"MATLAB PARA COMUNICACIONES Y SIMULACIÓN MONTECARLO"

Trabajo Preparatorio N°4
Laboratorio de Comunicaciones Inalámbricas

Melanny Cecibel Dávila Pazmiño
Ingeniería en Telecomunicaciones
Facultad de Eléctrica y Electrónica
Quito, Ecuador
melanny.davila@epn.edu.ec

***Abstract*—En el siguiente preparatorio se abordaran temas acerca de tipos de modulación, ruido AWGN, SNR, BER y otros parámetros que influyen en el entorno de un sistema de comunicación.**

***Index Terms*—MatLab, SNR, modulación, BER, E_b/N_0, probabilidad.**

I. Introducción

Una simulación Montercarlo es un método estadístico que trabaja con la generación de variables aleatorias para resolver problemas matemáticos y de esta manera analizar las características o comportamiento de un sistema. Este método es útil para el análisis de riesgos dado a que hace uso de distribuciones de probabilidad permitiendo así que la simulación no sólo presente lo que puede suceder, sino lo probable que es un resultado.

II. Objetivos

- Familiarizar al estudiante con el método Monte Carlo
- Calcular el Bit Error Rate (BER) para diferentes modulaciones en un canal AWGN usando Montecarlo

III. Cuestionario

A. Consultar los comandos randi, awgn, comm.BPSKModulator, qammod, qamdemod, comm.BPSKDemodulator, biterr, berawgn. Este ítem no deberá superar las 2 páginas.

randi

- La función randi devuelve un valor aleatorio entero entre el 1 y un valor máximo; siempre que se defina como $x = randi(max)$. Los números obtenidos siguen una distribución uniforme.
- Si $x = randi(max, n)$ se crea una matriz de dimensión nxn con valores aleatorios entre 1 y max. También, se puede definir el tipo de datos ('single', 'double', 'int8', 'uint8', 'int16', 'uint16', 'int32' o 'uint32') [1].

awgn

- Mediante este comando se añade ruido blanco Gaussiano a una señal de datos, como uno de sus principales argumentos se puede definir el valor de SNR (relación señal a ruido) en dB mientras que si no es definido por defecto tendrá valor de 0 [dBW].
- Además de definir el valor de SNR se puede definir la potencia de la señal de datos en dBW, entre otras propiedades como la unidad de potencia (decibelos o escala lineal) [2].

comm.BPSKModulator

- Este objeto permite realizar una modulación BPKS, su salida es una señal modulada en banda base. Para realizar esto primero se debe definir objeto y establecer sus propiedades como: desfasamiento, tipo de dato de salida, entre otros. Una vez hecho esto, se podrá llamar al objeto como si fuera una función en cualquier parte del script [3].

qammod

- Este comando realiza modulación por amplitud de cuadratura; con $Y = qammod(X, M)$ se modula la señal de entrada X con el orden de modulación especificado por la variable M. La salida Y es la señal modulada.
- $Y = qammod(X, M, symOrder)$ especifica el orden de los símbolos. Finalmente, usando el comando $Y = qammod(---, Nombre, Valor)$ se permite especificar opciones usando argumentos y su valor [4].

qamdemod

- $Z = qamdemod(Y, M)$ devuelve una señal Z demodulada en quadratura en base a una señal de entrada Y cuyo orden de modulación es M. Se puede especificar el orden de los símbolos para la demodulación.
- En el caso de usar la línea de código $Z = qamdemod(- - -, Nombre, Valor)$ se puede especificar opciones usando uno o más argumentos [5].

comm.BPSKDemodulator

- Mediantela creación de este objeto se puede realizar la demodulación de una señal modulada previamente con BPSK; pueden definirse los mismos argumentos que en el caso del objeto comm.BPSKModulator [6].

biterr

- El comando devuelve el número de bits que son diferentes en la comparación entre dos vectores y la relación entre el número total de bits diferentes entre los bits totales.
- La función determina el orden en el que se comparan los dos vectores en función de sus tamaños [7].

berawgn

- Este comando permite obtener BER (tasa de bits errados) y SER (tasa de símbolos errados) sobre señales de datos no condificadas que trabajan sobre canales AWGN. Como argumento recibe E_b/N_o que es la relación entre la energía de bits y la densidad de potencia del ruido en decibelios [8]. Como un parámetro se puede definir el tipo de modulación: M-PSK, OQPSK, M-FSK, MSK o CPFSK.

B. Se debe crear una función llamada modulador.m que tendrá como parámetros de entrada 1) tren de bits, y 2) m. La salida de esta función es un vector de longitud 1xN con los símbolos modulados. Esta función debe modular los btis en BPSK, 4-QAM, 16-QAM según el valor de m a la entrada. Para la modulación BPSK se debe usar el comando comm.BPSKModulator. Para las modulaciones 4-QAM y 16-QAM se debe utilizar el comando qammod con la opción de entrada tipo bit.

La función solicitada se presneta en la figura 1.

```
function [informacionModulada] = modulador(informacion, estados)
%Creacion del objeto de modulador BPSK
moduladorbpsk = comm.BPSKModulator;
switch estados
    case 2 %BPSK
        informacionModulada = moduladorbpsk(informacion);
    case 4 %4QAM
        informacionModulada = qammod(informacion, 4);
    case 16 %16QAM
        informacionModulada = qammod(informacion, 16);
end
end
```

Fig. 1. Función creada para modular

*C. Se debe crear una función llamada demodulador.m que tendrá como parámetros de entrada 1) símbolos a demodular, y 2) m. La salida de esta función es un vector de longitud 1 × (N*m Para la demodulación BPSK se debe usar el comando comm.BPSKDemodulator. Para las modulaciones 4-QAM y 16-QAM se debe utilizar el comando qamdemod con la opción de salida tipo bit.*

A continuación, se presenta el segmento de código que realiza lo solicitado.

```
clc
clear all
close all
disp('Ingrese el numero de estados de modulacion a utilizar')
m = input('a. BPSK b.4-QAM c.16-QAM: ');
%Numero de simbolos
N = input('Ingrese el valor de N: ');
%Creacion del vector de informacion
informacion = randi([0 m-1],1, N*m)';
%Validacion de los estados de modulacion
if m == 2 || m == 4 || m == 16
    %Modulacion los datos
    informacionModulada = modulador(informacion, m);
else
    disp('Opcion no valida')
end
%Demodulacion de la senal modulada
informacionDemodulada = demodulador(informacionModulada, m);
```

Fig. 2. Función demodulador.m

*D. Se debe crear un script llamado calculo_BER.m que genere un vector de bits aleatorios de longitud 1 × (N*m) usando el comando randi. Luego module los bits con la función modulador.m y luego demodule los símbolos con la función demodulador.m*

El segmento de código mostrado en la figura 3, realiza la modulación y demodulación de datos.

```
clc
clear all
close all
disp('Ingrese el numero de estados de modulacion a utilizar')
m = input('a. BPSK b.4-QAM  c.16-QAM:');
%Numero de simbolos
N = input('Ingrese el valor de N');
%Creacion del vector de informacion
informacion = randi([0 m-1],1, N*m)';
%Validacion de los estados de modulacion
if m == 2 || m == 4 || m == 16
    %Modulacion los datos
    informacionModulada = modulador(informacion, m);
else
    disp('Opcion no valida')
end
%Demodulacion de la senal modulada
informacionDemodulada = demodulador(informacionModulada, m);
```

Fig. 3. Script desarrollado

REFERENCES

[1] "Pseudoaleatorio enteros distribuidos uniformemente - MATLAB randi - MathWorks América Latina". https://la.mathworks.com/help/matlab/ref/randi.html (accedido jun. 19, 2021).

[2] "Add white Gaussian noise to signal - MATLAB awgn - MathWorks América Latina". https://la.mathworks.com/help/comm/ref/awgn.html?searchHighlight=awgn&s_tid=srchtitle (accedido jun. 21, 2021).

[3] "Modulate using BPSK method - MATLAB - MathWorks América Latina". https://la.mathworks.com/help/comm/ref/comm.bpskmodulator-system-object.html?searchHighlight=comm.bpskmodulator&s_tid=srchtitle (accedido jun. 19, 2021).

[4] “Quadrature amplitude modulation (QAM) - MATLAB qammod - MathWorks América Latina”. https://la.mathworks.com/help/comm/ref/qammod.html?searchHighlight=qammod&s_tid=srchtitle#bu39xr3 (accedido jun. 19, 2021).
[5] “Quadrature amplitude demodulation - MATLAB qamdemod - MathWorks América Latina”. https://la.mathworks.com/help/comm/ref/qamdemod.html?searchHighlight=qamdemod&s_tid=srchtitle (accedido jun. 19, 2021).
[6] “Demodulate using BPSK method - MATLAB - MathWorks América Latina”. https://la.mathworks.com/help/comm/ref/comm.bpskdemodulator-system-object.html?searchHighlight=comm.bpskdemodulator&s_tid=srchtitle (accedido jun. 21, 2021).
[7] “Number of bit errors and bit error rate (BER) - MATLAB biterr - MathWorks América Latina”. https://la.mathworks.com/help/comm/ref/biterr.html?searchHighlight=biterr&s_tid=srchtitle (accedido jun. 21, 2021).
[8] “BER and SER for uncoded data over AWGN channels - MATLAB berawgn - MathWorks América Latina”. https://la.mathworks.com/help/comm/ref/berawgn.html?searchHighlight=berawgn&s_tid=srchtitle (accedido jun. 21, 2021).