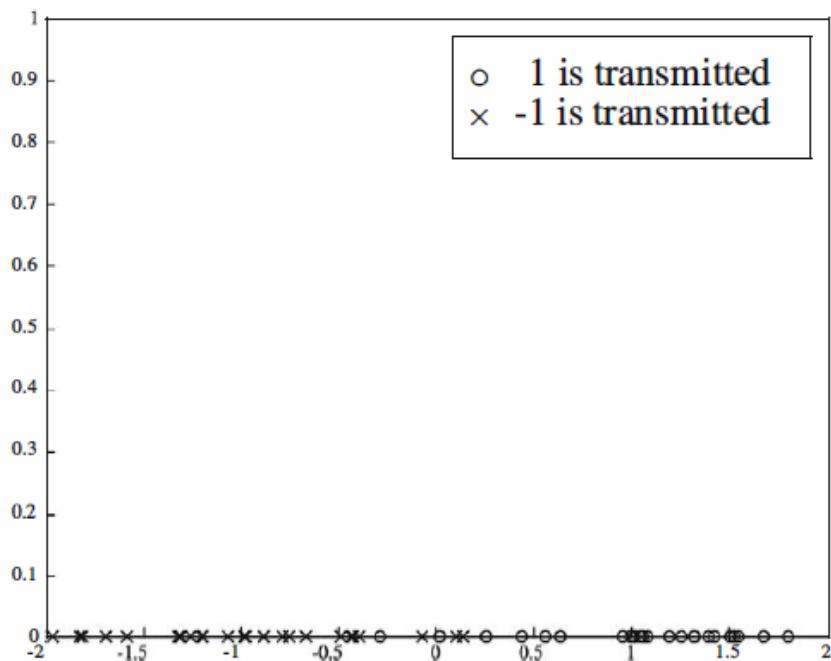
ruido. Sin embargo, cuando el ruido aditivo es suficientemente grande, se producen errores.

En el caso de Canal 2, la secuencia libre de ruido y ruidoso ($\sigma = 0,1$) $\{y_n\}$ es como se ilustra en la Figura 6.20. Ahora, se observa que la ISI puede causar errores en el detector que compara la secuencia recibida $\{y_n\}$ con el umbral fijado en cero, incluso en la ausencia de ruido. Por lo tanto, para esta característica de canal, el ojo está completamente cerrado.

6.5 Diseño de sistemas de comunicación para los canales de banda limitada

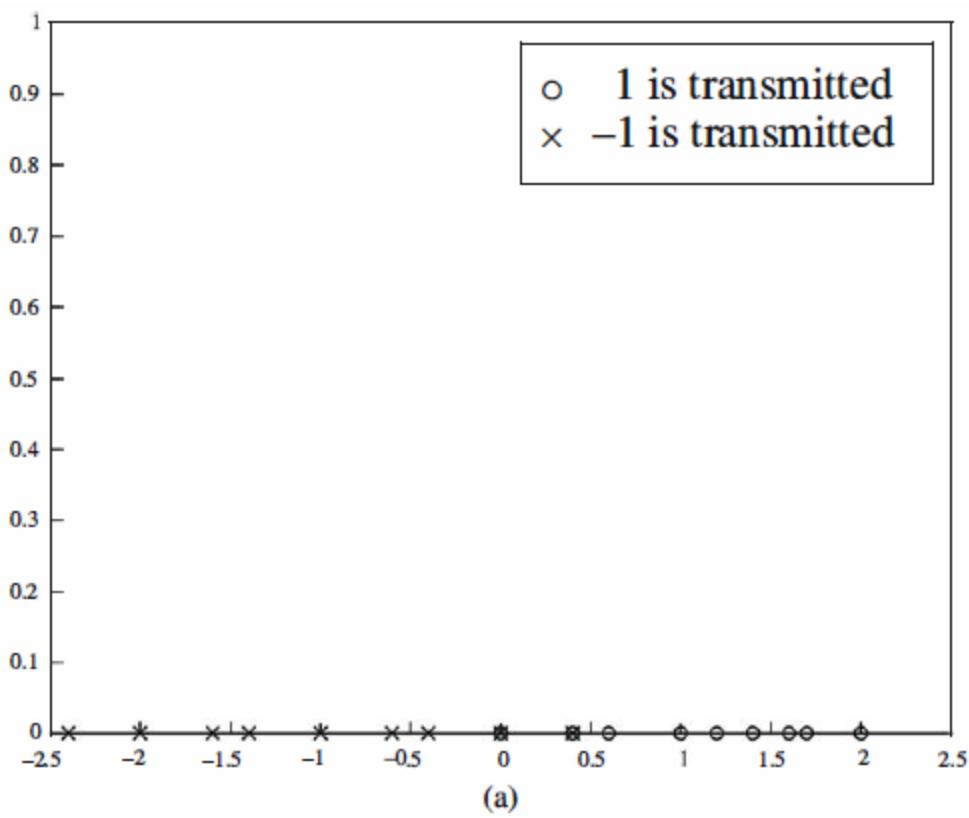
En esta sección se considera el diseño de los filtros transmisor y receptor que son adecuados para un canal de banda base de banda limitada. Se consideran dos casos. En el primero





(b)

Figura 6.19: La salida del modelo de canal 1 sin y con AWGN (ruido gaussiano blanco aditivo). (a) Ningún ruido. (b) el ruido gaussiano aditivo con un $\sigma^2 = 0.1$



(a)

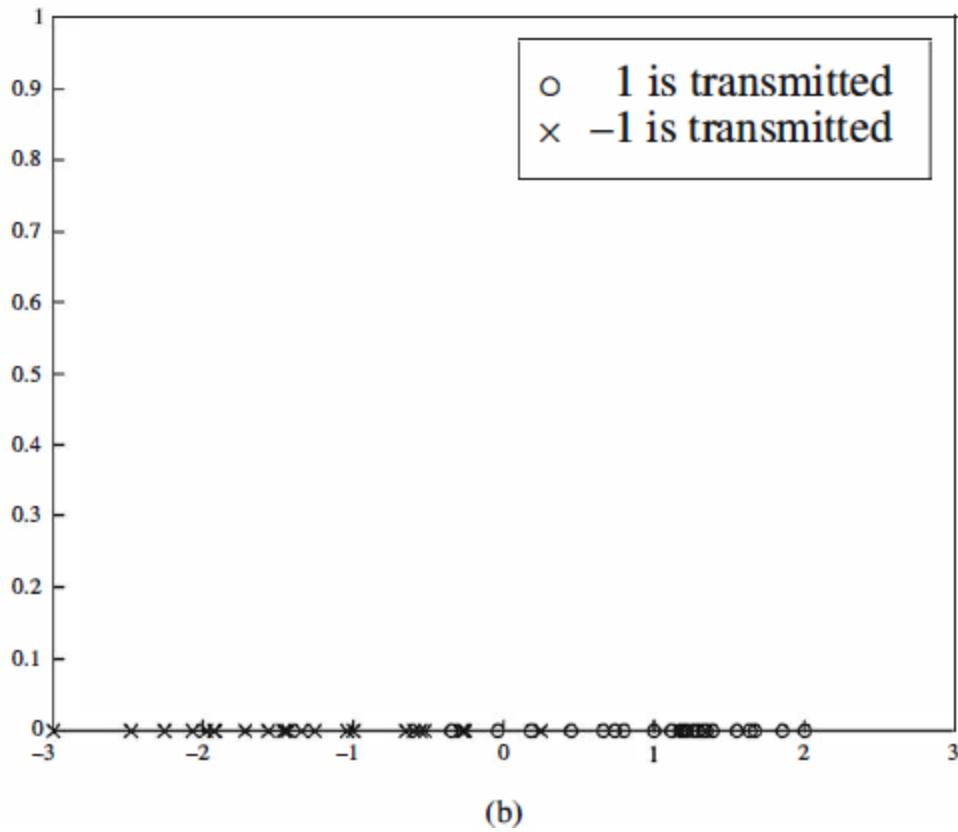


Figura 6.20: La salida del modelo de canal 2 sin y con AWGN. (a) Ningún ruido. (b) el ruido gaussiano aditivo con varianza $\sigma^2 = 0.1$

En el primer caso, el diseño se basa en filtros de transmisor y receptor que resultan en cero ISI. En el segundo caso, el diseño se basa en filtros de transmisor y receptor que tienen un (predeterminada) cantidad de ISI especificada. Por lo tanto, el segundo enfoque de diseño conduce a una cantidad controlada de la ISI. Las señales correspondientes transmitidas se denominan señales de respuesta parciales. En ambos casos se supone que el canal es ideal; es decir, $A(f)$ y $\tau(f)$ son constantes dentro de la anchura de banda de canal W . Por simplicidad, suponemos que $A(f) = 1$ y $\tau(f) = 0$.

6.5.1 Diseño de señal para ISI cero:

El diseño de las señales de banda limitada con cero ISI es un problema considerado por Nyquist hace unos 70 años. Se demostró que una condición necesaria y suficiente para una señal $x(t)$ para tener cero ISI - es que,

$$x(nT) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \quad (6.5.1)$$

Es que su transformada de Fourier $X(f)$ satisface

$$\sum_{m=-\infty}^{\infty} X\left(f + \frac{m}{T}\right) = T \quad (6.5.2)$$

Donde $1/T$ es la tasa de símbolo.

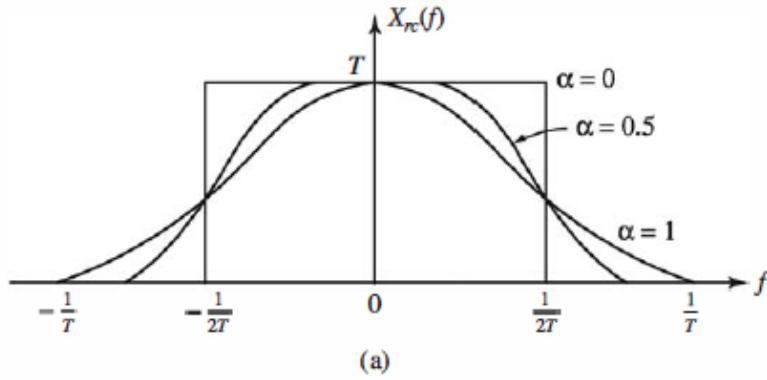
En general, hay muchas señales que pueden ser diseñadas para tener esta propiedad. Una de las señales más comúnmente utilizados en la práctica que tiene una característica en respuesta en frecuencia es el coseno-realzado, que se define como

$$X_{rc}(f) = \begin{cases} T, & 0 \leq |f| \leq \frac{(1-\alpha)}{2T} \\ \frac{T}{2} \left[1 + \cos \frac{\pi T}{\alpha} \left(|f| - \frac{1-\alpha}{2T} \right) \right], & \frac{1-\alpha}{2T} < |f| \leq \frac{1+\alpha}{2T} \\ 0, & |f| > \frac{1+\alpha}{2T} \end{cases} \quad (6.5.3)$$

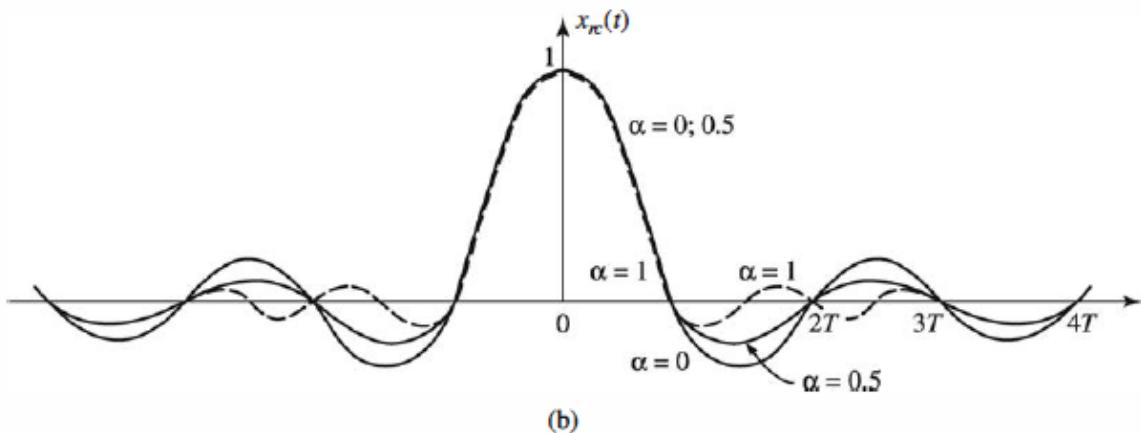
Donde α se llama el factor de atenuación(o factor *rolloff*), que toma valores en el rango de $0 \leq \alpha \leq 1$, y $1/T$ es la tasa de símbolo. La respuesta de frecuencia $X_{rc}(f)$ se ilustra en la Figura 6.21 (a) para $\alpha=0$, $\alpha=1/2$, y $\alpha=1$. Obsérvese que cuando $\alpha=0$, $X_{rc}(f)$ se reduce a un ideal "barrera de bloqueo", físicamente respuesta de frecuencia no alcanzable con una ocupación media del ancho de banda $1/2T$. La frecuencia $1/2T$ se denomina *frecuencia de Nyquist*. Para $\alpha > 0$, el ancho de banda ocupado por la señal deseada $X_{rc}(f)$ más allá de la frecuencia de Nyquist $1/2T$ se llama el *exceso de ancho de banda*, normalmente expresado como un porcentaje de la frecuencia Nyquist. Por ejemplo, cuando $\alpha=1/2$, el exceso de ancho de banda es 50%, y cuando $\alpha = 1$, el exceso de ancho de banda es 100%. El impulso de la señal $x_{rc}(t)$ que tiene el espectro de coseno realzado es

$$x_{rc}(t) = \frac{\sin \pi t/T}{\pi t/T} \frac{\cos(\pi \alpha t/T)}{1 - 4\alpha^2 t^2/T^2} \quad (6.5.4)$$

La Figura 6.21 (b) ilustra $x_{rc}(t)$ para $\alpha = 0, 1/2, 1$. Porque $X_{rc}(f)$ satisface (6.5.2), observamos que $x_{rc}(t) = 1$ en $t = 0$ y $x_{rc}(t) = 0$ en $t = kT$, para $k = \pm 1, \pm 2, \dots$. en consecuencia, en los instantes de muestreo $t = kT$, $k \neq 0$, no hay ninguna ISI de símbolos adyacentes cuando no hay distorsión del canal.



(a)



(b)

Figura 6.21: respuesta de frecuencia del coseno-realizado y la forma del pulso correspondiente. (a) respuesta de frecuencia del coseno-realizado. (b) las formas de pulso del coseno realizado en su respuesta en frecuencia

Sin embargo, en la presencia de distorsión del canal, el ISI dada por (6.4.7) ya no es cero, y se necesita un ecualizador de canal para reducir al mínimo su efecto sobre el rendimiento del sistema. Los ecualizadores de canal son considerados en la Sección 6.6.

En un canal ideal, los filtros de transmisión y recepción están diseñados de manera conjunta para cero ISI en los instantes de muestreo deseadas $t = nT$. Por lo tanto, si $G_T(f)$ es la respuesta de frecuencia del filtro transmisor y $G_R(f)$ es la respuesta de frecuencia del filtro del receptor, entonces el producto (cascada de los dos filtros) $G_T(f) G_R(f)$ está diseñado para producir cero ISI. Por ejemplo, si el producto $G_T(f) G_R(f)$ se selecciona como

$$G_T(f) G_R(f) = X_{rc}(f) \quad (6.5.5)$$

Donde $X_{rc}(f)$ es la característica de respuesta de frecuencia de coseno realizado, a continuación, el ISI en los tiempos de muestreo $t = nT$ es cero.

Problema ilustrativo 6.8 Deseamos diseñar una implementación digital de los filtros de transmisor y receptor $G_T(f)$ y $G_R(f)$ de manera que sus productos satisface (6.5.5) y $G_R(f)$ es el filtro adaptado a $G_T(f)$.

Solución: La forma más sencilla de diseñar e implementar los filtros de transmisión y recepción en forma digital es emplear filtros FIR con fase lineal (respuesta al impulso simétrico). La respuesta de magnitud deseada es

$$|G_T(f)| = |G_R(f)| = \sqrt{X_{rc}(f)} \quad (6.5.6)$$

Donde $X_{rc}(f)$ viene dada por (6.5.3). La respuesta de frecuencia está relacionada con la respuesta de impulso del filtro digital por la ecuación

$$G_T(f) = \sum_{n=-(N-1)/2}^{(N-1)/2} g_T(n) e^{-j2\pi f n T_s} \quad (6.5.7)$$

Donde T_s es el intervalo de muestreo y N es la longitud del filtro. Tenga en cuenta que N es impar. Debido a que $G_T(f)$ está limitada en banda, podemos seleccionar los F_s frecuencia de muestreo para ser al menos $2/T$. Nuestra elección es

$$F_s = \frac{1}{T_s} = \frac{4}{T}$$

O, equivalentemente, $T_s = T/4$. Por lo tanto la frecuencia de muestreo es $F_s/2 = 2/T$. Debido a que $G_T(f) = \sqrt{X_{rc}(f)}$, se puede muestrear $X_{rc}(f)$ en puntos equidistantes en frecuencia, con separación de frecuencia $\Delta f = F_s/N$. Así tenemos

$$\sqrt{X_{rc}(m\Delta f)} = \sqrt{X_{rc}(mF_s/N)} = \sum_{n=-(N-1)/2}^{(N-1)/2} g_T(n) e^{-j2\pi mn/N} \quad (6.5.8)$$

La relación de la transformada inversa es

$$g_T(n) = \sum_{m=-(N-1)/2}^{(N-1)/2} \sqrt{X_{rc}\left(\frac{4m}{NT}\right)} e^{j2\pi mn/N}, \quad n = 0, \pm 1, \dots, \pm \frac{N-1}{2} \quad (6.5.9)$$

Desde $g_T(n)$ es simétrica, la respuesta de impulso del filtro de fase lineal del transmisor deseado se obtiene retrasando $g_T(n)$ por $(N-1)/2$ muestras. Las secuencias de comandos de MATLAB para este cálculo se dan a continuación.

M-FILE

```
% MATLAB script for Illustrative Problem 6.8.
echo on
N=31;
T=1;
alpha=1/4;
n=-(N-1)/2:(N-1)/2; % the indices for g_T
% The expression for g_T is obtained next.
for i=1:length(n),
    g_T(i)=0;
    for m=-(N-1)/2:(N-1)/2,
        g_T(i)=g_T(i)+sqrt(xrc(4*m/(N*T),alpha,T))*exp(j*2*pi*m*n(i)/N);

    echo off ;
end;
end;
echo on;
g_T=real(g_T) ; % The imaginary part is due to the finite machine precision
% Derive g_T(n-(N-1)/2).
n2=0:N-1;
% Get the frequency response characteristics.
[G_T,W]=freqz(g_T,1);
% normalized magnitude response
magG_T_in_dB=20*log10(abs(G_T)/max(abs(G_T)));
% impulse response of the cascade of the transmitter and the receiver filters
g_R=g_T;
imp_resp_of_cascade=conv(g_R,g_T);
% Plotting commands follow.
```

M-FILE

```
function [y] = xrc(f,alpha,T);
% [y]=xrc(f,alpha,T)
%           Evaluates the expression Xrc(f). The parameters alpha and T
%           must also be given as inputs to the function.
if (abs(f) > ((1+alpha)/(2*T))),
    y=0;
elseif (abs(f) > ((1-alpha)/(2*T))),
    y=(T/2)*(1+cos((pi*T/alpha)*(abs(f)-(1-alpha)/(2*T))));
else
    y=T;
end;
```

La Figura 6.22 (a) ilustra $G_T(n - \frac{N-1}{2})$, para $n = 0, 1, \dots, N - 1$ y para $\alpha = 1/4$ y $N = 31$. Las características de respuesta de frecuencia correspondientes se muestran en la Figura 6.22 (b). Tenga en cuenta que la respuesta de frecuencia ya no es cero porque $|f| \geq (1 + \alpha)$ que es, porque el filtro digital tiene duración finita. Sin embargo, los lóbulos laterales en el espectro son relativamente pequeñas. Una mayor reducción en los lóbulos laterales se puede lograr mediante el aumento de N . Finalmente, en la figura 6.23, se muestra la respuesta de impulso de la cascada de los filtros *FIR* de transmisor y receptor. Esto puede ser comparado con la respuesta de impulso ideales obtenida mediante el muestreo de $X_{rc}(t)$ a una tasa $F_s = 4/T$.

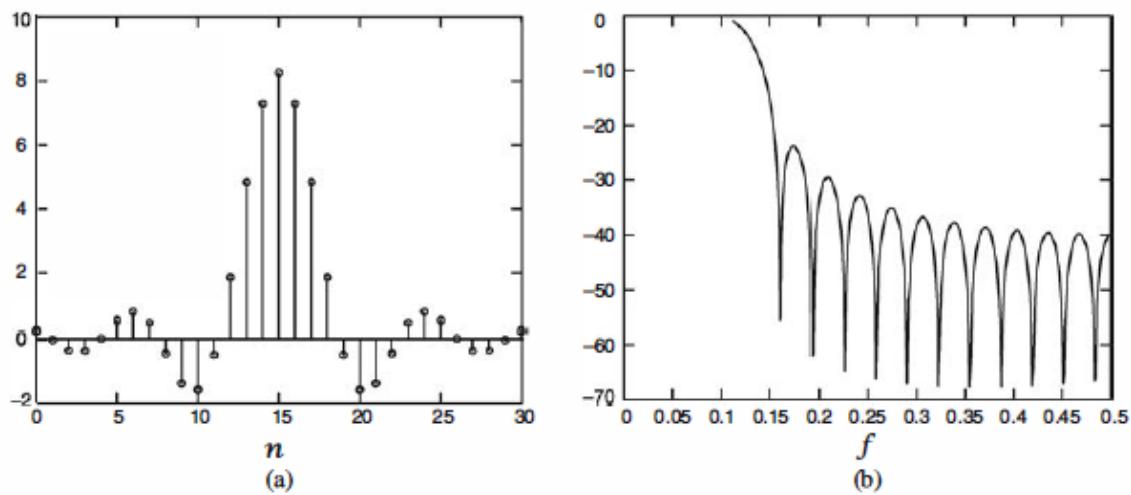


Figura 6.22: Respuesta al impulso y respuesta en frecuencia del filtro *FIR* de tiempo discreto truncado en el transmisor

6.6 Ecualizadores Lineales:

El tipo más común de ecualizador de canal utilizado en la práctica para reducir la ISI es un filtro *FIR* lineal con coeficientes ajustables $\{C_i\}$, como se muestra en la Figura 6.28. En los canales cuyas características de respuesta de frecuencia son desconocidos pero invariante en el tiempo, podemos medir las características del canal y ajustar los parámetros del ecualizador; una vez ajustado, los parámetros permanecen fijos durante la transmisión de datos. Estos ecualizadores se llaman ecualizadores preestablecidos. Por otro lado, ecualizadores adaptativos actualizan sus parámetros sobre una base periódica durante la transmisión de datos, por lo que son capaces de seguimiento de una respuesta de canal variable en el tiempo lentamente.

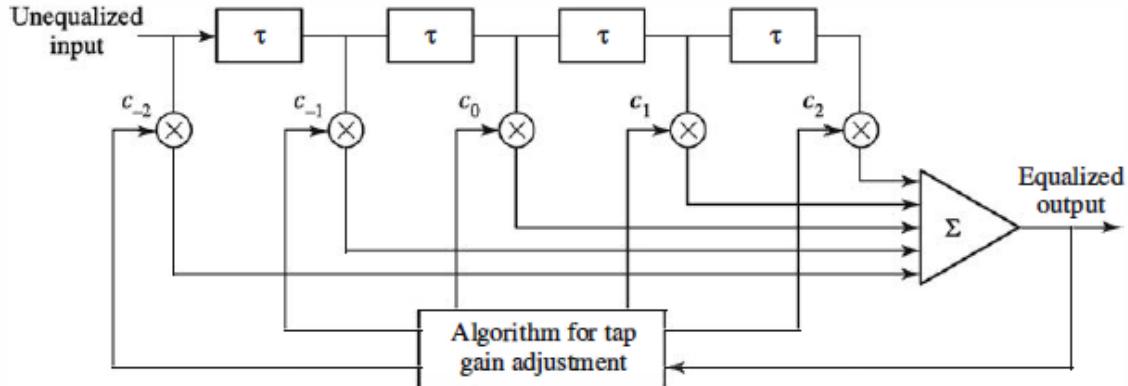


Figura 6 .28: Filtro transversal lineal

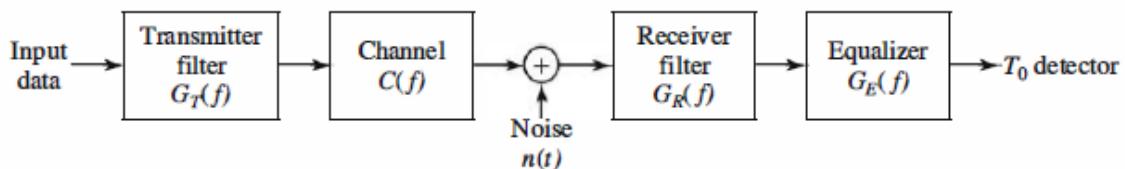


Figura 6.29: Diagrama de bloques de un sistema con un ecualizador

En primer lugar, vamos a considerar las características de diseño de un ecualizador lineal desde un punto de vista del dominio de la frecuencia. La Figura 6.29 muestra un diagrama de bloques de un sistema que emplea un filtro lineal como un ecualizador de canal. El demodulador consta de un filtro receptor con respuesta de frecuencia $G_R(f)$ en cascada con un filtro de compensación de canal que tiene una respuesta de frecuencia $G_E(f)$. Como se indica en la sección anterior, la respuesta del filtro receptor $G_R(f)$ se corresponde con la respuesta de que el transmisor es, $G_R(f) = G_T^*(f)$ -- y el producto $G_R(f) G_T(f)$ por lo general está diseñado de modo que o bien no es cero ISI en los instantes de muestreo como, por ejemplo, cuando $G_R(f) G_T(f) = X_{rc}(f)$, o ISI controlada para las señales de respuesta parcial.

Para el sistema mostrado en la Figura 6.29, en la que la respuesta de frecuencia de canal no es ideal, la condición deseada para el cero ISI es:

$$G_T(f)C(f)G_R(f)G_E(f) = X_{rc}(f) \quad (6.6.1)$$

Donde $X_{rc}(f)$ es la característica espectral deseada de coseno realizado. Debido $G_T(f) G_R(f) = X_{rc}(f)$ por diseño, la respuesta de frecuencia del ecualizador que compensa la distorsión del canal es:

$$G_E(f) = \frac{1}{C(f)} = \frac{1}{|C(f)|} e^{-j\theta_c(f)} \quad (6.6.2)$$

Por lo tanto, la respuesta de amplitud del ecualizador es $|G_E(f)| = 1 / |C(f)|$, y su respuesta de fase es $\Theta_E(f) = -\Theta_C(f)$. En este caso, el ecualizador se dice que es el filtro de canal inverso a la respuesta del canal.

Tomamos nota de que el filtro de canal inverso completamente elimina ISI causada por el canal. Puesto que obliga a la ISI a ser cero en los instantes de muestreo $t = kT$ para $k = 0, 1, \dots$, el ecualizador se llama un *ecualizador forzado a cero*. Por lo tanto, la entrada al detector es simplemente

$$z_k = a_k + \eta_k, \quad k = 0, 1, \dots \quad (6.6.3)$$

Donde η_k representa el ruido aditivo y a_k es el símbolo deseado.

En la práctica, el ISI causada por la distorsión de canal se limita generalmente a un número finito de símbolos a cada lado del símbolo deseado. Por lo tanto, el número de términos que constituyen el ISI en la suma dada por (6.4.7) es finito. Como consecuencia, en la práctica, el ecualizador de canal se implementa como un filtro *FIR* o un filtro transversal, con coeficientes de toma ajustables $\{c_n\}$, como se ilustra en la Figura 6.28. El tiempo de retardo τ entre pulsos adyacentes se puede seleccionar tan grande como T , el intervalo de símbolo, en cuyo caso el *ecualizador FIR* se llama un *ecualizador de símbolos separados*. En este caso, la entrada al ecualizador es la secuencia muestreada dada por (6.4.6). Sin embargo, observamos que cuando la tasa de símbolos es $1/T < 2W$, las frecuencias de la señal recibida por encima de la frecuencia de plegado $1/T$ son alias en frecuencias por debajo de $1/T$. En este caso, el ecualizador compensa la señal interferida (alias) del canal distorsionador.

Por otro lado, cuando se selecciona el retardo de tiempo τ entre pulsos adyacentes de tal manera que $1/\tau \geq 2W > 1/T$, no se produce aliasing; por lo tanto, el ecualizador de canal inverso compensa la verdadera distorsión del canal. Debido a $\tau < T$, se dice que el ecualizador de canal tiene pulsos fraccionadamente espaciados, y se llama *ecualizador fraccionadamente espaciado*. En la práctica, τ se selecciona a menudo a $\tau = T/2$. Observe que, en este caso, la tasa de muestreo en la entrada al filtro de $G_E(f)$ es $2/T$.

La respuesta al impulso de la *FIR* ecualizador es

$$g_E(t) = \sum_{n=-K}^K c_n \delta(t - n\tau) \quad (6.6.4)$$

y la respuesta de frecuencia correspondiente es

$$G_E(f) = \sum_{n=-K}^K c_n e^{-j2\pi f n\tau} \quad (6.6.5)$$

donde $\{C_n\}$ son los $2K + 1$ coeficientes de ecualizador y K se elige suficientemente grande para que el ecualizador se extiende por la longitud de la ISI que es, $2K + 1 \geq L$, donde L es el número de muestras de señal abarcado por el ISI. Debido a que $X(f) = G_y(f) C(f) G_R(f)$ y $x(t)$ es la señal de pulso que corresponde a $X(f)$, el pulso de la señal de salida es igualado

$$q(t) = \sum_{n=-K}^{K} c_n x(t - n\tau) \quad (6.6.6)$$

La condición de cero forzado ahora se puede aplicar a las muestras de $q(t)$ tomadas en los tiempos $t = mT$. Estas muestras son

$$q(mT) = \sum_{n=-K}^{K} c_n x(mT - n\tau), \quad m = 0, \pm 1, \dots, \pm K \quad (6.6.7)$$

Debido a que hay $2K + 1$ coeficientes de ecualizador, podemos controlar sólo $2K + 1$ valores muestrados de $q(t)$. En concreto, podemos forzar a las condiciones

$$\begin{aligned} q(mT) &= \sum_{n=-K}^{K} c_n x(mT - n\tau) \\ &= \begin{cases} 1, & m = 0 \\ 0, & m = \pm 1, \pm 2, \dots, \pm K \end{cases} \end{aligned} \quad (6.6.8)$$

Que puede expresarse en forma matricial como $Xc = q$, donde X es un $(2K + 1) \times (2K + 1)$ matriz con los elementos de $x(mT - n\tau)$, c es el $(2K + 1)$ Coeficiente de vector, y q es el $(2K + 1)$ vector columna con un elemento distinto de cero. Por lo tanto, se obtiene un conjunto de $(2K + 1)$ ecuaciones lineales para los coeficientes del ecualizador cero forzado.

Debemos destacar que el FIR ecualizador forzado a cero no elimina completamente la ISI porque tiene una longitud finita. Sin embargo, a medida que aumenta K , el ISI residual se puede reducir y en el límite de $K \rightarrow \infty$, el ISI se elimina completamente.

Problema ilustrativo 6.11 Considere un canal distorsionado por el pulso $x(t)$, en la entrada al ecualizador, dado por la expresión

$$x(t) = \frac{1}{1 + (2t/T)^2}$$

Donde $1 / T$ es la tasa de símbolo. El pulso se muestrea a la velocidad de $2 / T$ y se iguala por un ecualizador cero forzado. Vamos a determinar los coeficientes de un ecualizador forzado a cero de cinco pulsos.

Solución: De acuerdo con (6.6.8), el ecualizador cero forzado debe satisfacer las ecuaciones

$$q(mT) = \sum_{n=-2}^2 c_n x(mT - \frac{nT}{2}) = \begin{cases} 1, & m = 0 \\ 0, & m = \pm 1, \pm 2 \end{cases}$$

La matriz X con elementos $x(mT - nT/2)$ se da como

$$X = \begin{bmatrix} \frac{1}{5} & \frac{1}{10} & \frac{1}{17} & \frac{1}{26} & \frac{1}{37} \\ 1 & \frac{1}{2} & \frac{1}{5} & \frac{1}{10} & \frac{1}{17} \\ \frac{1}{5} & \frac{1}{2} & 1 & \frac{1}{2} & \frac{1}{5} \\ \frac{1}{17} & \frac{1}{10} & \frac{1}{5} & \frac{1}{2} & 1 \\ \frac{1}{37} & \frac{1}{26} & \frac{1}{17} & \frac{1}{10} & \frac{1}{5} \end{bmatrix} \quad (6.6.9)$$

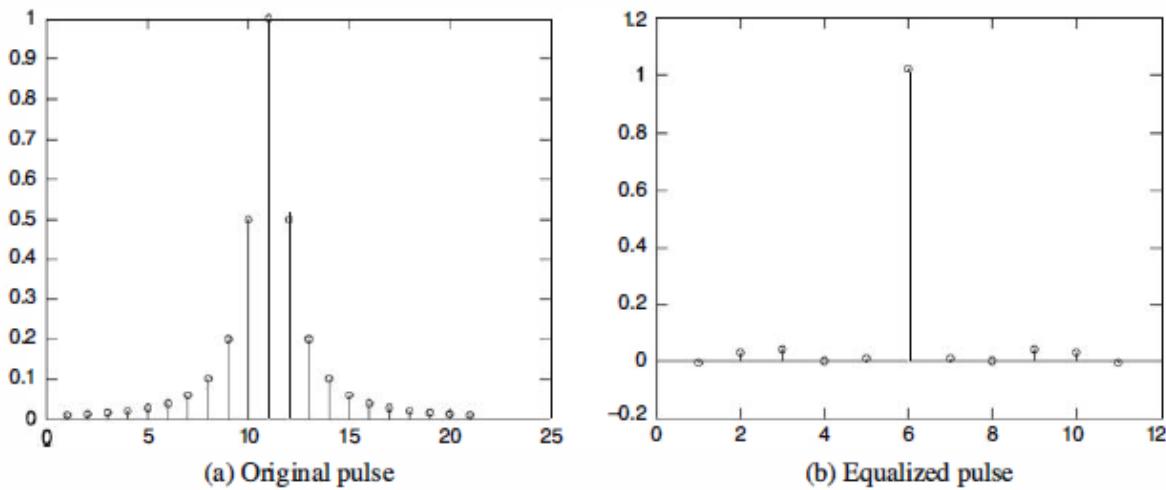


Figura 6.30: Representación gráfica del pulso original y pulso igualado en el Problema Ilustrativo 6.11

El vector de coeficientes c y el vector q se dan como

$$\mathbf{c} = \begin{bmatrix} c_{-2} \\ c_{-1} \\ c_0 \\ c_1 \\ c_2 \end{bmatrix} \quad \mathbf{q} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad (6.6.10)$$

Entonces las ecuaciones lineales $\mathbf{Xc} = \mathbf{q}$ puedan ser resueltas mediante la inversión de la matriz \mathbf{X} . Por lo tanto obtenemos

$$\mathbf{c}_{\text{opt}} = \mathbf{X}^{-1}\mathbf{q} = \begin{bmatrix} -2.2 \\ 4.9 \\ -3 \\ 4.9 \\ -2.2 \end{bmatrix} \quad (6.6.11)$$

La Figura 6.30 ilustra el pulso original $x(t)$ y el pulso ecualizado. Tenga en cuenta la pequeña cantidad de la ISI residual en el pulso ecualizado. La secuencia de comandos MATLAB para este cálculo se da a continuación.

Escritura de MATLAB para Problema ilustrativo 6.11

M-FILE

```
% MATLAB script for Illustrative Problem 6.11.
echo on
[T=1;
Fs=2/T;
Ts=1/Fs;
c_opt=[-2.2 4.9 -3 4.9 -2.2];
```

```

t=-5*T:T/2:5*T;
x=1./((1+((2/T)^t)).^2); % sampled pulse
equalized_x=filter(c_opt,1,[x 0 0]); % since there will be a delay of two samples at the output
% to take care of the delay
equalized_x=equalized_x(3:length(equalized_x));
% Now, let us downsample the equalizer output.
for i=1:2:length(equalized_x),
    downsampled_equalizer_output((i+1)/2)=equalized_x(i);
    echo off;
end;
echo on ;
% Plotting commands follow.

```

Un inconveniente para el ecualizador forzado a cero es que ignora la presencia de ruido aditivo. Como consecuencia de ello, su uso puede resultar en una mejora significativa de ruido. Esto se ve fácilmente observando que en un rango de frecuencia donde $C(f)$ es pequeño, el ecualizador de canal $G_E(f) = 1 / C(f)$ compensa mediante la colocación de una gran ganancia en ese rango de frecuencias. En consecuencia, el ruido en ese rango de frecuencias es mucho mayor. Una alternativa es relajar la condición cero ISI y seleccionar el ecualizador de canal característica tal que la potencia combinada en el ISI residual y el ruido aditivo en la salida del ecualizador se minimiza. Un ecualizador de canal que está optimizado basado en el criterio de error cuadrático medio mínimo (MMSE) logra el objetivo deseado.

Elaborar, consideremos la salida de ruido corrompido de la FIR ecualizador, que es

$$z(t) = \sum_{n=-K}^{K} c_n y(t - n\tau) \quad (6.6.12)$$

Donde $y(t)$ es la entrada al ecualizador, dado por (6.4.3). La salida del ecualizador se muestrea a los tiempos $t = mT$. Por lo tanto, obtenemos

$$z(mT) = \sum_{n=-K}^{K} c_n y(mT - n\tau) \quad (6.6.13)$$

La respuesta deseada en la salida del ecualizador en $t = mT$ es el símbolo transmitido am . El error se define como la diferencia entre los am y $z(mT)$. Luego, el error mínimo cuadrático (MSE) entre la z de muestreo de salida real (mT) y los valores deseados am son cuadrados:

$$\begin{aligned}
\text{MSE} &= E |z(mT) - a_m|^2 \\
&= E \left[\left| \sum_{n=-K}^K c_n y(mT - n\tau) - a_m \right|^2 \right] \\
&= \sum_{n=-K}^K \sum_{k=-K}^K c_n c_k R_y(n-k) - 2 \sum_{k=-K}^K c_k R_{ay}(k) + E(|a_m|^2) \quad (6.6.14)
\end{aligned}$$

Donde las correlaciones se definen como

$$\begin{aligned}
R_y(n-k) &= E[y^*(mT - n\tau)y(mT - k\tau)] \\
R_{ay}(k) &= E[y(mT - k\tau)a_m^*] \quad (6.6.15)
\end{aligned}$$

Y la expectativa se toma con respecto a la secuencia de información aleatoria $\{a_m\}$ y el ruido aditivo.

La solución MSE mínimo se obtiene diferenciando (6.6.14) con respecto a los coeficientes del ecualizador $\{C_n\}$. De este modo se obtienen las condiciones necesarias para el MSE como mínimo

$$\sum_{n=-K}^K c_n R_y(n-k) = R_{ay}(k), \quad k = 0, \pm 1, \pm 2, \dots, \pm K \quad (6.6.16)$$

Estas son las ecuaciones lineales $2k + 1$ para los coeficientes del ecualizador. En contraste con la solución de ecualizador forzado a cero se ha descrito anteriormente, estas ecuaciones dependen de las propiedades estadísticas (la autocorrelación) del ruido, así como el ISI a través de la autocorrelación $R_y(n)$.

En la práctica, la matriz de autocorrelación $R_y(n)$ y el vector de correlación cruzada $R_{ay}(n)$ son desconocidos a priori. Sin embargo, estas secuencias de correlación pueden estimarse mediante la transmisión de una señal de prueba a través del canal y el uso de las estimaciones de tiempo de la media.

$$\begin{aligned}
\hat{R}_y(n) &= \frac{1}{N} \sum_{k=1}^N y^*(kT - n\tau)y(kT) \\
\hat{R}_{ay}(n) &= \frac{1}{N} \sum_{k=1}^N y(kT - n\tau)a_k^* \quad (6.6.17)
\end{aligned}$$

En lugar de los promedios Conjuntos para solucionar los coeficientes del ecualizador dadas por (6.6.16).

Problema ilustrativo 6.12 Consideremos el mismo pulso del canal distorsionado $x(t)$ como el Problema Ilustrativo 6.11, pero ahora vamos a diseñar el ecualizador de cinco pulsos basado en el criterio mínimo MSE. Los símbolos tienen media en cero y varianza en la unidad y no están correlacionados, es decir,

$$\begin{aligned} E(a_n) &= 0 \\ E(a_n a_m) &= 0, \quad n \neq m \\ E(|a_n|^2) &= 1 \end{aligned}$$

El ruido aditivo $v(t)$ tiene media cero y autocorrelación

$$R_{vv}(\tau) = \frac{N_0}{2} \delta(\tau)$$

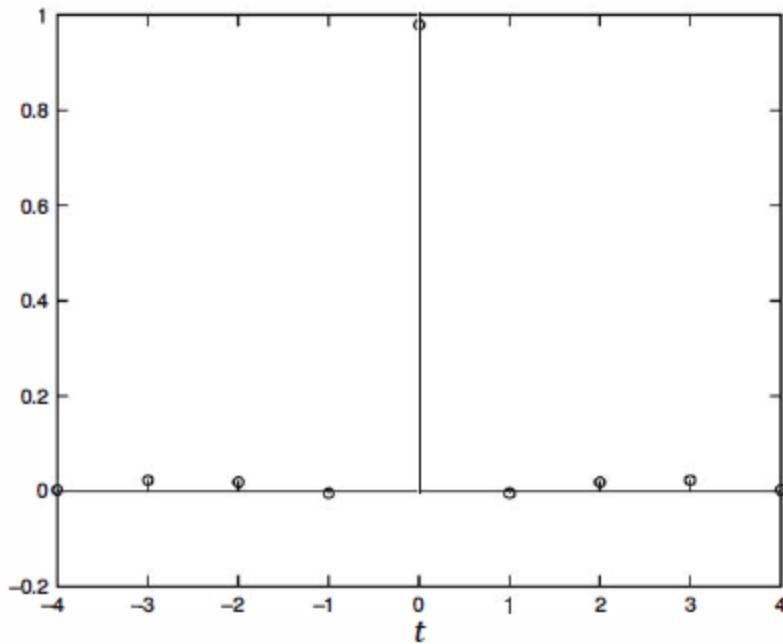


Figura 6.31: gráfica del pulso ecualizado en Problema Ilustrativo 6.12

Solución: Los coeficientes de derivación del ecualizador se obtienen resolviendo (6.6.16) con $K = 2$ y con $\tau = T/2$. La matriz con elementos $R_y(n - k)$ es simplemente

$$R_y = X^t X + \frac{N_0}{2} I$$

Donde X está dada por (6.6.9) e I es la matriz identidad. El vector con elementos $R_y(k)$ se da como

$$R_{ay} = \begin{bmatrix} \frac{1}{5} \\ \frac{1}{2} \\ 1 \\ \frac{1}{2} \\ \frac{1}{5} \end{bmatrix}$$

Los coeficientes de ecualización obtenidos resolviendo (6.6.16) son los siguientes:

$$c_{\text{opt}} = \begin{bmatrix} 0.0956 \\ -0.7347 \\ 1.6761 \\ -0.7347 \\ 0.0956 \end{bmatrix}$$

Se muestra la trama del pulso ecualizado en la Figura 6.31. La secuencia de comandos de MATLAB para este cálculo es la siguiente.

M-FILE

```
% MATLAB script for Illustrative Problem 6.12.
echo on
T=1;
for n=-2:2,
    for k=-2:2,
        temp=0;
        for i=-2:2, temp=temp+(1/(1+(n-i)^2))*(1/(1+(k-i)^2)); end;
        X(k+3,n+3)=temp;
        echo off ;
    end;
end;
echo on;
N0=0.01;                                % assuming that N0=0.01
Ry=X+(N0/2)*eye(5);
Riy=[1/5 1/2 1 1/2 1/5].';
c_opt=inv(Ry)*Riy;                      % optimal tap coefficients
% find the equalized pulse...
t=-3:1/2:3;
x=1./(1+(2*t/T).^2);                   % sampled pulse
equalized_pulse=conv(x,c_opt);
% Decimate the pulse to get the samples at the symbol rate.
decimated_equalized_pulse=equalized_pulse(1:2:length(equalized_pulse));
% Plotting command follows.
```

6.6.1 Ecualizadores lineales adaptativos

Hemos demostrado que los coeficientes de pulsos de un ecualizador lineal se pueden determinar mediante la resolución de un conjunto de ecuaciones lineales. En el criterio de optimización forzado a cero, las ecuaciones lineales están dadas por (6.6.8). Por otro lado, si el criterio de optimización se basa en reducir al mínimo el *MSE*, los coeficientes del ecualizador óptimo se determinan por la solución del conjunto de ecuaciones lineales dado por (6.6.16).

En ambos casos, podemos expresar el conjunto de ecuaciones lineales en la forma general de la matriz

$$\mathbf{B}\mathbf{c} = \mathbf{d} \quad (6.6.18)$$

donde \mathbf{B} es una matriz $(2K + 1) \times (2K + 1)$, \mathbf{c} es un vector columna que representa los $2K + 1$ coeficientes del ecualizador, y \mathbf{d} es un vector columna dimensional $(2K + 1)$. La solución de (6.6.18) produce

$$\mathbf{c}_{\text{opt}} = \mathbf{B}^{-1} \mathbf{d} \quad (6.6.19)$$

En implementaciones prácticas de ecualizadores, la solución de (6.6.18) para el vector de coeficientes óptimo se obtiene normalmente mediante un procedimiento iterativo que evita el cálculo explícito de la inversa de la matriz \mathbf{B} . El procedimiento iterativo más simple es el método de descenso más agudo, en el que uno empieza por elegir arbitrariamente el vector de coeficientes \mathbf{c} , por ejemplo, \mathbf{c}_0 . Esta elección inicial del coeficiente \mathbf{c}_0 del vector corresponde a un punto en la función de criterio que se está optimizando. Por ejemplo, en el caso del criterio *MSE*, la suposición \mathbf{c}_0 inicial corresponde a un punto en la superficie *MSE* cuadrática de los coeficientes del espacio dimensional $(2K + 1)$. El vector gradiente, definido como \mathbf{g}_0 , que es la derivada de la *MSE* con respecto a los coeficientes de filtro $2K + 1$, se calcula a continuación, en este punto de la superficie del criterio, y cada coeficiente de pulso se cambia en la dirección opuesta a su componente de gradiente correspondiente. El cambio en el coeficiente de pulso j -ésimo es proporcional al tamaño del componente de gradiente j -ésimo.

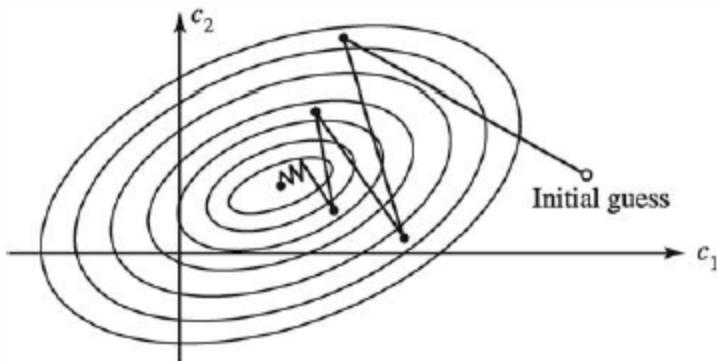


Figura 6.32: Ejemplo de las características de convergencia de un algoritmo de gradiente

Por ejemplo, el vector gradiente, denotado como \mathbf{g}_k , para el criterio *MSE*, que se encuentra mediante la adopción de los derivados de la *MSE* con respecto a cada uno de los $2K + 1$ coeficientes, es

$$\mathbf{g}_k = \mathbf{B}\mathbf{c}_k - \mathbf{d}, \quad k = 0, 1, 2, \dots \quad (6.6.20)$$

A continuación, el coeficiente de vector \mathbf{c}_k se actualiza de acuerdo con la relación

$$\mathbf{c}_{k+1} = \mathbf{c}_k - \Delta \mathbf{g}_k \quad (6.6.21)$$

donde Δ es el parámetro de tamaño de paso para el procedimiento iterativo. Para asegurar la convergencia del proceso iterativo, Δ es elegido para ser un número positivo pequeño. En este caso, el vector de gradiente \mathbf{g}_k converge hacia cero, es decir, $\mathbf{g}_k \rightarrow 0$ cuando $k \rightarrow \infty$ y el coeficiente de vector $\mathbf{c}_k \rightarrow \mathbf{c}_{\text{opt}}$, como se ilustra en la Figura 6.32 sobre la base de la optimización de dos dimensiones. En general, la convergencia de los

coeficientes del ecualizador de pulsos a \mathbf{c}_{opt} no puede alcanzarse en un número finito de iteraciones con el método de descenso más agudo. Sin embargo, la óptima solución de \mathbf{c}_{opt} se puede abordar tan de cerca como se desee en unos pocos cientos de iteraciones. En sistemas de comunicación digitales que emplean ecualizadores de canal, cada iteración corresponde a un intervalo de tiempo para el envío de un símbolo; por lo tanto, unos pocos cientos de iteraciones para lograr la convergencia a \mathbf{c}_{opt} corresponde a una fracción de un segundo.

La ecualización de canal adaptativo se requiere para los canales cuyas características cambian con el tiempo. En tal caso, el ISI varía con el tiempo. El ecualizador de canal debe realizar un seguimiento de este tipo de variaciones de tiempo en la respuesta del canal y adaptar sus coeficientes para reducir el ISI. En el contexto de la discusión anterior, el vector de coeficientes óptimo \mathbf{c}_{opt} varía con el tiempo debido a las variaciones de tiempo en la matriz \mathbf{B} y, para el caso de el criterio *MSE*, las variaciones en el tiempo en el vector \mathbf{d} . En estas condiciones, el método iterativo descrito anteriormente puede ser modificado para utilizar estimaciones de los componentes de gradiente. Por lo tanto, el algoritmo de ajuste de los coeficientes del ecualizador puede expresarse como

$$\hat{\mathbf{c}}_{k+1} = \hat{\mathbf{c}}_k - \Delta \hat{\mathbf{g}}_k \quad (6.6.22)$$

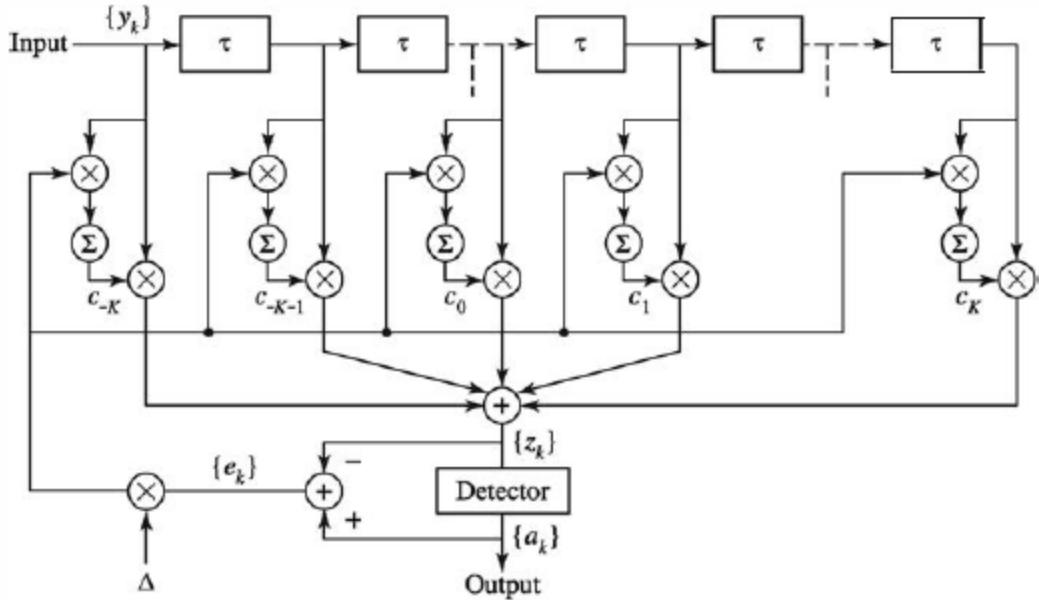


Figura 6.33: Ecualizador Lineal adaptativo basado en el criterio MSE

donde $\overline{\mathbf{g}_K}$ denota una estimación del vector gradiente \mathbf{g}_K y $\overline{\mathbf{c}_k}$ indica la estimación del vector de coeficientes del pulso.

En el caso del criterio *MSE*, el vector gradiente de \mathbf{g}_K dada por (6.6.20) también puede expresarse como

$$\mathbf{g}_k = -E(e_k \mathbf{y}_k^*)$$

Una estimación $\overline{g_K}$ del vector gradiente en la iteración k-ésima se calcula como

$$\hat{\mathbf{g}}_k = -e_k \mathbf{y}_k^* \quad (6.6.23)$$

donde e_k indica la diferencia entre la salida deseada del ecualizador en el instante de tiempo k-ésimo de la salida real $z(kT)$ y \mathbf{y}_k denota el vector columna de $2K + 1$ valores de señales recibidas contenidas en el ecualizador en el instante de tiempo k . La señal de error e_k se expresa como

$$e_k = \mathbf{a}_k - z_k \quad (6.6.24)$$

donde $z_k = z(kT)$ es la salida del ecualizador dada por (6.6.13) y \mathbf{a}_k es el símbolo deseado. Por lo tanto, mediante la sustitución de (6.6.23) en (6.6.22), se obtiene el algoritmo adaptativo para la optimización de los coeficientes de pulsos (basado en el criterio *MSE*) como

$$\hat{\mathbf{c}}_{k+1} = \hat{\mathbf{c}}_k + \Delta e_k \mathbf{y}_k^* \quad (6.6.25)$$

Debido a una estimación del vector gradiente se utiliza en (6.6.25), el algoritmo se llama *algoritmo de gradiente estocástico*. También se conoce como el algoritmo de cuadrados mínimos de media o Least Mean Square (LMS).

Un diagrama de bloques de un ecualizador adaptativo que se adapta a sus coeficientes de pulsos de acuerdo con (6.6.25) se ilustra en la Figura 6.33. Tenga en cuenta que la diferencia entre la salida \mathbf{a}_k deseada y la salida real z_k del ecualizador se utiliza para formar la señal de error e_k .

Este error se escala por el parámetro de tamaño de paso Δ y la señal de error escalado Δe_k multiplica los valores de las señales recibidas $\{y(KT - n\tau)\}$ con los $2k + 1$ pulsos. Los productos $\Delta e_k \mathbf{y}^*(KT - n\tau)$ en los $2k + 1$ pulsos se añaden a continuación a los valores anteriores de los coeficientes de derivación para obtener los coeficientes de los pulsos actualizados, de acuerdo con (6.6.25). Este cálculo se repite después de recibir cada nuevo símbolo. Por lo tanto, los coeficientes del ecualizador se actualizan al tipo de símbolo.

Inicialmente, el ecualizador adaptativo es entrenado por la transmisión de una secuencia pseudoaleatoria conocida $\{am\}$ sobre el canal. En el demodulador, el ecualizador emplea la secuencia conocida para ajustar sus coeficientes. Tras el ajuste inicial, el ecualizador adaptativo cambia de un modo de entrenamiento a un modo de decisión dirigida, en cuyo caso las decisiones en la salida del detector son suficientemente fiables para que la señal de error se forme mediante el cálculo de la diferencia entre la salida del detector y la salida que ecualizador que es,

$$e_k = \hat{a}_k - z_k \quad (6.6.26)$$

Donde \overline{a}_K es la salida del detector. En general, los errores de decisión en la salida del detector ocurren con poca frecuencia; en consecuencia, estos errores tienen poco efecto en el rendimiento del algoritmo de seguimiento dado por (6.6.25).

Una regla de oro para seleccionar el parámetro de tamaño de paso con el fin de asegurar la convergencia y buenas capacidades de seguimiento en los canales de variación lenta es

$$\Delta = \frac{1}{5(2K + 1)P_R} \quad (6.6.27)$$

donde P_R indica la potencia de la señal más ruido recibida, que se puede estimar a partir de la señal recibida.

Problema ilustrativo 6.13 Vamos a implementar un ecualizador adaptativo basado en el algoritmo LMS dada en (6.6.25). El número del canal de las líneas seleccionadas para el ecualizador es $2K + 1 = 11$. La señal más el ruido de la fuente P_R recibida es normalizada a la unidad. La característica del canal está dada por el vector x como

$$x = [0.05 \ -0.063 \ 0.088 \ -0.126 \ -0.25 \ 0.9047 \ 0.25 \ 0 \ 0.126 \ 0.038 \ 0.088]$$

Solución: Las características de convergencia del algoritmo de gradiente estocástico en (6.6.25) se ilustran en la Figura 6.34. Estos gráficos han sido obtenidas a partir de una simulación del ecualizador adaptativo de 11 pulsos, hecha en un ordenador. Los gráficos representan la MSE promedio durante varias realizaciones. Como se muestra, cuando Δ disminuye, la convergencia se retarda un poco, pero se consigue un MSE inferior, lo que indica que los coeficientes estimados están más cerca de c_{opt} .

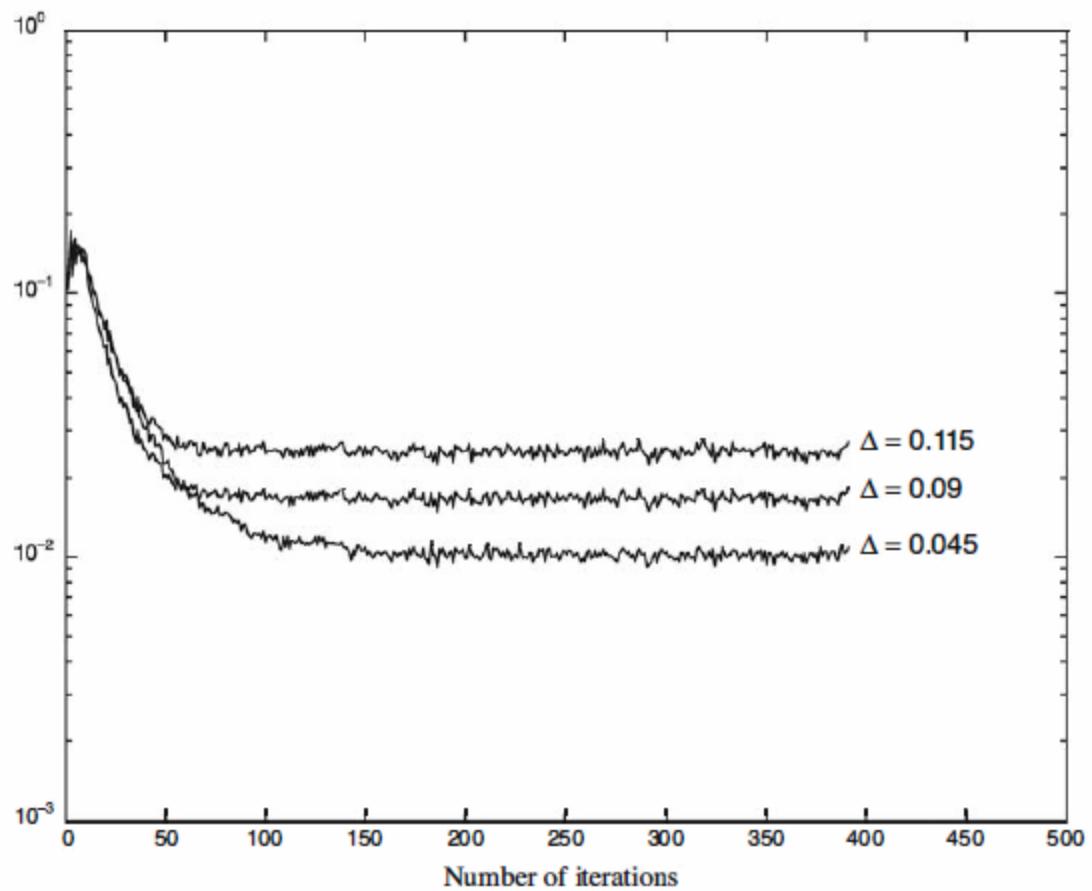


Figura 6.34: características de convergencia iniciales del algoritmo LMS con diferentes tamaños de paso

La secuencia de comandos de MATLAB para este ejemplo es la siguiente.

M-FILE

```
% MATLAB script for Illustrative Problem 6.13.
echo on
N=500; % length of the information sequence
K=5;
actual_isi=[0.05 -0.063 0.088 -0.126 -0.25 0.9047 0.25 0 0.126 0.038 0.088];
sigma=0.01;
delta=0.115;
Num_of_realizations=1000;
mse_av=zeros(1,N-2*K);
for j=1:Num_of_realizations, % Compute the average over a number of realizations.
    % the information sequence
    for i=1:N,
        if (rand<0.5),
            info(i)=-1;
        else
            info(i)=1;
        end;
        echo off ;
    end;
    if (j==1) ; echo on ; end
    % the channel output
    y=filter(actual_isi,1,info);
    for i=1:2:N, [noise(i) noise(i+1)]=gngauss(sigma); end;
    y=y+noise;
    % Now the equalization part follows.
    estimated_c=[0 0 0 0 0 1 0 0 0 0 0]; % initial estimate of ISI
    for k=1:N-2*K,
        y_k=y(k:k+2*K);
        z_k=estimated_c*y_k.' ;
        e_k=info(k)-z_k;
        estimated_c=estimated_c+delta*e_k*y_k;
        mse(k)=e_k^2;
        echo off ;
    end;
    if (j==1) ; echo on ; end
    mse_av=mse_av+mse;
    echo off ;
end;
echo on ;
mse_av=mse_av/Num_of_realizations; % mean-squared error versus iterations
% Plotting commands follow.
```

Problema ilustrativo 6.14 [Rendimiento del Ecualizador adaptado MSE] Un sistema de comunicación transmite símbolos binarios $\{a_n = \pm 1\}$ a través de un canal de ISI cuya

característica viene dada por el vector x del Problema ilustrativo 6.13. La salida del canal es la secuencia

$$y_n = \sum_{k=0}^{10} x_k a_{n-k} + w_n, \quad n = 1, 2, \dots$$

donde $a_n = 0$, $n < 0$, y $\{W_n\}$ es una secuencia de ruido blanco gaussiano de varianza σ^2 . El ecualizador es el ecualizador adaptativo lineal basada en el criterio MSE como se muestra en la Figura 6.33. Su salida es la secuencia

$$z_n = \sum_{k=-K}^{K} c_k y_{n-k}, \quad n = 1, 2, \dots$$

Donde $\{c_k\}$ son los coeficientes del ecualizador.

Realizar una simulación de Monte Carlo para determinar la probabilidad de error para el ecualizador como una función de la $SNR = 1/2\sigma^2$. Use los primeros 200 símbolos transmitidos para formar el ecualizador con el uso del algoritmo LMS y un tamaño de paso $\Delta = 0,045$. A continuación, lleve a cabo la simulación de 10.000 símbolos de transmisión para cada elección de varianza σ^2 .

Solución: El rendimiento del ecualizador se muestra en la Figura 6.35.

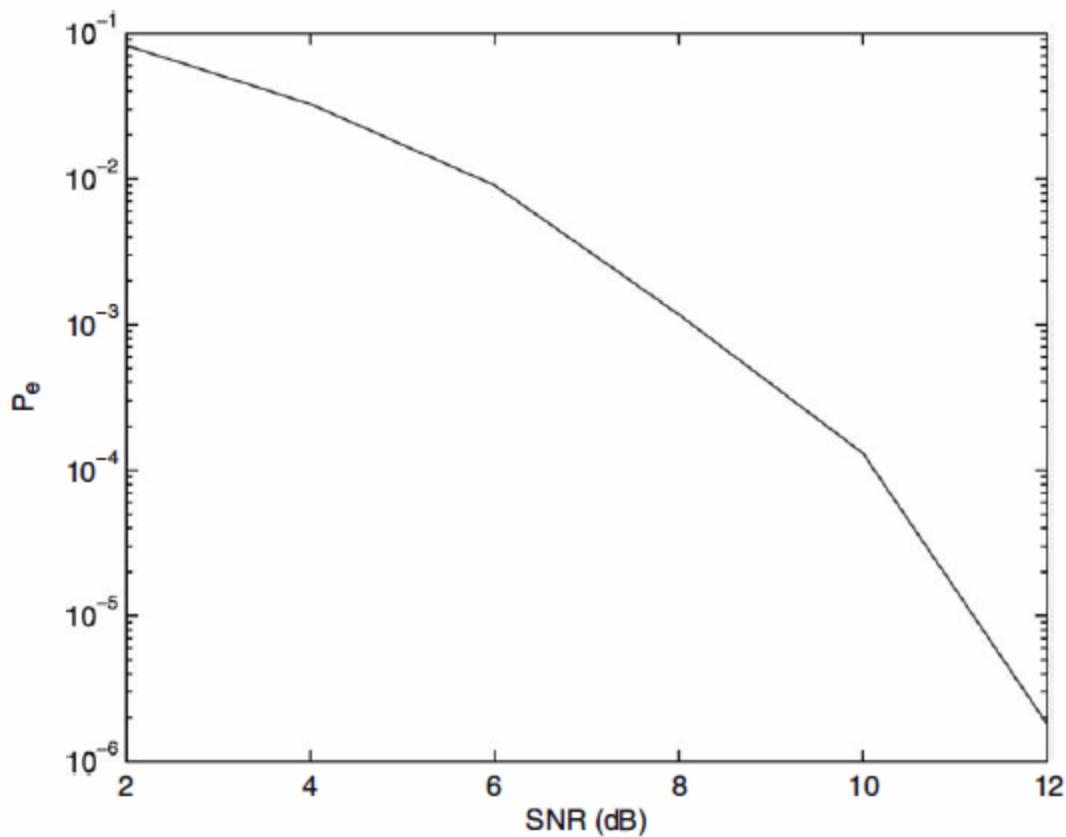


Figura 6.35: El rendimiento del ecualizador del Problema Ilustrativo 6.14

La secuencia de comandos de MATLAB para el problema es la siguiente.

M-FILE

```
% MATLAB script for Illustrative Problem 6.14

N=20000; % Length of the training sequence
% Number of transmitted symbols for each choice of SNR
Nt=[1000000 1000000 1000000 1000000 1000000 5000000 10000000];
delta=0.0045;
K=5;
actual_isi=[0.05 -0.063 0.088 -0.126 -0.25 0.9047 0.25 0 0.126 0.038 0.088];
SNR_dB=2:2:14;
SNR=10.^{SNR_dB/10};
LSNR=size(SNR,2);
var=1/2./SNR;
sigma=sqrt(var);
Pe=zeros(1,LSNR);
for idx=1:LSNR
    % (A) The training sequence:
    training_s=ones(1,N);
    for i=1:N
        if (rand<0.5)
            training_s(i)=-1;
        end
    end
    % (B) The received signal:
    % (C) The estimated ISI sequence:
    % (D) The estimated symbol sequence:
end
```

```

% The channel output:
y=filter(actual_isi,1,training_s);
noise=zeros(1,N);
for i=1:2:N
    noise(i) =random('Normal',0,sigma(idx));
    noise(i+1) = noise(i);
end
y=y+noise;
% The equalization part follows:
estimated_c=[0 0 0 0 1 0 0 0 0 0]; % initial estimate of ISI
for k=1:N-2*K
    y_k=y(k:k+2*K);
    z_k=estimated_c*y_k';
    e_k=training_s(k)-z_k;
    estimated_c=estimated_c+delta*e_k*y_k;
end
% (B) The transmitted information sequence:
info=ones(1,Nt(idx));
for i=1:Nt(idx)
    if (rand<0.5)
        info(i)=-1;
    end
end
% The channel output:
y=filter(actual_isi,1,info);
noise=sigma(idx)*randn(1,Nt(idx));
y=y+noise;
% The equalization part:
count = 0;
err_count=0;
z_k_vec=ones(1,Nt(idx)-2*K);
for k=1:Nt(idx)-2*K;
    y_k=y(k:k+2*K);
    z_k=estimated_c*y_k';
    if z_k<0
        z_k_vec(k)=-1;
    end
    err_count=err_count+0.5*abs(info(k)-z_k_vec(k));
end
Pe(idx)= err_count/length(z_k_vec);
clear y; clear noise
end
% Plot the results:
semilogy(SNR_dB,Pe)
grid
ylabel('P_e')
xlabel('SNR (dB)')

```

Aunque hemos descrito con cierto detalle el funcionamiento de un ecualizador adaptado que está optimizado sobre la base del criterio MSE, el funcionamiento de un ecualizador adaptado basado en el método de forzado a cero es muy similar. La principal diferencia radica en el método para estimar los vectores de gradiente g_k en cada iteración. Un diagrama de bloques de un ecualizador adaptado forzado a cero se muestra en la Figura 6.36.

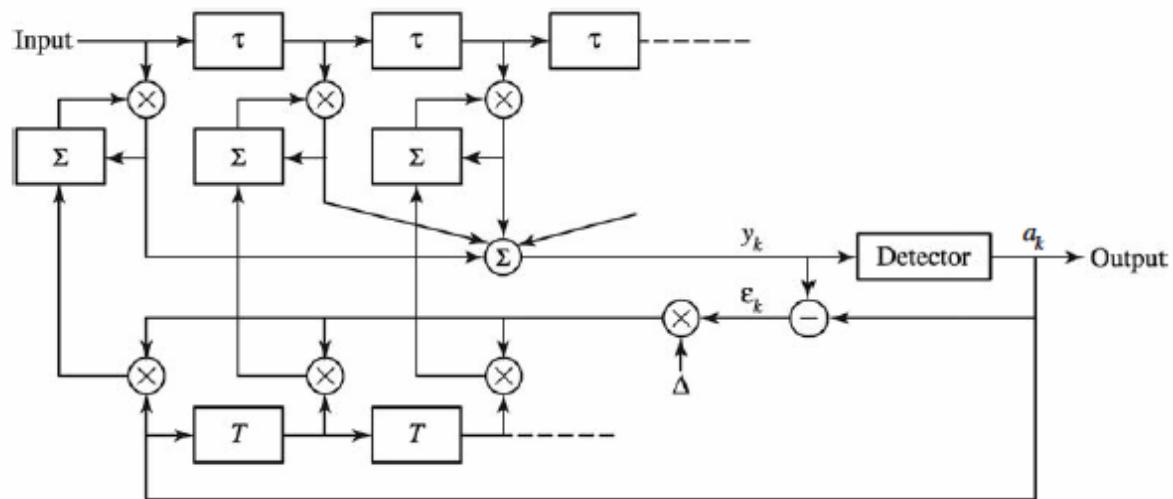


Figura 6.36: Un ecualizador adaptado forzado a cero

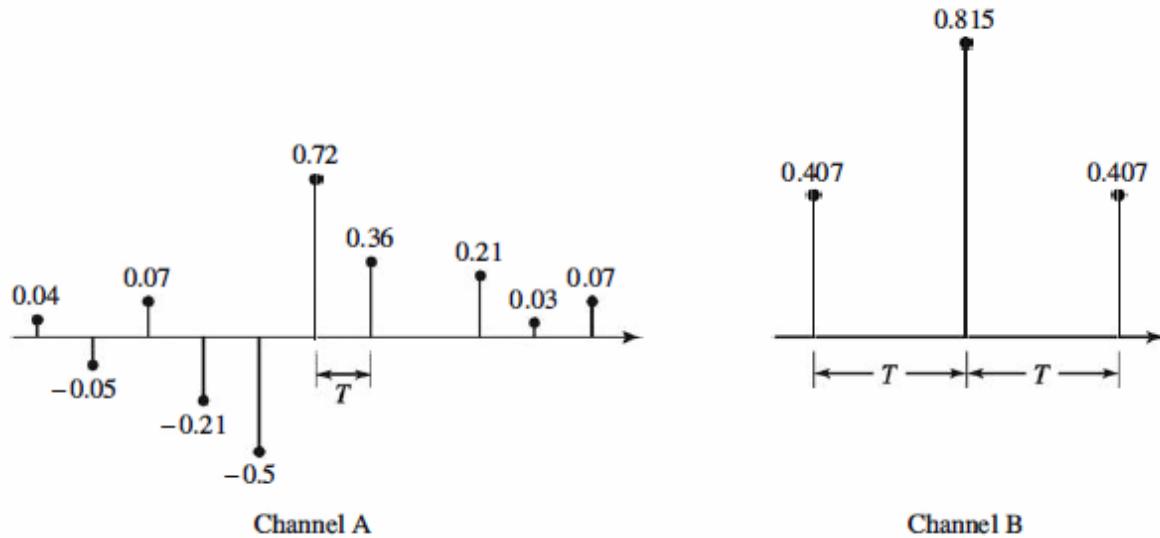


Figura 6.37: Dos canales con ISI