

UNIVERSIDAD NACIONAL DE CÓRDOBA

FACULTAD DE CIENCIAS EXACTAS, FÍSICAS y NATURALES

COMUNICACIONES DIGITALES



Trabajo práctico N2:

Simulaciones de Sistemas de Comunicaciones

Profesora:

- **Dra. Gabriela Corral Briones.**

Alumnos

- | | | |
|-------------------------------|----------|-------|
| • Blanco, Lucas | 39056912 | ICOMP |
| • Bosack, Federico Ezequiel | 39267560 | ICOMP |
| • Collante, Gerardo Alexis | 39022782 | ICOMP |
| • Oliva Arias, Carlos Agustín | 35915867 | ICOMP |
| • Vargas Rodríguez, Diego | 39983867 | ICOMP |

Fecha

24/05/2018

Índice

Introducción.....	3
Desarrollo.....	3
Consigna 1.....	3
Implementación.....	3
Cantidad de estados.....	3
Máquina finita de estados.....	4
Trellis.....	4
Algoritmo de Viterbi.....	4
Vector secuencia.....	5
Matriz de estados.....	5
Matriz de sobrevivientes.....	5
Matriz de métricas.....	5
Obtención del sobreviviente.....	6
Traceback.....	6
Matriz de decodificación.....	6
BER/SNR.....	6
Consigna 2:.....	7
Implementación.....	7
Cantidad de estados.....	8
Maquina finita de estados.....	8
Trellis.....	8
Algoritmo de Viterbi.....	8
Vector de secuencias.....	8
Matriz de metricas.....	8
BER/SNR con codificación.....	9
Consigna 3.....	9
BER/SNR sin codificación.....	9
Conclusiones.....	10

Introducción

Mediante lo aprendido a lo largo de la materia, en este práctico de laboratorio se representará mediante simulaciones dos sistemas de comunicaciones digitales, uno con filtro FIR y otro con código convolucional, que se les agregará ruido gaussiano blanco y luego se intentará decodificar a través del Algoritmo de Viterbi. Se irán variando los valores de la relación señal-ruido (SNR) para ver como afecta al porcentaje de errores.

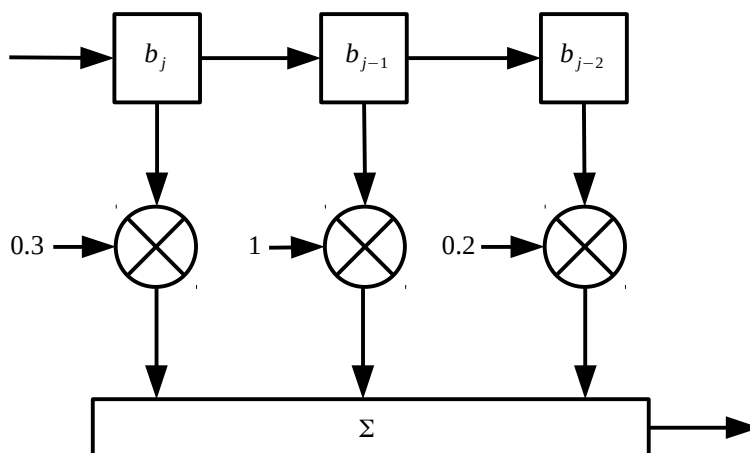
Desarrollo

Consigna 1

1. Realizar el simulador de un sistema de comunicaciones $BPSK = \{\pm 1\}$ donde el canal es modelado mediante un FIR con tres coeficiente $[0.3, 1, 0.2]$ y el receptor es un detector óptimo implementado con el algoritmo de Viterbi. Obtener la curva de BER para $SNR_{dB} = \{1, 2, 3, 4, 5\}$.

Implementación

Se implementó un código de generación aleatoria de símbolos $BPSK = \{\pm 1\}$ Para modelar el canal, empleamos un filtro FIR con los coeficientes 0.3 , 1 y 0.2 que se muestra en la figura donde b_j es el bit que entra, y b_{j-1} y b_{j-2} son las memorias que guardan los bits anteriores.



Cantidad de estados

Esto nos permite determinar cuantos estados poseerá la transmisión. La cantidad de estados C_e está determinada por la siguiente ecuación donde λ es el tamaño del alfabeto y m es la cantidad de memorias.

$$C_e = \lambda^m$$

Ya que al ser BSPK el tamaño del alfabeto es 2 y la cantidad de memorias es 2 obtenemos que la cantidad de estados es 4.

Máquina finita de estados

Una vez obtenida la cantidad de estados se procederá a armar la maquina finita de estados para representar el sistema que se muestra en la fig. 2.

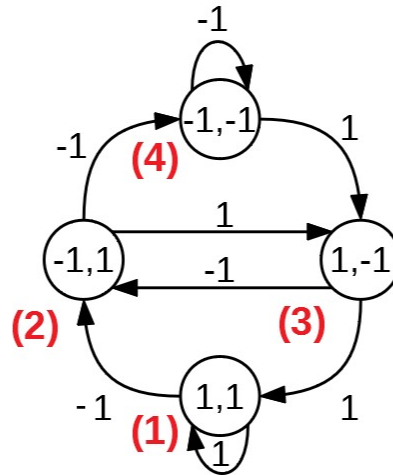


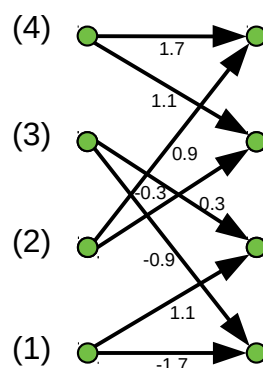
Fig. 2. Máquina de estados

Los números en rojo representan un número simbólico de estado para obtener una mejor comprensión. La manera de armarla es que b_{j-1} y b_{j-2} representan el estado actual, b_j y b_{j-1} representan el estado siguiente y x_j el valor de la transición, que luego será llamada métrica.

Trellis

Es posible definir a un trellis como un diagrama desplegado dinámico de transiciones y nos sirve para representar de manera más clara la evolución de una máquina finita de estados en el tiempo. Se realizó el trellis para este sistema en la siguiente figura.

Los números a la izquierda representan los estados y los valores en las flechas las métricas ya calculadas.



Algoritmo de Viterbi

Para el detector, se empleó el algoritmo de Viterbi; un método que dada una determinada secuencia de entrada es capaz de obtener los símbolos decodificados sin necesidad de probar todas las combinaciones posible (es decir, con una cantidad de cálculos que crece de manera exponencial) para obtenerlos con una cantidad de operaciones que crece de manera lineal.

Vector secuencia

Posee la siguiente forma y son los bits de entrada al decodificador.

1	2	3	4	5	.	.	.	N-1	N

Matriz de estados

Servirá para representar los estados del trellis. Los valores inf significan un valor negativo muy grande para representar estados imposibles.

		ITERACIÓN									
		1	2	3	4	5	.	.	.	N-1	N
E S T A D O	1	inf	inf							inf	inf
	2	inf	inf								inf
	3	inf								inf	inf
	4										

Matriz de sobrevivientes

Se guardaran en la iteración i, de donde provino, es decir su origen. Esto es necesario para más adelante realizar el traceback.

		ITERACIÓN								
		1	2	3	4	5	.	.	.	N-1
E S T A D O	1									
	2									
	3									
	4									

Matriz de métricas

A partir de esto es posible realizar una matriz de métricas que luego será escalada con el valor secuencia[iteracion que llega al decodificador, para obtener una nueva matriz de metricas.

		ORIGEN			
		1	2	3	4
D E S T I N O	1	1.7	X	0.3	X
	2	1.1	X	0.9	X
	3	X	0.3	X	-1.1
	4	X	0.9	X	-1.7

Obtención del sobreviviente

Una vez obtenida la matriz de nuevas metricas se debe sumar a cada columna origen el valor correspondiente a Estado[origen, iteracion. Luego se debe obtener el mayor de la fila (es decir, de los que llegan al mismo destino el mayor) y actualizar el valor Estados destino, iteracion+1. A su vez se debe actualizar Sobrevivientes origen, iteracion+1.

De esta manera se va completando el algoritmo hasta llegar al ultimo bit.

Traceback

Utilizando la matriz de sobrevivientes, es posible ir saltando hacía atrás desde el ultimo valor e ir “recolectando” el camino más pesado en forma inversa.

3	1	3	3	←1	←1
3	1	3	3	3	1
4	2	4	4	4	2
4	←4	4	4	4	2

Matriz de decodificación

Con la maquina de estados realizada es posible realizar una matriz de decodificación mostrada en la siguiente figura, donde las cruces significan transiciones imposibles:

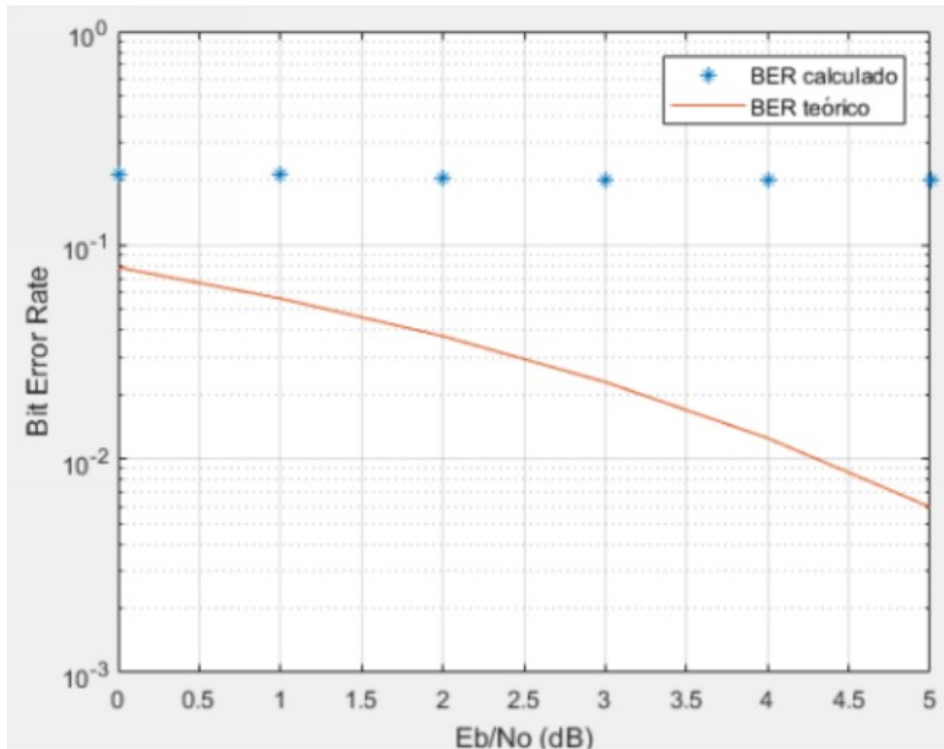
		ORIGEN			
		1	2	3	4
D E S T I N O	1	1	X	1	X
	2	-1	X	-1	X
	3	X	1	X	1
	4	X	-1	X	-1

Entonces con junto con el camino más pesado obtenido es posible obtener los símbolos que fueron enviados.

BER/SNR

La tasa de error de bit (BER) es la cantidad de errores de bit por unidad de tiempo y se define como la relación entre los bits errados sobre bits enviados. La SNR es la relación señal ruido y cuantifica cuanto ruido gaussiano blanco tiene una señal.

Es útil graficar BER en función de SNR para un determinado sistema de comunicación para determinar que tan inmune al ruido es y las posibilidades de enviar bits sin errores según la cantidad de ruido.



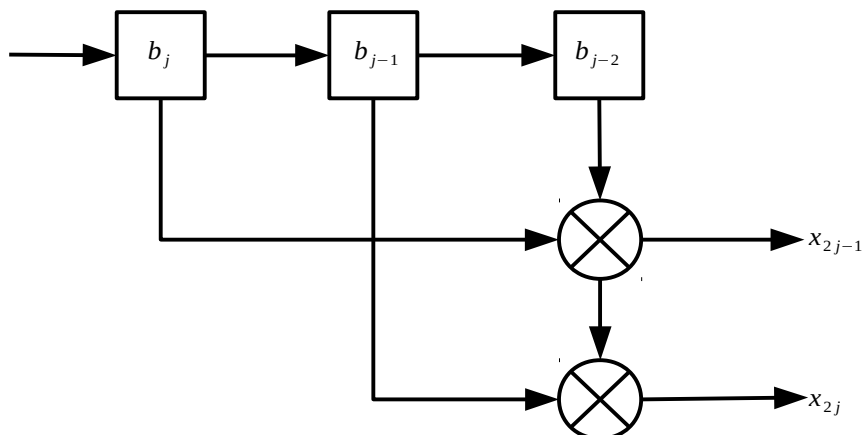
A mayor SNR el BER no decrece, esto indica que quizás el algoritmo no está funcionando de manera eficiente y por tanto no es posible sacar conclusiones de este gráfico.

Consigna 2:

Realizar el simulador de un sistema de comunicaciones BPSK en donde la señal a transmitir es codificada mediante el código convolucional implementado en clase. El canal es AWGN. La señal codificada debe ser escalada por $1/\sqrt{2}$ para mantener la energía del bit de información transmitido. Obtener la curva de BER en función de la SNR_{dB} para la señal sin codificación y con codificación.

Implementación

Se implementó un código de generación aleatoria de símbolos $BPSK = \{\pm 1\}$. Para modelar el canal, empleamos un código convolucional que se muestra en la figura donde b_j es el bit que entra, y b_{j-1} y b_{j-2} son las memorias que guardan los bits anteriores.



Cantidad de estados

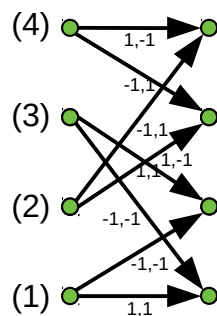
El calculo es el mismo que se realizó con anterioridad debido a que poseen el mismo alfabeto y la misma cantidad de memorias.

Maquina finita de estados

El grafico de la maquina de estados también es el mismo que el realizado para el FIR porque poseen el mismo alfabeto y cantidad de memorias, sin embargo al tener dos salidas difieren las métricas y su modo de calcularse, ya que ahora la secuencia estará conformada por dos bits (uno por cada salida) y debe aplicarse el producto punto para este vector.

Trellis

Por tanto el trellis difiere del anterior en que se debe realizar el producto punto para obtener cada una de las métricas.



Algoritmo de Viterbi

Se utiliza el mismo explicado anteriormente, se mostrará únicamente el vector de secuencia y la matriz de métricas a utilizarse.

Vector de secuencias

Cada una de los valores de la secuencia tendrá dos bits en lugar de uno como en el caso anterior.

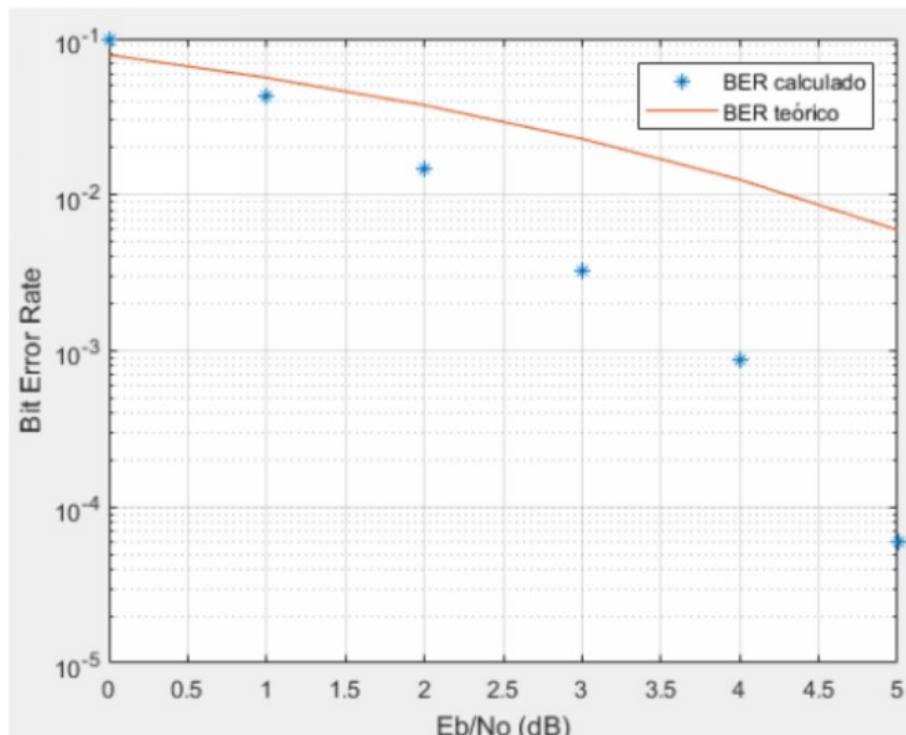
0	1	2	.	.	.	N-1	N

Matriz de métricas

Se observa que cada uno tiene dos bits, debe escalarse con el vector de secuencias para obtener la nueva matriz de métricas, el procedimiento sigue como se explico anteriormente.

		ORIGEN							
		0		1		2		3	
D E S T I N O	0	1	1	X	-1	-1	X		
	1	-1	-1	X	1	1	X		
	2	X	1	-1	X	-1	1		
	3	X	-1	1	X	1	-1		

BER/SNR con codificación

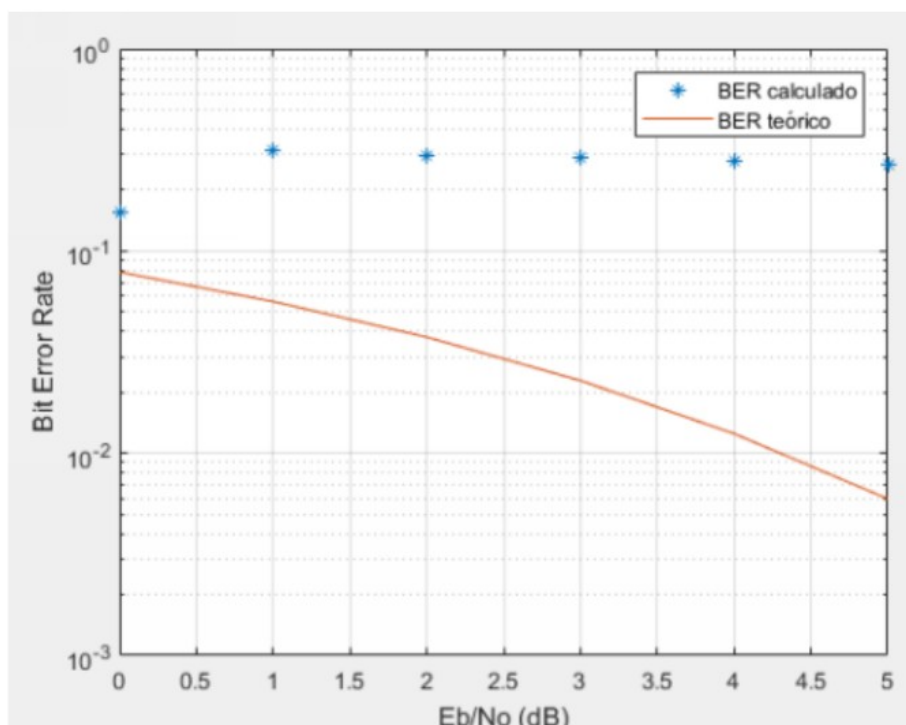


Esto se interpreta que a mayor SNR (eje de abscisas) el BER decrece, es decir, se obtiene una menor tasa de bits de errores lo que mejora bastante la comunicación.

Consigna 3

Se pide enviar datos igual que antes pero sin codificación, para ser enviados solo con ruido. Luego se decodificó de la manera que si era positivo el valor era 1 y caso contrario, -1. Así se llegó al siguiente gráfico.

BER/SNR sin codificación



Conclusiones

Se obtuvo una grata experiencia en diseñar el algoritmo de Viterbi durante la materia para entenderlo de mejor manera y aplicarlo sin limitaciones. Sin embargo hubo complicaciones tales como la migración del código desde JAVA a MATLAB, lo que llevo mucho tiempo de prueba y error. Esto se debió a que en JAVA nos resultó complicado implementar AWGN en las respectivas señales y obtener el plot de BER/SNR solicitado, lo que llevo a una gran demora de tiempo que pese a nuestros esfuerzos no pudimos evitar.

Cerrando se podría decir que esta metodología nos ayudo mucho para obtener conocimientos en conjunto para aprender a trabajar en equipo y lograr una mejor performance.

Bibliografía

[Principles of Digital Communications - Bixio Rimoldi]

[Notas de clase]