

Probabilidad de error y Capacidad de un canal discreto

Se supone un canal discreto cuya entrada dependerá de los datos proporcionados por el usuario. Se le pide ingresar la amplitud de la señal que será modulada así los niveles de cuantificación N ($\#\Omega$) para poder calcular el alfabeto Ω .

En este código se generan 50,000 muestras de la señal cuyos resultados serán simulados promediados sobre 1000 intentos.

```

1  clc
2  clear
3  close all
4
5  samples= 50000;
6  N = input('Ingrese cardinalidad del alfabeto: ');
7  amplitude = input('Ingrese amplitud de la senial: ');
8  alphabet = zeros(1,N);
9  slot = amplitude/(N-1);
10 umbral=slot/2;
11 for i=2:N
12     alphabet(i) = alphabet(i-1)+ slot ;
13 end
14
15 fprintf(['\nIntervalo q   = ' num2str(slot) ' V \n\n' ] )
16 fprintf(['El alfabeto es:  {' num2str(alphabet) '} \n' ])
17 x= randsrc(1,samples ,alphabet);

```

Fuente equiprobable

Apartir de una función de probabilidad equiprobable, se genera la secuencia x de entrada del canal y se calcula la entropía $H(X)$ y la potencia de la señal σ_x .

```

1  % ----- FDP DE X -----
2  fdp_x=ones(1,N)./N;
3  x= randsrc(1,samples ,[alphabet;fdp_x]);
4
5
6  % ENTROPIA DE ENTRADA
7  H_input = -sum(fdp_x.*log2(fdp_x));

```

Las siguientes líneas son ajustables directamente en el código, ya que determinan la calidad de la simulación en función del rango de SNR que se desea visualizar en las gráficas, así como el número de intentos en la simulación de Montecarlo.

```

1  %-----
2  %VALORES DINAMICOS                               %|
3  SNR_range=-20:20;                                %|
4  SNR_vector_length = length(SNR_range);           %|
5  montecarlos_attempts=100;                         %|
6  errors = zeros(1,montecarlos_attempts);          %|
7  %-----

```

Antes de comenzar con la parte fuerte del código, se deben inicializar correctamente algunas variables:

- `index` : Índice para almacenar valores en las celdas que contienen matrices o arreglos en función de la SNR.
- `error_probability` : vector fila de probabilidades de error.
- `H_output` : vector columna de entropías de salida $H(Y)$
- `I_M=` : vector fila para la Información mutua o Capacidad del Canal.
- `P_matrixes_array` : celda que contiene las matrices P del canal.
- `P_outs_array` : celda que contiene vectores columna de las probabilidades de salida $P(Y)$
- `P_outsC_array` : celda que contiene matrices de tamaño $N \times N$ para las probabilidades condicionales o A posteriori $P(X|Y)$

```

1  index=1;
2
3  error_probability =zeros(1,SNR_vector_length);
4  H_output=zeros(SNR_vector_length,1);
5  I_M=zeros(1,SNR_vector_length);
6
7  P_matrixes_array = cell(1, SNR_vector_length);
8  P_outs_array = cell(1,SNR_vector_length);
9  P_outsC_array = cell(1,SNR_vector_length);
10
11
12  for i = 1:SNR_vector_length
13      P_matrixes_array{i} = zeros(N); % Almacenar la matriz en la celda del arreglo
14      P_outs_array{i} = zeros(N,1);
15  end
16

```

Para cada iteración del rango de SNR se inicializa

```

1  for SNR = SNR_range
2
3      P_matrix=zeros(N);
4      P_out = zeros(N,1);
5      P_out_conditional=zeros(N);
6      P_apriori = zeros(N,1);

```

- `P_matrix` : matriz P del canal.
- `P_out` : vector columna de probabilidades de salida $P(Y)$
- `P_out_conditional` : Matriz de probabilidades a posteriori
- `P_apriori` Probabilidades a priori

El siguiente ciclo `for` realiza las 1000 simulaciones para promediar el número de errores de acuerdo análisis de la salida $y = x + n$. Se decodifica a partir del establecimiento de un umbral en las primeras líneas, por convención, es la mitad del intervalo q de cuantificación.

```

1 for SNR = SNR_range
2     P_matrix=zeros(N);
3     P_out = zeros(N,1);
4     P_out_conditional=zeros(N);
5     P_apriori = zeros(N,1);
6
7     for k=1:montecarlos_attempts
8         power_n = power_x*10^(-SNR/10);
9         noise = randn(1,samples).*sqrt(power_n);
10        y=x+noise;
11
12        % ----- UMBRAL -----
13        y_decode = zeros(1,samples);
14        for i=1:samples
15            if y(i)<= umbral
16                y_decode(i)= alphabet(1);
17            elseif y(i)>= alphabet(N-1)+umbral
18                y_decode(i)= alphabet(N);
19            elseif y(i) < alphabet(1)
20                y_decode(i)= alphabet(1);
21            elseif y(i) > alphabet(N)
22                y_decode(i)= alphabet(N);
23            else
24                for j=2:N-1
25                    if alphabet(j) - umbral <= y(i) && y(i) < alphabet(j)+umbral
26                        y_decode(i)= alphabet(j);
27                    end
28                end
29            end
30        end
31
32        %----- CONTEO DE ERRORES -----
33
34        errors(k) = sum ~(y_decode==x);
35
36    end

```

Después de decodificar la salida, se puede construir la matriz del canal mediante el uso de tres ciclos for anidados. En primer lugar, se llenarán las filas y luego las columnas de la matriz. Para hacerlo, se pueden extraer las posiciones en `y_decode` donde se encuentran los símbolos del alfabeto deseado y guardarlos en un vector fila para posteriormente compararlos y contabilizar cuántos de ellos había en la entrada. En la línea 13 se dividen las filas entre el número de ocurrencias de cada símbolo para calcular sus probabilidades.

```

1 %----- Primer metodo P(Y|X) -----
2
3 for i=1:N
4     matrix_row = find(x== alphabet(i));
5     num_symbols_out = length(matrix_row);
6     for j=1:num_symbols_out
7         for k=1:N
8             if(y_decode(matrix_row(j)) == alphabet(k))
9                 P_matrix(i,k) = P_matrix(i,k) + 1;
10            end
11        end
12    end
13    P_matrix(i,:)=P_matrix(i,:)./num_symbols_out;
14 end

```

```

15
16 P_matrixes_array{index}= P_matrix;

```

De acuerdo con el sistema de ecuaciones para obtener el conjunto de probabilidades de salida:

$$\begin{aligned}
 P(x = \Omega_1)P(y = \Omega_1|x = \Omega_1) + P(x = \Omega_2)P(y = \Omega_1|x = \Omega_2) + \dots + P(x = \Omega_N)P(y = \Omega_1|x = \Omega_N) &= P(y = \Omega_1) \\
 P(x = \Omega_1)P(y = \Omega_2|x = \Omega_1) + P(x = \Omega_2)P(y = \Omega_2|x = \Omega_2) + \dots + P(x = \Omega_N)P(y = \Omega_2|x = \Omega_N) &= P(y = \Omega_2) \\
 \vdots & \\
 P(x = \Omega_1)P(y = \Omega_N|x = \Omega_1) + P(x = \Omega_2)P(y = \Omega_N|x = \Omega_2) + \dots + P(x = \Omega_N)P(y = \Omega_N|x = \Omega_N) &= P(y = \Omega_N)
 \end{aligned}$$

Se implementa en Matlab como:

```

1 %PROBABILIDADES DE SALIDA
2     for i=1:N
3         P_out(i) = sum(fdp_x(i)'.* P_matrix(:,i)');
4     end

```

Mediante el teorema de Bayes se calcula la probabilidad hacia atrás y se guarda en el vector P_outsC_array.

```

1 %PROPABILIDADES CONDICIONALES
2
3     for i=1:N
4         for j=1:N
5             P_out_conditional(i,j) = (P_matrix(i,j)'.*fdp_x(j))./ P_out(j);
6         end
7     end
8
9     P_outs_array{index} = P_out';
10    P_outsC_array{index} = sum(P_out_conditional');
11

```

Finalmente se calcula la entropía hacia atrás y se hace la resta de $H(A) - H(A|B)$ para obtener la capacidad del canal:

```

1
2 %ENTROPIA CONDICIONAL
3 H_AB = -sum(P_outsC_array{index}.*log2(P_outsC_array{index}));
4
5
6 % %-----Para el segundo metodo P(X|Y) -----
7 % for i=1:N
8 %     matrix_row = find(y_decode== alphabet(i));
9 %     num_symbols_out = length(matrix_row);
10 %     for j=1:num_symbols_out
11 %         for k=1:N
12 %             if(x(matrix_row(j)) == alphabet(k))
13 %                 P_matrix(i,k) = P_matrix(i,k) +1;
14 %             end
15 %         end
16 %     end
17 %     P_matrix(i,:)=P_matrix(i,:)./num_symbols_out;
18 % end
19 % P_matrixes_array2{index}= P_matrix;
20 % %-----
21

```

```

22
23 I_M(index) = H_input - H_AB;
24
25 index=index+1;
26 end

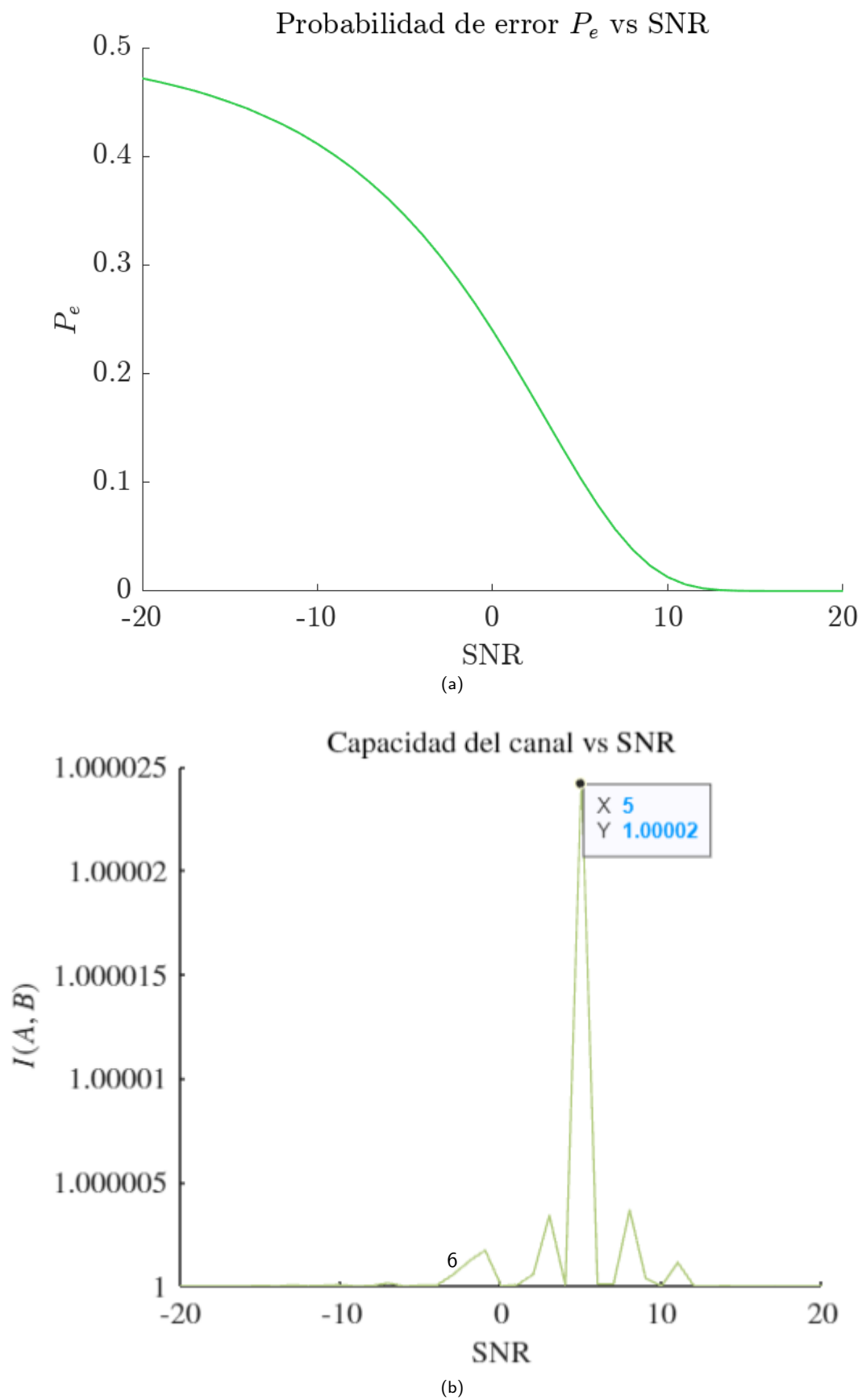
```

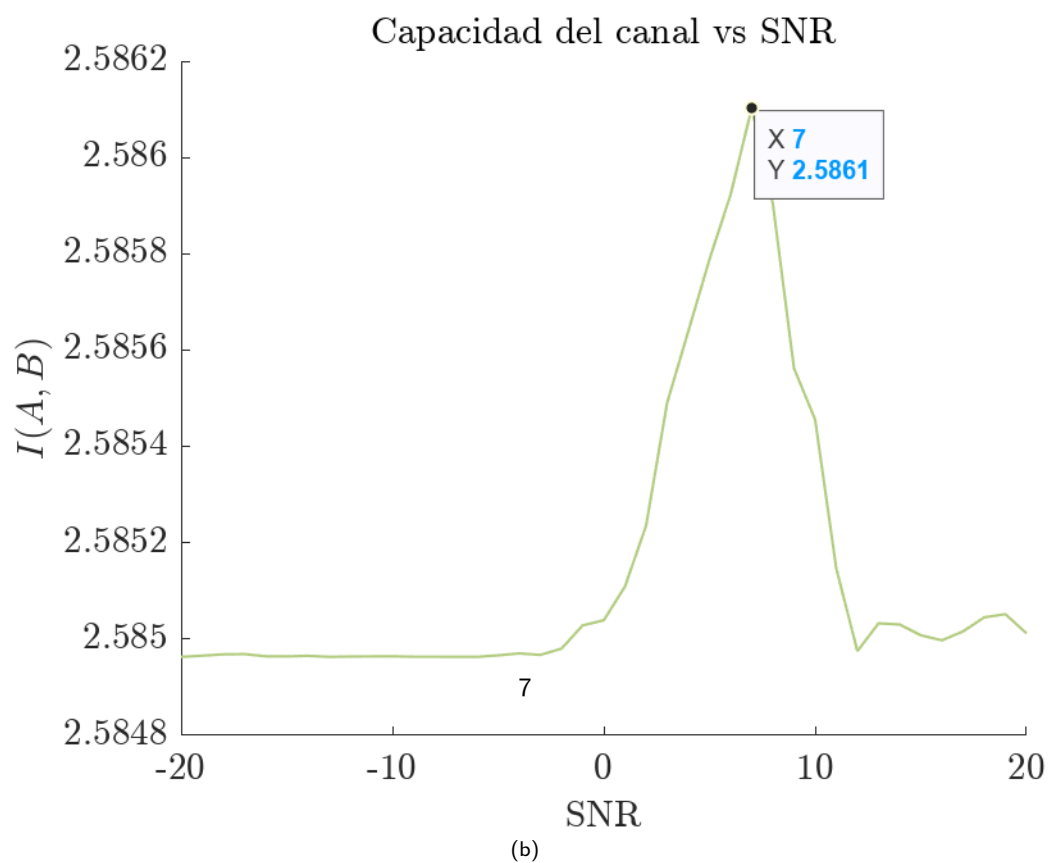
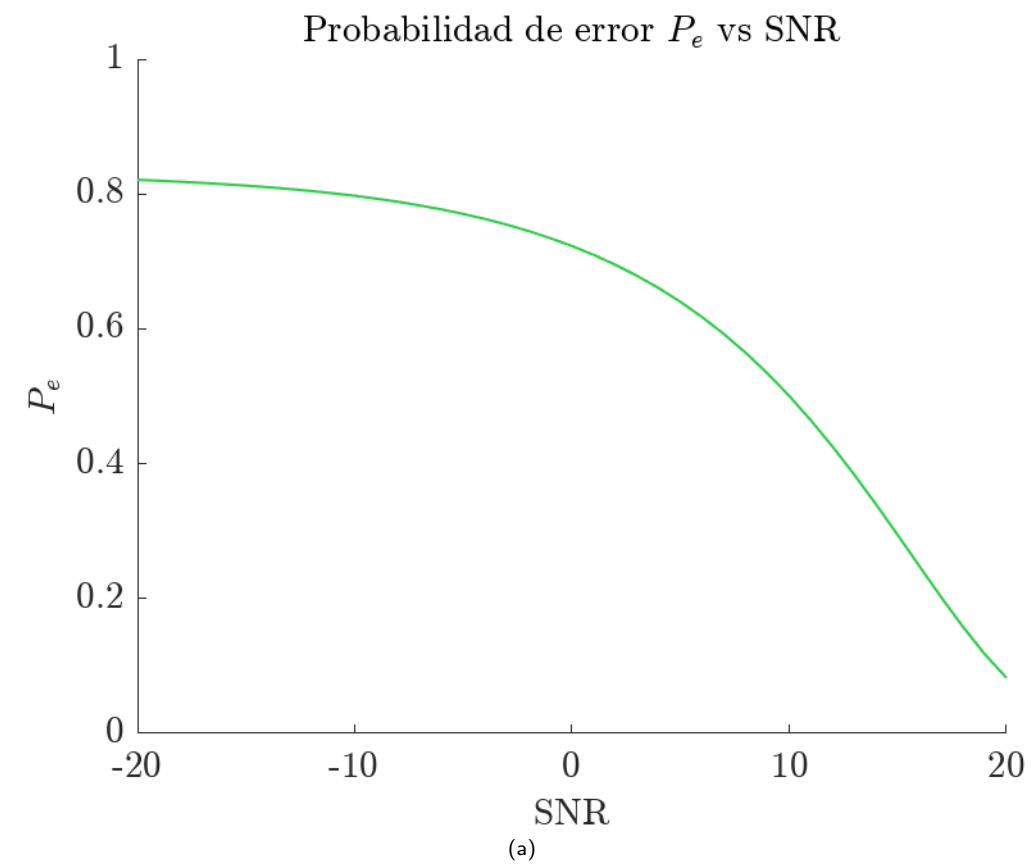
El resto del código son instrucciones para graficar la P_e y la Capacidad del canal vs SNR y darles formato.

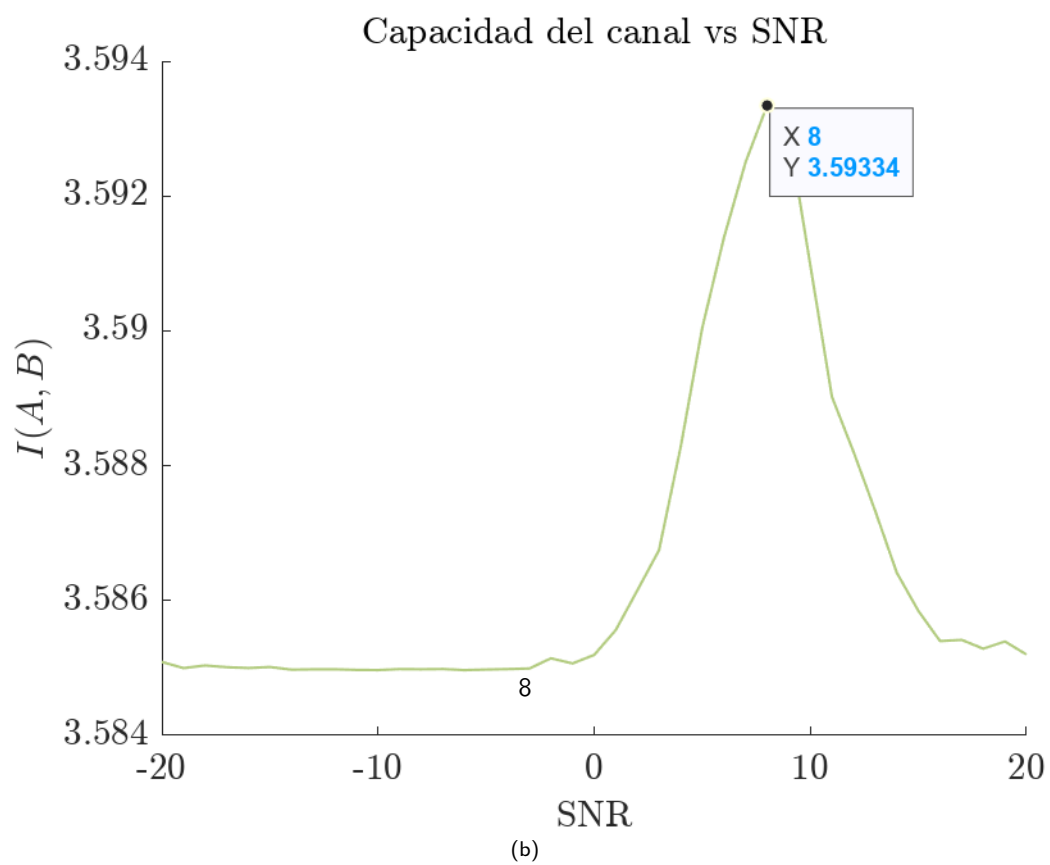
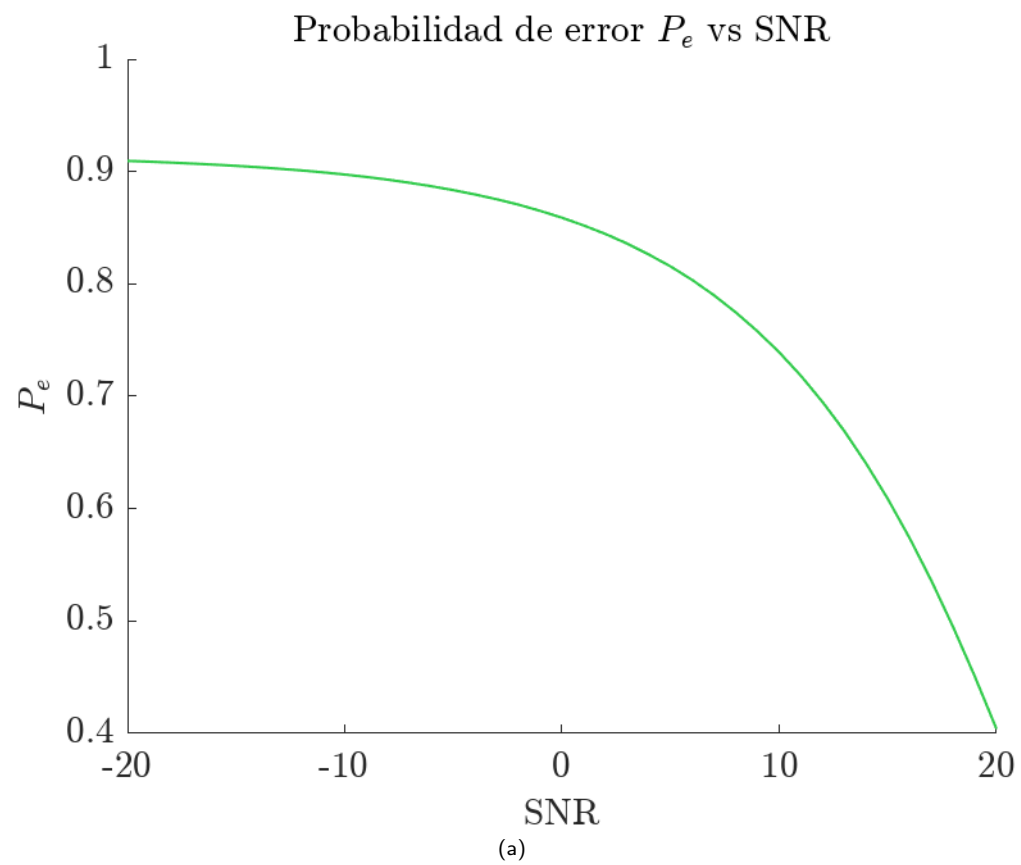
```

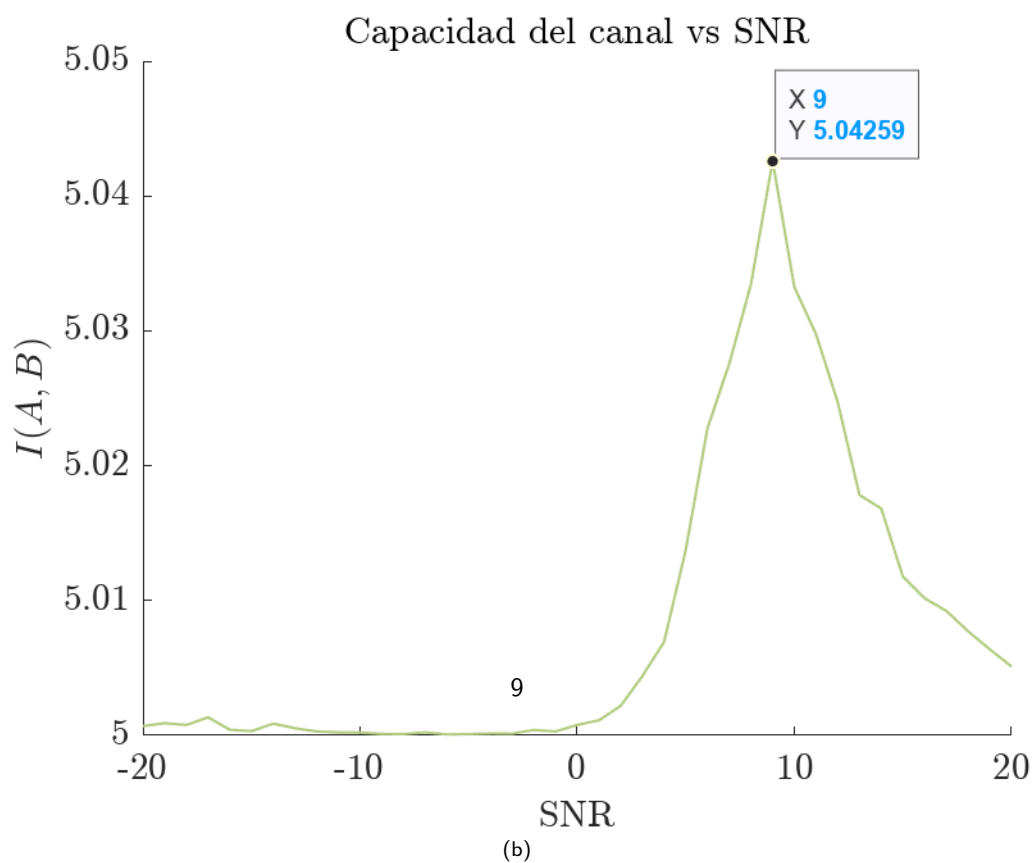
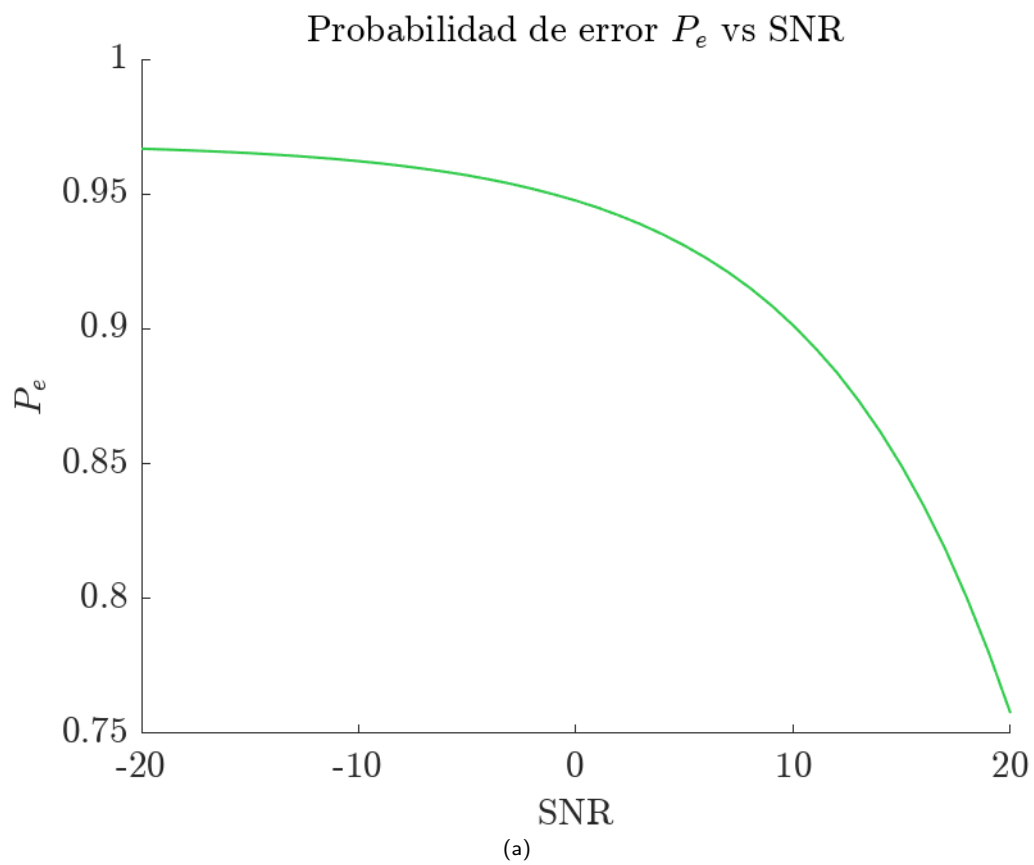
1 T=table(SNR_range',error_probability');
2 T.Properties.VariableNames = ["SNR","Probabilidad de error"]
3
4 hfig = figure;
5 figure(1)
6 plot(SNR_range,error_probability,'LineWidth',1,'Color', [0.2, 0.8, 0.3]);
7 xlabel('SNR')
8 ylabel('$P_e$')
9 title('Probabilidad de error $P_e$ vs SNR')
10 figure_style(hfig)
11
12 hfig = figure;
13 figure(2)
14 stem(x,'filled', 'LineWidth', 3)
15 hold on
16 stem(y,'filled','LineWidth',2)
17 axis([samples/2 (samples/2)+50 -0.5 amplitud+1 ])
18 ylabel('$\Omega(S_i)$')
19 xlabel('$S_i$')
20 title('Entrada X vs salida Y')
21 figure_style(hfig)
22
23 hfig = figure;
24 figure(3)
25 plot(SNR_range,I_M,'LineWidth',1,'Color', [0.7, 0.8, 0.5]);
26 xlabel('SNR')
27 ylabel('$I(A, B)$')
28 title('Capacidad del canal vs SNR')
29 figure_style(hfig)
30
31
32
33 %----- CODIGO PARA FIGURAS -----
34
35 function figure_style(hfig)
36 picturewidth = 20; % set this parameter and keep it forever
37 hw_ratio = 0.65; % feel free to play with this ratio
38 set(findall(hfig,'-property','FontSize'),'FontSize',14) % adjust fontsize to your
    document
39
40 set(findall(hfig,'-property','Box'),'Box','off') % optional
41 set(findall(hfig,'-property','Interpreter'),'Interpreter','latex')
42 set(findall(hfig,'-property','TickLabelInterpreter'),'TickLabelInterpreter','latex
    ')
43 end

```

Figura 1: Señal de $A=5V$ modulada a 2 Niveles

Figura 2: Señal de $A=5V$ modulada a 6 Niveles



Figura 3: Señal de $A=9V$ modulada a 32 Niveles

Resultados

Conclusiones

De las gráficas presentadas puedo concluir lo siguiente: es evidente que a cuanto mayor sea la relación SNR habrá menor probabilidad de que el canal cometa errores, sin embargo, se puede observar que hay un punto en donde la capacidad del canal es la máxima y esta no necesariamente aumenta con relación a la SNR; de hecho, empieza a decrecer. Al aumentar los niveles de cuantificación o en este caso la cardinalidad del alfabeto, mayor será la información pero también aumenta el tiempo de procesamiento para la computadora ya que hace análisis de 5000 muestras en iteraciones de 1000×41 . Parece que se estudiará próximamente cuál (y por qué) hay un valor de SNR ideal para el canal, en esta simulación parece estar entre los 5 y 10 dB para los valores de amplitud dados.