Capítulo 2. Procesos Aleatorios

1. Objetivos

- Conocer el concepto de variable y vector aleatorio, así como su descripción estadística
- Comprender el concepto de proceso y secuencia aleatoria y conocer los tipos principales en el campo de las telecomunicaciones
- Conseguir el estudio el comportamiento de procesos y secuencias aleatorias que se transmiten a través de sistemas lineales invariantes en el tiempo.

2. Variables Aleatorias.Generación

2.1. Introducción

Es habitual en la práctica el uso de generadores de números aleatorios para simular el efecto de señales similares a ruido y otros fenómenos aleatorios que aparecen en el mundo físico. Dicho ruido está presente en los dispositivos electrónicos y en los sistemas y normalmente limita nuestra capacidad para comunicarnos a lo largo de grandes distancias o de detectar señales relativamente débiles. Generando este ruido en un ordenador, podemos estudiar sus efectos durante la simulación de sistemas de comunicación y comprobar las prestaciones de dichos sistemas en presencia de un ruido.

MATLAB dispone de un generador de números aleatorios que permite obtener variables aleatorias con funciones de densidad de probabilidad de muchos tipos,incluidas las variables aleatorias uniformes y Gaussianas, de gran importancia.
Sabemos que los valores numéricos generados por un ordenador tienen una precisión limitada y, por lo tanto, es imposible representar la totalidad de los valores continuos que hay en un intervalo $a \le X \le b$. Sin embargo, podemos suponer que el ordenador representa cada valor con un gran número de bits tanto en punto fijo como en punto flotante. Así, con fines prácticos, el número de valores posibles generados en el intervalo $a \le X \le b$ es lo suficientemente grande para justificar la suposición de que es posible obtener cualquier valor en el intervalo deseado.

2.2. Generación de Variables Aleatorias

2.2.1. Introducción teórica

La función de densidad de probabilidad que caracteriza a una variable aleatoria X se denomina $f(X)$. Para una variable aleatoria uniforme U en el intervalo $[0,1]$ esta función se ilustra en la Figura 2.1(a). La integral de la función de densidad de probabilidad, que representa el área bajo $f(X)$, se denomina función de distribución de probabilidad de la variable X y se define como:

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

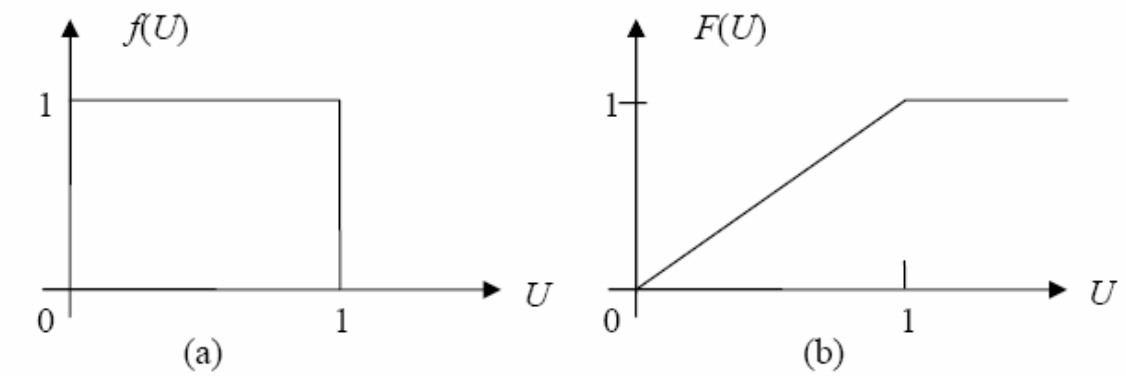


Figura 2.1. Función de densidad de probabilidad f(U) y función de distribución de probabilidad F(U) de una variable aleatoria uniforme U

2.2.2. Comandos en Matlab

Las funciones `rand(N)` y `rand(M,N)` devuelven matrices de orden `NxN` y `MxN` respectivamente de los valores aleatorios independientes uniformemente distribuidos entre `0` y `1`. Si deseamos obtener un conjunto de variables aleatorias uniformemente distribuidas en el intervalo $[a,b]$ basta con hacer:

```
>> x = a + (b - a)*rand(M,N);
```

Las funciones `randn(N)` y `randn(M,N)` devuelven matrices de orden `NxN` y `MxN` respectivamente de valores aleatorios independientes con una distribución normal de media `0` y varianza `1`. Si deseamos obtener un conjunto de variables aleatorias normales con media `m` y desviación típica `sigma` basta con hacer:

```
>> n = m + sigma*randn(M,N);
```

Una vez generado un vector de valores aleatorios es posible estimar su media y su varianza, entre otros muchos estadísticos, mediante las funciones `mean` y `var` respectivamente.

```
>> mean (x)
>> var (x)
```

La orden `R=RANDOM(NAME,A,B,C,M,N,...)` o `R=RANDOM(NAME,A,B,C,[M,N,...])` devuelve una matriz `NxM` de valores aleatorios distribuidos de la manera que indique el parámetro `NOMBRE` (`>>help random`) que utiiza los parámetros `A`, `B` y `C`.

La orden de MATLAB `hist(x,N)` proporciona el histograma de los valores contenidos en el vector `x`, representando el número de elementos que hay en cada uno de los N contenedores igualmente espaciados que genera. El histograma puede considerarse una aproximación de la función de densidad de probabilidad de las muestras del proceso aleatorio, aunque con los valores del eje vertical escalados.

Ejercicio 2.1.

Enunciado

Realizar un fichero en MATLAB 'rand_amp_cos.m' que dibuje cuatro experimentos de un coseno cuya amplitud sea una variable aleatoria discreta uniformemente distribuida entre $[1,4]$ *y de frecuencia 4 Hz.*

Resultados

```
%Ejercicio 3.1 rand_amp_cos.m dibuja cuatro senoides cuya
%amplitud es una variable aleatoria uniforme entre [1,4]

NUM_EXPERIMENTOS=4;
FREC_REPRESENTACION=100;
Tfinal=1;
Tinicial=0;
a=1;
b=4;
t=Tinicial:(1/FREC_REPRESENTACION):(Tfinal-
1/FREC_REPRESENTACION);
experimentos=zeros(NUM_EXPERIMENTOS,FREC_REPRESENTACION);
A=zeros(NUM_EXPERIMENTOS,FREC_REPRESENTACION);
figure(1);
clf;
for n=1:NUM_EXPERIMENTOS
    A(n,:)= a + (b - a)*rand(FREC_REPRESENTACION,1);
    experimentos(n,:)= A(n,1)*cos(2*pi*4*t);

    %----otra forma -----
    %experimentos(n,:)=random('unid',b,a,1)*cos(2*pi*4*t);
    %--------------------

    subplot(NUM_EXPERIMENTOS,1,n);
    plot(t,experimentos(n,:));
end
    subplot(NUM_EXPERIMENTOS,1,1);
title('Experimentos de una senoide de Amplitud Aleatoria')
```

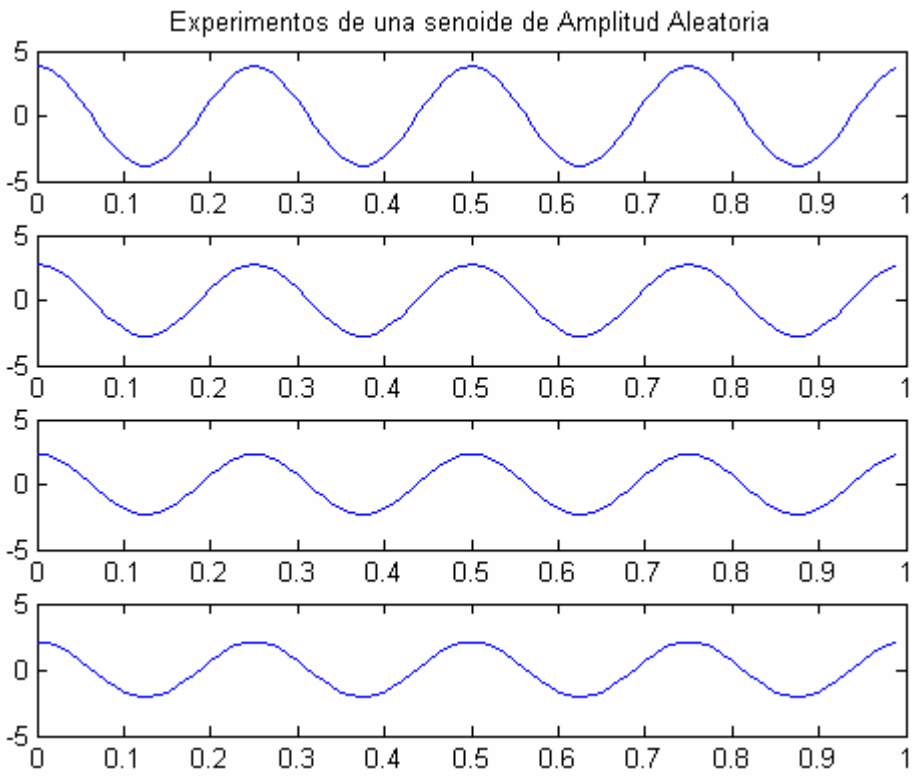


Figura 2.2. Experimentos de una senoide de Amplitud Aleatoria

3. Procesos Aleatorios

3.1. Definición

Para estudiar señales aleatorias, tenemos que observar una colección de señales, más que el caso de una sola. Esta colección de señales se llama proceso aleatorio. A cada señal de esta colección se le llama realización o función muestra del proceso.

3.2. Procesos Gaussianos

Los procesos Gaussianos juegan un papel muy importante en los sistemas de comunicación. La razón principal la importancia de los procesos Gaussianos es el ruido térmico en los dispositivos electrónicos, que es producido por el movimiento aleatorio de los electrones debido a la agitación térmica, este movimiento aleatorio puede ser modelado por un proceso Gaussiano.

La razón por la que el ruido térmico se comporte como un proceso gaussiano se debe a que la corriente producida por el movimiento de los electrones en un circuito eléctrico se puede ver como la suma de pequeñas corrientes de un gran número de fuentes (electrones individuales). Se puede asumir que la mayoría de estas fuentes se comportan independientemente, con lo cual la corriente total es la suma de un gran numero de variables aleatorias independientes e idénticamente distribuidas. Si aplicamos el teorema central del límite, la corriente total tiene una distribución Gaussiana.

Una de las razones del estudio de procesos aleatorios gaussianos, es el modelado del canal de comunicación. Generalmente, la clase de ruido que puede ser encontrado en un canal de comunicaciones es el ruido Gaussiano, las fuentes de ruido gaussiano son muchas e inevitables. En ellas se incluyen, entre otros, el ruido térmico introducido en el medio de transmisión. De acuerdo con el teorema central del límite, cualquier ruido que resulta de los efectos combinados de muchas fuentes independientes tiende a ser distribuido de forma gaussiana. Como consecuencia de ello, y debido a la relativa facilidad con la que se puede tratar matemáticamente, el ruido Gaussiano ha sido empleado casi exculusivamente en los modelos de modulación de canal en la moderna teoría de las comunicaciones. Así pues en todo el proyecto supondremos un canal AWGN (se verá más adelante).

Aparte del ruido térmico los procesos Gaussianos ofrecen también buenos modelos de fuentes de información. Los procesos Gaussianos tienen algunas propiedades interesantes, las cuales se muestran a continuación, que hacen que estos procesos sean manejables matemáticamente y fáciles de tratar.

Un proceso aleatorio $X(t)$ es un proceso si para todo n y para todos los instantes $(t1, t2, ..., tn)$ las variables aleatorias $\{X(t_1)\}_{i=1}^{n}$ tienen una función de densidad gaussiana que se puede expresar en función de la media μ y varianza σ^2:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[\frac{-x^2}{2\sigma^2}\right] \qquad (2.3.1)$$

Las principal característica de un proceso aleatorio gaussiano es la siguiente:

Si el proceso Gaussiano $X(t)$ atraviesa un sistema lineal e invariante en el tiempo (LTI); el proceso aleatorio obtenido a la salida $Y(t)$ es también un proceso Gaussiano.

Ejercicio 2.2.

Enunciado

Realizar una función en MATLAB que se guarde en el archivo 'exp_gauss.m' de su directorio de trabajo que dibuje cuatro experimentos (realizaciones) de un proceso Gaussiano en función de 2 parámetros de entrada: media (m) y desviación típica (σ).

```
% Uso:
%  exp_gauss(media,varianza)
% donde :
%          media          =    media del Proceso Aleatorio Gaussiano (probar con
0)
%          desv_tipica    =   desviación típica del PA (probar con 1)
```

```
function exp_gauss(media,desv_tipica);
NUM_EXPERIMENTOS=4;
FREC_REPRESENTACION=1024;
Tfinal=1;
Tinicial=0;
t=Tinicial:(1/FREC_REPRESENTACION):(Tfinal-
1/FREC_REPRESENTACION);
experimentos=zeros(NUM_EXPERIMENTOS,FREC_REPRESENTACION);
figure(1);
clf;
for n=1:NUM_EXPERIMENTOS
    experimentos(n,:)= media +
(desv_tipica)*randn(1,FREC_REPRESENTACION);
    subplot(NUM_EXPERIMENTOS,1,n);
    plot(t,experimentos(n,:));
end
    subplot(NUM_EXPERIMENTOS,1,1);
title('Experimentos de un Proceso Aleatorio Gaussiano');
```

Resultados

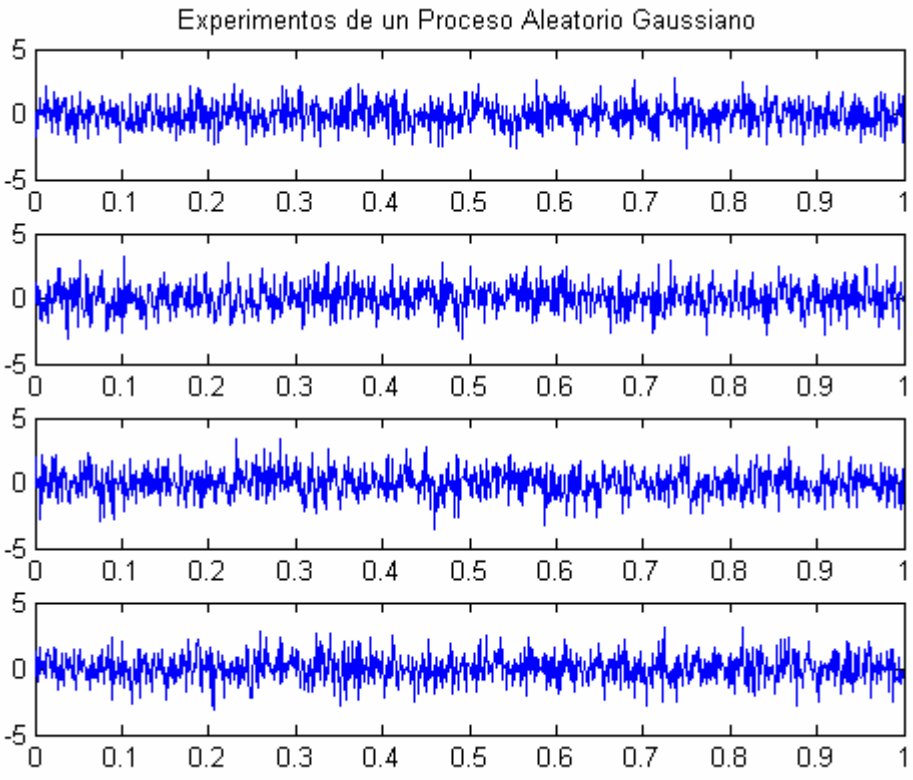


Figura 2.3. Experimentos de un Proceso Aleatorio Gaussiano

Ejercicio 2.3.

Enunciado

Realizar un fichero en MATLAB 'cos_y_ruido.m' que dibuje cuatro experimentos de un coseno aleatorio con ruido. La amplitud y frecuencia del coseno será una variable aleatoria discreta uniformemente distribuida entre [1,4] y la fase una variable aleatoria discreta uniformemente distribuida entre [-π,π]. El ruido será una variable aleatoria gaussiana de media 0, varianza 1 y amplitud 0.1.

Resultados

```
NUM_EXPERIMENTOS=4;
FREC_REPRESENTACION=1000;
Tfinal=1;
Tinicial=0;
a=1;
b=4;
t=Tinicial:(1/FREC_REPRESENTACION):(Tfinal-
1/FREC_REPRESENTACION);
experimentos=zeros(NUM_EXPERIMENTOS,FREC_REPRESENTACION);
amp_frec=zeros(NUM_EXPERIMENTOS,FREC_REPRESENTACION);
fase=zeros(NUM_EXPERIMENTOS,FREC_REPRESENTACION);
figure(1);
clf;
for n=1:NUM_EXPERIMENTOS
    amp_frec(n,:)= a + (b - a)*rand(FREC_REPRESENTACION,1);
    fase(n,:)=-pi + (pi-(-pi))*rand(FREC_REPRESENTACION,1);

experimentos(n,:)=amp_frec(n,1)*cos(2*pi*amp_frec(n,1)*t+fase(
n,1))+0.1*randn(1,FREC_REPRESENTACION);
    subplot(NUM_EXPERIMENTOS,1,n);
    plot(t,experimentos(n,:));
end
    subplot(NUM_EXPERIMENTOS,1,1);;
```

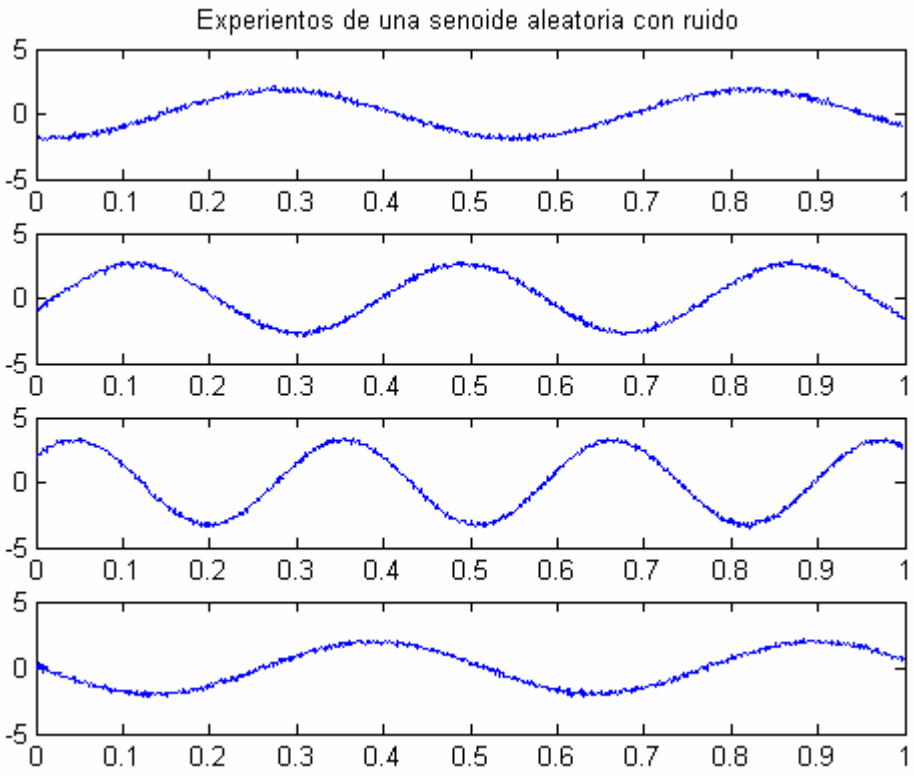


Figura 2.4. Experimentos de una aleatoria senoide con ruido

3.3. Autocorrelación de los Procesos Aleatorios.

La autocorrelación de un proceso aleatorio es el valor esperado del producto de la variable aleatoria (o realización) con una versión desplazada en el tiempo de si misma.

$$R_{xx}(t_1,t_2) = E\left[x_1 x_2\right] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1 x_2 f\left(x_1,x_2\right) dx_2 dx_1 \qquad (2.3.2)$$

La ecuación anterior es válida para los procesos aleatorios estacionarios y no estacionarios. Para los procesos estacionarios podemos simplificar más la ecuación. En los procesos estacionarios la función autocorrelación depende únicamente de la diferencia de tiempo.
De esta forma para procesos aleatorios continuos:

$$R_{xx}(t,t+\tau) = R_{xx}(\tau) = E\left[X(t)X(t+\tau)\right] \quad (2.3.3)$$

Para procesos discretos:

$$R_{xx}\left[n,n+m\right] = R_{xx}\left[m\right] = \sum_{n=-\infty}^{\infty} x\left[n\right]x\left[n+m\right] \qquad (2.3.4)$$

3.3.1. Aproximación de la Función Autocorrelación y del Espectro de Densidad de Potencia para el caso discreto

Supongamos que el proceso discreto en el tiempo *X(n)* procede del muestreo de un proceso continuo en el tiempo *X*(*t*). Si el período de muestreo es *Ts* y llamamos $R_x^c(\tau)$ a la función de autocorrelación y $S_x^C(f)$ a la PSD del proceso continuo entonces se cumple:

$$R_x(m) = R_x^C(mT_S) \qquad (2.3.5)$$

$$S_x(f) = \frac{1}{T}\sum_{l=-\infty}^{\infty} S_x^C\left(\frac{f+l}{T}\right) \qquad (2.3.6)$$

Si el proceso está limitado en banda y muestreamos adecuadamente es posible recuperar la PSD del proceso continuo a partir del discreto en el tiempo. En un ordenador trabajamos con secuencias de muestras de los procesos aleatorios. Por lo tanto el cálculo de las esperanzas hay que realizarlo suponiendo que procesos son ergódicos. Es decir, que sus promedios temporales coinciden con esperanzas. En ese caso, la función de autocorrelación se puede estimar como:

$$\hat{R}'_X(m+M+1) = \begin{cases} \dfrac{1}{N-m}\sum_{n=1}^{N-m} X_{n+m}X_n^*, & m = 0...,M \\ \dfrac{1}{N-m}\sum_{n=-m+1}^{N} X_{n-m}X_n^*, & m = -1,-2,....-M \end{cases} \qquad (2.3.7)$$

siendo *N* el número de muestras de la secuencia y *M* el número de desplazamientos positivos y negativos para el cálculo de la autocorrelación. Se ha tenido en cuenta que MATLAB considera los índices empezando por el 1 (no existen índices negativos ni 0).

Podemos observar que se cumple la propiedad de simetría:

$$\hat{R}'_X(-m+M+1) = \hat{R}'_X(m+M+1)^* \qquad (2.3.8)$$

La secuencia es una estimación de la autocorrelación desplazada (ya que en la muestra 1 aparecerá la correspondiente a *m* = –*M*). Para evitar ese desplazamiento (que tendría el efecto de producir una fase no nula en la PSD) definimos la estimación de la autocorrelación como

$$\begin{aligned} \hat{R}_X(m+1) &= \hat{R}'_X(m+M+1) \qquad m = 0,1,...,M \\ \hat{R}_X(m+2M+2) &= \hat{R}'_X(m+M+1) \qquad m = -1,-2,...,-M \end{aligned} \qquad (2.3.9)$$

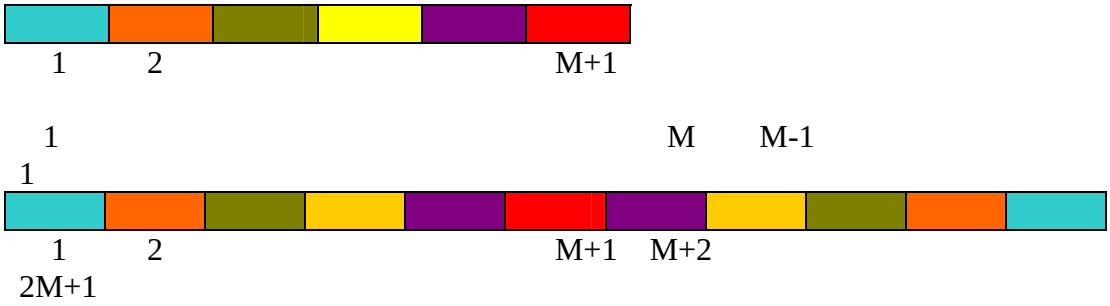


Figura 2.5. Simetría de la función Autocorrelación

Las secuencias con que trabajamos son finitas. Se trata de ventanas de datos correspondientes a los procesos aleatorios. Las frecuencias también han de ser discretas. Por lo tanto la transformada que utilizamos para estimar los espectros de densidad de potencia es la Transformada Discreta de Fourier (DFT), que considera la secuencia original periódica de período igual a su número de muestras y utiliza secuencias discretas. Partimos de la estimación de la autocorrelación $\hat{R}_x'(m+M+1)$, cuyas muestras se numeran de 1 a 2*M*+1. La estimación de la PSD se realiza del siguiente modo

$$\hat{S}_X(k+1) = \sum_{l=0}^{2M} \hat{R}_X(l+1) e^{-j\frac{2\pi}{2M+1}kl} \qquad \text{k=0,..2M} \qquad (2.3.10)$$

Lo que es equivalente a

$$\hat{S}_X(k+1) = \sum_{n=-M}^{M} \hat{R}_X(m+M+1) e^{-j\frac{2\pi}{2M+1}k(m+M)} \qquad \text{k=0,..2M} \qquad (2.3.11)$$

Las frecuencias para las que se realiza la estima serán, en ciclos $f = k/(2M+1)$.

En MATLAB este cálculo se realiza con la función `fft`, que implementa el algoritmo FFT, que estima la Transformada de Fourier de una secuencia de datos. La orden `fft` permite realizar esta estimación utilizando el mismo número de puntos que tiene la secuencia discreta o bien utilizando un número distinto. Si se utilizan más puntos se rellena con ceros. Si se utilizan menos se trunca la secuencia. Nosotros utilizaremos los mismos puntos (vea la ayuda de MATLAB).

Aunque la PSD de un proceso tiene fase igual a 0 para todas las frecuencias, debido a la precisión del ordenador aparece una fase muy pequeña (despreciable), pero que nos impide representar la PSD en dos dimensiones frente a la frecuencia. Para evitar esto tendremos que solicitar a MatLab de forma explícita que represente el módulo de la PSD. El cálculo del módulo de un número complejo se realiza mediante la orden `abs`, mientras que para calcular la fase se utiliza `angle`.

Tal como hemos definido el cálculo de la PSD (es decir, tal como lo hace MATLAB con la orden `fft`) las frecuencias están definidas de 0 a 2*M*/(2*M*+1). Para que en la representación de la PSD las frecuencias se encuentren en el intervalo [–*M*/(2*M*+1),*M*/(2*M*+1)] es necesario utilizar la orden `fftshift`. Si tenemos una secuencia de un número par de muestras 2*L*, la orden `fftshift` realiza el cambio observado en la siguiente figura.

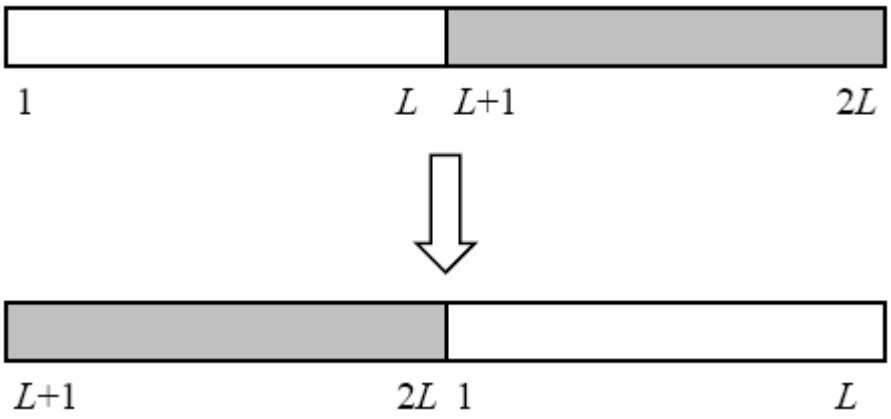


Figura 2.6. Efecto de la orden fftshift sobre una secuencia par

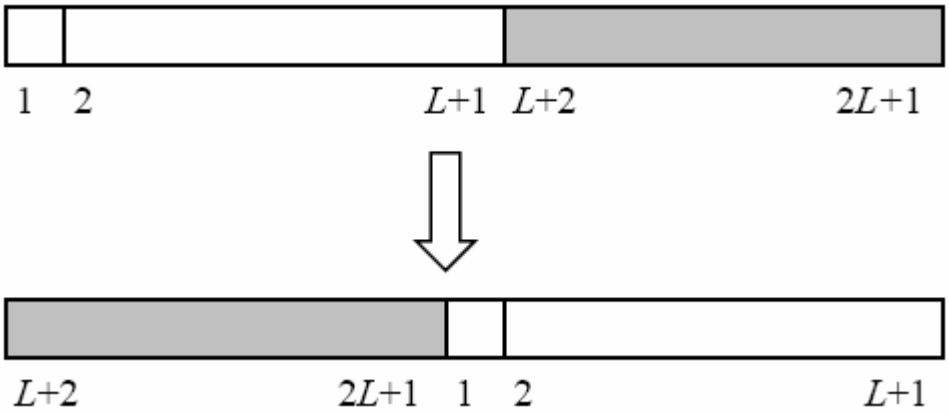


Figura 2.7. Efecto de la orden fftshift sobre una secuencia impar

Si hacemos `fftshift` a una PSD (o a cualquier secuencia obtenida de una orden `fft`) con un número impar de puntos las frecuencias asociadas a sus muestras serán $k/(2L+1)$ para $k = -L,\ldots, L$. Si se trata de un número par de puntos las frecuencias serán $k/(2L)$ para $k = -L,\ldots, L-1$ (recordemos que la Transformada de Fourier en Tiempo Discreto es periódica con período 2π rad o 1 ciclo).

Ejercicio 2.4.

Enunciado

a) Generar una función en MATLAB llamada autocorrelacion.m que estimen $\hat{R}_x(m)$ y $\hat{R}'_x(m)$. *Para ello en aucorrelacion.m implemente la siguiente ecuación:*

$$Rxx[m] = \frac{1}{N-m}\sum_{n=0}^{N-m-1}(x[n]x[n+m]) \quad \text{m=0,..M}$$

```
% Uso:
%  [Rx_des,Rx]= aucorrelacion(x,M)
% donde :
%          x        =    función muestra del proceso aleatorio o realización)
%          M       =    cantidad de muestras de la función autocorrelación
%          Rx      =    Estimación de la autocorrelación desde 1 hasta M+1
%          Rx_des =   Estimacion de la autocorrelacón desde [-M,M]
```

Para el calculo de Rx_des haga uso de la propiedad de simetría mostrada en la figura 2.5.

b) Una vez generada la función crear un fichero en MATLAB 'ejercicio2_4a.m' que haga lo siguiente :

1- *Genere una secuencia discreta de N=1000 muestras de variables aleatorias independientes e idénticamente distribuidad (i.i.d) con función de densidad de probabilidad uniforme en el intervalo [-1,1].*
2- *Utilice la función anterior para el calculo de la autocorrelación desplazada desplazada con M=50*
3- *Dibuje la función muestra de la variable aleatoria y la función autocorrelación desplazada*

c) Modifique el fichero anterior ejercicio2_4b.m para que realice los pasos anteriores 100 veces y que además estime el espectro de densidad de potencia. Promedie los resultados obtenidos y dibújelos.

Resultado

a)

```
function [Rx_des,Rx]=autocorrelacion(X,M)
N=length(X);
Rx=zeros(1,M+1);
for m=1:M+1,
  for n=1:N-m+1,
    Rx(m)=Rx(m)+X(n)*X(n+m-1);
  end;
  Rx(m)=Rx(m)/(N-m+1);
end;
for i=1:M,
    Rx_des(i)=Rx(M+2-i);
end
Rx_des(M+1:2*M+1)=Rx(1:M+1);
```

b)

```
N=1000;
X=2*randn(1,N)-1;
M=50;
[Rx_des,Rx]=autocorrelacion(X,M);
subplot(2,1,1);
plot(X)
title('Realización del Proceso')
subplot(2,1,2);
plot([-M:M],Rx_des)
title('Función autocorrelación del Proceso')
```

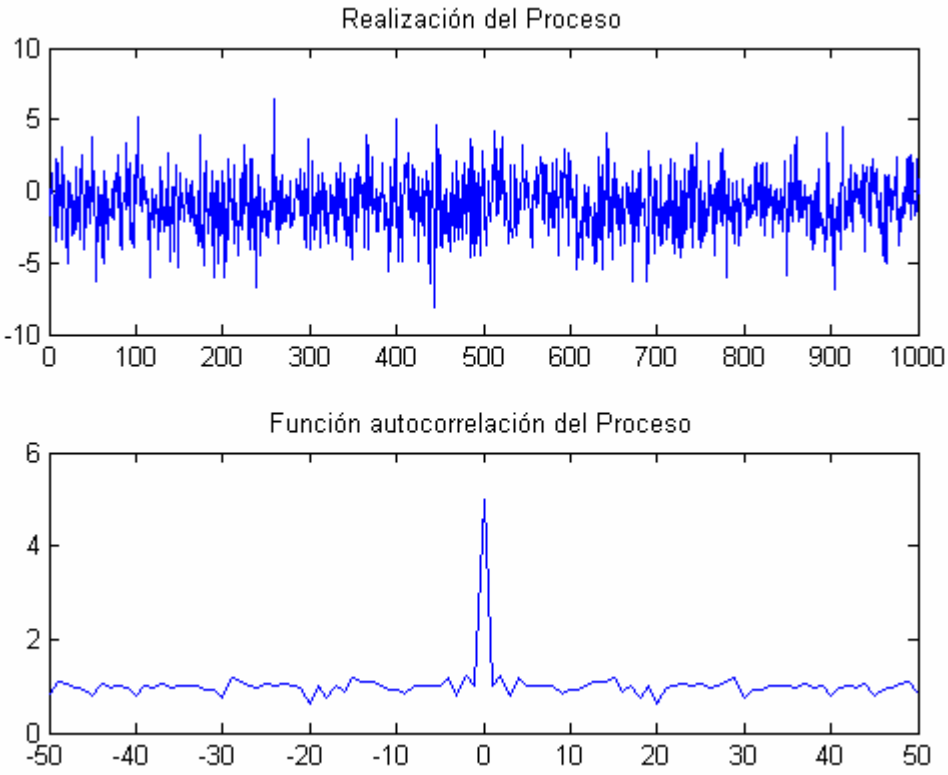


Figura 2.8. Función Autocorrelación de un Proceso Aleatorio Uniforme entre [-1,1]

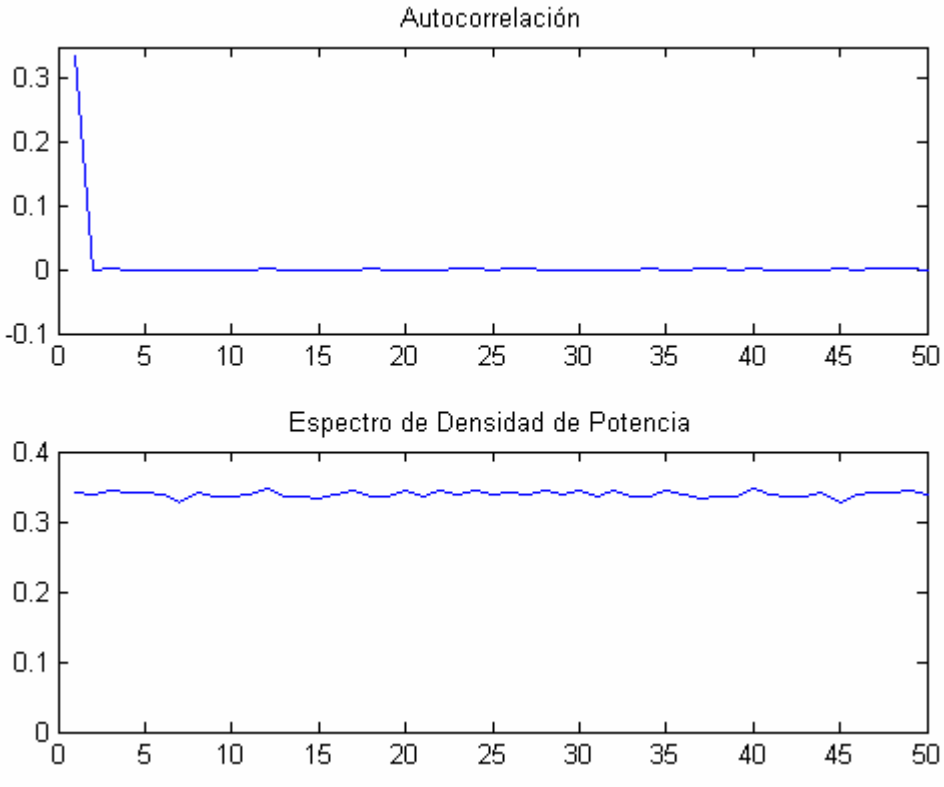
c)

```
clear all;
N=1000;
M=50;
Rx_av=zeros(1,M+1);
Sx_av=zeros(1,M+1);

for j=1:100,              % Media de 100 realizaciones
   X=2*rand(1,N)-1;       % N variables aleatorias uniformes
                  %e identícamente distribuidas entre -1 y 1
   [Rx_des,Rx]=autocorrelacion(X,M);     % Autocorrelacion
   Sx=fftshift(abs(fft(Rx)));       % Espectro de Potencia
   Rx_av=Rx_av+Rx;        % Suma de las autocorrelaciones
   Sx_av=Sx_av+Sx;        % Suma de los espectros
end;

Rx_av=Rx_av/100;          % Promedio de la Autocorrelacion
Sx_av=Sx_av/100;          % Promedio del Espectro

figure;
subplot(2,1,1);
plot(Rx_av);
axis([0 50 -0.1 0.35]);
title('Autocorrelación');
subplot(2,1,2);
plot(Sx_av);
axis([0 50 0 0.4]);
title('Espectro de Densidad de Potencia');
```



v

Figura 2.9. Función Autocorrelación y Espectro de Densidad de Potencia de un Proceso Aleatorio Uniforme entre [-1,1]

3.4. Procesos Blancos

Un proceso aleatorio $X(t)$ se dice que es blanco cuando su densidad espectral de potencia es constante para toda frecuencia. Al igual que los procesos aleatorios gaussianos, la importancia de este tipo de procesos reside en que el ruido térmico se puede modelar como un proceso aleatorio blanco, al menos para las frecuencias empleadas en las comunicaciones convencionales. En dichos sistemas la densidad espectral de potencia del ruido térmico se puede considerar $S_x(F) = \frac{No}{2}$ W/rad, siendo $No = kT$ donde T es la temperatura en grados kelvin y k la constante de Boltzmann (1.38×10-23 J/K).

En un proceso aleatorio blanco la función de autocorrelación es:

$$R_x(f) = \int_{-\infty}^{\infty} S_X(f) e^{-j2\pi f\tau} df = \frac{N_0}{2}\delta(\tau) \qquad (2.4.1)$$

Esto significa que si muestreamos un proceso aleatorio blanco en dos puntos diferentes t_1 y t_2, las variables aleatorias resultantes serán incorreladas.

3.5. Canal de Ruido Aditivo

El modelo más simple de canal es el ruido aditivo, se muestra en la figura 3.1.

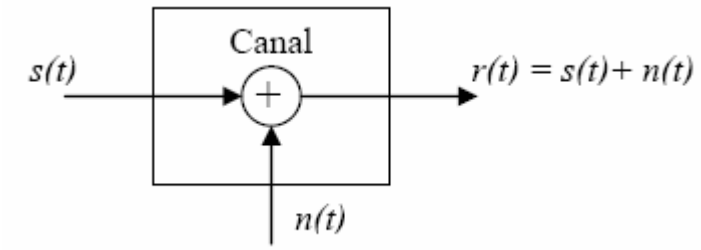


Figura 2.10. Modelo simple de canal

En este modelo, la señal transmitida s(t) se corrompe por una señal de ruido que modelamos como la función muestra de un proceso aleatorio n(t). Este proceso deberá caracterizarse estadísticamente, y en muchas ocasiones supondremos que es un proceso blanco y gaussiano, hablaremos entonces de ruido AWGN.

4. Filtrado Lineal de Procesos Aleatorios

4.1. Procesos Aleatorios Continuos

Si un proceso aleatorio estacionario *X(t)* atraviesa un filtro lineal e invariante en el tiempo que se caracteriza en el tiempo por una respuesta al impulso *h(t)* y en el dominio de la frecuencia por su respuesta en frecuencia:

$$H(f) = \int_{-\infty}^{\infty} h(\tau) e^{-j2\pi ft} dt \tag{2.4.1}$$

El proceso aleatorio a la salida del filtro lineal es:

$$Y(f) = \int_{-\infty}^{\infty} X(t) h(t-\tau) dt \tag{2.4.2}$$

El valor medio de Y(t) es:

$$m_y = H(0) m_X \tag{2.4.3}$$

En el dominio de la frecuencia, el espectro de densidad de potencia de Y(t) está relacionado con el espectro de densidad de potencia del proceso de entrada X(t) y con el filtro lineal mediante la expresión:

$$S_Y(f) = S_x(f) \left| H(f) \right|^2 \tag{2.4.4}$$

4.2. Procesos Aleatorios Discretos

Si consideramos el problema en el dominio discreto del tiempo. Suponemos que el proceso aleatorio X(t) se muestra y que sus muestras atraviesan un flitro lineal discreto en el tiempo con respuesta impulsiva h(n). El proceso a la salida viene dado por la siguiente suma de convolución:

$$Y(n) = \sum_{K=0}^{\infty} h(k) X(n-k) \tag{2.4.5}$$

Las expresiones para el valor medio y para el espectro de densidad de potencia coinciden con el caso continuo.

Ejercicio 2.5.

Enunciado

Suponer que un Proceso Aleatorio Blanco X(t) con Espectro de potencia Sx(f)=1 para todas ls frecuencias atraviesa un filtro lineal con respuesta al impulso:

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

Deterimina la densidad espectral del filtro Sy(f).

Resultado

```
paso=0.1;
Finicial=-2;
Ffinal=2;
f=[Finicial:paso:(Ffinal-paso)];
Sx=ones(1,length(f));
H=1./(1+(2*pi*f).^2);
Sy=Sx.*H.^2;
plot(Sy)
xlabel('F')
title('Espectro de Potencia de Y(t)')
```

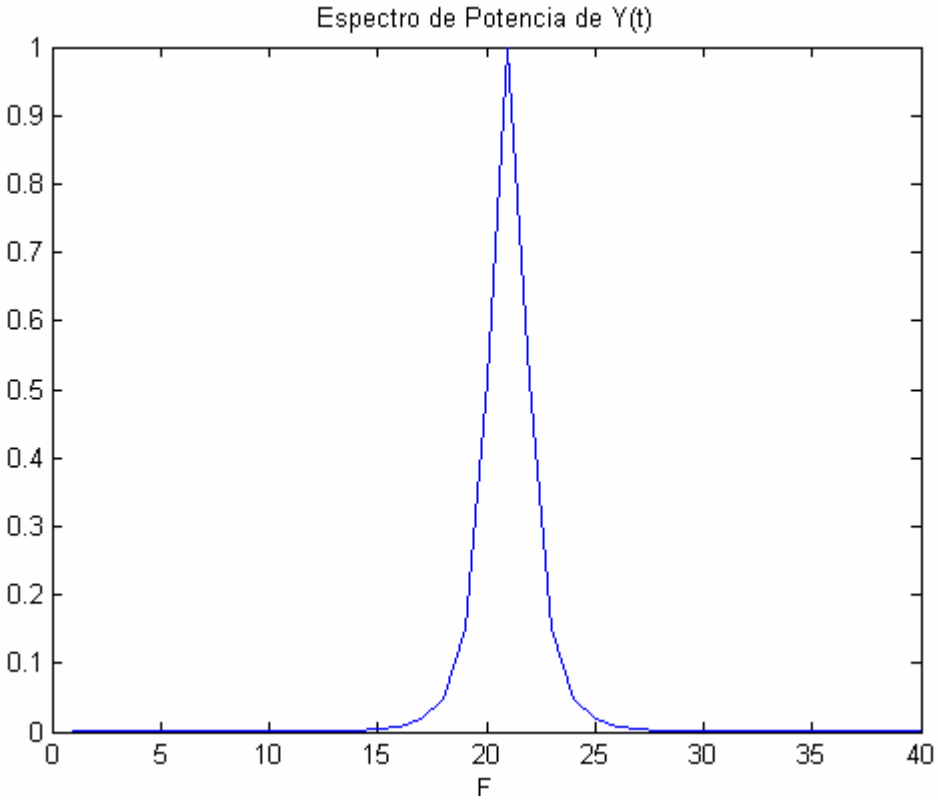


Figura 2.11. Resultado del Ejercicio 2.5

5. Simulación de Monte Carlo

5.1. Introducción

El método fundamental de evaluación de prestaciones mediante simulación se denomina Simulación Monte Carlo. En concreto, este método consiste en la repetición de ensayos independientes del experimento simulado de interés y la estimación de prestaciones promedio. Los ensayos independientes lo deben ser en el sentido estadístico, es decir las muestras de las variables aleatorias y los procesos aleatorios utilizados para simular el comportamiento del sistema han de ser necesariamente independientes. De esta manera, el bucle de la simulación Monte Carlo queda como puede verse en la siguiente figura.

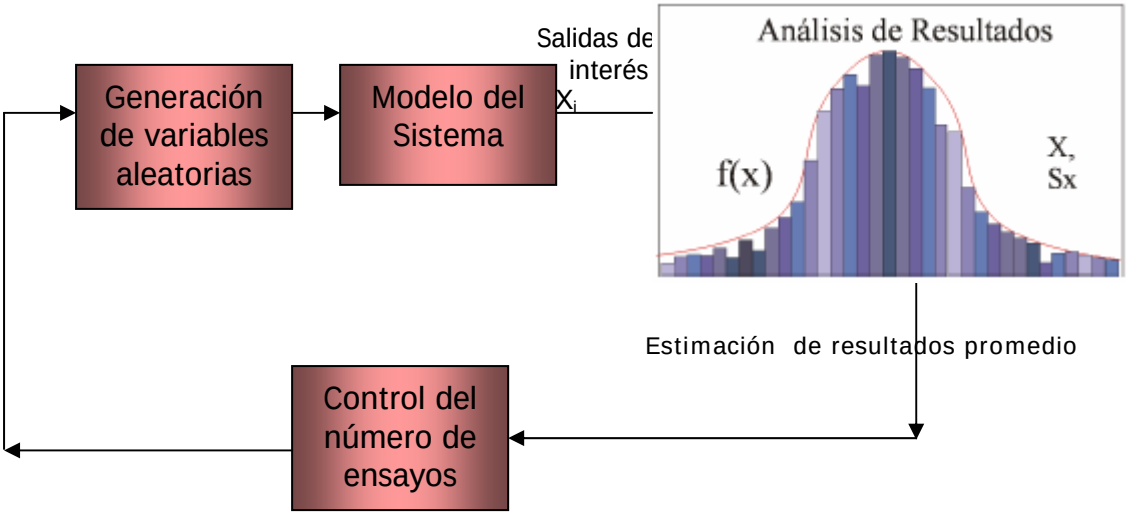


Figura 2.12. Bucle de simulación Monte Carlo

Tras generar las variables aleatorias precisas (independientes para cada ensayo), se inyectan en los puntos precisos del modelo del sistema. El sistema simulado proporciona una serie de salidas. Estas salidas se guardan para cada ensayo independiente. El número de ensayos independientes que se deben generar depende de los resultados que se deban promediar y de la calidad de la evaluación deseada.

5.2. Estimación de prestaciones. Calidad de Estimador

Las prestaciones que se suelen evaluar en los sistemas son prestaciones medias así como la dispersión de dichas prestaciones. En algunas ocasiones, también se evalúan comportamientos en casos peores o mejores, con el objetivo de acotar el funcionamiento del sistema, pero muy a medido tener únicamente medidas del comportamiento medio y de la varianza de una magnitud de salida es más que suficiente, Esta salida del sistema simulado en principio será una variable aleatoria

producto de transformar las variables aleatorias de entrada por una función dependiente del modelo del sistema.

Analizaremos por tanto los estimadores de la media y de la varianza de una variable aleatoria. Dichos estimadores no son más que funciones, que se aplican sobre las muestras de la variable de salida y permiten obtener una estimación de la media y la varianza de esta variable de salida. Llamaremos X a la variable aleatoria de interés (variable de salida) y $\{X\}_{i=1,..N}$ al conjunto de N muestras de la variable independiente obtenidas en los N ensayos independientes simulados. Las funciones más habituales para obtener la media y la varianza de las variables aleatorias son:

- Estimador de la Media Muestral: $$\langle x \rangle = \frac{1}{N} \sum_{i=1}^{N} x_i \qquad (2.5.1)$$

- Estimador de la Varianza: $$S_X = \frac{1}{N-1} \sum_{i=1}^{N} \left(x_i - \langle x \rangle \right)^2 \qquad (2.5.2)$$

5.3. Estimación de prestaciones de sistemas digitales de comunicaciones.

Típicamente, los sistemas de comunicaciones se caracterizan por su probabilidad de error de bit (BER: Bit Error Rate). En ocasiones lo que se analiza es la probabilidad de error de símbolo, en vez de la de error de bit. En concreto, la técnica habitual para comparar calidad de los distintos sistemas y codificadores es dibujar las curvas de BER frente a SNR (relación señal a ruido), bajo unas mismas condiciones de trabajo (entorno de interferencias, posibilidad de multitrayecto en el caso de sistemas de radiocomunicaciones,…). Para obtener una curva BER/SNR, únicamente se precisará tomar un conjunto de valores de la SNR razonables, y generar con cada uno de ellos un conjunto de ensayos que permitan estimar la BER correspondiente.

El método básico de estimación de la BER es el método de Monte Carlo. En este caso, lo que se hace es repetir reiteradamente el ensayo de enviar un bit (o ristra de bits), detectar si hay errores en la detección del bit enviado, contar los errores de bit, y dividir por el número de bits (ensayos) transmitidos. En definitiva, se utiliza la definición de frecuencia de la probabilidad, como número de casos favorables entre número de casos posibles.

La BER puede verse como el valor medio de un “detector de errores” de bit, que denominaremos H(y), done y es la salida sobre la que se realiza la decisión. Existen dos situaciones posibles para cada ensayo independiente de la simulación Monte Carlo:

- -H(y) vale 0 si no hay error.
- -H(y) vale 1 si hay error.

Por lo tanto, el estimador de la BER se podría realizar como:

$$BER = \frac{1}{N_{ensayos}} \sum_{i=1}^{N_{ensayos}} H(y_i) = \frac{N_{errores}}{N_{ensayos}} \qquad (2.5.3)$$

El estimador cumple*: $E\{BER\}=P_B$,* pero tiene un problema, es que tiene una varianza muy grande. Puede demostrarse que la varianza de este estimador es:

$$Var\{BER\} = \frac{P_B(1-P_B)}{N_{ensayos}} \qquad (2.5.4)$$

Para tener un estimador razonable de la probabilidad de error, desearíamos que la desviación típica del estimador (la raíz cuadrada de la varianza) fuese mucho menor que el valor estimado. Asumamos que el estimador sigue una distribución aproximadamente gaussiana (lo que solo es válido en realidad para un número muy grande de ensayos). De la descripción de la gaussiana podemos deducir que existe una probabilidad bastante alta de que el estimador quede por encima de $P_B + \sqrt{Var\{BER\}}$, para asegurar que el etimador sea poco errático buscaremos que $P_B >> \sqrt{Var\{BER\}}$. Para valores pequeños de P_B el valor de la varianza del estimador de la BER puede aproximarse por:

$$Var\{BER\}\ ;\ \frac{P_B}{N_{ensayos}} \qquad (2.5.5)$$

Por lo tanto, y operando, puede llegar a verse que la condición anterior queda como:

$$N_{ensayos} >> \frac{1}{P_B} \qquad (2.5.6)$$

Una regla que se utiliza habitualmente es que el número de ensayos independientes debe estar en el orden de 10/ P_B ensayos para tener un estimador cuya varianza sea mucho menor que la media (y que por tanto el error relativo sea pequeño).

La simulación de Montecarlo sigue por lo tanto la siguiente Figura.

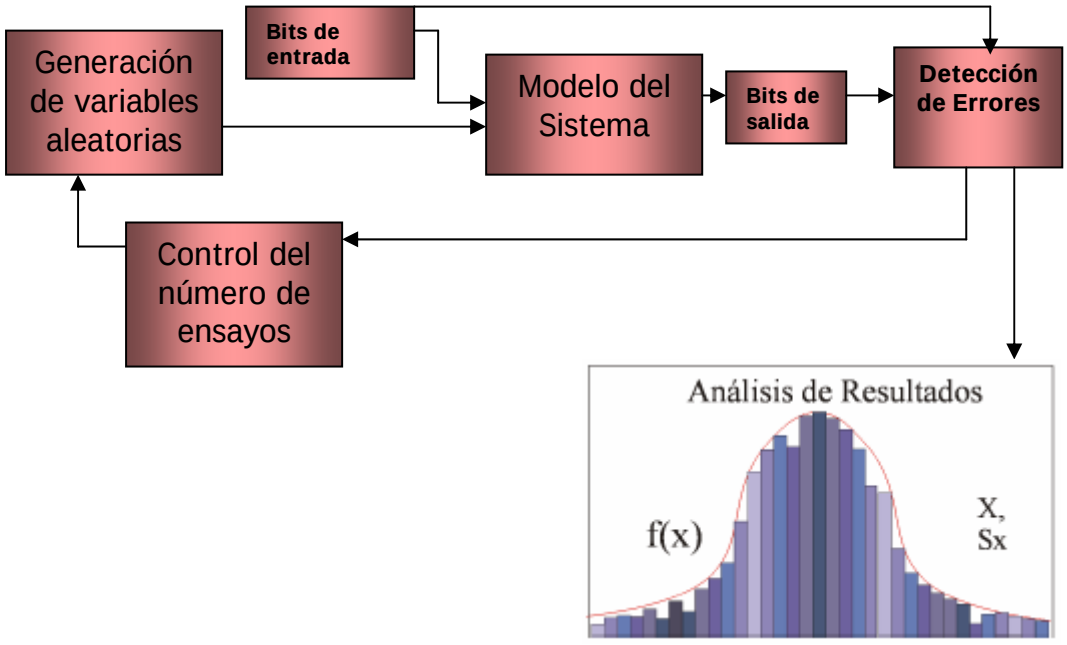


Figura 2.13. Estimación de la BER mediante el Método de Monte Carlo