

Comunicaciones Digitales

INFORME TALLER 3

Teoría de Información: Codificación

GRUPO: 1

Nombre	CI	email
Héctor Garrido	51070701	hectoreduardogarridoalvarez@gmail.com
Santiago Avila	49017096	s.avila07@outlook.com

1. Parte I: Canal simétrico con borrado

a)

En esta sección se busca determinar cuánto vale el valor de la probabilidad de que un símbolo sea borrado, en otras palabras, se busca determinar el valor de $P(Y = e)$. Aplicando el teorema de la probabilidad total se llega a que el resultado es el siguiente:

$$P(Y = e) = P(Y = e|X = x_0)P(X = x_0) + P(Y = e|X = x_1)P(X = x_1) = p \quad (1)$$

b)

Se busca ahora obtener los bits de información por símbolo transmitido, esto se consigue al calcular la entropía del transmisor antes de que sea pasado por el canal. Se utiliza la entropía dado que se puede interpretar como una forma indicar la información que da cada bit, siendo los más probables los que menos información aportan y los más raros los que aportan más.

Teniendo esto en consideración se puede observar el valor de la entropía en la siguiente ecuación, en donde $\Omega(a)$ es una forma de denominar este tipo de entropías debido a la forma que presenta su gráfica, como se ve en la [Figura 1](#).

$$H(x) = -a \log_2(a) - (1 - a) \log_2(1 - a) = \Omega(a)$$

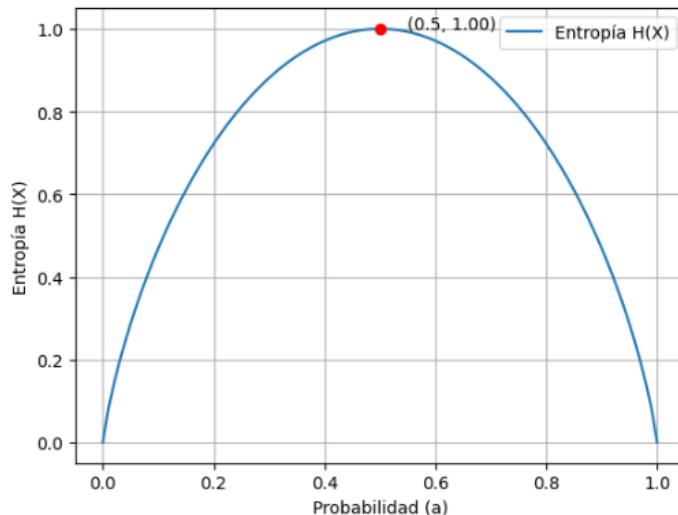


Figura 1: Entropía de una señal binaria.

c)

La mayor incertidumbre sobre el símbolo a transmitir se presenta cuando los símbolos son equiprobables, debido a que no se puede tener ninguna certeza sobre cual es el símbolo que se recibirá luego. A su vez esto se puede comprobar cuando se observa la [Figura 1](#) en donde se logra ver como el máximo se alcanza cuando la probabilidad de enviar x_1 es igual a la de x_0 .

d)

Una modificación posible es agregar un bit extra de tal forma que se codifique un nuevo símbolo en la modulación BPSK, este símbolo se colocaría en el medio de los símbolos pre-existentes como se puede ver en el ejemplo de la [Figura 2](#).

Sabiendo que el canal es AWGN, se calcula las probabilidades p y q de la siguiente forma:

$$P(Y = e|X = x_0) = \int_{-\frac{A}{2}}^{+\infty} p_{x_0}(t)dt = Q\left(\frac{A}{2\sigma}\right) = p \quad (2)$$

$$P(Y = x_1|X = x_0) = \int_{\frac{A}{2}}^{+\infty} p_{x_0}(t)dt = Q\left(\frac{3A}{2\sigma}\right) = q \quad (3)$$

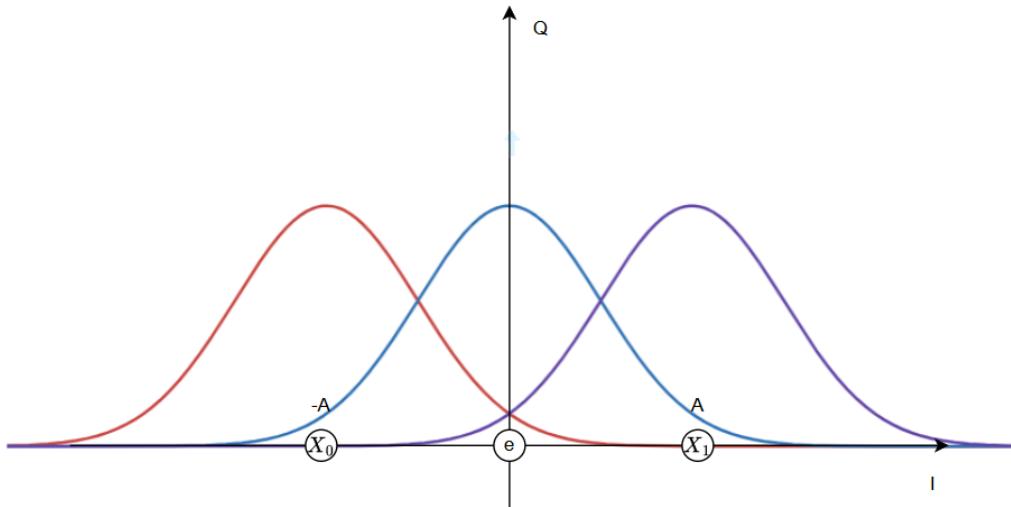


Figura 2: Ejemplo de distribución BPSK modificada.

En donde $Q(x)$ hace referencia a la formulación de la cola Gaussiana, A es la energía de transmisión, σ es la varianza del ruido y p_{x_0} es la distribución de probabilidad asociada al símbolo x_0 .

e)

En este caso se plantea la hipótesis de que p es pequeño, bajo esta condición parece razonable asumir que q será despreciable. En particular, teniendo en cuenta los resultados de la parte anterior, se puede ver que q es más baja que p , considerando el decrecimiento de la distribución normal lejos de su centro, se puede concluir que despreciar q es razonable.

f)

A continuación se mostrarán los resultados de las cantidades solicitadas en esta sección, los cálculos detallados se encontrarán en el anexo.

$$\begin{aligned} H\{Y\} &= \Omega(a)(1-p) + \Omega(p) \\ H\{Y|X\} &= \Omega(p) \\ H\{X|Y\} &= \Omega(a)p \\ I\{X,Y\} &= \Omega(a)(1-p) \end{aligned}$$

Una posible interpretación de $I\{X,Y\}$ es tomarlo como el grado de incertidumbre que se tiene sobre el valor de X dado que se conoce el valor de Y .

A partir de $I\{X,Y\}$ se puede determinar la capacidad del canal como $C = \max_{P(X=x)} I\{X,Y\}$. Sabiendo que la máxima incertidumbre se da cuando los símbolos son equiprobables se llega a que el canal tiene la forma siguiente:

$$C = 1 - p \quad (4)$$

g)

Para el caso de un canal tradicional, se considera una modulación BPSK en donde su diagrama es similar al que se encuentra en la [Figura 2](#) sólo que sin el valor en el origen. Bajo estas condiciones el valor de q es el siguiente:

$$P(Y = x_1 | X = x_0) = Q\left(\frac{A}{\sigma}\right) = q \quad (5)$$

A su vez la capacidad del canal es ya conocida con la siguiente forma:

$$C = 1 - \Omega(q) \quad (6)$$

A partir de estos resultados se pueden llegar a la siguiente lista de conclusiones.

- Comparando las ecuaciones (3) y (5) se ve que la probabilidad de elegir el símbolo equivocado es mayor para el canal tradicional.
- Viendo las ecuaciones (2) y (5) se ve que la probabilidad de que un símbolo sea borrado es mayor que la probabilidad de elegir un símbolo erróneo en el canal tradicional.
- La capacidad del canal modificado es menor a la capacidad del canal tradicional.

Entonces, en resumen, si se está trabajando en aplicaciones en donde el confundir símbolos es crítico, lo mejor es optar por el canal modificado, pagando el costo de la menor capacidad de transmisión. Mientras que si lo que importa más es transmitir a una mayor capacidad, entonces el canal tradicional presenta más ventajas.

2. Parte II: Algoritmos de decisión iterativa

a)

Siendo Y_k el patrón obtenido de la palabra x_k , $H\{X|Y_k\}$ representa la entropía de la solución X , puede verse como el tamaño del espacio de posible X dado que el intento x_k dió el patrón y_k .

b)

Y_k es el patrón que dice si cada letra de x_k pertenece a X y está en el lugar correcto o no, es un mapeo de $x_k = a, b, c, d$, a un patrón que depende directamente de $X = \alpha, \beta, \gamma, \epsilon, \sigma$.

c)

Se puede utilizar la siguiente ecuación, dado que ambos se toman de un conjunto finito de valores:

$$H\{X, y_k\} = H\{y_k\} + H\{X|y_k\} = H\{X\} + H\{y_k|X\} \quad (7)$$

Se puede despejar

$$H\{X|y_k\} = H\{y_k|X\} + H\{X\} - H\{y_k\} \quad (8)$$

Tomando que $H\{X, y_k\} \leq H\{X\} + H\{y_k\}$ y la ecuación 7, en el peor caso:

$$H\{X|y_k\} \leq H\{X\} \quad (9)$$

Donde $H\{X\}$ no depende de y_k . Esta entropía $H\{X|y_k\}$ es la que se quiere minimizar en cada intento x_k para reducir el espacio donde se encuentra X , por lo que viendo 8, se desea maximizar $H\{y_k\}$.

d)

La probabilidad de cada patrón y_j , $j = 1, \dots, 3^5$, dado x_k será la suma de la probabilidad de cada palabra permitida $x_k \in X_k$ que cumple el mismo patrón, sobre la cantidad de todas las palabras posibles $x \in X$, es decir:

$$p(y_j) = \frac{\sum_{x_k \in X_k} p(x_k) \text{ tq } y_j = f(X, x_k)}{\#X} \quad (10)$$

Para cada patrón posible $p(y_j)$. Y la entropía de y_k para cada x_k será definida como

$$H\{y_k\} = - \sum_{y_k}^{Y_k} p(y_k) \log_2(p(y_k)) \quad (11)$$

Tomando $y_j = y_k$ en la ecuación anterior.

e)

Tomando el primer intento con la palabra crane, se obtiene:

- Puntaje promedio maximizando entropía y sin considerar la frecuencia de las palabras: 3.94.

Considerando la frecuencia del uso de las palabras en el idioma inglés: 3.63.

- Puntaje promedio sin maximizar entropía y sin considerar la frecuencia de las palabras: 4.72.

Considerando la frecuencia del uso de las palabras en el idioma inglés: 4.73.

El promedio de puntajes en wordle es 3.97. Por lo que, sin maximizar la entropía (y empezando con la palabra crane) el resultado es un poco peor. Maximizando la entropía, el resultado mejora.

3. Parte III: Compresión de la fuente

a)

El código de Huffman obtenido no se puede considerar un código óptimo dado que para alcanzar un largo promedio igual a la entropía es necesario que el alfabeto sea uniformemente probable, sin embargo, dado que estamos trabajando con palabras salidas de idiomas, se tienen combinaciones de letras más probables que otras. Por ejemplo, en el caso del inglés, las letras que tienen más probabilidad son *E*, *T* y *A*.

b)

En esta sección se busca comparar la entropía de diferentes archivos considerando diferentes extensiones de fuente, en la tabla siguiente se puede ver un resumen de dichas cantidades.

Cantidad de extensión	Entropía por bit		
	adams.txt	mensaje1.txt	mensaje2.txt
1	0.99372	0.81127	0.99986
2	0.97827	0.51320	0.99712
4	0.84254	0.26320	0.97332

Tabla 1: Entropías por bit para cada codificación de archivo.

La justificación para la diferencia en las entropías se puede ilustrar en la [Figura 3](#) en donde se puede ver que para el caso de *mensaje2.txt* los símbolos estaban cerca de ser equiprobables, lo que explica su poca variación en la entropía debido a que la condición de mayor entropía se da con la equiprobabilidad de sus símbolos.

En el caso de *mensaje1.txt* se nota una clara diferencia en probabilidades entre sus símbolos, por lo tanto se tiene una menor incertidumbre y esto genera una baja entropía.

Y finalmente para *adams.txt* se ve una distribución de probabilidad que no es enteramente equiprobable, pero que tampoco presenta una diferencia de probabilidad tan marcada como *mensaje1.txt*, lo cual justifica que haya sufrido una disminución de la entropía mejor que *mensaje2.txt* pero peor que *mensaje1.txt*.

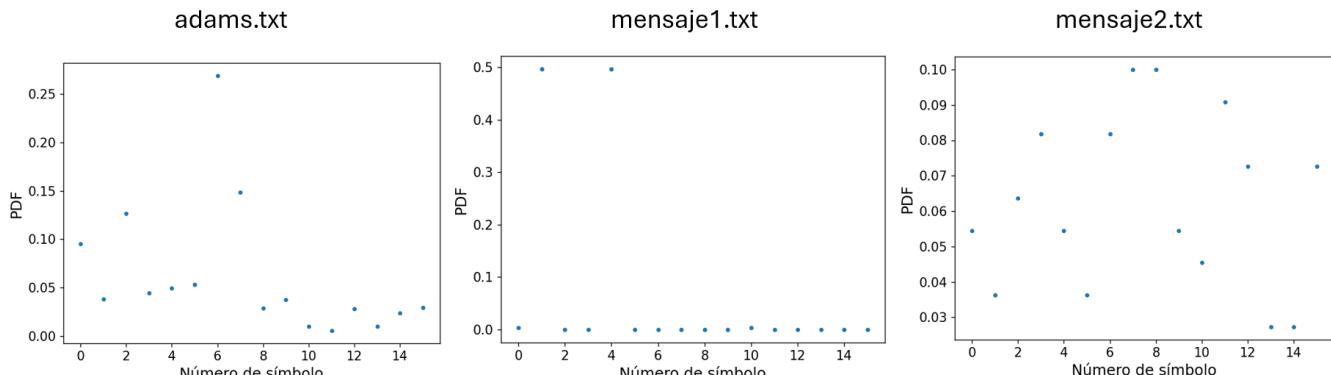


Figura 3: PDF de símbolo luego de haber sido extendido 4 veces.

c)

Al ordenar los códigos por largo de símbolo se puede ver lo siguiente.

Para *mensaje1.txt* se logra observar un aumento desigual en la asignación de bits, esto se debe principalmente a que cuando se intenta priorizar los más probables, en este caso, se tienen muy pocos símbolos con probabilidad alta y la mayoría con probabilidad baja, haciendo que en cada nuevo símbolo su codificación agregue un nuevo bit. En el caso particular de ampliar la fuente 4 veces, se ve que el símbolo más probable tiene 1 bit y el menos probable 15 bits.

En el caso de *adams.txt* y *mensaje2.txt* se puede ver una distribución de bits más equitativa dado que se tienen una mayor diversidad de probabilidades. Si se considera el caso de ampliar la fuente 4 veces se puede ver que *adams.txt* tiene un rango de bits que va de 2 a 7 mientras que para *mensaje2.txt* el rango va de 3 a 5.

d)

En la [Tabla 2](#) se presenta la tasa de compresión obtenida con diferentes herramientas. Como se puede observar, el resultado del código Huffman varía con respecto al utilizado con otras herramientas, esto se debe principalmente a que tanto WinRAR como el explorador de archivos de windows utilizan el código Lempel-Ziv para hacer las compresiones.

La principal diferencia entre ambos tipos de codificación es que Lempel-Ziv no depende de la probabilidad de cada símbolo sino que realiza una codificación símbolo a símbolo de forma secuencial. Para una comparación más en detalle de ambos métodos se puede ver [\[1\]](#).

	Tasa de compresión		
	Código extendido 4 veces	WinRAR	Explorador de windows
adams.txt	0.1519	0.3967	0.3976
mensaje1.txt	0.6220	0.4491	0.2275
mensaje2.txt	0.0181	-1.3818	-2.218

Tabla 2: Tasas de compresión para los distintos archivos utilizando distintas herramientas.

4. Parte IV: Codificación para corrección de errores

4.1. Ejercicio 9.4

a)

Teniendo como palabra recibida $r = (110110)$ y como matriz de paridad H de la forma:

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \implies H^T = \begin{pmatrix} 111 \\ 110 \\ 001 \\ 101 \\ 010 \\ 100 \end{pmatrix}$$

Para obtener la palabra de código enviada es necesario obtener el síndrome del código, esto se obtiene como $rH^T = s$. Sabiendo entonces la forma de H^T se obtiene que el síndrome es $s = (110)$, y dado que por letra existe un único bit erróneo, el error se encuentra en la fila que posee ese mismo valor en la matriz H^T . Por lo tanto se obtiene que la palabra enviada originalmente fue $x = (100110)$.

b.1)

Dado las dimensiones de la matriz generadora, se sabe que existen 2^4 palabras posibles, por lo tanto multiplicando todos los posibles vectores de 4 bits por la matriz generadora se obtienen las palabras de la [Tabla 3](#)

Palabra a codificar	Palabras de código
0000	0000000
0001	1100001
0010	0110010
0011	1010011
0100	1010100
0101	0110101
0110	1100110
0111	0000111
1000	0101100
1001	1001101
1010	0011110
1011	1111111
1100	1111000
1101	0111001
1110	1001010
1111	0101011

Tabla 3: Palabras de código generadas a partir de la matriz generadora.**b.2)**

Al analizar la [Tabla 3](#) se puede comprobar que la distancia mínima de código, sin ser el vector nulo, es 3. Estando esta en los códigos 1100001, 0110010, 1010100, 0000111, 0101100 y 1001010.

b.3)

En este caso, para obtener la matriz de paridad a partir de una matriz generadora que no se encuentra en su forma sistemática, se aplica el hecho de que $GH^T = 0$, G y H^T definidas en [\(12\)](#)

$$G = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad H^T = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \\ h_{41} & h_{42} & h_{43} \\ h_{51} & h_{52} & h_{53} \\ h_{61} & h_{62} & h_{63} \\ h_{71} & h_{72} & h_{73} \end{pmatrix} \quad (12)$$

Realizando las operaciones entre matrices se obtienen las relaciones siguientes $\forall i = 1, 2, 3$:

$$\begin{aligned} h_{2i} + h_{4i} + h_{5i} &= 0 \\ h_{1i} + h_{3i} + h_{5i} &= 0 \\ h_{2i} + h_{3i} + h_{6i} &= 0 \\ h_{1i} + h_{2i} + h_{7i} &= 0 \end{aligned}$$

Luego se buscó una posible matriz H^T cuyos coeficientes cumplieran las ecuaciones anteriores y se la transpuso para obtener la matriz de paridad buscada.

$$H^T = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \implies H = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

b.4)

La obtención del síndrome se realizará de forma análoga a lo realizado en secciones anteriores, tomando en este caso en recepción $R = (1101011)$ se tiene un síndrome $s = (001)$ por lo tanto, asumiendo que existe un único error de bit, se deduce que el mensaje original era $x = (0101011)$.

b.5)

Para transformar la matriz G (12) en su versión sistemática G_s la idea es sumar la cuarta columna con la quinta, obteniendo así que las columnas 4 a 7 formen la identidad, luego esas columnas fueron movidas hacia la izquierda de tal forma que $G_s = [I_k|P]$ siendo P el bloque formado con las columnas restantes como se puede ver en (13).

$$G_s = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} \Rightarrow P = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \quad (13)$$

Sabiendo lo anterior, se puede obtener la matriz de paridad paridad H como $H = [P^T | I_{n-k}]$, el resultado se puede ver en (14)

$$H = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (14)$$

Si bien no es algo común, los códigos de bloques también pueden ser representados como polinomios, en este caso se buscará representar a la matriz de paridad calculada anteriormente como un filtro. Para hacer esto lo primero es notar que la matriz de paridad en formato polinomial tiene la forma $H_p = [x, 1 + x^2, x + x^2, 1 + x, 1, x, x^2]$.

Sea entonces $c_i^{(k)} = m_i H_p \forall k = 1 \dots 7$ las distintas salidas del filtro, m_i la secuencia de valores de entrada y c_i la secuencia final de salida. Entonces el filtro obtenido se puede visualizar en la Figura 4, en donde se utilizaron los bloques D para representar el retardo de 1 bit en los datos.

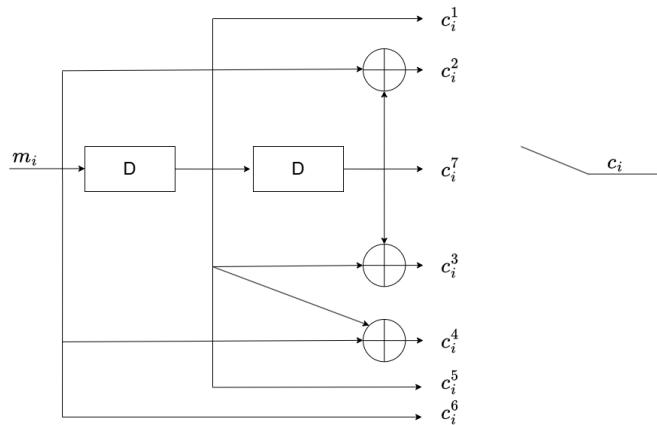
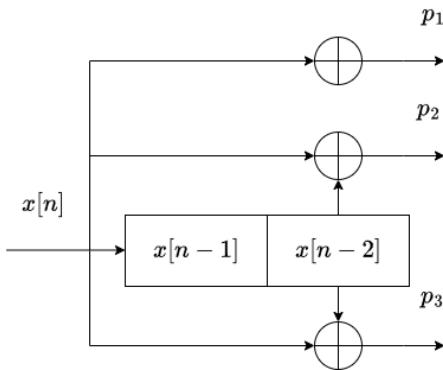


Figura 4: Implementación de la matriz de paridad como un filtro.

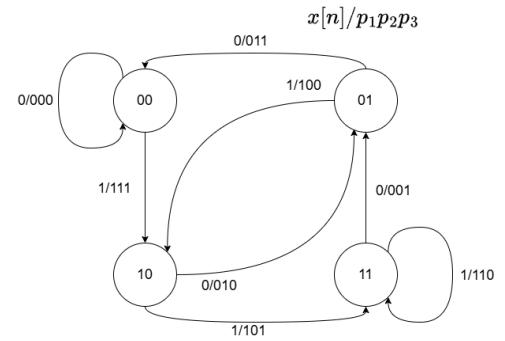
4.2. Ejercicio 9.6

a.1)

A partir de la función de transferencia en transformada-D: $[1, 1 + D + D^2, 1 + D^2]$ se puede deducir que el sistema va a poseer dos registros que permitan almacenar los valores de la entrada hasta $x[n - 2]$. Con esto en mente se realizaron las representaciones correspondientes considerando que $p_1 = x[n]$, $p_2 = x + x[n - 1] + x[n - 2]$ y $p_3 = x[n] + x[n - 2]$, las representaciones como shift register y diagrama de estados se pueden ver en la Figura 5.



(a) Representación como shift register.

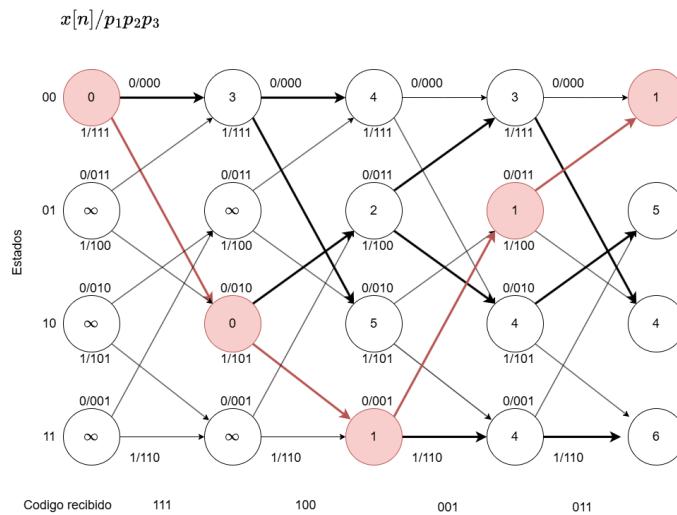


(b) Representación como diagrama de estados.

Figura 5: Distintas representaciones de un mismo código convolucional.**a.2) & a.3)**

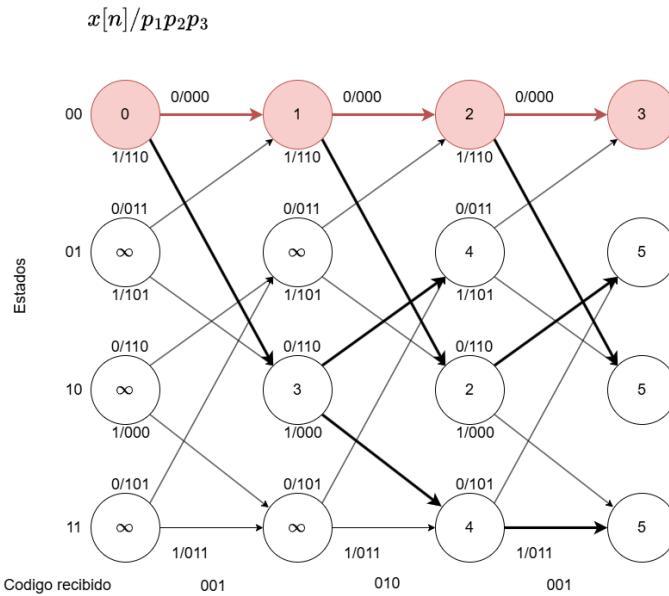
A partir de los diagramas realizados anteriormente, se puede realizar un diagrama de Trellis como se puede ver en la [Figura 6](#), en donde a su vez se detallan los caminos seguidos utilizando el algoritmo de Viterbi para obtener el mensaje original al haber recibido la secuencia 111, 100, 001, 011.

Analizando el resultado se puede ver que la secuencia obtenida tuvo un error en el segundo valor de la secuencia, teniendo que ser este 101 en lugar de 100. De esta forma se puede ver que el mensaje enviado antes de la codificación fue $x[n] = 1100$.

**Figura 6:** Diagrama de Trellis del sistema.**b.1) & b.2) & b.3)**

En la [Figura 7](#) se puede ver el diagrama de Trellis para el código propuesto, en donde a su vez se detallan los caminos seguidos utilizando el algoritmo de Viterbi para obtener el mensaje original al haber recibido la secuencia 001, 100, 001, 011.

Analizando el resultado se puede ver que se obtuvieron tres errores en el código recibido, asumiendo que se siguiera el camino indicado, la secuencia enviada más probable sería 000, 000, 000, siendo la palabra enviada original $x[n] = 0000$. Luego como un camino siguiente más cercano se puede elegir cualquiera de los otros partiendo de 00 dado que al llegar al final de la secuencia todos se encuentran a la misma distancia de Hamming, siendo esta mayor a la obtenida si se va por el camino elegido anteriormente.

**Figura 7:** Diagrama de Trellis del sistema.

4.3. Ejercicio 9.7

a)

Para el caso de este archivo, no se tiene un sistema de corrección de errores, únicamente se posee un decodificador, en la [Tabla 4](#) se puede ver la variación del BER para distintos valores de SNR.

SNR	5	4	3	2	1	0	-1	-2	-3	-4	-5
BER	-37.929783	-3.6	-2.5	-1.7	-1.26	-0.96	-0.76	-0.609	-0.52	-0.46	-0.4

Tabla 4: BER para archivo *ejercicio_7_a.grc*.

b)

En este caso el código trae implementada una corrección de errores al generar un diagrama de Trellis de la constelación de llegada y aplicar el algoritmo de Viterbi para encontrar la representación más probable del mensaje original. En la [Tabla 5](#) se puede ver la variación del BER para distintos valores de SNR y es posible notar una mejoría respecto al caso de la [Tabla 4](#).

SNR	5	4	3	2	1	0	-1	-2	-3	-4	-5
BER	-37.929783	-37.929783	-6	-4	-2.8	-1.9	-1.33	-0.966	-0.7	-0.606	-0.504

Tabla 5: BER para archivo *ejercicio_7_b.grc*.

Una comparación más gráfica de la [Tabla 4](#) y [Tabla 5](#) se puede ver en la [Figura 8](#) en donde se puede ver claramente como a medida que disminuye el SNR, el archivo que utilizó corrección de errores notó una menor disminución del BER del sistema.

c)

Al analizar el archivo que no poseía bloques correctores de errores fue posible detectar la presencia del ruido a alrededor de los -6.5 dB mientras que para el que posee corrección se comenzó a detectar a partir de -3.2 dB. Esto muestra que cuando se utiliza un algoritmo de corrección de errores, el ruido tiene que tener el doble de potencia respecto a la versión sin corrección para que pueda ser sentido por el humano.

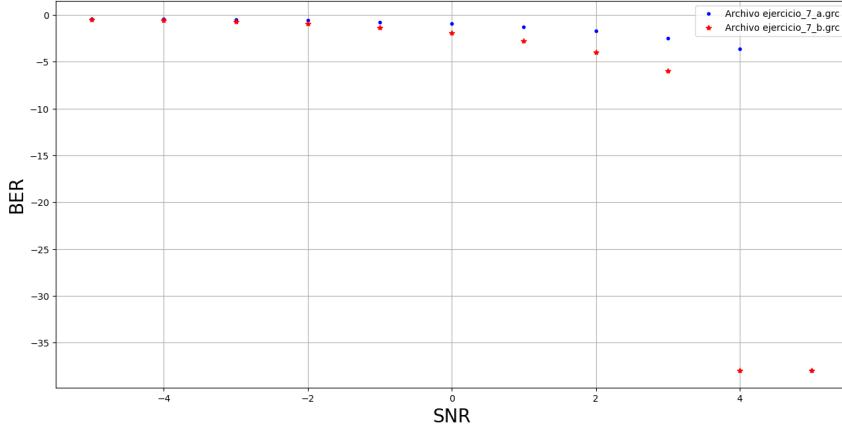


Figura 8: Gráfica de los BER de cada archivo en función del SNR.

5. Anexo

5.1. Parte I: f

A continuación se enlistarán las cuentas más detalladas de los resultados de la sección f de la parte I, ordenados según su orden de aparición.

$$H\{Y\} = -P_{x_0} \log_2(P_{x_0}) - P_e \log_2(P_e) - P_{x_1} \log_2(P_{x_1})$$

Siendo P_{x_0} , P_e y P_{x_1} las probabilidades de que en la salida se vea x_0, e y x_1 respectivamente. Estas cantidades se calculan de la siguiente forma:

$$\begin{aligned} P_{x_0} &= P(Y = x_0) = a(1 - q - p) + (1 - a)q \approx a(1 - p) \\ P_e &= P(Y = e) = p \\ P_{x_1} &= P(Y = x_1) = aq + (1 - a)(1 - p - q) \approx (1 - a)(1 - p) \end{aligned}$$

En donde la aproximación sale de despreciar el término q frente a el resto de parámetros.

Con estos parámetros obtenidos se puede obtener $H\{Y\}$ de la siguiente forma:

$$\begin{aligned} H\{Y\} &= -a(1 - p) \log_2(a(1 - p)) - p \log_2(p) - (1 - a)(1 - p) \log_2((1 - a)(1 - p)) \\ &= a(1 - p)[\log_2(\frac{1}{a}) + \log_2(\frac{1}{1 - p})] + p \log_2(\frac{1}{p}) + (1 - a)(1 - p)[\log_2(\frac{1}{1 - a}) + \log_2(\frac{1}{1 - p})] \\ &= \Omega(a)(1 - p) + a(1 - p) \log_2(\frac{1}{1 - p}) + p \log_2(\frac{1}{p}) + (1 - a)(1 - p) \log_2(\frac{1}{1 - p}) \\ &= \Omega(a)(1 - p) + (1 - p) \log_2(\frac{1}{1 - p}) + p \log_2(\frac{1}{p}) \\ &= \Omega(a)(1 - p) + \Omega(p) \quad \square \end{aligned}$$

$$H\{Y|X\} = P(X = x_0)H\{Y|X = x_0\} + P(X = x_1)H\{Y|X = x_1\}$$

En donde se cumple que al despreciar q las entropías condicionales tienen los siguientes valores:

$$\begin{aligned} H\{Y|X = x_0\} &= -(1 - p) \log(1 - p) - p \log_2(p) = \Omega(p) \\ H\{Y|X = x_1\} &= -(1 - p) \log(1 - p) - p \log_2(p) = \Omega(p) \end{aligned}$$

Entonces finalmente se tiene que:

$$H\{Y|X\} = a\Omega(p) + (1 - a)\Omega(p) = \Omega(p) \quad \square$$

$$H\{X|Y\} = H\{Y|X\} + H\{X\} - H\{Y\} = \Omega(p) + \Omega(a) - \Omega(a)(1 - p) - \Omega(p) = \Omega(a)p \quad \square$$

$$I\{X, Y\} = H\{Y\} - H\{Y|X\} = \Omega(a)(1 - p) + \Omega(p) - \Omega(p) = \Omega(a)(1 - p) \quad \square$$

Bibliografía

- [1] R. A. Bedruz and A. R. Quiros, “Comparison of huffman algorithm and lempel-ziv algorithm for audio, image and text compression,” 12 2015, pp. 1–6.