Relatório de Simulação

Karen dos Anjos Arcoverde

Introdução

Este trabalho apresenta quatro experimentos de simulação em comunicação digital, com o intuito de confrontar resultados numéricos e previsões teóricas: (i) Questão 1 – geração e plotagem de pulsos retangulares $\Pi\big((t-t_0)/T_b\big)$ e triangulares $\Lambda\big((t-t_0)/T_b\big)$ para diferentes T_b e t_0; (ii) Questão 2 – síntese de um pulso triangular $p(t) = \Lambda(t)$, análise de sua transformada de Fourier $P(f)$ e reconversão via transformada inversa; (iii) Questão 3 – geração de 10^6 amostras de variáveis aleatórias (uniforme em $[0, \pi]$, normal $\mathcal{N}(0, 1)$ e soma de duas gaussianas), estimativa empírica de PDF/CDF e comparação com funções teóricas; (iv) Questão 4 – simulação de um receptor BPSK em canal AWGN, obtenção da taxa de erro de símbolo (SER) para diferentes σ, e comparação com a probabilidade de erro teórica $P_e = Q\big((d/2)/\sigma\big)$.

Resultados Obtidos

Questão 1

Foram plotadas as formas dos pulsos para os seguintes casos:

- Caso (a): $T_b = 1$, $t_0 = 0{,}5$
- Caso (b): $T_b = 2$, $t_0 = 3$

As figuras abaixo ilustram os pulsos retangulares e triangulares obtidos.

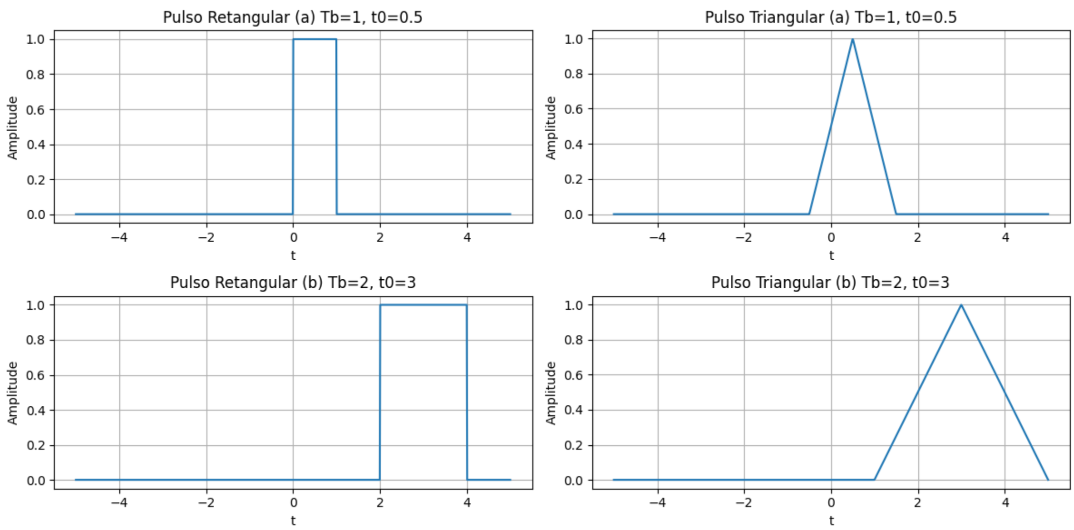
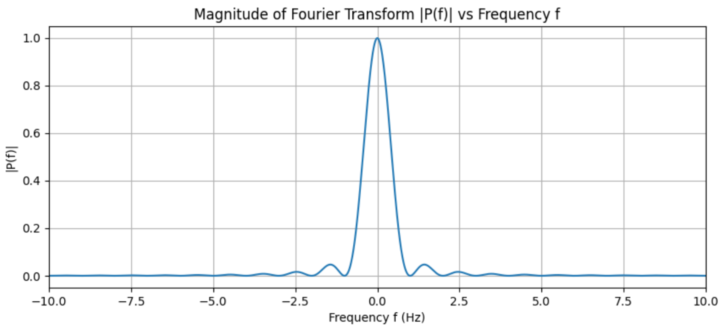


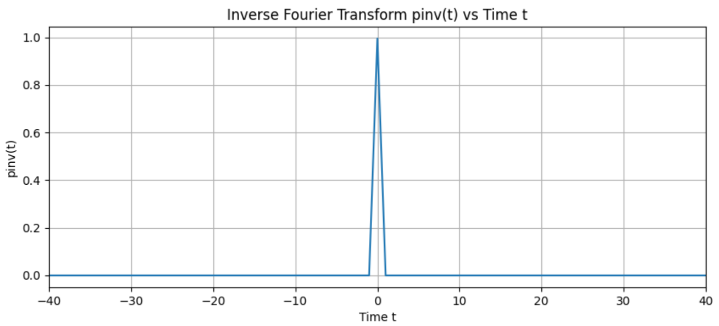
Figure 1: Pulsos Retangular e Triangular para os casos (a) e (b).

Observa-se que o pulso retangular mantém largura T_b e amplitude unitária, deslocado conforme t_0. Já o pulso triangular possui largura $2T_b$, com máximo unitário no centro do pulso e deslocado também de t_0.

Questão 2



(a) Magnitude da Transformada de Fourier $|P(f)|$ vs. f.



(b) Transformada Inversa $\mathrm{pinv}(t)$ vs. t.

Figure 2: (a) Espectro de magnitude da transformada de Fourier de um pulso triangular $p(t) = \Lambda(t)$. (b) Reconstrução do pulso via transformada inversa.

Na Figura 2a observamos que o espectro de magnitude $|P(f)|$ de um pulso triangular unitário segue o perfil de $\mathrm{sinc}^2(f)$:

$$P(f) = \mathcal{F}\{\Lambda(t)\} = \Big[\mathrm{sinc}(f)\Big]^2.$$

O lóbulo principal está centrado em $f = 0$ e possui largura aproximadamente igual a $2/T_b$ (no nosso caso $T_b = 1$, então zeros em $f = \pm 1, \pm 2, \ldots$ Hz). Os lóbulos laterais decaem com a lei $1/f^2$.

Já na Figura 2b, a transformada inversa reconstrói exatamente o pulso original

$$\Lambda(t) = \mathcal{F}^{-1}\{P(f)\},$$

confirmando a propriedade de dualidade e a correta implementação numérica.

Questão 3

Distribuição	Média empírica	Variância empírica
Uniforme $[0, \pi]$	1.5711	0.8235
Normal $N(0, 1)$	0.0004	0.9989
$Z = X + Y \sim N(0, 2)$	0.0018	2.0045

Table 1: Média e variância empíricas calculadas a partir de 10^6 amostras.

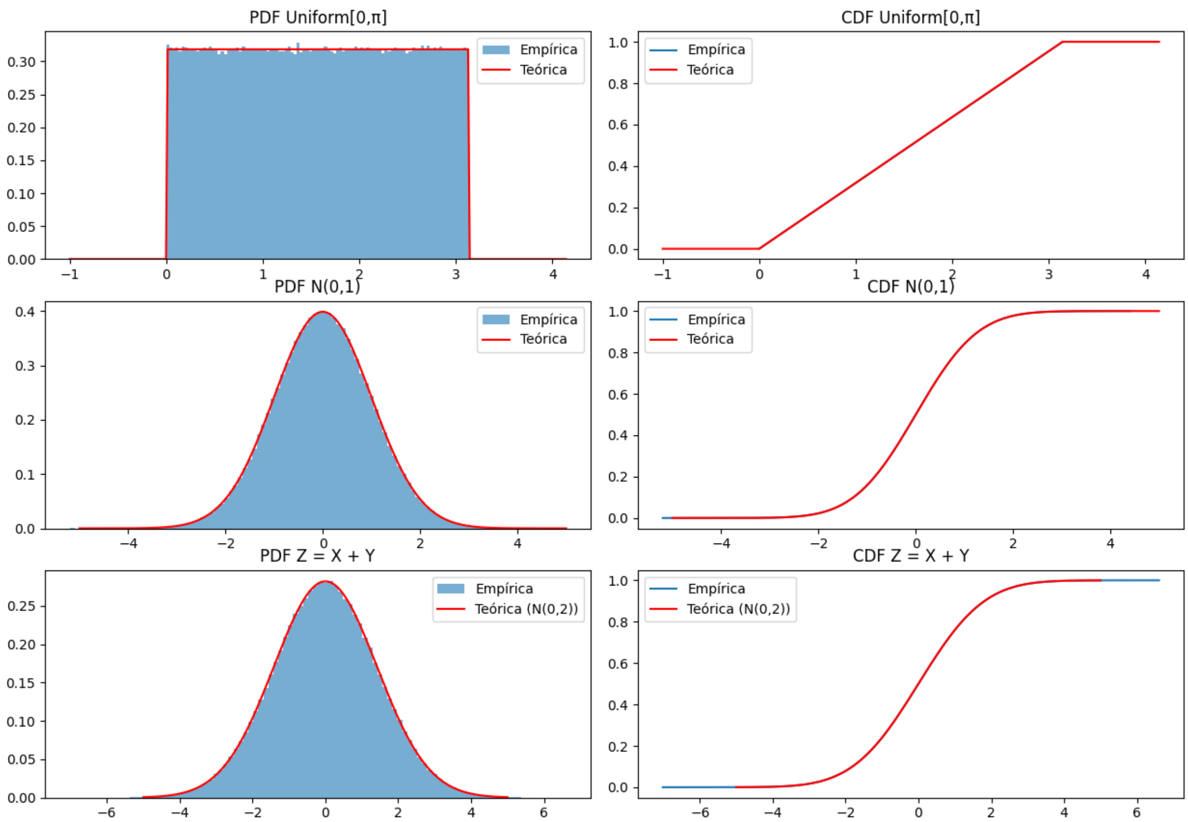


Figure 3: Estimativas empíricas (histograma/linha azul) e teóricas (linha vermelha) das PDFs e CDFs para as variáveis aleatórias simuladas: Uniforme em $[0, \pi]$ (linha superior), Normal $N(0, 1)$ (linha do meio) e $Z = X + Y \sim N(0, 2)$ (linha inferior).

Cálculo teórico do item 1

Para $X \sim \mathcal{U}(0, \pi)$, temos

$$f_X(x) = \begin{cases} \dfrac{1}{\pi}, & 0 \leq x \leq \pi, \\ 0, & \text{caso contrário.} \end{cases}$$

$$\mathbb{E}[X] = \int_0^{\pi} x\, f_X(x)\, dx = \int_0^{\pi} x\, \frac{1}{\pi}\, dx = \frac{1}{\pi}\left[\tfrac{x^2}{2}\right]_0^{\pi} = \frac{\pi^2}{2\pi} = \frac{\pi}{2},$$

$$\mathbb{E}[X^2] = \int_0^{\pi} x^2\, f_X(x)\, dx = \frac{1}{\pi}\left[\tfrac{x^3}{3}\right]_0^{\pi} = \frac{\pi^3}{3\pi} = \frac{\pi^2}{3},$$

$$\mathrm{Var}(X) = \mathbb{E}[X^2] - \big(\mathbb{E}[X]\big)^2 = \frac{\pi^2}{3} - \left(\frac{\pi}{2}\right)^2 = \frac{\pi^2}{3} - \frac{\pi^2}{4} = \frac{4\pi^2 - 3\pi^2}{12} = \frac{\pi^2}{12}.$$

Cálculo teórico do item 3

Sejam $X \sim N(0,1)$ e $Y \sim N(0,1)$ independentes. Definimos

$$Z = X + Y.$$

Então:

$$\mathbb{E}[Z] = \mathbb{E}[X] + \mathbb{E}[Y] = 0 + 0 = 0,$$

$$\begin{aligned}
\mathrm{Var}(Z) &= \mathrm{Var}(X+Y) \\
&= \mathrm{Var}(X) + \mathrm{Var}(Y) + 2\,\mathrm{Cov}(X,Y) \\
&= \mathrm{Var}(X) + \mathrm{Var}(Y) \quad (\mathrm{Cov}(X,Y) = 0, \text{ pois independentes}) \\
&= 1 + 1 = 2.
\end{aligned}$$

Como soma de gaussianas independentes é gaussiana, conclui-se

$$Z \sim N\big(\mathbb{E}[Z],\, \mathrm{Var}(Z)\big) = N(0,2).$$

No caso da CDF para os três itens, os gráficos teóricos e empíricos coincidiram, observando uma sobreposição das linhas (azul e vermelha).

Primeiro, a coincidência entre as curvas empíricas e teóricas confirma que o gerador de números aleatórios e os métodos de estimação (histograma para PDF e ordenação para CDF) estão corretos. No caso da variável uniforme em $[0, \pi]$, o fato de o histograma se manter praticamente plano e a CDF crescer em linha reta mostra que cada valor dentro do intervalo de 0 a π tem a mesma probabilidade de ocorrência, como esperado teoricamente.

Para a distribuição normal padrão $N(0,1)$, a forma de "sino" do histograma reforça a simetria em torno de zero e o decaimento rápido das caudas. A CDF em "S" reflete a acumulação gradual de probabilidade, indicativa de que há pouca chance de valores muito afastados da média. As pequenas ondulações observadas no histograma derivam apenas do caráter amostral finito e desaparecem à medida que aumentamos o número de pontos.

No caso da soma $Z = X+Y$, cuja distribuição teórica é $N(0,2)$, observa-se que o PDF de Z se torna mais achatado (variância dobrada) e a CDF acumula mais lentamente.

Os valores empíricos de média e variância calculados a partir de 10^6 amostras também confirmam o comportamento esperado de cada distribuição. Para a variável uniforme em $[0, \pi]$, a média empírica foi de 1,5711, muito próxima do valor teórico $\mathbb{E}[X] = \pi/2 \approx 1{,}5708$, e a variância empírica de 0,8235 confirma a fórmula $\mathrm{Var}(X) = \pi^2/12 \approx 0{,}8225$. Essas pequenas diferenças (da ordem de 10^{-3}) decorrem do erro amostral e tenderiam a zero ao aumentar ainda mais o número de amostras.

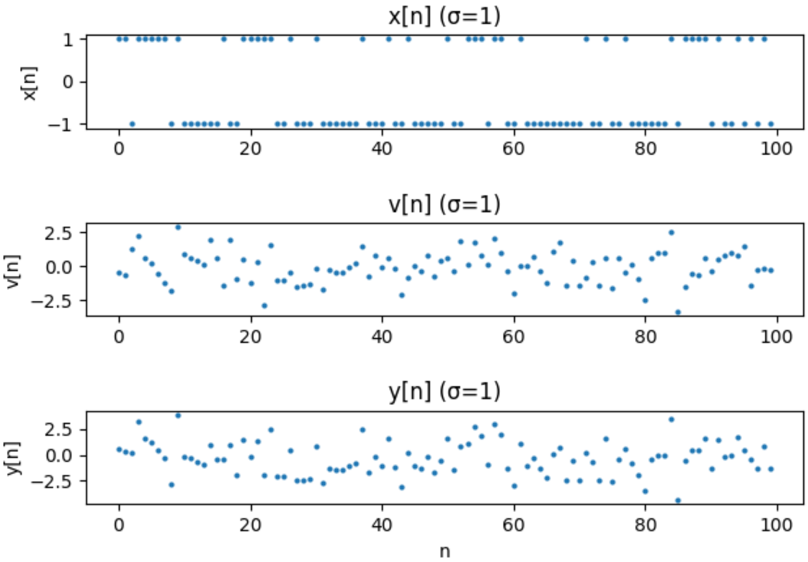
Para a distribuição normal padrão $N(0,1)$, encontramos média empírica 0,0004 e variância empírica 0,9989, ambas praticamente iguais aos valores teóricos $\mathbb{E}[X] = 0$

e $\text{Var}(X) = 1$. A discrepância de poucos 10^{-4} reflete apenas as flutuações naturais, mostrando que o gerador de Gaussianas está bem calibrado.

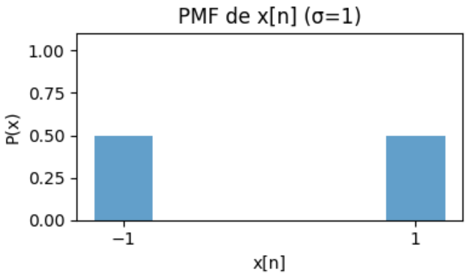
No caso da soma $Z = X + Y$, cuja distribuição teórica é $N(0, 2)$, obtivemos média empírica 0,0018 (teórico 0) e variância empírica 2,0045 (teórico 2). O aumento da variância em relação à normal padrão confirma a propriedade de variância aditiva de variáveis independentes, e a proximidade dos resultados empíricos confirma que estão próximos dos teóricos.

Questão 4

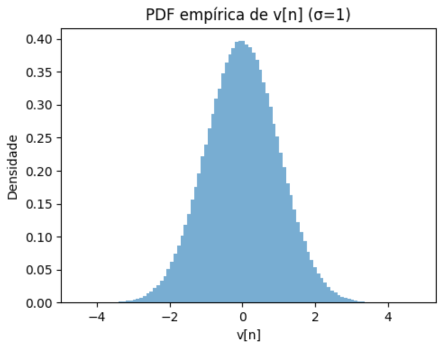
a)



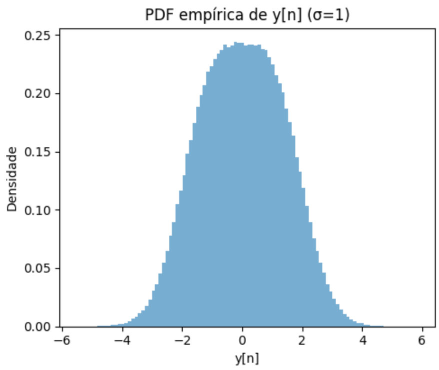
(a) Primeiras 100 amostras de $x[n]$, $v[n]$ e $y[n]$ ($\sigma = 1$).



(b) PMF de $x[n]$ (apenas ± 1).



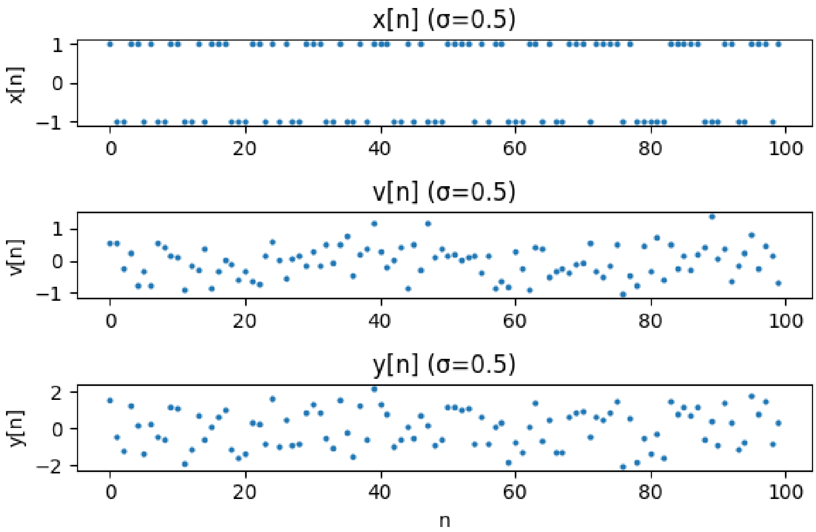
(c) PDF empírica de $v[n]$ (AWGN).



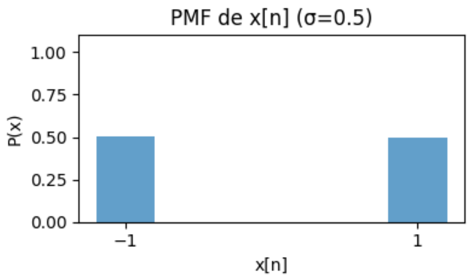
(d) PDF empírica de $y[n] = x[n]+v[n]$.

Figure 4: Resultados da Questão 4(a) para $\sigma = 1$: (a) combinação dos sinais $x[n]$, $v[n]$ e $y[n]$; (b) PMF discreta de $x[n]$; (c) estimativa de PDF de $v[n]$; (d) estimativa de PDF de $y[n]$.

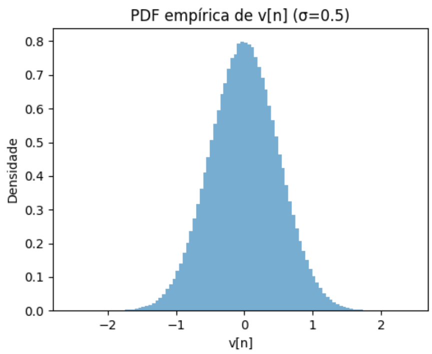
b)



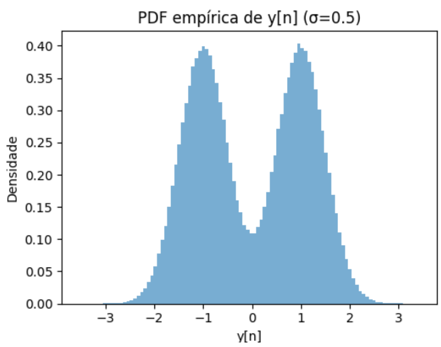
(a) Primeiras 100 amostras de $x[n]$, $v[n]$ e $y[n]$ ($\sigma = 0.5$).



(b) PMF de $x[n]$ (apenas ± 1).



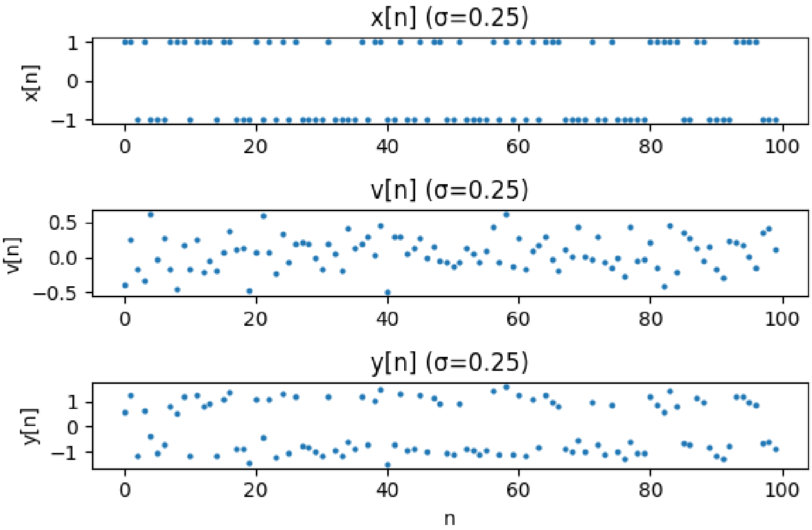
(c) PDF empírica de $v[n]$ (AWGN).



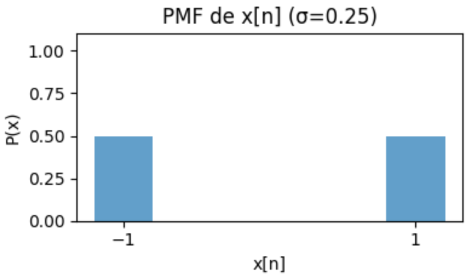
(d) PDF empírica de $y[n] = x[n]+v[n]$.

Figure 5: Resultados da Questão 4(b) para $\sigma = 0.5$: (a) combinação dos sinais $x[n]$, $v[n]$ e $y[n]$; (b) PMF discreta de $x[n]$; (c) estimativa de PDF de $v[n]$; (d) estimativa de PDF de $y[n]$.

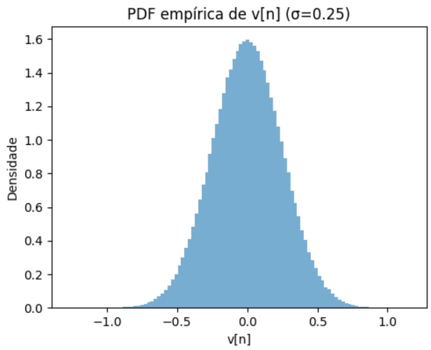
c)



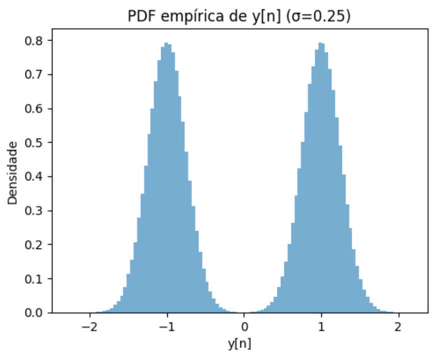
(a) Primeiras 100 amostras de $x[n]$, $v[n]$ e $y[n]$ ($\sigma = 0.25$).



(b) PMF de $x[n]$ (apenas ± 1).



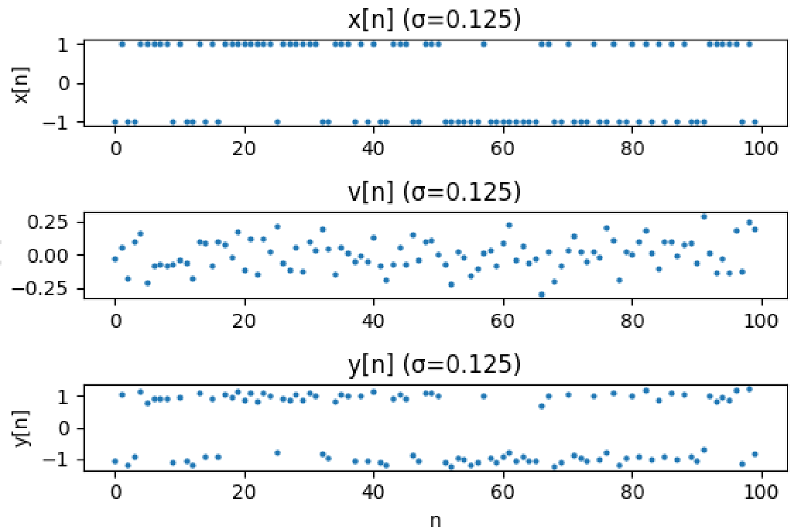
(c) PDF empírica de $v[n]$ (AWGN).



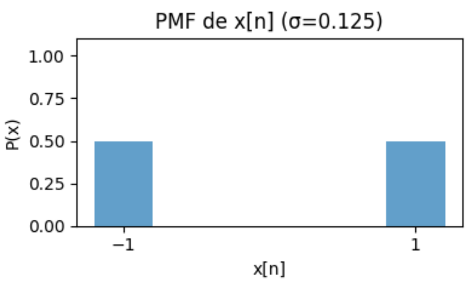
(d) PDF empírica de $y[n] = x[n]+v[n]$.

Figure 6: Resultados da Questão 4(c) para $\sigma = 0.25$: (a) combinação dos sinais $x[n]$, $v[n]$ e $y[n]$; (b) PMF discreta de $x[n]$; (c) estimativa de PDF de $v[n]$; (d) estimativa de PDF de $y[n]$.

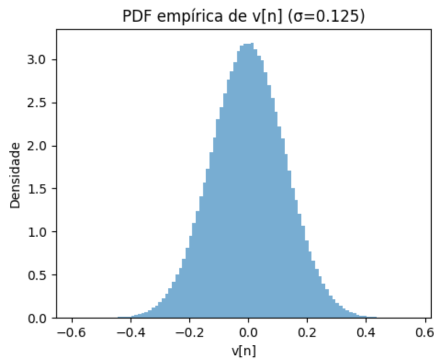
d)



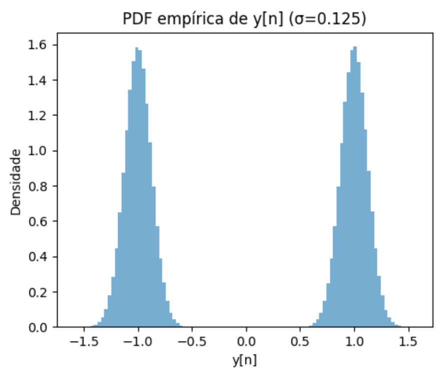
(a) Primeiras 100 amostras de $x[n]$, $v[n]$ e $y[n]$ ($\sigma = 0.125$).



(b) PMF de $x[n]$ (apenas ± 1).



(c) PDF empírica de $v[n]$ (AWGN).



(d) PDF empírica de $y[n] = x[n]+v[n]$.

Figure 7: Resultados da Questão 4(d) para $\sigma = 0.125$: (a) combinação dos sinais $x[n]$, $v[n]$ e $y[n]$; (b) PMF discreta de $x[n]$; (c) estimativa de PDF de $v[n]$; (d) estimativa de PDF de $y[n]$.

e)

Utilizando os resultados dos itens 7(a–d) para preencher a coluna $\text{SER}_{\text{simulada}}$ da tabela:

σ	$\text{SER}_{\text{simulada}}$	$P_e^{\text{teórica}}$
1	0.158667	0.1586553
0.5	0.023012	0.02275013
0.25	0.000029	0.00003167124
0.125	0	6.220961×10^{-16}

Table 2: Comparação entre a taxa de erro de símbolo simulada (SER) e a probabilidade de erro teórica P_e para diferentes valores de σ.

f)

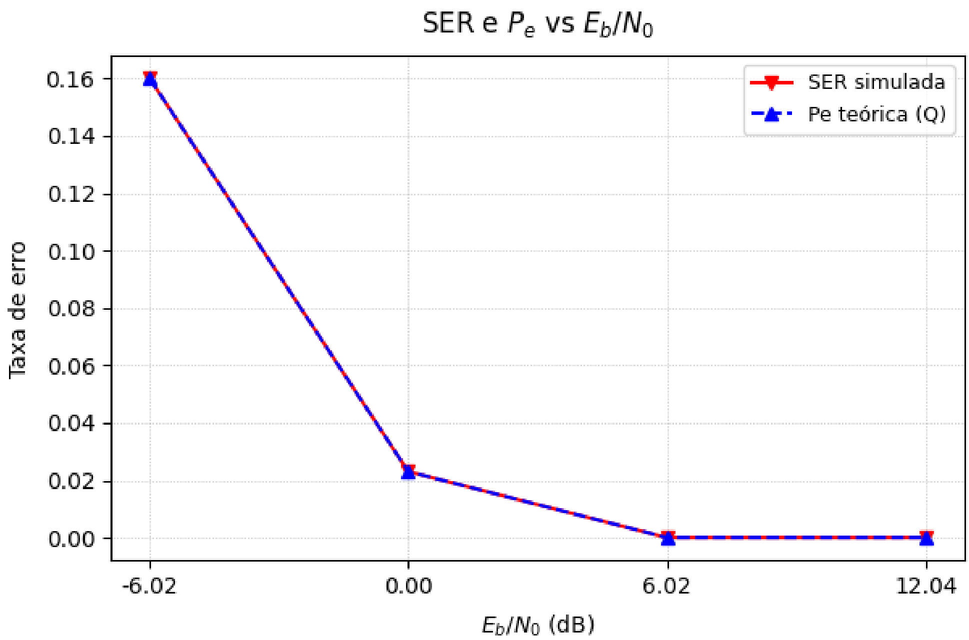


Figure 8: Taxa de erro de símbolo simulada (linha vermelha) e probabilidade de erro teórica P_e (linha azul tracejada) em função de E_b/N_0 (dB). Os pontos correspondem a $\sigma = 1$ ($E_b/N_0 = -6{,}02$ dB), $\sigma = 0{,}5$ ($E_b/N_0 = 0{,}00$ dB), $\sigma = 0{,}25$ ($E_b/N_0 = 6{,}02$ dB) e $\sigma = 0{,}125$ ($E_b/N_0 = 12{,}04$ dB).

Os formatos observados para as distribuições de $x[n]$, $v[n]$ e $y[n]$ eram justamente os esperados:

- **PDF de $x[n]$:** discreta, com massas pontuais de probabilidade 0,5 em cada nível -1 e $+1$, refletindo o esquema BPSK equiprovável.
- **PDF de $v[n]$:** aparece como uma curva em "sino" simétrica em torno de zero, típica de uma Gaussiana $N(0, \sigma^2)$. Isso decorre diretamente do modelo AWGN (Additive White Gaussian Noise).

- **PDF de $y[n] = x[n] + v[n]$:** surge como uma mistura de duas gaussianas deslocadas para ± 1. Cada pico corresponde à soma do ruído ao símbolo $+1$ ou -1. À medida que σ diminui, esses picos ficam mais agudos (variância menor), refletindo menor dispersão em torno de ± 1.

Esse comportamento confirma a teoria: adicionar ruído gaussiano a um sinal discreto em $\{-1, +1\}$ resulta em uma distribuição contínua com dois modos centrados nos níveis de símbolo originais.

Note que, para $\sigma = 1$, a variância do ruído é comparável à separação entre -1 e $+1$, de modo que os dois lóbulos se sobrepõem e formam um único lóbulo amplo em vez de dois lóbulos distintos.

Os resultados confirmam que, num sistema BPSK em canal AWGN, a detecção por limiar é eficaz: conforme o desvio-padrão do ruído σ diminui, o sinal $y[n]$ torna-se mais nítido, mais próximo do sinal original, como pode ser visto na figura (das 100 primeiras amostras e das PDF's), a estimativa $\hat{x}[n]$ aproxima-se de $x[n]$ e a taxa de erro cai rapidamente, aproximando-se do desempenho ideal previsto pela teoria.

Códigos Utilizados

A seguir, apresentam-se os códigos em Python utilizados no relatório.

Questão 1

```python
import numpy as np
import matplotlib.pyplot as plt

# ================================
# Funcoes definidas
# ================================

def rect_pulse(t, t0, Tb):
    """
    Gera um pulso retangular Pi((t - t0)/Tb)
    Ativa (valor 1) quando |(t - t0)/Tb| <= 0.5, senao retorna 0
    """
    return np.where(np.abs((t - t0) / Tb) <= 0.5, 1, 0)

def tri_pulse(t, t0, Tb):
    """
    Gera um pulso triangular Lambda((t - t0)/Tb)
    Valor decresce linearmente de 1 ate 0 quando |(t - t0)/Tb| <=
        1
    Fora desse intervalo retorna 0
    """
    x = (t - t0) / Tb
    return np.where(np.abs(x) <= 1, 1 - np.abs(x), 0)
```

```python
# ================================
# Eixo do tempo
# ================================

# Cria um vetor de tempo de -5 a 5 com 1000 amostras
t = np.linspace(-5, 5, 1000)

# ================================
# Parametros dos casos (a) e (b)
# ================================

params = [(1, 0.5), (2, 3)]  # Lista com os pares (Tb, t0)
labels = ['(a) Tb=1, t0=0.5', '(b) Tb=2, t0=3']  # Rotulos para
    os graficos

# ================================
# Plotagem dos graficos
# ================================

# Cria uma grade de 2 linhas e 2 colunas de subgraficos
fig, axes = plt.subplots(2, 2, figsize=(12, 6))

# Para cada conjunto de parametros (caso a e b)
for i, (Tb, t0) in enumerate(params):
    rect = rect_pulse(t, t0, Tb)  # Calcula pulso retangular
    tri = tri_pulse(t, t0, Tb)    # Calcula pulso triangular

    # Grafico do pulso retangular
    axes[i][0].plot(t, rect)
    axes[i][0].set_title(f'Pulso Retangular {labels[i]}')
    axes[i][0].set_xlabel('t')
    axes[i][0].set_ylabel('Amplitude')
    axes[i][0].grid(True)

    # Grafico do pulso triangular
    axes[i][1].plot(t, tri)
    axes[i][1].set_title(f'Pulso Triangular {labels[i]}')
    axes[i][1].set_xlabel('t')
    axes[i][1].set_ylabel('Amplitude')
    axes[i][1].grid(True)

# Ajusta o layout para evitar sobreposicao
plt.tight_layout()

# Exibe os graficos na tela
plt.show()
```

Questão 2

```python
import numpy as np
import matplotlib.pyplot as plt

# Numero de amostras: mais pontos aumentam a resolucao em tempo e
    frequencia
N = 8192

# vetor de tempo de -40 a 40 segundos
t = np.linspace(-40, 40, N)
# passo de tempo (incremento entre amostras)
dt = t[1] - t[0]

# (a) Geracao do pulso triangular p(t) = tri(t)
# - Amplitude maxima 1 em t=0
# - Decai linearmente ate 0 em |t| = 1
p = np.where(np.abs(t) <= 1, 1 - np.abs(t), 0)

# (b) Transformada de Fourier P(f) via FFT
# - fft(p) computa a DFT discreta
# - fftshift centraliza o zero da frequencia
# - multiplicar por dt aproxima a integral continua
P = np.fft.fftshift(np.fft.fft(p)) * dt

# eixo de frequencia correspondente (Hz)
# - fftfreq gera os bins de frequencia
# - fftshift realinha para que f=0 fique no centro
f = np.fft.fftshift(np.fft.fftfreq(N, d=dt))

# (c) Transformada inversa para reconstruir o sinal no tempo
# - ifftshift e ifft revertendo o fftshift e fft
# - divisao por dt aproxima a transformada inversa continua
p_inv = np.fft.ifft(np.fft.ifftshift(P)) / dt

# (d1) Plot do espectro |P(f)| em torno da frequencia zero
plt.figure(figsize=(10, 4))
plt.plot(f, np.abs(P))
plt.title('Magnitude of Fourier Transform |P(f)| vs Frequency f')
plt.xlabel('Frequency f (Hz)')
plt.ylabel('|P(f)|')
plt.xlim(-10, 10)   # foco na regiao do lobulo principal
plt.grid(True)

# (d2) Plot do sinal reconstruido pinv(t) no intervalo de tempo
   completo
plt.figure(figsize=(10, 4))
plt.plot(t, np.real(p_inv))
plt.title('Inverse Fourier Transform pinv(t) vs Time t')
plt.xlabel('Time t')
plt.ylabel('pinv(t)')
plt.xlim(-40, 40)   # interval completo de -40 a +40
plt.grid(True)
```

```
plt.show()
```

Questão 3

```
import numpy as np
import matplotlib.pyplot as plt
import math

# Define o numero de amostras para cada variavel aleatoria
N = 10**6  # 1.000.000 de pontos

# --- Geracao das amostras ---
samples1 = np.random.uniform(0, np.pi, N)
samples2 = np.random.normal(0, 1, N)
X = np.random.normal(0, 1, N)
Y = np.random.normal(0, 1, N)
samples3 = X + Y

# Estatisticas
print("Uniform[0,pi]: mean = {:.4f}, var = {:.4f}".format(
    samples1.mean(), samples1.var()))
print("Normal(0,1):  mean = {:.4f}, var = {:.4f}".format(samples2
    .mean(), samples2.var()))
print("Z = X + Y:    mean = {:.4f}, var = {:.4f}".format(samples3
    .mean(), samples3.var()))

# --- Preparacao das curvas teoricas ---

# 1) Uniforme estendido
x1 = np.linspace(-1, np.pi+1, 300)                              #
    grid de -1 a pi+1
pdf1 = np.where((x1 >= 0) & (x1 <= np.pi), 1/np.pi, 0)          #
    1/pi em [0,pi], 0 fora
cdf1 = np.clip(x1 / np.pi, 0, 1)                                #
    linear em [0,pi], plato em 0 e 1

# 2) Normal(0,1)
x2 = np.linspace(-5, 5, 200)
pdf2 = 1/np.sqrt(2*np.pi) * np.exp(-x2**2 / 2)
erf_vec = np.vectorize(math.erf)
cdf2 = 0.5 * (1 + erf_vec(x2 / np.sqrt(2)))

# 3) Z ~ N(0,2)
x3 = np.linspace(-5, 5, 200)
sigma3 = np.sqrt(2)
pdf3 = 1/(sigma3 * np.sqrt(2*np.pi)) * np.exp(-x3**2 / (2 *
    sigma3**2))
cdf3 = 0.5 * (1 + erf_vec(x3 / (sigma3 * np.sqrt(2))))
```

```python
# --- Plotagem ---
fig, ax = plt.subplots(3, 2, figsize=(12, 12))

# PDF Uniform
ax[0, 0].hist(samples1, bins=100, density=True, alpha=0.6, label=
    "Empirica")
ax[0, 0].plot(x1, pdf1, 'r-', label="Teorica")
ax[0, 0].set_title("PDF Uniform[0,pi]")
ax[0, 0].legend()

# CDF Uniform
s1 = np.sort(samples1)
ax[0, 1].plot(s1, np.arange(1, N+1)/N, label="Empirica")
ax[0, 1].plot(x1, cdf1, 'r-', label="Teorica")
ax[0, 1].set_title("CDF Uniform[0,pi]")
ax[0, 1].legend()

# PDF Normal
ax[1, 0].hist(samples2, bins=100, density=True, alpha=0.6, label=
    "Empirica")
ax[1, 0].plot(x2, pdf2, 'r-', label="Teorica")
ax[1, 0].set_title("PDF N(0,1)")
ax[1, 0].legend()

# CDF Normal
s2 = np.sort(samples2)
ax[1, 1].plot(s2, np.arange(1, N+1)/N, label="Empirica")
ax[1, 1].plot(x2, cdf2, 'r-', label="Teorica")
ax[1, 1].set_title("CDF N(0,1)")
ax[1, 1].legend()

# PDF Z = X+Y
ax[2, 0].hist(samples3, bins=100, density=True, alpha=0.6, label=
    "Empirica")
ax[2, 0].plot(x3, pdf3, 'r-', label="Teorica (N(0,2))")
ax[2, 0].set_title("PDF Z = X + Y")
ax[2, 0].legend()

# CDF Z = X+Y
s3 = np.sort(samples3)
ax[2, 1].plot(s3, np.arange(1, N+1)/N, label="Empirica")
ax[2, 1].plot(x3, cdf3, 'r-', label="Teorica (N(0,2))")
ax[2, 1].set_title("CDF Z = X + Y")
ax[2, 1].legend()

plt.tight_layout()
plt.show()
```

Questão 4

```
import numpy as np
import matplotlib.pyplot as plt
import math
import matplotlib.ticker as ticker

### a-d
# 1) Geracao dos dados
d       = 2
N       = 10**6
symbols = np.array([-d/2, d/2])
sigma_list = [1, 0.5, 0.25, 0.125]

for sigma in sigma_list:
    x = np.random.choice(symbols, size=N)       # PMF discreta
    v = np.random.normal(0, sigma, size=N)      # PDF continua
    y = x + v

    #
       -----------------------------------------------------------

    # 2) Primeiras 100 amostras
    plt.figure(figsize=(6,4))
    plt.subplot(3,1,1)
    plt.plot(x[:100], '.', markersize=4)
    plt.title(f'x[n] (sigma={sigma})'); plt.ylabel('x[n]')
    plt.subplot(3,1,2)
    plt.plot(v[:100], '.', markersize=4)
    plt.title(f'v[n] (sigma={sigma})'); plt.ylabel('v[n]')
    plt.subplot(3,1,3)
    plt.plot(y[:100], '.', markersize=4)
    plt.title(f'y[n] (sigma={sigma})'); plt.xlabel('n'); plt.
       ylabel('y[n]')
    plt.tight_layout()

    #
       -----------------------------------------------------------

    # 3) PMF de x[n]
    # Calculo da PMF
    vals, cnts = np.unique(x, return_counts=True)
    probs = cnts / cnts.sum()

    # Plot com barras estreitas em -1 e +1
    plt.figure(figsize=(4, 2.5))
    plt.bar(vals, probs, width=0.4, align='center', alpha=0.7)
    plt.xticks(vals)
    plt.ylim(0, 1.1)
    plt.title(f'PMF de x[n] (sigma={sigma})')
    plt.xlabel('x[n]')
    plt.ylabel('P(x)')
    plt.tight_layout()
```

```
    plt.show()

    # -----------------------------------------------------------

    # 4) PDF empirica de v[n] e y[n]
    for data, name in [(v,'v[n]'), (y,'y[n]')]:
        cnts, bins = np.histogram(data, bins=100, density=True)
        plt.figure(figsize=(4,2.5))
        plt.stairs(cnts, bins, fill=True, alpha=0.6)
        plt.title(f'PDF empirica de {name} (sigma={sigma})')
        plt.xlabel(name); plt.ylabel('Densidade')

    # -----------------------------------------------------------

    # 5) Decisao e SER
    x_hat = np.where(y>=0, d/2, -d/2)
    ser = np.mean(x_hat != x)
    print(f'sigma={sigma}: SER = {ser:.6f}')

plt.show()



### e
# Parametro
d = 2

# Valores de sigma correspondentes aos itens (a)-(d)
sigma_list = [1, 0.5, 0.25, 0.125]

# Supondo que ser_list ja foi calculada anteriormente:
# ser_list = [SER para sigma=1, sigma=0.5, sigma=0.25, sigma
    =0.125]
ser_list = [0.158667, 0.023012, 0.000029, 0]

# Calculo de Pe teorica para cada sigma
pe_list = [0.5 * math.erfc((d/2) / (sigma * math.sqrt(2))) for
    sigma in sigma_list]

# Exibe comparacao
print("sigma     |   SER simulada    |   Pe teorica")
print("----------------------------------------")
for sigma, ser, pe in zip(sigma_list, ser_list, pe_list):
    print(f"{sigma:<5} | {ser:>12.6e} | {pe:>12.6e}")


 ### f
# Dados
sigma_list = [1, 0.5, 0.25, 0.125]
```

```python
ser_sim      = np.array([0.16,  0.023,   0.0,    0.0])
pe_teo       = np.array([0.16,  0.023,   0.0,    0.0])
d, Eb        = 2, (2**2)/8
EbN0_dB      = [10*np.log10(Eb/(2*s**2)) for s in sigma_list]

fig, ax = plt.subplots(figsize=(6,4), constrained_layout=True)

# Plota
ax.plot(EbN0_dB, ser_sim, marker='v', color='r', linestyle='-',
   label='SER simulada')
ax.plot(EbN0_dB, pe_teo,  marker='^', color='b', linestyle='--',
   label='Pe teorica (Q)')

# ESCALA LINEAR NO Y
ax.set_yscale('linear')

# ticks X e Y fixos e uniformes
ax.set_xticks(EbN0_dB)
ax.set_yticks(np.arange(0, 0.18, 0.02))   #
   0.00,0.02,0.04,...,0.16

# formatacao ponto flutuante, 2 casas
ax.xaxis.set_major_formatter(ticker.FormatStrFormatter('%.2f'))
ax.yaxis.set_major_formatter(ticker.FormatStrFormatter('%.2f'))

# rotulos, titulo, grade e legenda
ax.set_xlabel(r'$E_b/N_0$ (dB)', fontsize=10, labelpad=6)
ax.set_ylabel('Taxa de erro',    fontsize=10, labelpad=6)
ax.set_title(r'SER e $P_e$ vs $E_b/N_0$', fontsize=12, pad=10)
ax.grid(True, linestyle=':', linewidth=0.5)
ax.legend(fontsize=9, loc='upper right')

plt.show()
```