

## HƯỚNG DẪN GIẢI BÀI TẬP VÀ MÔ PHỎNG

### CHƯƠNG 2

#### Bài 1:

Cho một dãy xung chữ nhật biên độ  $A$ , chu kỳ  $T$  và thời gian xung  $T_1$  ( $T_1 < T$ ).

- a) Tìm năng lượng xung
- b) Tìm công suất trung bình của xung

#### Hướng dẫn giải

- a) Vì đây là hàm tuần hoàn nên

$$E[\infty] = \int_{-\infty}^{\infty} |s(t)|^2 dt = \int_0^{T_1} A^2 dt = A^2 T_1 \quad \text{Joule}$$

- b)  $\bar{P}[\infty] = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\infty}^{\infty} |s(t)|^2 dt = \frac{1}{T} \int_0^{T_1} |s(t)|^2 dt = \frac{A^2 T_1}{T} \quad \text{W}$

#### Bài 2:

Cho hàm bậc thang

$$s(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases}$$

- a) Tìm tổng năng lượng
- b) Tìm công suất trung bình
- a) Là hàm kiểu gì?

#### Hướng dẫn giải

- a)  $E[\infty] = \int_{-\infty}^{\infty} |s(t)|^2 dt = \infty$

- b)  $\bar{P}[\infty] = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\infty}^{\infty} |s(t)|^2 dt = 1 \text{ W}$

- c) Là hàm kiểu công suất

#### Bài 3:

Cho hàm mũ

$$s(t) = \begin{cases} 0, & t < 0 \\ \exp(-2t), & t \geq 0 \end{cases}$$

- a) Tìm tổng năng lượng
- b) Tìm công suất trung bình
- a) Là hàm kiểu gì?

#### Hướng dẫn giải

- a)  $E[\infty] = \int_{-\infty}^{\infty} |s(t)|^2 dt = \int_0^{\infty} \exp(-2t) dt = 1/2 \text{ Joule}$

b)

$$\begin{aligned}\bar{P}[\infty] &= \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T |s(t)|^2 dt = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \exp(-2t) dt \\ &= \lim_{T \rightarrow \infty} \frac{1 - \exp(-2T)}{2T} = 0 \text{ W}\end{aligned}$$

c) Là hàm kiểu năng lượng

**Bài 4:**

Cho hàm

$$1/\sqrt{1+t}$$

a) Tìm tổng năng lượng

b) Tìm công suất trung bình

a) Là hàm kiểu gì?

**Hướng dẫn giải**

a)  $E[\infty] = \int_{-\infty}^{\infty} |s(t)|^2 dt = \int_{-\infty}^{\infty} \frac{1}{1+t} dt = \log_e(1+t) \Big|_0^{\infty} = \infty \text{ Joule}$

b) 
$$\begin{aligned}\bar{P}[\infty] &= \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\infty}^{\infty} |s(t)|^2 dt = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\infty}^{\infty} \frac{1}{1+t} dt \\ &= \lim_{T \rightarrow \infty} \frac{1}{T} \log_e(1+T) = 0 \text{ W}\end{aligned}$$

c) Là hàm kiểu công suất

**Bài 5:**

Tìm ACF và PSD của hàm cosin dưới đây

$$s(t) = A \cos(2\pi f_1 t + \theta)$$

Tìm

a) ACF

b) PSD

c) Công suất trung bình

**Hướng dẫn giải**

a) Vì đây là hàm tuần hoàn có giá trị thực nên

$$\begin{aligned}R_s(\tau) &= \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\alpha}^{\alpha+T} s(t)s(t+\tau) dt = \frac{1}{T} \int_0^T s(t)s(t+\tau) dt \\ &= \frac{A^2}{T} \int_0^T \{\cos(2\pi f_1 t + \theta) \cos[2\pi f_1 (t+\tau) + \theta]\} dt = \frac{A^2}{2} \cos(2\pi f_1 \tau)\end{aligned}$$

b) Ta có thể biểu diễn

$$\frac{1}{2} \cos(2\pi f_1 \tau) = e^{j2\pi f_1 \tau} + e^{-j2\pi f_1 \tau}$$

Vì thế

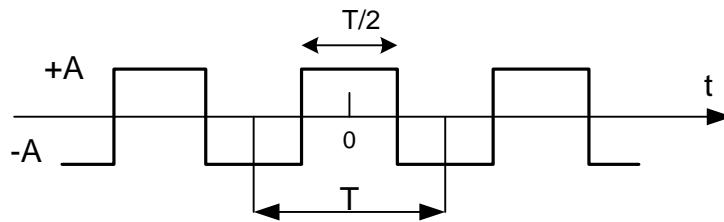
$$\Phi(f) = F\left[\frac{A^2}{2} \cos(2\pi f\tau)\right] = \frac{A^2}{4} [\delta(f + f_1) + \delta(f - f_1)]$$

c) Công suất trung bình tính như sau:

$$\bar{P}(\infty) = R_s(\tau=0) = \int_{-\infty}^{\infty} \Phi(f) df = \frac{A^2}{2} W$$

### Bài 6:

Cho dãy xung chữ nhật biên độ  $\pm A$ , chu kỳ  $T$  như ở hình vẽ dưới đây



- Tìm biến đổi Fourier
- Tìm PSD
- Tìm ACF
- Tìm công suất trung bình

### Hướng dẫn giải

a) Biến đổi Fourier có dạng sau:

$$S(f) = ASinc(Tf/2)\delta_{f1}(f) - A\delta(f)$$

trong đó  $f_1=1/T$

$$\delta_{f1}(f) = \sum_k \delta(f - kf_1)$$

b)  $\Phi(f)=A^2 Sinc^2(Tf/2)\delta_{f1}(f)- A^2\delta(f)$

c) Từ quan hệ biến đổi Fourier:  $\int_{-\infty}^{\infty} s_1(z)s_2(z-\tau)dz = s_1(\tau) \otimes s_2(\tau) \Leftrightarrow S_1(f).S_2(f)$

Ta được  $R_s(\tau) = 2A^2 \Lambda(2\tau/T) \otimes \delta_T(\tau) = 2A^2 \Lambda(2\tau/T) p_T(\tau-kT)-A^2$

d)  $P=R_s(0)=A^2$

### Bài 7:

Cho dãy xung trong là quá trình ngẫu nhiên được biểu diễn theo công thức sau:

$$X(t) = \sum_{k=-\infty}^{\infty} A_k p_T(t + \frac{T}{2} - kT)$$

trong đó  $A_k=\{+A,-A\}$  với xác suất xuất hiện  $+A$  và  $-A$  bằng nhau và bằng  $1/2$ . Tìm:

- ACF
- PSD
- Công suất trung bình

**Hướng dẫn giải**

a)

$$\begin{aligned} R_x(\tau) &= E[X(t)X(t+\tau)] \\ &= \begin{cases} A^2 \left[ 1 - \frac{|\tau|}{T} \right], & |\tau| \leq T \\ 0, & \text{nếu khác} \end{cases} \\ &= A^2 \Lambda_T(\tau) \end{aligned}$$

b)  $\Phi_x(f) = A^2 T \text{sinc}^2(fT)$

c)  $P = R_x(0) = A^2$

**Bài 8:**

Một đường truyền dẫn băng gốc trong đó mỗi ký hiệu truyền được 2 bit có thừa số dốc  $\alpha=1$ . Nếu tốc độ số liệu cần truyền là 9600 bps, tìm:

- a) Tốc độ truyền dẫn
- b) Băng thông Nyquist

**Hướng dẫn giải**

a)  $R_s = 2R_b = 2 \times 9600 \text{ kps} = 19200 \text{ sps}$

b) Băng thông Nyquist

$$B_N = (1 + \alpha)R_s / 2 = 19200 \text{ Hz}$$

**Bài 9:**

Một đường truyền dẫn băng thông có dữ liệu như ở bài 8. Tìm:

- a) Tốc độ truyền dẫn
- b) Băng thông Nyquist

**Hướng dẫn giải**

a)  $R_s = 2R_b = 2 \times 9600 \text{ kps} = 19200 \text{ sps}$

b) Băng thông Nyquist

$$B_N = (1 + \alpha)R_s = 2 \times 19200 = 38400 \text{ Hz}$$

**Bài 10:**

Một tín hiệu được đo tại đầu ra của bộ lọc băng thông lý tưởng có băng thông là B Hz. Khi không có tín hiệu tại đầu vào bộ lọc, công suất đo được là  $1 \times 10^{-6} \text{ W}$ . Khi có tín hiệu NRZ lưỡng cực công suất đo được là  $1,1 \times 10^{-5} \text{ W}$ . Tạp âm có dạng tạp âm trắng. Tính:

- a) Tỷ số tín hiệu trên tạp âm theo dB
- b) Xác suất máy thu nhận biết sai xung NRZ

**Hướng dẫn giải**

a)  $\text{SNR} = (11 - 1) / 1 = 10 \rightarrow \text{SNR}[\text{dB}] = 10 \lg 10 = 10 \text{ dB}$

b) Ta có:  $A^2 / \sigma^2 = \text{SNR} = 10$  và tra bảng trong phụ lục

$$P_e = Q(\sqrt{10}) = Q(3,16) = 8 \times 10^{-4}$$

**Bài 11:**

Nếu băng thông bộ lọc trong bài 10 tăng gấp đôi và đo mức công suất tín hiệu tại đầu ra bộ lọc. Hỏi:

- a) Khi không có tín hiệu thì công suất đo được tại đầu ra của bộ lọc là bao nhiêu ? và tỷ số tín hiệu trên tạp âm là bao nhiêu?
- c) Xác suất lỗi xung NRZ là bao nhiêu ?

**Hướng dẫn giải**

- a) Trong trường hợp này công suất tạp âm tăng gấp đôi và bằng  $2 \times 10^{-6}$  còn công suất tín hiệu không đổi và công suất đo tại đầu ra của bộ lọc sẽ bằng  $1,2 \times 10^{-5}$ , Vì thế tỷ số tín hiệu trên tạp âm như sau:

$$SNR = (12-2)/2 = 5 \rightarrow SNR[dB] = 10 \lg 5 = 6,98970 \text{ dB}$$

- b) Xác suất lỗi xung như sau:

$$P_e = Q(\sqrt{5}) = Q(2,24) = 0,0125$$

**Bài 12:**

Giả sử các mức công suất tín hiệu và tạp âm giống như trong bài 11 và tín hiệu là một tần số duy nhất tại tâm của bộ lọc băng thông. Tìm băng thông bộ lọc khi tỷ số tín hiệu trên tạp âm đo được là  $SNR = 30 \text{ dB}$ .

**Hướng dẫn giải**

Chuyển SNR vào số lần:  $SNR = 10^{30/10} = 1000$  lần

$$\text{Từ công thức SNR ta có thể viết: } SNR = 1000 = \frac{1 \times 10^{-5}}{\frac{B'}{2B} 10^{-6}} = \frac{2B}{B'} 10$$

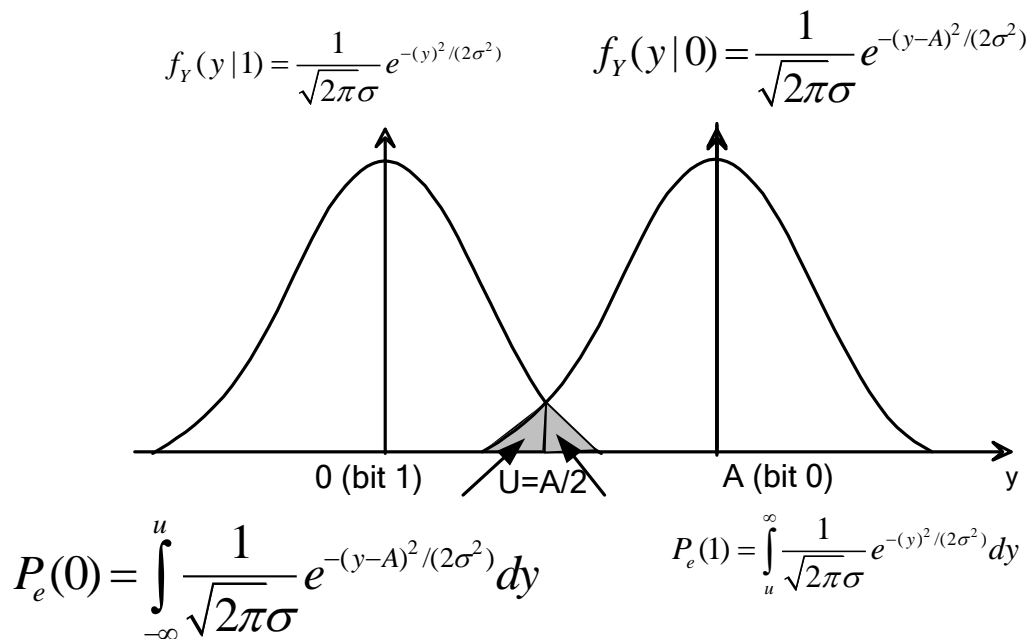
Vậy băng thông của bộ lọc là:  $B' = 2B \times 10^{-2} \text{ Hz}$

**Bài 13:**

Cho một chuỗi nhị phân dài vô tận có phân bố 1 và 0 ngẫu nhiên đi qua kênh AWGN. Tìm xác suất lỗi xung khi: Các xung là NRZ đơn cực  $\{0, A\}$  với  $SNR = 10 \text{ dB}$ .

**Hướng dẫn giải**

Do NRZ đơn cực nhận hai mức tín hiệu  $\{0, A\}$ . Nên các hàm mật độ xác suất trong trường hợp này có dạng sau:



$$P_e = P_0 P_e(0) + P_1 P_e(1) = (1/2)[P_e(0) + P_e(1)],$$

trong đó  $P_0 = P_1 = 1/2$ ,  $P_e(0) = P_e(1)$ .

Vì thế ta có thể viết

$$P_e = \int_u^{\infty} \frac{1}{\sqrt{2\pi}\sigma} \exp(-y^2/2\sigma^2) dy$$

Đặt  $y/\sigma = z$  ta được

$$P_e = \int_{A/2\sigma}^{\infty} \frac{1}{\sqrt{2\pi}} \exp(-z^2/2) dz = Q\left(\frac{A}{2\sigma}\right)$$

Do  $A^2/\sigma^2 = \text{SNR} = 10$ , nên xác suất lỗi xung bằng:  $P_e = Q(\sqrt{10}/2) = Q(1,58) = 0,0571$

#### Bài 14:

Hãy đọc mã chương trình Matlab dưới đây, làm sáng tỏ phổ của tín hiệu tuần hoàn (phổ vạch) ở dạng khảo sát tín hiệu xung chữ nhật tuần hoàn bằng cách chạy chương trình, thay đổi các tham số đầu vào chương trình, biểu diễn đánh giá kết quả mô phỏng.

#### Hướng dẫn giải:

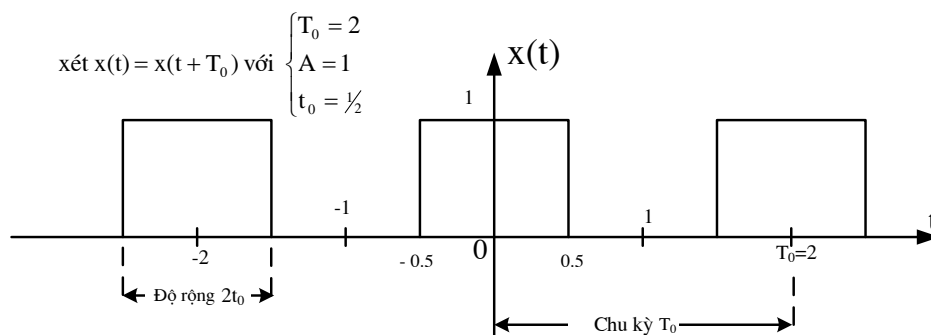
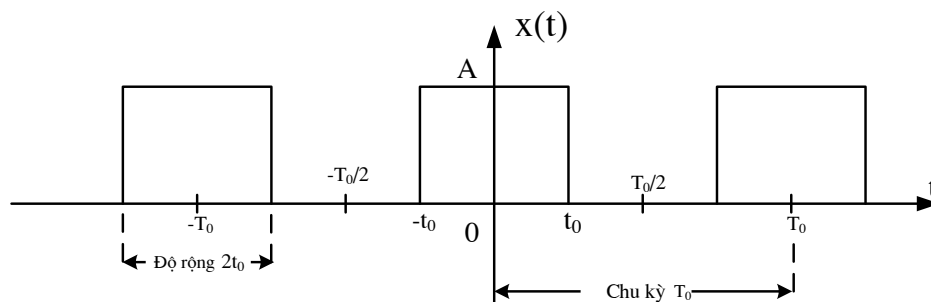
Lấy chuỗi xung chữ nhật tuần hoàn dưới đây làm ví dụ minh họa:

Cho tín hiệu chu kỳ  $x(t)$  có chu kỳ  $T_0$  được định nghĩa bởi.

$$x(t) = A \Pi\left(\frac{t}{2t_0}\right) = \begin{cases} A, & |t| < t_0 \\ \frac{A}{2}, & t = \pm t_0 \\ 0, & \neq \end{cases}$$

Với  $|t| \leq T_0/2$ , trong đó  $t_0 < T_0/2$ . Tín hiệu xung chữ nhật  $\Pi(t)$  được xác định là

$$\Pi(t) = \begin{cases} 1, & |t| < \frac{1}{2} \\ \frac{1}{2}, & t = \pm \frac{1}{2} \\ 0, & \neq \end{cases}$$



Vẽ  $x(t)$  được cho ở hình trên. Nếu  $A=1, T_0=4$  và  $t_0=1 \Rightarrow$  cho cùng giá trị kết quả

- Tìm các hệ số chuỗi Fourier dưới dạng lượng giác và mũ.
- Vẽ phổ rời rạc của  $x(t)$ .

- Tìm các hệ số chuỗi Fourier dưới dạng lượng giác và mũ  
Các hệ số chuỗi Fourier.

$$x_n = \frac{1}{T_0} \int_{\alpha}^{\alpha+T_0} x(t) \cdot e^{-j2\pi \frac{n}{T_0} t} dt$$

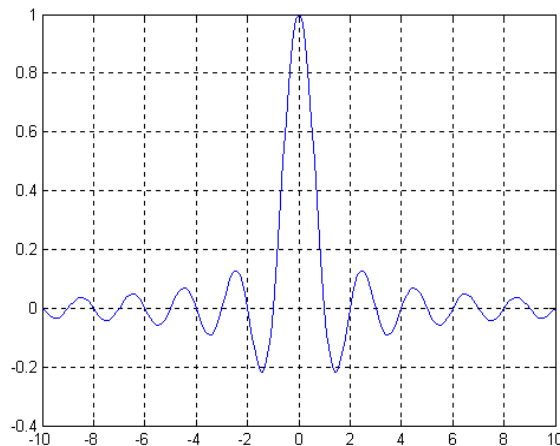
$$\text{Thành phần DC: } x_0 = \frac{1}{2} \int_{-1}^1 x(t) dt = \frac{1}{2} \int_{-0.5}^{0.5} (1) dt = \frac{1}{2}$$

$$\begin{aligned} x_n &= \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} x(t) \cdot e^{-j2\pi n t / T_0} dt = \frac{1}{2} \int_{-1}^1 x(t) \cdot e^{-j2\pi n t / 2} dt \\ &= \frac{1}{2} \int_{-0.5}^{0.5} e^{-jn\pi t} dt = -\frac{1}{j2n\pi} e^{-jn\pi t} \Big|_{t=-0.5}^{t=0.5} \\ &= -\frac{1}{j2n\pi} \left( -j \sin \frac{n\pi}{2} - j \sin \frac{n\pi}{2} \right) \\ &= \frac{1}{n\pi} \sin \left( \pi \frac{n}{2} \right) = \left( \frac{1}{2} \right) \times \frac{\sin \left( \pi \frac{n}{2} \right)}{\left( \pi \frac{n}{2} \right)} \end{aligned}$$

$$= \frac{1}{2} \text{SinC}\left(\frac{n}{2}\right),$$

$$= \begin{cases} 0, & n = \pm 2, \pm 4, \dots \\ \frac{1}{n\pi} (-1)^{\left| \frac{(n-1)}{2} \right|}, & n = \pm 1, \pm 3, \dots \end{cases}$$

Hàm SinC(x) được định nghĩa là:  $\text{SinC}(x) = \frac{\sin(\pi x)}{\pi x}$  và hàm sinc(x) có dạng hình sau:



Thấy rõ, tất cả các hệ số của  $x_n$  là thực (vì  $x(t)$  là thực chẵn), vì vậy

$$\begin{cases} a_n = \sin c\left(\frac{n}{2}\right) \\ b_n = 0 \\ c_n = \sqrt{a_n^2 + b_n^2} = \left| \sin c\left(\frac{n}{2}\right) \right| \\ \theta_n = -\arctan\left(\frac{b_n}{a_n}\right) = 0. \pi \end{cases}$$

Lưu ý rằng, khi n chẵn, thì  $x_n = 0$  (trừ  $n=0$  khi đó  $a_0 = c_0 = 1$  và  $x_0 = 1/2$ ). Vì vậy, ta có.

$$\begin{aligned} x(t) &= \sum_{n=-\infty}^{\infty} \underbrace{\frac{1}{2} \text{SinC}\left(\frac{n}{2}\right)}_{x_n=0, \text{ khi chẵn}} \cdot e^{j2\pi \frac{n}{2} t} \\ &= \frac{1}{2} + \sum_{n=1}^{\infty} \underbrace{\frac{1}{2} \text{SinC}\left(\frac{n}{2}\right) \text{Cos}(n\pi t)}_{\text{do } b_n=0, \text{ và } x_n=0 \text{ khi } n \text{ chẵn} \neq 0} \\ &= \frac{1}{2} + \sum_{\substack{n=1 \\ n \text{ lẻ}}}^{\infty} \frac{2}{n\pi} \cos\left(n\pi t + \left[(-1)^{\frac{(n-1)}{2}} - 1\right] \frac{\pi}{2}\right) \end{aligned}$$

**Nhận xét:** Từ kết quả khai triển chuỗi fourier, cho thấy chuỗi xung chữ nhật tuần hoàn chu kỳ  $T_0$  chứa tổng vô hạn các hàm điều hoà (là bội số của tần số cơ bản  $f_0=1/T_0$ ). Để nêu bật quan hệ giữa chuỗi xung chữ nhật tuần hoàn và các hàm điều hoà được phân tích từ nó và đặc biệt ứng dụng trong thực tế. Do các phần tử và hệ thống viễn thông điều



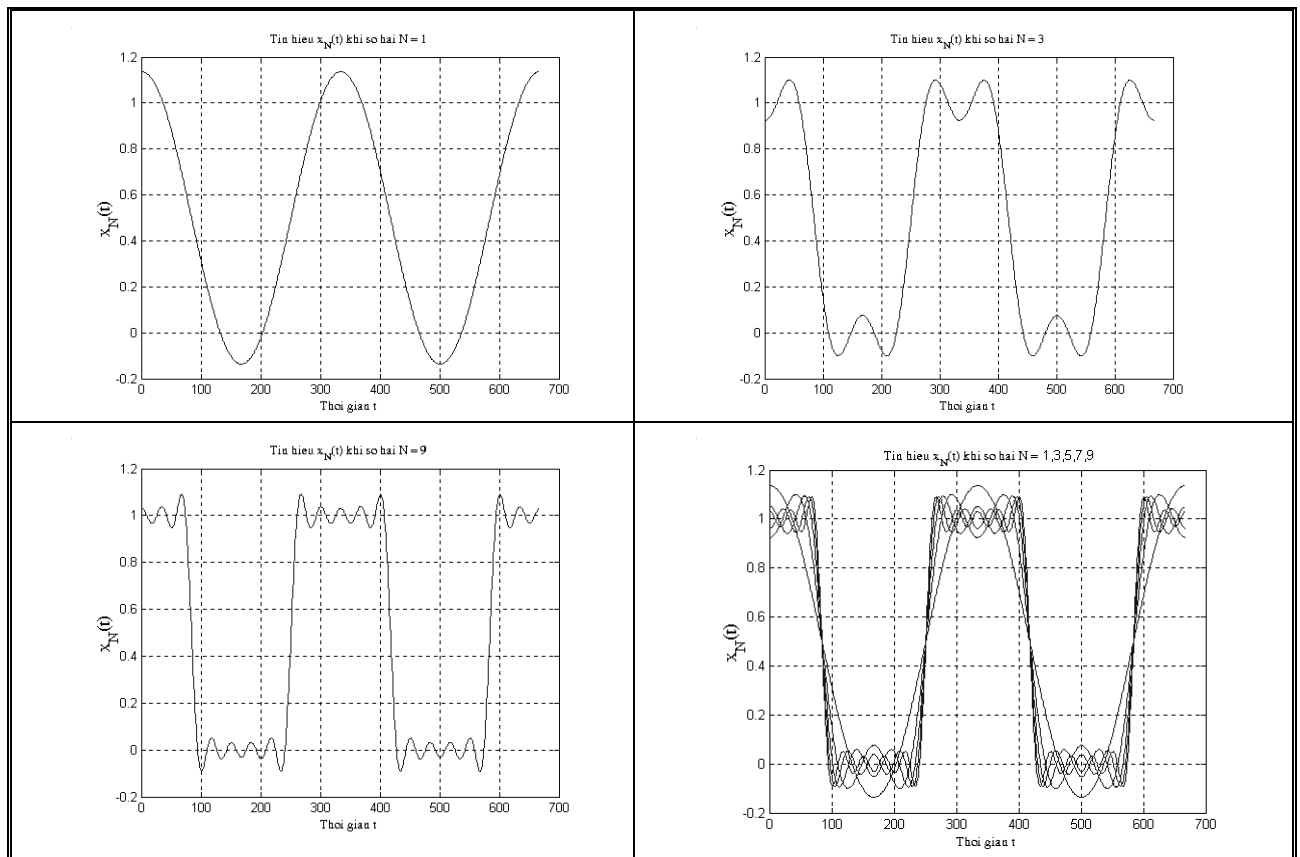
có độ băng tần hữu hạn nên khi cho xung vuông qua chúng, xung này sẽ bị loại bỏ một số thành phần tần số. Vì thế cần phải xét  $x(t)$  theo số hàm điều hoà  $N$ .

Nếu ký hiệu  $N$  là số hàm điều hoà chứa trong chuỗi xung chữ nhật  $X_N(t)$ , thì  $X(t)$  được xác định bởi

$$x_N(t) = \frac{1}{2} + \sum_{\substack{n=1 \\ n \text{ lẻ}}}^N \frac{2}{n\pi} \cos\left(n\pi t + \left[(-1)^{(n-1)/2} - 1\right] \frac{\pi}{2}\right), \quad -\infty < t < \infty$$

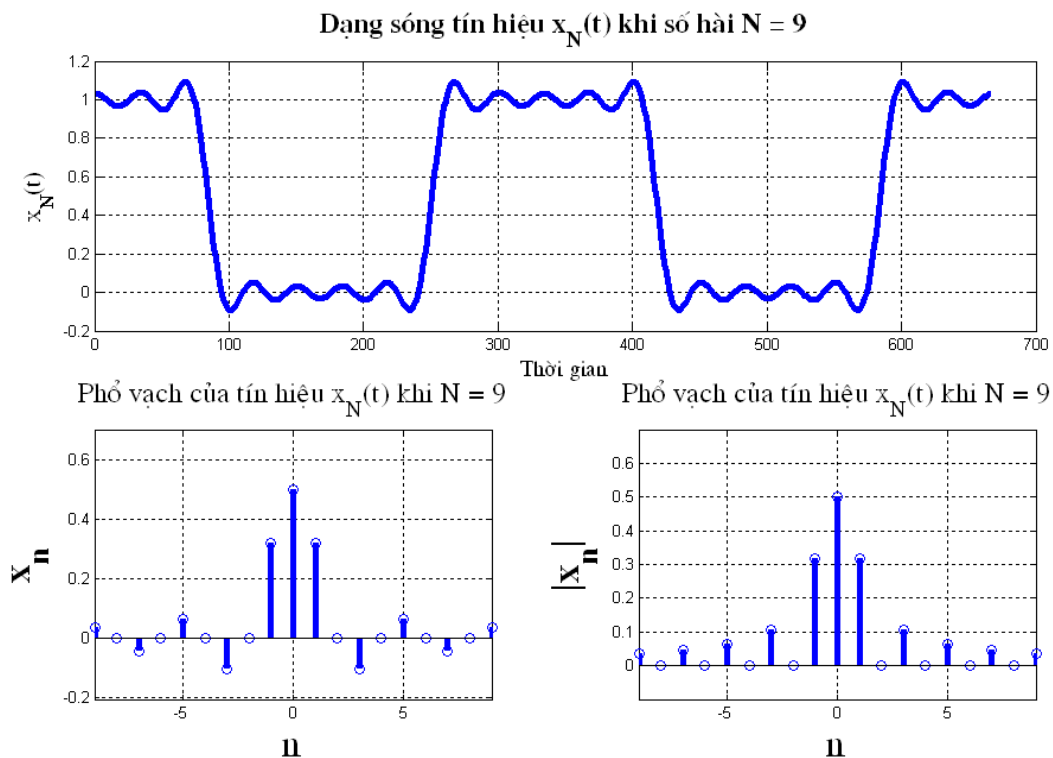
Theo định lý của Fourier, thì  $x_N(t)$  hội tụ về  $x(t)$  khi  $N \Rightarrow \infty$ . Nói cách khác,  $|x_N(t) - x(t)|$  về không với  $\forall t$  khi  $N$  tăng  $\Leftrightarrow$  khi  $N$  có giá trị càng lớn thì phép lấy xấp xỉ càng chính xác. Vẽ khảo sát xấp xỉ chuỗi Fourier cho tín hiệu này trên một chu kỳ tín hiệu với  $n = 0, 1, 3, 5, 7, 9$ . **Chương trình Matlab được cho ở File NVD4B\_sim1.m và NVD\_NC21.m**

Tổng kết các công thức mô phỏng	
Chuỗi xung chữ nhật chứa tổng <b>vô hạn</b> các hàm điều hoà	Chuỗi xung chữ nhật chứa tổng <b>hữu hạn</b> các hàm điều hoà
$x(t) = \sum_{n=-\infty}^{\infty} \underbrace{\frac{1}{2} \text{SinC}\left(\frac{n}{2}\right)}_{x_n=0, \text{ khi chẵn}} e^{j2\pi \frac{n}{2} t}$ $= \frac{1}{2} + \sum_{n=1}^{\infty} \underbrace{\frac{1}{2} \text{SinC}\left(\frac{n}{2}\right) \text{Cos}(n\pi)}_{\text{do } b_n=0, \text{ và } x_n=0 \text{ khi } n \text{ chẵn} \neq 0}$ $= \frac{1}{2} + \sum_{\substack{n=1 \\ n \text{ lẻ}}}^{\infty} \frac{2}{n\pi} \cos\left(n\pi t + \left[(-1)^{(n-1)/2} - 1\right] \frac{\pi}{2}\right)$	$x_N(t) = \frac{1}{2} + \sum_{\substack{n=1 \\ n \text{ lẻ}}}^N \frac{2}{n\pi} \cos\left(n\pi t + \left[(-1)^{(n-1)/2} - 1\right] \frac{\pi}{2}\right),$ $-\infty < t < \infty$
$x_n = \frac{1}{n\pi} \sin\left(\pi \frac{n}{2}\right) = \left(\frac{1}{2}\right) \times \frac{\sin\left(\pi \frac{n}{2}\right)}{\left(\pi \frac{n}{2}\right)}$ $= \frac{1}{2} \text{SinC}\left(\frac{n}{2}\right),$ $= \begin{cases} 0, & n = \pm 2, \pm 4, \dots \\ \frac{1}{n\pi} (-1)^{\left  \frac{(n-1)}{2} \right }, & n = \pm 1, \pm 3, \dots \end{cases}$	
$\text{SinC}(x) = \frac{\sin(\pi x)}{\pi x}$	



b) Vẽ phổ rời rạc của  $x(t)$ .

Ta lưu ý rằng,  $x_n$  là thực. Vì vậy, phụ thuộc vào dấu của nó, mà pha bằng 0 hay  $\pi$ . Độ lớn của  $x_n = \frac{1}{2} \left| \text{SinC} \left( \frac{n}{2} \right) \right|$ . Phổ rời rạc của  $x(t)$  được cho ở hình kết quả mô phỏng dưới đây.



Dạng sóng và phổ rời rạc của tín hiệu  $x_N(t)$

```

function y = NVD4B_sim1
%file NVD4B_sim1.m
% close all;
clear all;
t = -2:6/1000:2;
N = input('Nhap so hai = ');
c0 = 0.5;
w0 = pi;
xN = c0*ones(1,length(t)); % DC Compoment
for k=1:2:N % Odd function
    theta = ((-1)^((k-1)/2)-1)*pi/2;
    xN = xN + 2/k/pi*cos(k*w0*t + theta);
end
%=====
h1 = figure(1)
set(h1,'color','c','Name','H4B.1.3: NVD');
subplot(2,2,1:2);
plot(1:length(xN),xN,'LineWidth',2);
xlabel('thời gian','FontName','.VnTime','FontSize',12);
ylabel('x_N(t)','FontName','.VnTime','FontSize',16);
title(['Dạng sóng tín hiệu x_N(t) khi số hài N = ',...
    num2str(N)],'FontName','.VnTime','FontSize',14);
grid on;

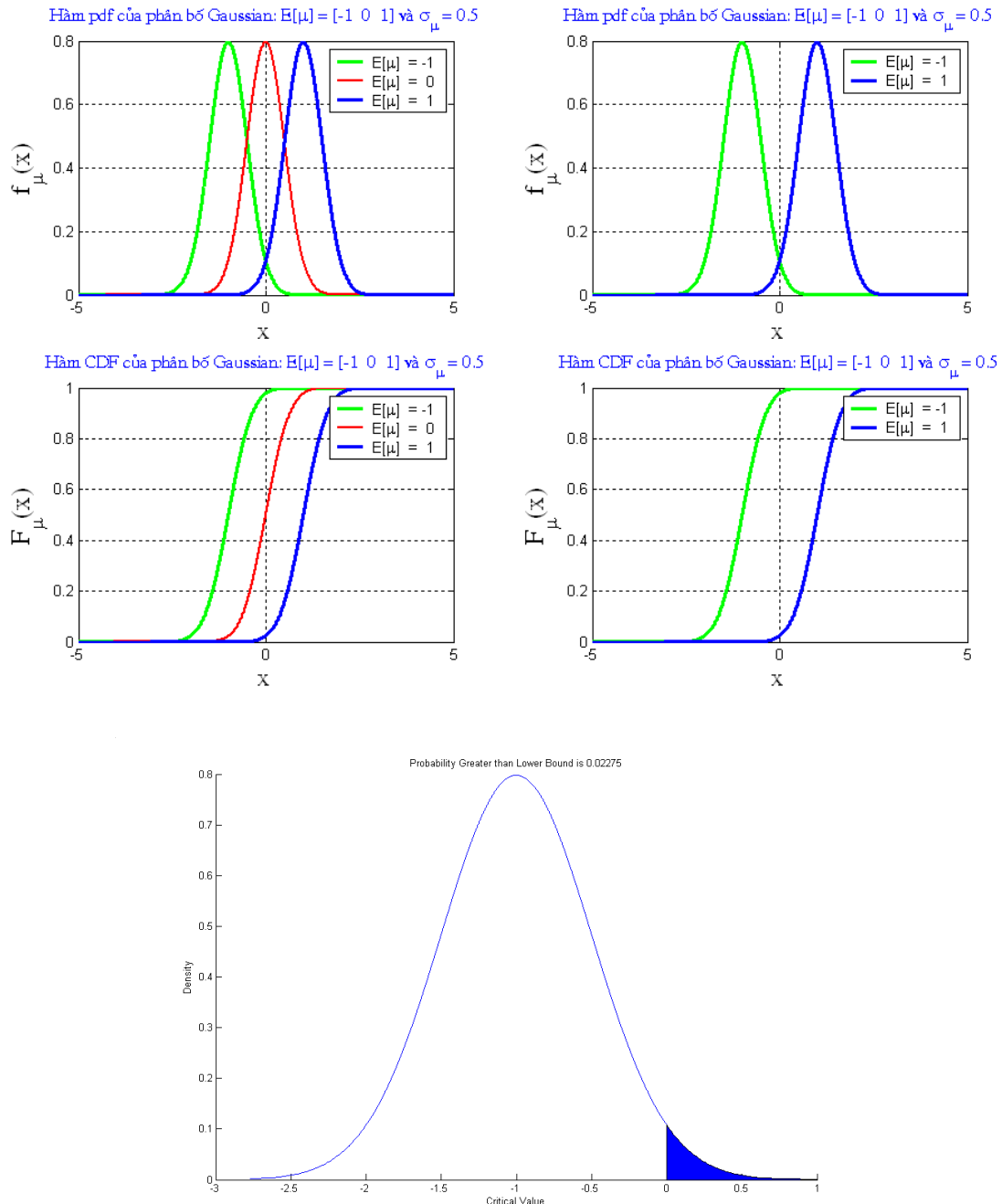
n = [-N:1:N];
x_n1 = 0.5*sinc(n/2);
x_n2 = 0.5*abs(sinc(n/2));
subplot(2,2,3);
% stem(n,x_n1,'k','LineWidth',2);
stem(n,x_n1);
xlabel('n','FontName','.VnTime','FontSize',14);
ylabel('x_n','FontName','.VnTime','FontSize',16);
axis([min(n) max(n) min(x_n1)-0.1 max(x_n1)+0.2])
title(['Phổ vạch của tín hiệu x_N(t) khi N = ',...
    num2str(N)],'FontName','.VnTime','FontSize',12);
subplot(2,2,4)
% stem(n,x_n2,'k','LineWidth',2);
stem(n,x_n2);
xlabel('n','FontName','.VnTime','FontSize',14);
ylabel('|x_n|','FontName','.VnTime','FontSize',16);
title(['Phổ vạch của tín hiệu x_N(t) khi N = ',...
    num2str(N)],'FontName','.VnTime','FontSize',12);
axis([min(n) max(n) min(x_n2)-0.1 max(x_n2)+0.2])
grid on;
%=====
h2 = figure(2); % Compared figure
set(h2,'color','g','Name','H4B.1.3: NVD');
subplot(211)
plot(1:length(xN),xN,'r','LineWidth',3);
xlabel('Thời gian','FontName','.VnTime','FontSize',12);
ylabel('x_N(t)','FontName','.VnTime','FontSize',16);
title(['Dạng sóng tín hiệu x_N(t) khi số hài
N=',num2str(N)],'FontName','.VnTime','FontSize',14);
grid on;
subplot(212)
plot(1:length(xN),xN,'LineWidth',1.5);
xlabel('Thời gian','FontName','.VnTime','FontSize',12);
ylabel('x_N(t)','FontName','.VnTime','FontSize',16);
title(['So sánh dạng sóng tín hiệu x_N(t) khi số hài khác
nhau','FontName','.VnTime','FontSize',14);
grid on;
hold on;

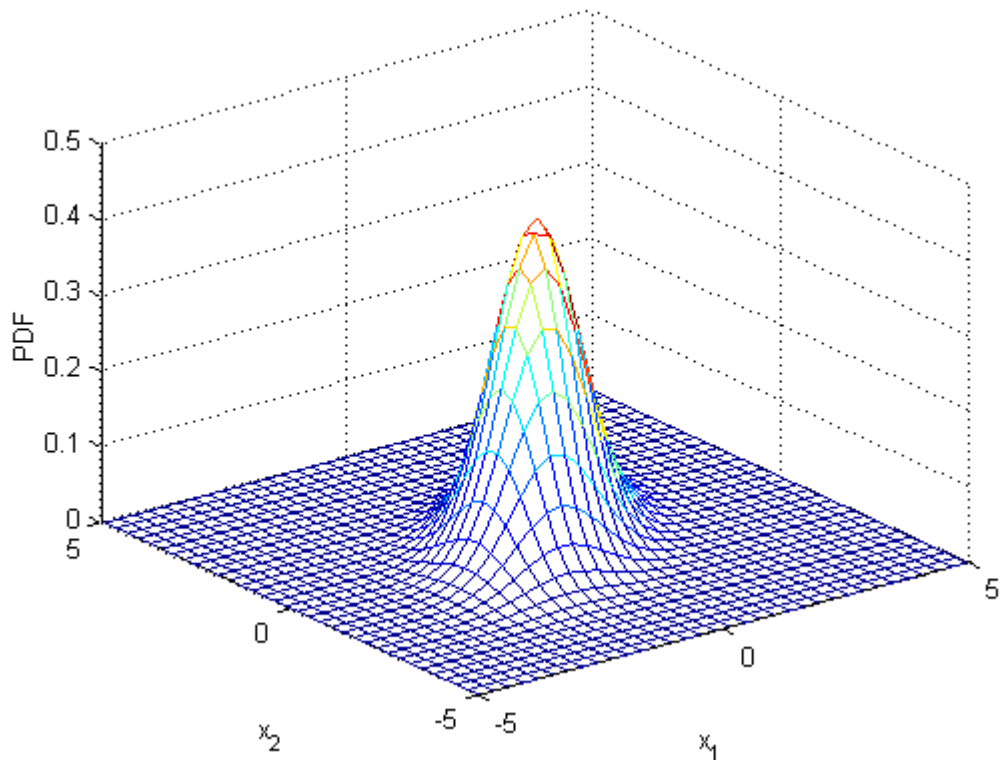
```

### Bài 15:

Hãy đọc mã chương trình Matlab dưới đây, để hiểu phân bố Gauss, chạy chương trình, thay đổi các tham số đặc trưng (trị trung bình và phương sai) của phân bố này và phân tích kết quả mô tả hàm.

### Hướng dẫn giải:





**Hàm phân bố xác suất CDF:**

```
x = 0:10;
y = unidcdf(x,10);
stairs(x,y) % Stairstep plot
set(gca,'Xlim',[0 11]);
xlabel('x');
ylabel('CDF');
title('Khảo sát hàm phân bố xác suất tích lũy CDF phân bố đều',
'FontName','.VnTime');
grid on;
```

**Hàm mật độ xác suất pdf:**

```
x = 0:10;
y = unidpdf(x,10);
stairs(x,y) % Stairstep plot
set(gca,'Xlim',[0 11]);
xlabel('x');
ylabel('PDF');
grid on;
```

Hàm **mfcnormspec**: Vẽ hàm mật độ xác suất trong khoảng xác định.

**function [p, h] = mfcnormspec(specs,mu,sigma)**

**p = mfcnormspec(specs,mu,sigma)** thực hiện vẽ hàm mật độ xác suất giữa các giới hạn trên và dưới được xác định bởi tham số specs (là vector hai phần tử xác định khoảng giá trị cần khảo sát), trong đó mu & sigma là các tham số của phân bố cần vẽ. Nếu dùng **[p,h] = mfcnormspec(specs,mu,sigma)**, thì hàm trả lại giá trị xác suất P của mẫu nằm trong khoảng giới hạn trên và giới hạn dưới và h để điều khiển các đối tượng đường vẽ.

```
if prod(size(specs)) ~= 2,
    error('Yêu cầu đối số thứ nhất là một vector hai phần tử');
end
```

```
lb = specs(1);
ub = specs(2);
if lb > ub
```

```
lb = specs(2);
ub = specs(1);
end

if lb == -Inf & ub == Inf
    error('Vector xác định giới hạn phải có ít nhất một phần tử hữu hạn');
end

if nargin < 2
    mu = 0;
    sigma = 1;
end

if max(size(mu)) > 1 | max(size(sigma)) > 1,
    error('Yêu cầu các đối số đầu vào thứ 2 & 3 vô hướng ');
end

prob = (0.0002:0.0004:0.9998)';

x = norminv(prob,mu,sigma);
y = normpdf(x,mu,sigma);
if lb == -Inf,
    p = normcdf(ub,mu,sigma);
elseif ub == Inf,
    p = 1 - normcdf(lb,mu,sigma);
else
    p = diff(normcdf([lb ub],mu,sigma));
end
nspecfig = figure;
nspecaxes = axes;
set(nspecaxes, 'Parent', nspecfig);
set(nspecaxes, 'Nextplot', 'add');
hh = plot(x,y,'b-');
xl = get(nspecaxes, 'Xlim');
lbinf = isinf(lb);
ubinf = isinf(ub);
if lbinf,
    lb = xl(1);
    yll = [0; eps];
else
    yll = normpdf(lb,mu,sigma);
    yll = [0; yll];
end
if ubinf,
    ub = xl(2);
    yul = [eps; 0];
else
    yul = normpdf(ub,mu,sigma);
    yul = [yul; 0];
end

l1 = [lb; lb];
ul = [ub; ub];

if ubinf
    title(['Xác suất lớn hơn giới hạn dưới =',num2str(p)], 'FontName', '.VnTime', 'color', 'b', 'FontSize', 12);
    k = find(x > lb);
    hh1 = plot(l1,yll,'b-');
elseif lbinf
    title(['Xác suất nhỏ hơn giới hạn trên =',num2str(p)], 'FontName', '.VnTime', 'color', 'b', 'FontSize', 12);
    k = find(x < ub);
    hh1 = plot(ul,yul,'b-');
```

```
else
    title(['Xác suất nằm trong khoảng giới hạn trên & dưới =', num2str(p)], 'FontName', '.VnTime', 'color', 'b', 'FontSize', 12);
    k = find(x > lb & x < ub);
    hh1 = plot(ll, yll, 'b-', ul, yul, 'b-');

end
xfill = x(k);
xfill = [ll; xfill; ul];
yfill = [yll; y(k); yul];
fill(xfill, yfill, 'b');

if nargin == 2
    h = [hh; hh1];
end

xaxis = reffline(0,0);
set(xaxis, 'Color', 'k');
ylabel('Mật độ xác suất', 'FontName', '.VnTime');
xlabel('Giá trị tới hạn', 'FontName', '.VnTime');
```

### Hàm chính

#### **function y=CS22**

```
mx = input('Nhap vector trung binh duoi dang MT cot [0 0] = ');
```

```
mx=mx';
```

```
Cx = input('Nhap ma tran hiệp biến Cx [1 1/2;1/2 1] = ');
```

```
x=multi_gp(mx,Cx);
```

```
% Tính pdf của (x1,x2)
```

```
delta = 0.3;
```

```
x1=-5:delta:5;
```

```
x2=-5:delta:5;
```

```
for i=1:length(x1)
```

```
    for j=1:length(x2)
```

```
        f(i,j)=(1/((2*pi)*det(Cx)^1/2))*exp((-1/2)*([x1(i) x2(j)] - mx')*inv(Cx)*([x1(i);x2(j)] -mx)));
```

```
    end
```

```
end
```

```
% Vẽ pdf
```

```
mesh(x1,x2,f);
```

```
xlabel('x_1');
```

```
ylabel('x_2');
```

```
zlabel('pdf')
```

#### **function [x]=multi\_gp(m,C)**

```
% MULTI_GP tạo quá trình ngẫu nhiên Gauss nhiều biến có trung bình m (vector cột) và ma trận hiệp biến C
```

```
N=length(m);
```

```
for i=1:N
```

```
    y(i)=gngauss;
```

```
end
```

```
y=y.';
```

```
x=sqrt(C)*y + m;
```

```
%-----
```

#### **function [gsrv1,gsrv2] = gngauss(m,sgma)**

```
% GNGASS tạo hai biến ngẫu nhiên độc lập có trung bình m và độ lệch chuẩn
sgma. Nếu một trong hai đối số vào không có thì gán trung bình là '0'. Nếu
có trung bình hoặc phương sai thì tạo hai biến ngẫu nhiên Gauss tiêu chuẩn

if nargin==0,
    m=0;
    sgma=1;
elseif nargin==1,
    sgma=m;
    m=0;
end;
u=rand; % Biến ngẫu nhiên phân phối đều trong khoảng (0,1)
z=sgma*(sqrt(2*log(1/(1-u)))); % Biến ngẫu nhiên phân phối Rayleigh.
u=rand; % Biến ngẫu nhiên phân phối đều trong khoảng (0,1)
khác
gsrv1=m+z*cos(2*pi*u);
gsrv2=m+z*cos(2*pi*u);
```

### Chương trình chính:

```
x = 0:0.1:3;
p = mfcraylpdf(x,1);
plot(x,p);
xlabel('x');
ylabel('pdf');
title('Khảo sát hàm mật độ xác suất của phân bố Rayleigh',
'FontName','.VnTime');
grid on;
```

### Chương trình con:

Xây dựng hàm mật độ xác suất cho phân bố Rayleigh:

**function Y = mfcraylpdf(x,b)**

Y = mfcraylpdf(X,B) thực hiện tính pdf của phân bố Rayleigh tại mỗi giá trị trong X cùng với các tham số tương ứng trong B. Các đầu vào Vector or matrix đối với X & B phải có cùng kích thước, cũng là kích thước của Y. Một đầu vào vô hướng đối với X hoặc B được khai triển thành ma trận hằng số với các chiều giống với các đầu vào khác.

Hàm được viết gọn

$$y = f(x|b) = \frac{x}{b^2} e^{\left(\frac{-x^2}{2b^2}\right)}$$

```
if nargin < 1,
    error('Yêu cầu ít nhất có một đối số đầu vào');
end
[errorcode x b] = distchck(2,x,b);
if errorcode > 0
    error('Yêu cầu các đối số không phải là vô hướng để phù hợp về kích thước');
end
% Khởi tạo Y =0.
y=zeros(size(x));
% Trở về NaN Nếu B không phải là số dương.
k1 = find(b <= 0);
if any(k1)
    tmp = NaN;
    y(k1) = tmp(ones(size(k1)));
end
k=find(b > 0 & x >= 0);
if any(k),
    xk = x(k);
    bk = b(k);
    y(k) = (xk./ bk.^ 2).* exp(-xk.^ 2./ (2*bk.^ 2));
end
end
```



**Chương trình chính:**

```
x = 0:0.1:3;
p = mfcraylcdf(x,1);
plot(x,p);
xlabel('x');
ylabel('CDF');
title('Khảo sát hàm phân bố tích lũy xác suất của phân bố Rayleigh',
'FontName','.VnTime');
```

**Chương trình con:**

Xây dựng hàm phân bố xác suất tích lũy cho phân bố Rayleigh:

**function p = mfcraylcdf(x,b)**

P = mfcraylcdf(X,B) thực hiện tính CDF của phân bố Rayleigh tại mỗi giá trị trong X cùng với các tham số tương ứng trong B. Các đầu vào Vector or matrix đối với X & B phải có cùng kích thước, cũng là kích thước của P. Một A đầu vào vô hướng đối với X hoặc B được khai triển thành ma trận hằng số với các chiều giống với các đầu vào khác.

Hàm được viết gọn

$$y = F(x|b) = \int_0^x \frac{t}{b^2} e^{\left(\frac{-t^2}{2b^2}\right)} dt$$

```
if nargin < 1,
    error('Yêu cầu ít nhất có một đối số đầu vào');
end
[errorcode x b] = distchck(2,x,b);
if errorcode > 0
    error('Yêu cầu các đối số không phải là vô hướng để phù hợp về kích thước');
end
% Khởi tạo P = 0.
p=zeros(size(x));
% Trở về NaN nếu B không phải là số dương.
k1 = find(b <= 0);
if any(k1)
    tmp = NaN;
    p(k1) = tmp(ones(size(k1))));
end
k=find(b > 0 & x >= 0);
if any(k),
    xk = x(k);
    bk = b(k);
    p(k) = 1 - exp(-xk.^ 2./ (2*bk.^ 2));
end
end
```

**Bài 16:**

Hãy đọc mã chương trình Matlab dưới đây, làm sáng tỏ phổ công suất của quá trình ngẫu nhiên bằng cách chạy chương trình, thay đổi các tham số cho chương trình, phân tích, biểu diễn kết quả và phân tích kết quả.

**Hướng dẫn giải**

Xét các đặc tính phổ của các tín hiệu PAM, tín hiệu PAM ở đầu vào của kênh truyền được thể hiện như sau:

$$v(t) = \sum_{n=-\infty}^{\infty} a_n g(t-nT) \quad (1)$$

trong đó:  $\{a_n\}$  là chuỗi biên độ tương ứng của các ký hiệu thông tin;  $g(t)$  là dạng sóng xung;  $T$  là khoảng thời gian ký hiệu. Mỗi phần tử của chuỗi  $\{a_n\}$  được chọn từ một trong các giá trị biên độ có thể có, đó là.

$$A_m = (2m - M + 1)d \quad m = 0, 1, 2, \dots, M-1. \quad (2)$$

trong đó:  $d$  là hệ số tỉ lệ nó xác định khoảng cách Euclid giữa một cặp biên độ tín hiệu bất kỳ nào đó;  $2d$  là khoảng cách Euclid giữa các mức biên độ tín hiệu lân cận bất kỳ.

Vì chuỗi thông tin là chuỗi ngẫu nhiên, nên chuỗi biên độ  $\{a_n\}$  tương ứng với các ký hiệu thông tin cũng là ngẫu nhiên. Do đó, tín hiệu PAM  $v(t)$  là một hàm mẫu của quá trình ngẫu nhiên  $V(t)$ . Để xác định đặc tính phổ của quá trình ngẫu nhiên  $V(t)$ , ta phải ước lượng được phổ công suất.

✓ Giá trị trung bình của quá trình ngẫu nhiên  $V(t)$  là:

$$E[V(t)] = \sum_{n=-\infty}^{\infty} E(a_n)g(t-nT) \quad (3)$$

Bằng cách chọn biên độ tín hiệu đối xứng qua điểm không như được cho ở (2) và đồng xác suất, ta được  $E(a_n) = 0$ , dẫn đến  $E[V(t)] = 0$ .

✓ Hàm tự tương quan của quá trình ngẫu nhiên  $V(t)$  là:

$$R_v(t + \tau; t) = E[V(t)V(t + \tau)] \quad (4)$$

Ta biết rằng, quá trình ngẫu nhiên mà có giá trị trung bình tuần hoàn và hàm tự tương quan tuần hoàn được gọi là dừng dừng tuần hoàn (hay *Cyclostationary*). Biến thời gian  $t$  được khử bằng cách lấy trung bình  $R_v(t + \tau; t)$  trên một chu kỳ, nghĩa là.

$$\bar{R}_v(\tau) = \frac{1}{T} \int_{-T/2}^{T/2} R_v(t + \tau; t) dt \quad (5)$$

Đối với tín hiệu PAM, nó được biểu diễn như sau:

$$\bar{R}_v(\tau) = \frac{1}{T} \sum_{m=-\infty}^{\infty} R_a(m)R_g(\tau - mT) \quad (6)$$

trong đó:  $R_a(m) = E(a_n a_{n+m})$  là tự tương quan của chuỗi  $\{a_n\}$ ;  $R_g(\tau)$  được định nghĩa là:

$$R_g(\tau) = \int_{-\infty}^{\infty} g(t)g(t + \tau) dt \quad (7)$$

Phổ công suất của quá trình ngẫu nhiên  $V(t)$  là biến đổi Fourier của hàm tự tương quan trung bình  $\bar{R}_v(\tau)$ , nghĩa là.

$$\begin{aligned} S_v(f) &= \int_{-\infty}^{\infty} \bar{R}_v(\tau) e^{-j2\pi f\tau} d\tau \\ &= \frac{1}{T} S_a(f) |G(f)|^2 \end{aligned} \quad (8)$$

trong đó:  $S_a(f)$  là phổ công suất của chuỗi biên độ  $\{a_n\}$ ;  $G(f)$  là biến đổi Fourier của xung  $g(t)$ ;  $S_a(f)$  được định nghĩa là.

$$S_a(f) = \sum_{m=-\infty}^{\infty} R_a(m) e^{-j2\pi f m T} \quad (9)$$

Từ biểu thức (8) cho thấy, phổ công suất của tín hiệu PAM là một hàm phổ công suất của các ký hiệu thông tin  $\{a_n\}$  và phổ của xung  $g(t)$ .

Trong trường hợp đặc biệt, khi chuỗi  $\{a_n\}$  không tương quan, nghĩa là.

$$R_a(m) = \begin{cases} \sigma_a^2 & m = 0 \\ 0 & m \neq 0 \end{cases} \quad (10)$$

trong đó:  $\sigma_a^2 = E(a_n^2)$ , theo đó  $S_a(f) = \sigma_a^2$  với mọi  $f$  và

$$S_v(f) = \frac{\sigma_a^2}{T} |G(f)|^2 \quad (11)$$

Trong trường hợp này, phổ công suất  $V(f)$  là hoàn toàn phụ thuộc vào đặc tính phổ của xung  $g(t)$ .

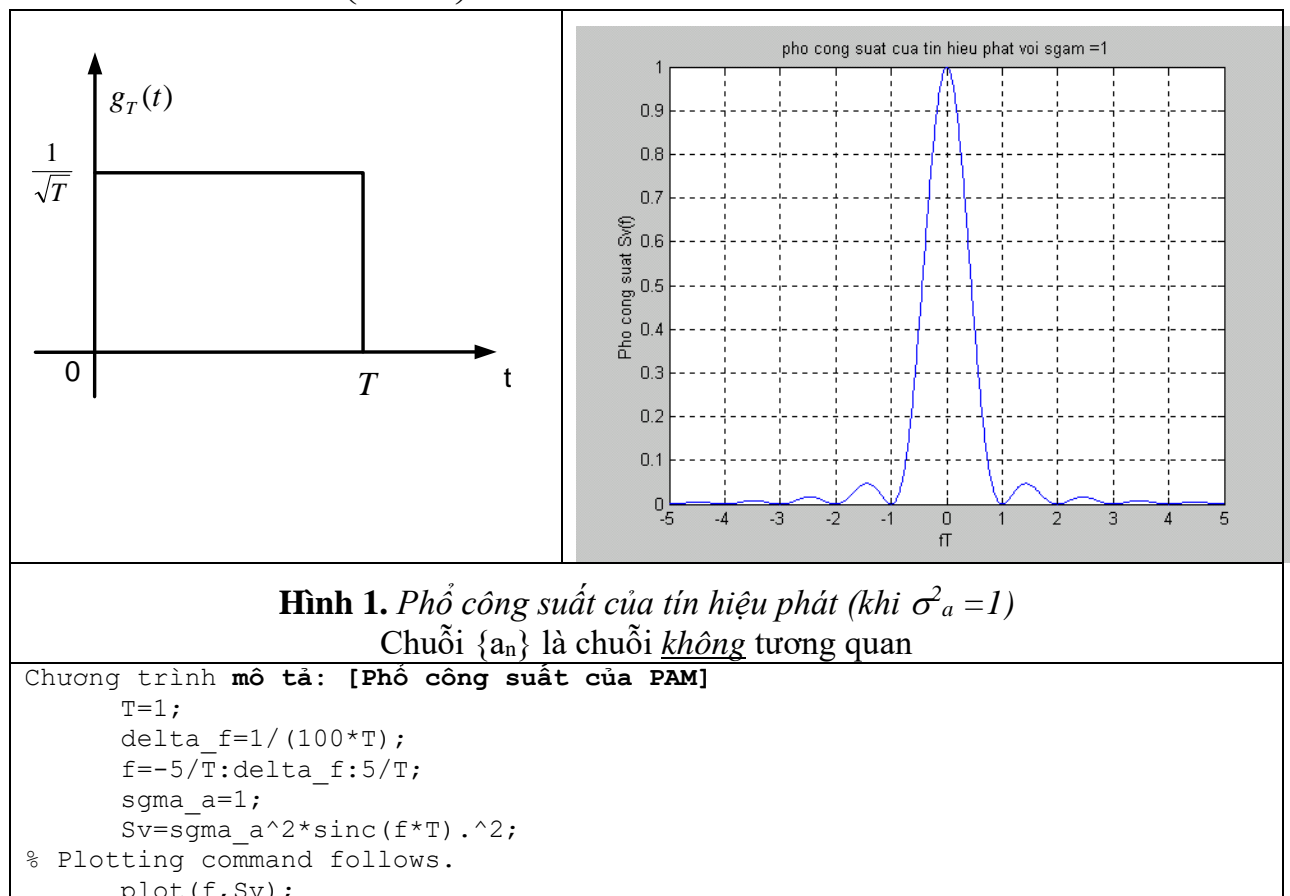
### ❖ Trường hợp 1: [Phổ công suất PAM]

Cho  $g(t)$  là xung chữ nhật ở hình vẽ và  $\{a_n\}$  là chuỗi không tương quan. Hãy xác định phổ công suất của  $V(f)$

Biến đổi Fourier của  $g(t)$  là

$$\begin{aligned} G(f) &= \int_{-\infty}^{\infty} g(t) e^{-j2\pi f t} dt \\ &= \sqrt{T} \frac{\sin \pi f T}{\pi f T} e^{-j2\pi f T} \end{aligned} \quad (12)$$

Và 
$$S_v(f) = \sigma_a^2 \left( \frac{\sin \pi f T}{\pi f T} \right)^2 = \sigma_a^2 \text{sinc}^2(fT) \quad (13)$$



❖ **Trường hợp 2: [Phổ công suất PAM].**

Giả sử hàm tự tương quan của chuỗi  $\{a_n\}$  và  $g_T(t)$  như sau:

$$R_a(m) = \begin{cases} 1 & m = 0 \\ \frac{1}{2} & m = 1, m = -1 \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

$$\text{và} \quad g_T(t) = \begin{cases} 1/\sqrt{T}, & 0 \leq t \leq T \\ 0, & t \neq \end{cases}$$

Hãy ước tính phổ công suất của quá trình ngẫu nhiên  $V(t)$ .

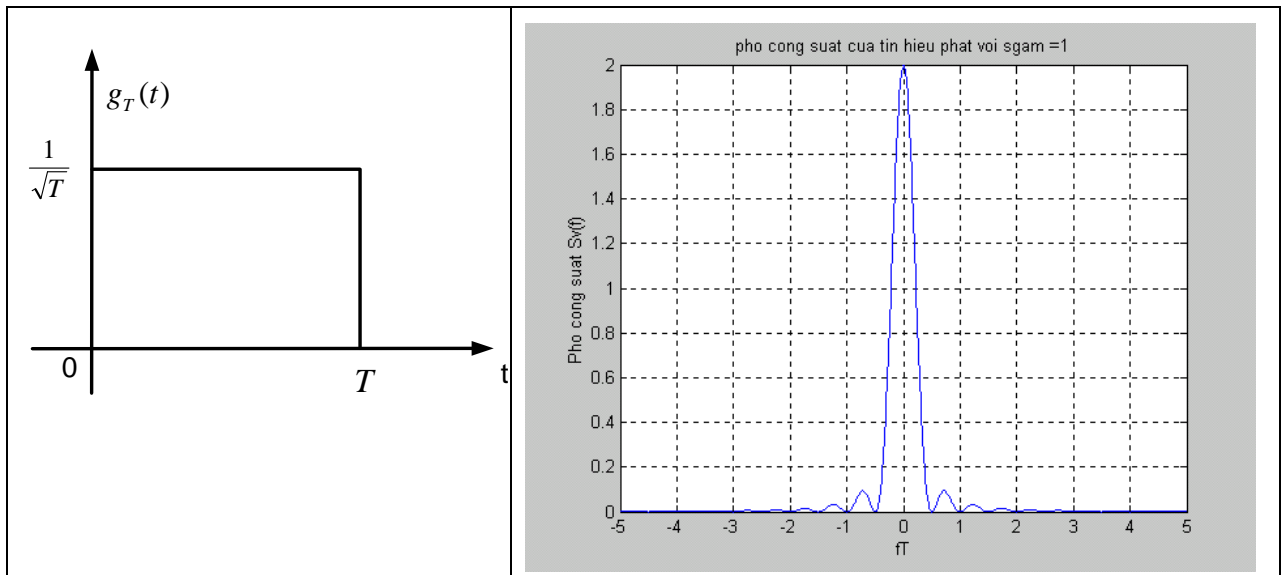
Phổ công suất của tín hiệu PAM  $V(t)$  được xác định theo (8). Phổ công suất của chuỗi  $\{a_n\}$  được xác định theo (9).

$$\begin{aligned} S_a(f) &= 1 + \cos 2\pi fT \\ &= 2 \cos^2 \pi fT \end{aligned} \quad (15)$$

Vậy.

$$S_v(f) = 2 \cos^2 \pi fT \left( \frac{\sin \pi fT}{\pi fT} \right)^2 \quad (16)$$

Chương trình Matlab thực hiện như sau. Trong trường hợp này toàn bộ phổ công suất của tín hiệu phát  $V(t)$  hẹp hơn đáng kể so với phổ công suất trường hợp 1:



**Hình 2.** Phổ công suất của tín hiệu phát (khi  $\sigma_a^2 = 1$ )

Chuỗi  $\{a_n\}$  là chuỗi tương quan

Chương trình mô tả: [Phổ công suất của PAM]

```
T=1;
delta_f=1/(100*T);
f=-5/T:delta_f:5/T;
Sv=2*(cos(pi*f*T).*sinc(f*T)).^2;
% Plotting command follows.
plot(f,Sv);
```

**Bài 17:**

Hãy đọc mã chương trình Matlab **NVD4B\_sim7.m** dưới đây, làm sáng tỏ quá trình điều chế biên độ hai băng **DSB-AM**, quá trình dịch phổ của tín hiệu băng tần gốc lên vùng tần số vô tuyến, ảnh hưởng của tạp âm Gauss trắng cộng lên tín hiệu phát trong miền thời gian và tần số ở dạng khảo sát tín hiệu bản tin, thay đổi các tham số đầu vào chương trình, biểu diễn đánh và phân tích kết quả mô phỏng.

**Hướng dẫn giải**

Ta dùng tín hiệu bản tin  $m(t)$  sau đây làm tín hiệu điều chế

$$m(t) = \begin{cases} 1, & 0 \leq t \leq \frac{t_0}{3} \\ -2, & \frac{t_0}{3} < t \leq \frac{2t_0}{3} \\ 0, & \text{nếu khác} \end{cases}$$

Bản tin này điều chế sóng mang  $c(t) = \cos(2\pi f_c t)$  theo phương pháp điều biên hia băng DSB-AM, và ký hiệu  $u(t)$  là tín hiệu sóng mang được điều chế. Nếu  $t_0 = 0,15$  và  $f_c = 250\text{Hz}$ . Ta sẽ

- Biểu diễn tín hiệu điều chế  $u(t)$ .
- Tìm phổ của  $m(t)$  và  $u(t)$ .
- Giả sử tín hiệu bản tin là tín hiệu tuần hoàn có  $T_0 = t_0$ . Ta tìm công suất trong tín hiệu điều chế.
- Nếu cộng tạp âm vào tín hiệu điều chế sao cho  $\text{SNR} = 10\text{ dB}$ , ta tìm công suất tạp âm.

❖ **Tín hiệu bản tin được viết như sau**

$$m(t) = \Pi\left(\frac{t - t_0/6}{t_0/3}\right) - 2\Pi\left(\frac{t - t_0/3}{t_0/3}\right)$$

Vì vậy thay  $t_0 = 0,15$  và  $f_c = 250\text{Hz}$  vào ta có tín hiệu điều chế

$$u(t) = \left[ \Pi\left(\frac{t - 0,025}{0,05}\right) - 2\Pi\left(\frac{t - 0,075}{0,05}\right) \right] \cos(500\pi t)$$

❖ **Dùng quan hệ FT chuẩn (tra bảng)  $\text{FT}[\Pi(t) = \text{sinc}(t)]$  cùng với các tính chất dịch và tỉ lệ của biến đổi Fourier, ta có**

$$\begin{aligned} \text{FT}[m(t)] &= \frac{t_0}{3} e^{-j\pi f t_0/3} \text{sinc}\left(\frac{t_0 f}{3}\right) - 2 \frac{t_0}{3} e^{-j\pi f t_0} \text{sinc}\left(\frac{t_0 f}{3}\right) \\ &= \frac{t_0}{3} e^{-j\pi f t_0/3} \text{sinc}\left(\frac{t_0 f}{3}\right) (1 - e^{-j2\pi f t_0/3}) \end{aligned}$$

thay  $t_0 = 0,15$  vào ta được

$$\text{FT}[m(t)] = 0,05 e^{-0,05j\pi f} \text{sinc}(0,05f) \cdot (1 - e^{-0,1j\pi f})$$

Với tín hiệu điều chế  $u(t)$ , ta có

$$U(f) = 0,025e^{-j\pi(f-f_c)} \operatorname{sinc}[0,05(f-f_c)] \cdot (1 - 2e^{-0,1j\pi(f-f_c)}) \\ + 0,025e^{-0,05j\pi(f+f_c)} \operatorname{sinc}[0,05(f+f_c)] \cdot (1 - 2e^{-0,1j\pi(f+f_c)})$$

Vẽ phổ biên độ của tín hiệu bản tin và phổ của tín hiệu điều chế được cho ở hình dưới đây, là kết quả chạy chương trình Matlab. Lưu ý rằng, thay đổi các tham số của chương trình để khảo sát:

❖ **Công suất trong tín hiệu điều chế được cho bởi**

$$P_U = \frac{A_c^2}{2} P_m = \frac{1}{2} P_m$$

trong đó  $P_m$  là công suất trong tín hiệu bản tin

$$P_m = \frac{1}{t_0} \int_0^{2t_0/3} m^2(t) dt \\ = \frac{1}{t_0} \left( \frac{t_0}{3} + \frac{4t_0}{3} \right) = \frac{5}{3} = 1,666$$

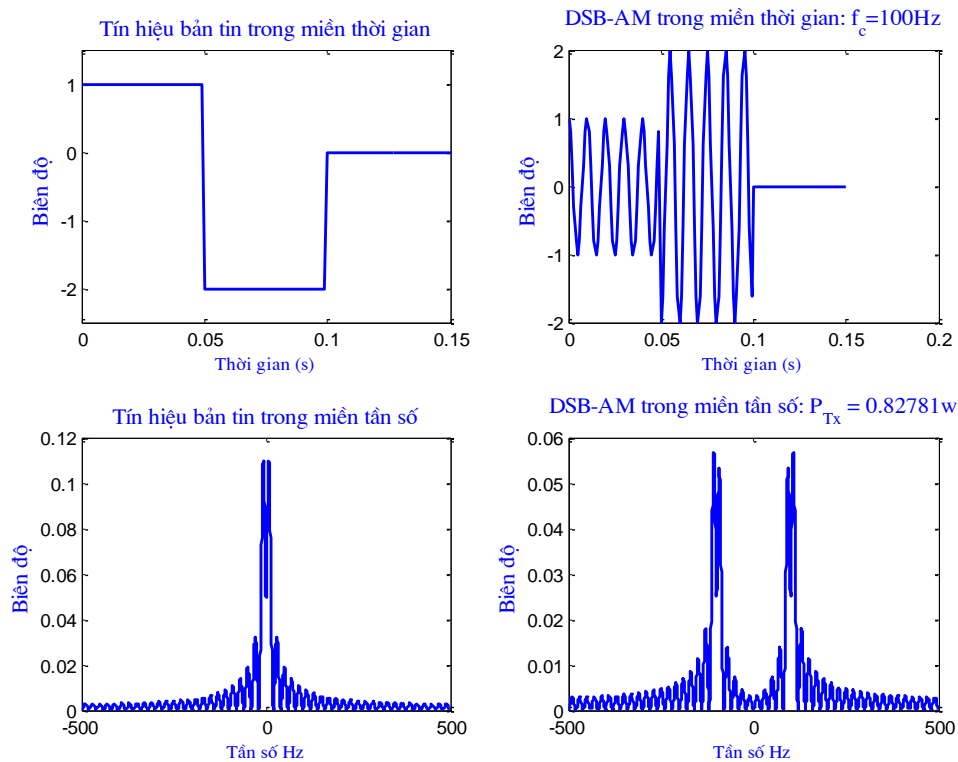
và  $P_U = \frac{P_m}{2} = 0,833$

❖ **Công suất tạp âm**

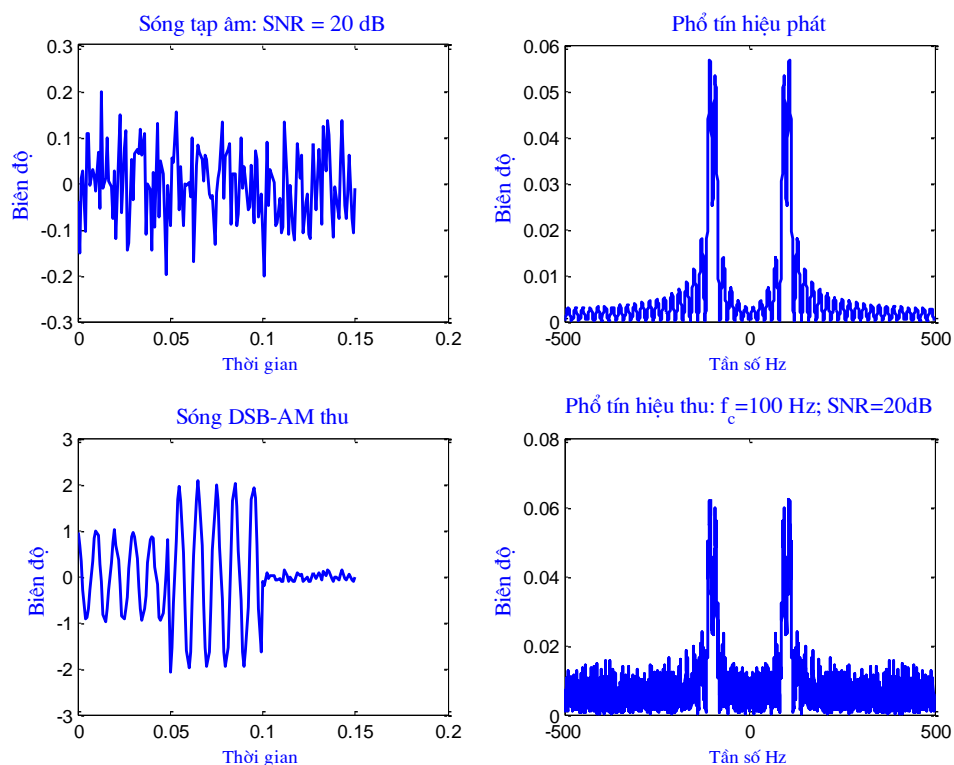
$$10 \log_{10} \left( \frac{P_R}{P_n} \right) = 10$$

hoặc  $P_R = P_U = 10P_n, \Rightarrow P_n = P_U/10 = 0,0833$

Chương trình Matlab mô phỏng điều chế DSB-AM được cho bởi **NVD4B\_sim7.m**. Các kết quả mô phỏng được cho ở các hình 1 và hình 2. Để được tường minh, ta thay đổi các giá trị của các tham số khi chạy chương trình như: tần số sóng mang, SNR,...



**Hình 1.** Tín hiệu bản tin và tín hiệu đầu ra bộ điều chế trong các miền thời gian và tần số



**Hình 2.** Mô phỏng tạp âm, tín hiệu thu trong các miền thời gian và tần số

```
function NVD4B_sim7
% demonstration for DSB-AM modulation.
%=====
clc;
clear all;
```

```

close all;
%=====
t0      = .15;           % signal duration
ts      = 0.001;        % sampling interval
fc      = input('Nhập tần số sóng mang = '); % Carrier frequency
snr      = input('Nhập SNR = ');
fs      = 1/ts;         % Sampling frequency
df      = 0.3;          % desired frequency resolution
t       = [0:ts:t0];    % time vector
snr_lin = 10^(snr/10);  % Linear SNR
%=====
% message signal
m       = [ones(1,t0/(3*ts)), -2*ones(1,t0/(3*ts)), zeros(1,t0/(3*ts)+1)];
c       = cos(2*fc*pi.*t); % Carrier signal
u       = m.*c;          % modulation signal
[M,m,df1] = fftseq(m,ts,df); % Fourier Transform
M        = M/fs;        % scaling
[U,u,df1] = fftseq(u,ts,df); % Fourier Transform
U        = U/fs;        % scaling
[C,c,df1] = fftseq(c,ts,df); % Fourier Transform
C        = C/fs;        % scaling
f        = [0:df1:df1*(length(m)-1)]-fs/2; % frequency vector
signal_power = spower(u(1:length(t))); % power in modulated signal
noise_power  = signal_power/snr_lin; % compute noise power
noise_std    = sqrt(noise_power); % Compute noise standard deviation
noise       = noise_std*randn(1,length(u)); % Generate noise
r           = u + noise; % add noise to the modulated signal
[R,r,df1]   = fftseq(r,ts,df); % spectrum of the signal + noise
R           = R/fs; % scaling
%=====
h1_27 = figure(1)
set(h1_27,'name','H1.27: NVD')
%=====
% the message signal in time domain
subplot(221);
plot(t,m(1:length(t)), 'LineWidth',2);
xlabel('Thời gian (s)', 'FontName', '.VnTime', 'color', 'b', 'FontSize',10);
ylabel('Biên độ ', 'FontName', '.VnTime', 'color', 'b', 'FontSize',12);
title('Tín hiệu bản tin trong miền thời gian',...
'FontName', '.VnTime', 'color', 'b', 'FontSize',12);
axis([min(t) max(t) min(m)-0.5 max(m)+0.5])
%=====
% the message signal in frequency domain
subplot(223);
plot(f,abs(fftshift(M)), 'LineWidth',1.5);
xlabel('Tần số Hz', 'FontName', '.VnTime', 'color', 'b', 'FontSize',10);
ylabel('Biên độ ', 'FontName', '.VnTime', 'color', 'b', 'FontSize',12);
title('Tín hiệu bản tin trong miền tần số', 'FontName', '.VnTime', 'color', 'b', 'FontSize',12);
%=====
% the modulated signal in time domain
subplot(222);
plot(t,u(1:length(t)), 'LineWidth',1.5);
xlabel('Thời gian (s)', 'FontName', '.VnTime', 'color', 'b', 'FontSize',10);
ylabel('Biên độ ', 'FontName', '.VnTime', 'color', 'b', 'FontSize',12);
title(['DSB-AM trong miền thời gian: f_c=', num2str(fc), 'Hz'],...
'FontName', '.VnTime', 'color', 'b', 'FontSize',12);
%=====
% the modulated signal in the frequency domain
subplot(224);

```



```
plot(f,abs(fftshift(U)), 'LineWidth',1.5);
xlabel('Tần số Hz','FontName','.VnTime','color','b','FontSize',10);
ylabel('Biên độ ','FontName','.VnTime','color','b','FontSize',12);
title(['DSB-AM trong miền tần số: P_T_x = ',num2str(signal_power),'w'],...
      'FontName','.VnTime','color','b','FontSize',12);
%=====
h1_28 = figure(2)
set(h1_28,'name','H1.28: NVD')

subplot(221);
plot(t,noise(1:length(t)), 'LineWidth',1.5);
xlabel('Thời gian','FontName','.VnTime','color','b','FontSize',10);
ylabel('Biên độ ','FontName','.VnTime','color','b','FontSize',12);
title(['Sóng tạp âm: SNR = ',num2str(snr),' dB'],...
      'FontName','.VnTime','color','b','FontSize',12);

subplot(222);
plot(f,abs(fftshift(U)), 'LineWidth',1.5);
xlabel('Tần số Hz','FontName','.VnTime','color','b','FontSize',10);
ylabel('Biên độ ','FontName','.VnTime','color','b','FontSize',12);
title('Phổ tín hiệu phát',...
      'FontName','.VnTime','color','b','FontSize',12);

subplot(223);
plot(t,r(1:length(t)), 'LineWidth',1.5);
xlabel('Thời gian','FontName','.VnTime','color','b','FontSize',10);
ylabel('Biên độ ','FontName','.VnTime','color','b','FontSize',12);
title('Sóng DSB-AM thu',...
      'FontName','.VnTime','color','b','FontSize',12);

subplot(224);
plot(f,abs(fftshift(R)), 'LineWidth',1.5);
xlabel('Tần số Hz','FontName','.VnTime','color','b','FontSize',10);
ylabel('Biên độ ','FontName','.VnTime','color','b','FontSize',12);
title(['Phổ tín hiệu thu: f_c=',num2str(fc),' Hz; SNR=',num2str(snr),'dB'],...
      'FontName','.VnTime','color','b','FontSize',12);
```

## CHƯƠNG 3

### Bài 1:

Cho một tín hiệu bốn mức  $a_i = \{-3a/2, -a/2, a/2, 3a/2\}$ ,  $i=1,2,3,4$  với thời gian truyền mỗi mức là  $T$ . Hãy biểu diễn tín hiệu này trong không gian tín hiệu.

- Tìm vector đơn vị
- Biểu diễn tín hiệu trong không gian tín hiệu

#### Hướng dẫn giải

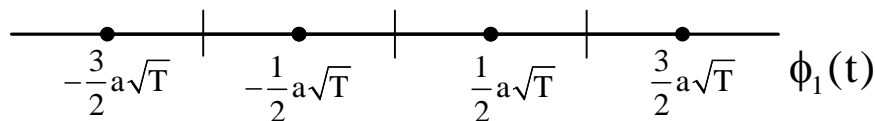
- Vector đơn vị được xác định như sau:

$$\phi_1(t) = \frac{a_3}{\sqrt{E_3}} = \frac{a/2}{\sqrt{\int_0^T (a/2)^2 dt}} = \frac{1}{\sqrt{T}}$$

Ta có:

$$s_{i1} = \int_0^T s_i(t) \frac{1}{\sqrt{T}} dt = a_i \sqrt{T}$$

- 



### Bài 2:

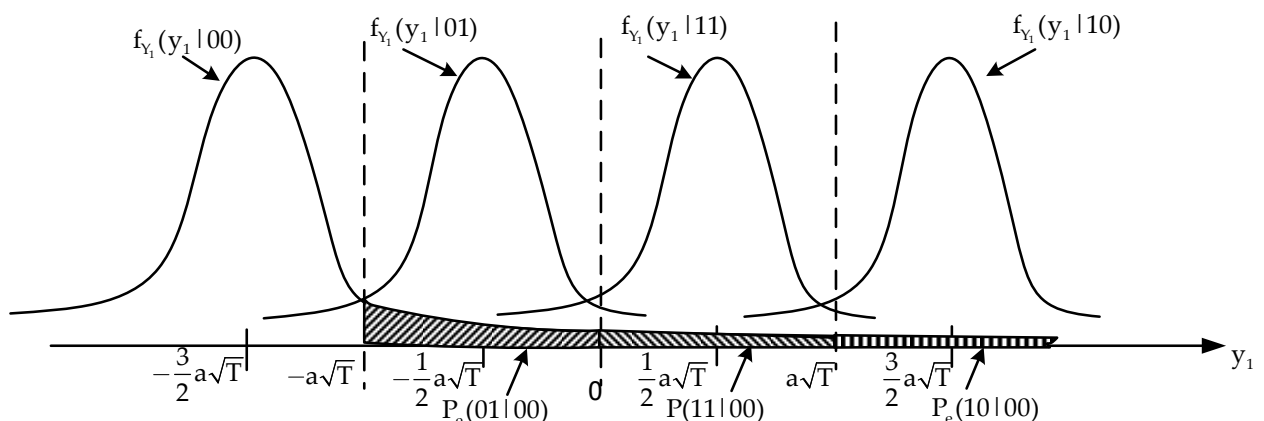
Giả sử mỗi mức của tín hiệu trong bài 1 truyền hai bit tương ứng là:  $\{00, 01, 11, 10\}$ . Tìm xác suất có điều kiện thu sai cặp bit 00.

#### Hướng dẫn giải

Xác suất có điều kiện thu sai cặp bit 00 như sau:

$$P_e(00) = P(01|00) + P(10|00) + P(11|00)$$

Các hàm mật độ xác suất có điều kiện được biểu diễn trên hình vẽ sau.



Phần diện tích tô chéo trên hình vẽ là xác suất sai có điều kiện cặp bit 00 là:

$$P_e(00) = \int_{-a\sqrt{T}}^{\infty} \frac{1}{\sqrt{\pi N_0}} \exp\left[-\frac{1}{N_0} \left(y_1 + \frac{3}{2}a\sqrt{T}\right)^2\right] dy_1$$

Đặt

$z = (y_1 + \frac{3}{2}a\sqrt{T})/\sqrt{N_0}$ , sau khi biến đổi ta được:

$$P_e(00) = \frac{1}{2} \frac{2}{\sqrt{\pi}} \int_{\frac{a}{2}\sqrt{\frac{T}{N_0}}}^{\infty} \exp[-z^2] dz = \frac{1}{2} \operatorname{erfc}\left(\frac{a}{2}\sqrt{\frac{T}{N_0}}\right)$$

**Hoặc**

Đặt

$z = \sqrt{2} \left(y_1 + \frac{3}{2}a\sqrt{T}\right) / \sqrt{N_0}$ , ta được

$$P_e(00) = \frac{1}{\sqrt{2\pi}} \int_{a\sqrt{\frac{T}{2N_0}}}^{\infty} \exp[-z^2/2] dz = Q\left(a\sqrt{\frac{T}{2N_0}}\right)$$

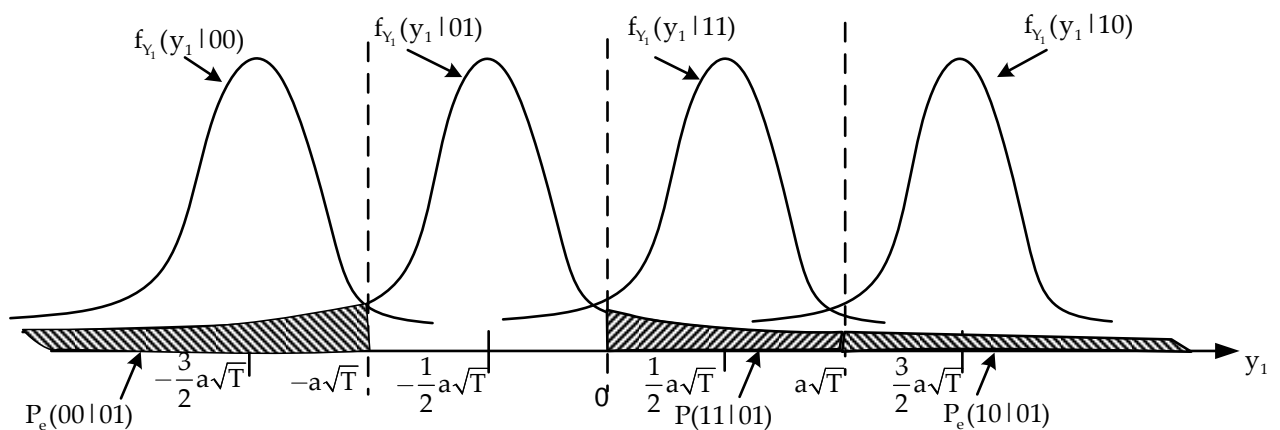
### Bài 3: (Tiếp)

Tìm xác suất có điều kiện thu sai cặp bit 01.

#### Hướng dẫn giải

$$P_e(01) = P(00|01) + P(11|01) + P(10|01)$$

Các hàm mật độ xác suất có điều kiện được biểu diễn trên hình vẽ sau.



Phần diện tích tô chéo trên hình vẽ là xác suất sai có điều kiện cặp bit 01 là:

$$\begin{aligned} P_e(01) &= \int_{-\infty}^{-a\sqrt{T}} \frac{1}{\sqrt{\pi N_0}} \exp\left[-\frac{1}{N_0} \left(y_1 + \frac{1}{2}a\sqrt{T}\right)^2\right] dy_1 + \int_0^{\infty} \frac{1}{\sqrt{\pi N_0}} \exp\left[-\frac{1}{N_0} \left(y_1 + \frac{1}{2}a\sqrt{T}\right)^2\right] dy_1 \\ &= 2 \int_0^{\infty} \frac{1}{\sqrt{\pi N_0}} \exp\left[-\frac{1}{N_0} \left(y_1 + \frac{1}{2}a\sqrt{T}\right)^2\right] dy_1 \end{aligned}$$

Đặt

$z = \left( y_1 + \frac{1}{2} a\sqrt{T} \right) / \sqrt{N_0}$ , sau khi biến đổi ta được:

$$P_e(01) = \frac{2}{\sqrt{\pi}} \int_{\frac{a}{2}\sqrt{\frac{T}{N_0}}}^{\infty} \exp[-z^2] dz = \operatorname{erfc}\left(\frac{a}{2}\sqrt{\frac{T}{N_0}}\right)$$

**Hoặc**

Đặt

$$Z = \sqrt{2} \left( y_1 + \frac{1}{2} a\sqrt{T} \right) / \sqrt{N_0}$$

$$\Rightarrow P_e(01) = \frac{2}{\sqrt{2\pi}} \int_{a\sqrt{\frac{T}{2N_0}}}^{\infty} \exp[-z^2/2] dz = 2Q\left(a\sqrt{\frac{T}{2N_0}}\right)$$

#### Bài 4: (tiếp)

Tìm xác suất có điều kiện thu sai cặp bit 11

#### Hướng dẫn giải

Phân tích như bài 3 ta được:

$$\begin{aligned} P_e(11) &= \int_{-\infty}^0 \frac{1}{\sqrt{\pi N_0}} \exp\left[-\frac{1}{N_0} \left(y_1 - \frac{1}{2} a\sqrt{T}\right)^2\right] dy_1 + \int_{a\sqrt{T}}^{\infty} \frac{1}{\sqrt{\pi N_0}} \exp\left[-\frac{1}{N_0} \left(y_1 - \frac{1}{2} a\sqrt{T}\right)^2\right] dy_1 \\ &= 2 \int_{a\sqrt{T}}^{\infty} \frac{1}{\sqrt{\pi N_0}} \exp\left[-\frac{1}{N_0} \left(y_1 - \frac{1}{2} a\sqrt{T}\right)^2\right] dy_1 \end{aligned}$$

Đặt

$z = \left( y_1 - \frac{1}{2} a\sqrt{T} \right) / \sqrt{N_0}$ , sau khi biến đổi ta được:

$$P_e(11) = \frac{2}{\sqrt{\pi}} \int_{\frac{a}{2}\sqrt{\frac{T}{N_0}}}^{\infty} \exp[-z^2] dz = \operatorname{erfc}\left(\frac{a}{2}\sqrt{\frac{T}{N_0}}\right)$$

**Hoặc**

Đặt

$$z = \sqrt{2} \left( y_1 - \frac{1}{2} a\sqrt{T} \right) / \sqrt{N_0}$$

$$P_e(11) = \frac{2}{\sqrt{2\pi}} \int_{a\sqrt{\frac{T}{2N_0}}}^{\infty} \exp[-z^2/2] dz = 2Q\left(a\sqrt{\frac{T}{2N_0}}\right)$$

#### Bài 5: (tiếp)

Tìm xác suất có điều kiện thu sai cặp bit 10.

### Hướng dẫn giải

Phân tích như bài 3 ta được:

$$\begin{aligned} P_e(10) &= \int_{-\infty}^{a\sqrt{T}} \frac{1}{\sqrt{\pi N_0}} \exp\left[-\frac{1}{N_0}\left(y_1 - \frac{3}{2}a\sqrt{T}\right)^2\right] dy_1 \\ &= \int_{2a\sqrt{T}}^{\infty} \frac{1}{\sqrt{\pi N_0}} \exp\left[-\frac{1}{N_0}\left(y_1 - \frac{3}{2}a\sqrt{T}\right)^2\right] dy_1 \end{aligned}$$

Đặt

$$Z = \left(y_1 - \frac{3}{2}a\sqrt{T}\right) / \sqrt{N_0}, \text{ sau khi biến đổi ta được:}$$

$$P_e(10) = \frac{1}{2} \frac{2}{\sqrt{\pi}} \int_{\frac{a}{2}\sqrt{\frac{T}{N_0}}}^{\infty} \exp[-z^2] dz = \frac{1}{2} \operatorname{erfc}\left(\frac{a}{2}\sqrt{\frac{T}{N_0}}\right)$$

**Hoặc**

$$Z = \sqrt{2}\left(y_1 - \frac{3}{2}a\sqrt{T}\right) / \sqrt{N_0}$$

$$P_e(10) = \frac{1}{\sqrt{2\pi}} \int_{a\sqrt{\frac{T}{2N_0}}}^{\infty} \exp[-z^2/2] dz = Q\left(a\sqrt{\frac{T}{2N_0}}\right)$$

### Bài 6: (tiếp)

Tìm xác suất thu sai ký hiệu trung bình với giả thiết xác suất phát các mức  $a_i$  bằng nhau và bằng 1/4.

### Hướng dẫn giải

$$P_e = P(00)P_e(00) + P(01)P_e(01) + P(11)P_e(11) + P(10)P_e(10)$$

trong đó  $P(xx)$  và  $P_e(xx)$  là xác suất phát ký hiệu  $xx$  và xác suất có điều kiện thu sai ký hiệu  $xx$  tương ứng.

Từ các kết quả bài 2,3,4,5 và  $P(xx) = 1/4$  ta được:

$$\begin{aligned} P_e &= [P_e(00) + P_e(01) + P_e(11) + P_e(10)]/4 \\ &= \frac{3}{4} \operatorname{erfc}\left(\frac{a}{2}\sqrt{\frac{T}{N_0}}\right) \\ &= \frac{6}{4} Q\left(a\sqrt{\frac{T}{2N_0}}\right) \end{aligned}$$

### Bài 7:

Cho một tín hiệu điều chế 4-ASK được xác định như sau:

$$s_i(t) = \sqrt{\frac{2E}{T}} a_i \cos(2\pi f_c t + \theta)$$

trong đó:  $E$  là năng lượng trên một ký hiệu  $= 2E_b$ ;  $E_b$  là năng lượng trên một bit;  $T$  là thời gian của một ký hiệu  $= 2T_b$ ;  $T_b$  là thời gian của một bit;  $i = 1, 2, 3, 4$ ;  $f_c$  tần số sóng mang;  $\theta$  là góc pha ban đầu bất kỳ không ảnh hưởng lên quá trình phân tích nên ta sẽ bỏ qua;  $a_i = \{-3, -1, 1, 3\}$ .

a) Tìm vector đơn vị

b) Biểu diễn tín hiệu trong không gian tín hiệu

### Hướng dẫn giải

$$a) \quad \phi_1(t) = \frac{\sqrt{\frac{2E}{T}} a_3 \cos(2\pi f_c t)}{\sqrt{E_3}} = \frac{\sqrt{\frac{2E}{T}} a_3 \cos(2\pi f_c t)}{\sqrt{\int_0^T \left( \sqrt{\frac{2E}{T}} a_3 \cos(2\pi f_c t) \right)^2 dt}} = \sqrt{\frac{2}{T}} \cos(2\pi f_c t)$$

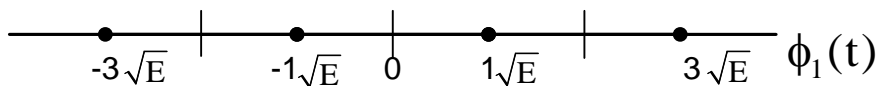
b) Ta có:

$$s_{i1} = \int_0^T s_i(t) \sqrt{\frac{2}{T}} \cos(2\pi f_c t) dt = \int_0^T \sqrt{\frac{2E}{T}} a_i \cos(2\pi f_c t) \sqrt{\frac{2}{T}} \cos(2\pi f_c t) dt$$

Sử dụng biến đổi lượng giác:

$$\sqrt{\frac{2E}{T}} a_i \cos(2\pi f_c t) \sqrt{\frac{2}{T}} \cos(2\pi f_c t) = \frac{\sqrt{E}}{T} a_i [1 + \cos(4\pi f_c t)]$$

ta được  $s_{i1} = \sqrt{E} a_i$  với biểu thị trên hình vẽ như sau:



### Bài 8: (Tiếp)

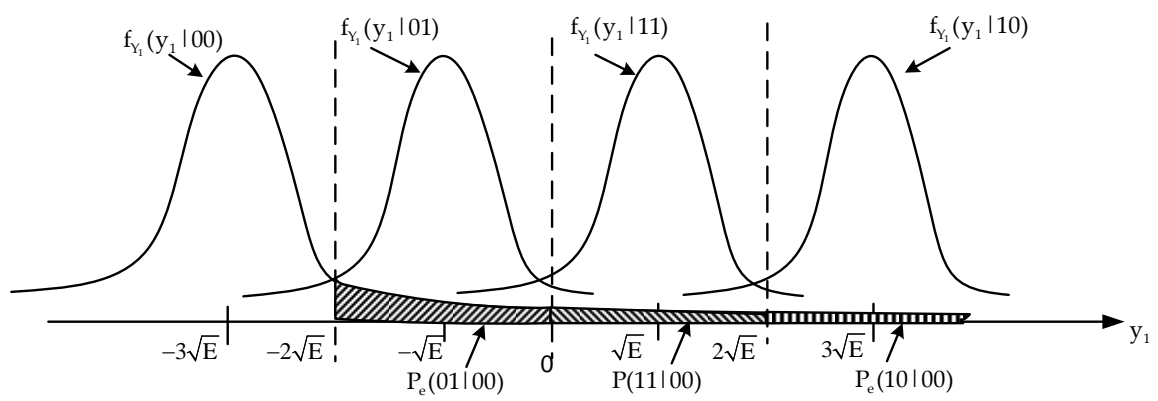
Giả sử mỗi mức của tín hiệu trong bài 7 truyền hai bit tương ứng như sau  $\{00, 01, 11, 10\}$ . Tìm xác suất có điều kiện thu sai cặp bit 00.

### Hướng dẫn giải

Xác suất có điều kiện thu sai cặp bit 00 như sau:.

$$P_e(00) = P(01|00) + P(10|00) + P(11|00)$$

Các hàm mật độ xác suất có điều kiện được biểu diễn trên hình vẽ sau.



Phần diện tích tô chéo trên hình vẽ là xác suất sai có điều kiện cặp bit 00 là:

$$P_e(00) = \int_{-2\sqrt{E}}^{\infty} \frac{1}{\sqrt{\pi N_0}} \exp\left[-\frac{1}{N_0}(y_1 + 3\sqrt{E})^2\right] dy_1$$

Đặt

$z = (y_1 + 3\sqrt{E}) / \sqrt{N_0}$ , sau khi biến đổi ta được:

$$P_e(00) = \frac{1}{2} \frac{2}{\sqrt{\pi}} \int_{\sqrt{\frac{E}{N_0}}}^{\infty} \exp[-z^2] dz = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E}{N_0}}\right)$$

**Hoặc**

Đặt

$z = \sqrt{2}(y_1 + 3\sqrt{E}) / \sqrt{N_0}$ , ta được

$$P_e(00) = \frac{1}{\sqrt{2\pi}} \int_{\sqrt{\frac{2E}{N_0}}}^{\infty} \exp[-z^2/2] dz = Q\left(\sqrt{\frac{2E}{N_0}}\right)$$

### Bài 9: (Tiếp)

Tìm xác suất có điều kiện thu sai cặp bit 01.

#### Hướng dẫn giải

$$P_e(01) = P(00|01) + P(11|01) + P(10|01)$$

Các hàm mật độ xác suất có điều kiện được biểu diễn trên hình vẽ sau.

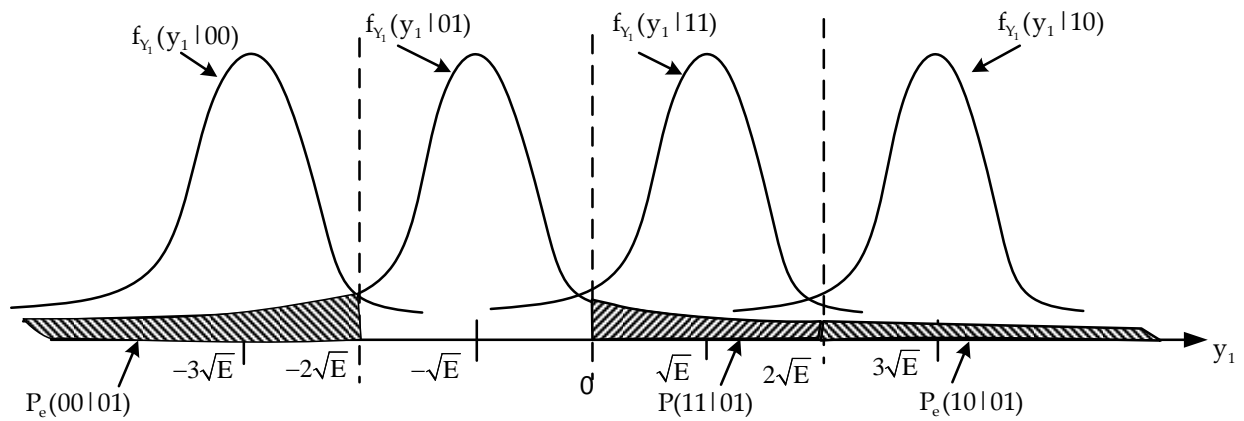
Phần diện tích tô chéo trên hình vẽ là xác suất sai có điều kiện cặp bit 01 là:

$$\begin{aligned} P_e(01) &= \int_{-\infty}^{-2\sqrt{E}} \frac{1}{\sqrt{\pi N_0}} \exp\left[-\frac{1}{N_0}(y_1 + \sqrt{E})^2\right] dy_1 + \int_0^{\infty} \frac{1}{\sqrt{\pi N_0}} \exp\left[-\frac{1}{N_0}(y_1 + \sqrt{E})^2\right] dy_1 \\ &= 2 \int_0^{\infty} \frac{1}{\sqrt{\pi N_0}} \exp\left[-\frac{1}{N_0}(y_1 + \sqrt{E})^2\right] dy_1 \end{aligned}$$

Đặt

$z = (y_1 + \sqrt{E}) / \sqrt{N_0}$ , sau khi biến đổi ta được:

$$P_e(01) = \frac{2}{\sqrt{\pi}} \int_{\sqrt{\frac{E}{N_0}}}^{\infty} \exp[-z^2] dz = \operatorname{erfc}\left(\sqrt{\frac{E}{N_0}}\right)$$



**Hoặc**

Đặt

$$z = \sqrt{2}(y_1 + \sqrt{E}) / \sqrt{N_0}$$

$$P_{e(01)} = \frac{2}{\sqrt{2\pi}} \int_{\sqrt{\frac{2E}{N_0}}}^{\infty} \exp\left[-z^2/2\right] dz = 2Q\left(\sqrt{\frac{2E}{N_0}}\right)$$

### Bài 10: (tiếp)

Tìm xác suất lỗi bit trung bình.

#### Hướng dẫn giải

Vì mỗi ký hiệu phát đồng thời hai bit nên xác suất lỗi bit trung bình sẽ bằng 1/2 xác suất lỗi ký hiệu và được tính như sau:

$$P_b = \frac{1}{2} [P(00)P_e(00) + P(01)P_e(01) + P(11)P_e(11) + P(10)P_e(10)]$$

Dựa trên các bài 8, 9 ta có thể viết:

$$P_e(00) = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E}{N_0}}\right) = Q\left(\sqrt{\frac{2E}{N_0}}\right)$$

$$P_e(01) = \operatorname{erfc}\left(\sqrt{\frac{E}{N_0}}\right) = 2Q\left(\sqrt{\frac{2E}{N_0}}\right)$$

$$P_e(11) = \operatorname{erfc}\left(\sqrt{\frac{E}{N_0}}\right) = 2Q\left(\sqrt{\frac{2E}{N_0}}\right)$$

$$P_e(10) = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E}{N_0}}\right) = Q\left(\sqrt{\frac{2E}{N_0}}\right)$$

Vì thế xác suất lỗi trung bình như sau:

$$P_e = \frac{3}{8} \operatorname{erfc}\left(\sqrt{\frac{E}{N_0}}\right) = \frac{3}{4} Q\left(\sqrt{\frac{2E}{N_0}}\right)$$



**Bài 11:**

Tìm số bit lỗi xảy ra trong một ngày đối với máy thu điều chế BPSK nhất quán hoạt động liên tục:  $R_b=10.000\text{bps}$ ;  $P=0,1 \cdot 10^{-6} \text{ W}$ ;  $N_0=10^{-11} \text{ W.Hz}^{-1}$ . Giả thiết rằng công suất và năng lượng bit được chuẩn hóa tại điện trở tải  $1 \Omega$ .

**Hướng dẫn giải**

$$E_b = P/R_b = 0,1 \cdot 10^{-10}; 2E_b/N_0 = 2; \text{ tra bảng ta được } P_b = 0,0228$$

Vậy số bit lỗi xảy ra trong một ngày bằng:

$$N_b = R_b \cdot (3600 \cdot 24) \cdot 0,0228 = 10.000(3600 \cdot 24) \cdot 0,0228 \approx 1970 \text{ bit}$$

**Bài 12:**

Một hệ thống BPSK hoạt động liên tục mắc lỗi trung bình 50000 bit lỗi trên một ngày.  $R_b=10000\text{bps}$ ,  $N_0=10^{-10} \text{ W.Hz}^{-1}$ .

a) Tìm xác suất lỗi bit

b) Tìm công suất thu tương ứng để được xác suất lỗi bit như a)

**Hướng dẫn giải**

a) 
$$P_e = \frac{50000}{10000(3600 \times 24)} = 5,79 \cdot 10^{-5}$$

b) Tra bảng hàm  $Q(x)$  trong phụ lục ta được:

$$2E_b/N_0 = 2P/(R_b N_0) = 3,2.$$

Vậy công suất thu tương ứng sẽ là:

$$P = 3,2 \times R_b N_0 / 2 = 3,2 \cdot 10000 \cdot 10^{-10} / 2 = 1,6 \cdot 10^{-4} \text{ W}$$

**Bài 13:**

Tín hiệu thu của hệ thống BPSK nhất quán được định nghĩa như sau:

$$y(t) = k \sqrt{\frac{2E_b}{T_b}} \sin(2\pi f_c t) \pm \sqrt{1-k^2} \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c t) + x(t), \quad 0 \leq t \leq T_b$$

trong đó: dấu cộng tương ứng với ký hiệu '0' và dấu trừ tương ứng với '1'; thành phần thứ nhất thể hiện sóng mang để đồng bộ máy thu với máy phát;  $T_b$  là độ rộng bit;  $E_b$  là năng lượng bit;  $x(t)$  là tạp âm Gauss trắng cộng.

a) Viết công thức liên hệ xác suất lỗi bit trung bình  $P_e$  với: xác suất phát ký hiệu 1,  $P(1)$ ; xác suất phát ký hiệu 0,  $P(0)$ ; xác suất có điều kiện  $P_e(0|1)$  là xác suất phát ký hiệu một nhưng quyết định thu ký hiệu 0; xác suất có điều kiện  $P_e(1|0)$  là xác suất phát ký hiệu 0 nhưng quyết định thu ký hiệu 1

b) Tìm các biểu thức xác định  $P_e(0|1)$  và  $P_e(1|0)$ .

**Hướng dẫn giải**

a) Công thức liên hệ xuất lỗi  $P_e$ :

$$P_e = P(1) \cdot P(0|1) + P(0) \cdot P(1|0) = 1/2 [P(0|1) + P(1|0)]$$

b) Sau tích phân để giải điều chế BPSK ta được:

$$Y_1 = \int_0^T s(t) \sqrt{\frac{2}{T}} \cos(2\pi f_c t) dt = \pm \sqrt{(1-k^2)E_b} + X_1$$

trong đó  $Y_1$  là tín hiệu sau mạch tích phân, '+' tương ứng ký hiệu '0' được phát, '-' tương ứng ký hiệu '1' được phát,  $W_1$  là tạp âm Gauss trắng cộng trung bình không.

Vì  $Y_1$  là biến ngẫu nhiên Gauss có giá trị trung bình  $\pm \sqrt{(1-k^2)E_b}$  nên:

$$f_{Y_1}(y_1|0) = \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{1}{N_0} \left(y_1 - \sqrt{(1-k^2)E_b}\right)^2\right)$$

$$f_{Y_1}(y_1|1) = \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{1}{N_0} \left(y_1 + \sqrt{(1-k^2)E_b}\right)^2\right)$$

Vậy xác suất có điều kiện mà máy thu quyết định 1 khi ký hiệu 0 được phát là:

$$\begin{aligned} P(1|0) &= \int_{-\infty}^0 f_{Y_1}(y_1|0) dy_1 = \int_{-\infty}^0 \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{1}{N_0} \left(y_1 - \sqrt{(1-k^2)E_b}\right)^2\right) dt \\ &= \int_0^{\infty} \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{1}{N_0} \left(y_1 - \sqrt{(1-k^2)E_b}\right)^2\right) dt \end{aligned}$$

và xác suất có điều kiện mà máy thu quyết định thu 0 khi ký hiệu 1 được phát:

$$P(0|1) = \int_0^{\infty} f_{Y_1}(y_1|1) dy_1 = \int_0^{\infty} \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{1}{N_0} \left(y_1 + \sqrt{(1-k^2)E_b}\right)^2\right) dt$$

Do tính đối xứng nên:  $P(1|0)=P(0|1)$  và vì thế:

$$P(1|0)=P(0|1) = \int_0^{\infty} \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{1}{N_0} \left(y_1 - \sqrt{(1-k^2)E_b}\right)^2\right) dt$$

Đặt

$$z = \sqrt{2} \left[ y_1 + \sqrt{(1-k^2)E_b} \right] / \sqrt{N_0}$$

ta được

$$P(1|0) = P(0|1) = \frac{1}{\sqrt{2\pi}} \int_{\sqrt{\frac{2(1-k^2)E_b}{N_0}}}^{\infty} \exp\left(-z^2/2\right) dt = Q\left(\sqrt{\frac{2(1-k^2)E_b}{N_0}}\right)$$

#### Bài 14: (Tiếp)

Đối với hệ thống BPSK như cho trong bài 13.

a) Chứng minh rằng xác suất lỗi trung bình bằng:  $P_e = \left( \sqrt{\frac{2E_b}{N_0} (1-k^2)} \right)$

trong đó:  $N_0$  là mật độ phổ công suất tạp âm Gauss trắng.

b) Giả thiết 10% công suất tín hiệu phát được phân bổ cho thành phần sóng mang chuẩn để đồng bộ tìm  $E_b/N_0$  để đảm bảo xác suất lỗi bit trung bình bằng  $3 \cdot 10^{-4}$ .

c) So sánh giá trị SNR hệ thống này đối với hệ thống BPSK thông thường

### Hướng dẫn giải

a) Xác suất lỗi bit trung bình được xác định như sau:

$$\begin{aligned} P_b &= P(1).P(0|1) + P(0).P(1|0) = 1/2[P(0|1)+P(1|0)] \\ &= P(0|1) = P(1|0) = Q\left(\sqrt{\frac{2(1-k^2)E_b}{N_0}}\right) \end{aligned}$$

b) Công suất trung bình chuẩn hóa cho điện trở 1Ôm của thành phần sóng mang để đồng bộ được xác định như sau;

$$P_c = \frac{1}{T_b} \int_0^{T_b} \left( k \sqrt{\frac{2E_b}{T_b}} \sin(2\pi f_c t) \right)^2 dt = k^2 P = 0,1P \rightarrow k^2 = 0,1,$$

nên công suất cho tín hiệu bản tin sẽ chiếm  $1-k^2=0,9$  tổng công suất. Rút ra:

$$u = \sqrt{\frac{2E_b}{N_0}(1-k^2)} = \sqrt{\frac{1,8E_b}{N_0}} \rightarrow \frac{E_b}{N_0} = \frac{u^2}{1,8}$$

Tra bảng trong phụ lục cho  $P_b=3.10^{-4}$  ta được  $u=3,4$ , Vì thế:

$$\frac{E_b}{N_0} = \frac{3,4^2}{1,8} = 6,42$$

c) Đối với hệ thống BPSK thông thường để đảm bảo xác suất lỗi bit  $3.10^{-4}$ , ta cần tỷ số  $E_b/N_0$  như sau:

$$\sqrt{\frac{2E_b}{N_0}} = 3,4 \rightarrow \frac{E_b}{N_0} = \frac{3,4^2}{2} = 5,78$$

Như vậy hệ thống BPSK sử dụng thành phần đồng bộ sóng mang cần có tỷ số SNR lớn hơn hệ thống thông thường là:  $(6,42)/(5,78) = 1,1$  lần

### Bài 15:

Một hệ thống BPSK có xác suất truyền bit "0" bằng xác suất truyền bit "1". Giả thiết rằng khi hệ thống đồng bộ tốt,  $E_b/N_0=9,6$  dB dẫn đến xác suất lỗi bit bằng  $10^{-5}$ . Trong trường hợp vòng khóa pha PLL bị mắc lỗi pha  $\gamma$ .

- Xác suất lỗi bit sẽ giảm cấp như thế vào nếu  $\gamma=25^\circ$
- Sai pha lớn nào sẽ dẫn đến xác suất lỗi bit bằng  $10^{-3}$

### Hướng dẫn giải

a) Sau tích phân để giải điều chế BPSK ta được:

$$\begin{aligned} Y_1 &= \int_0^T s(t) \sqrt{\frac{2}{T_b}} \cos(2\pi f_c t) dt = \int_0^T \pm \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c t) \sqrt{\frac{2}{T_b}} \cos(2\pi f_c t + \gamma) dt + X_1 \\ &= \pm \sqrt{E_b} \cos \gamma + X_1 \end{aligned}$$

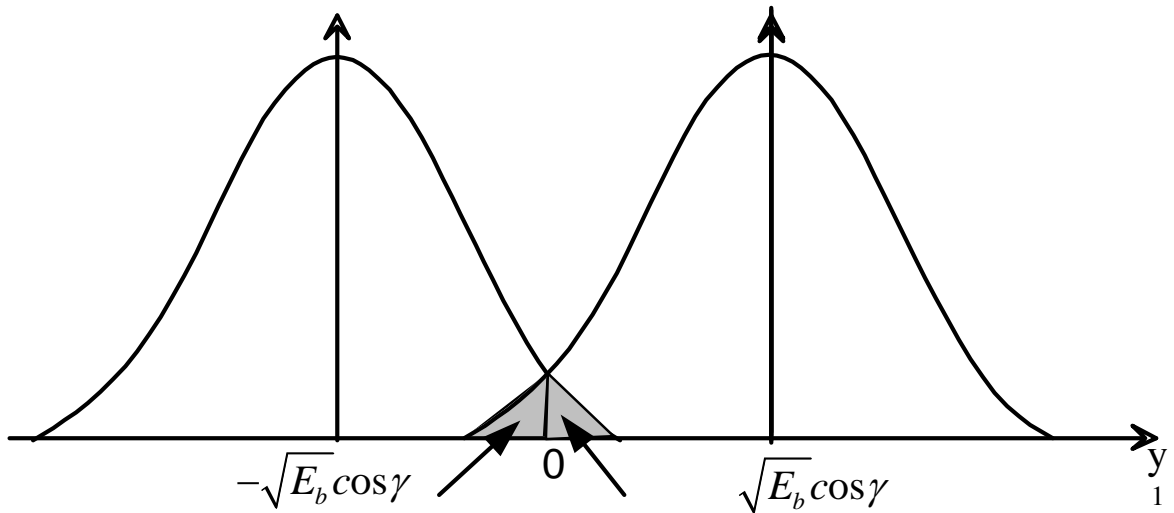
Vì  $Y_1$  là biến ngẫu nhiên Gauss có giá trị trung bình  $\pm \sqrt{E_b} \cos \gamma$  nên:

$$f_{Y1}(y_1|0) = \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{1}{N_0} (y_1 - \sqrt{E_b} \cos \gamma)^2\right)$$

$$f_{Y1}(y_1|1) = \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{1}{N_0} (y_1 + \sqrt{E_b} \cos \gamma)^2\right)$$

Ta có thể biểu diễn các hàm mật độ xác suất nói trên theo giá trị  $y_1$  như sau.

$$f_{Y1}(y_1|1) = \frac{1}{\sqrt{2\pi N_0}} e^{-(y_1 + \sqrt{E_b} \cos \gamma)^2 / N_0} \quad f_{Y1}(y_1|0) = \frac{1}{\sqrt{2\pi N_0}} e^{-(y_1 - \sqrt{E_b} \cos \gamma)^2 / N_0}$$



$$P_e(0) = \int_{-\infty}^0 \frac{1}{\sqrt{2\pi N_0}} e^{-(y_1 - \sqrt{E_b} \cos \gamma)^2 / N_0} dy_1 \quad P_e(1) = \int_0^{\infty} \frac{1}{\sqrt{2\pi N_0}} e^{-(y_1 + \sqrt{E_b} \cos \gamma)^2 / N_0} dy_1$$

Do xác suất phát bit "1" bằng xác suất phát bit "0" nên:  $P(0) = P(1) = 1/2$  và tính đối xứng của các hàm mật độ xác suất có điều kiện, nên các xác suất lỗi bit trong trường hợp này được tính như sau:

$$\begin{aligned} P_b &= P(0)P_e(0) + P(1)P_e(1) = 1/2[P_e(0) + P_e(1)] = P_e(0) = P_e(1) \\ &= \int_0^{\infty} \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{1}{N_0} (y_1 + \sqrt{E_b} \cos \gamma)^2\right) dy_1 \end{aligned}$$

Đặt

$$z = \sqrt{2} \frac{y_1 + \sqrt{E_b} \cos \gamma}{\sqrt{N_0}}$$

ta được:

$$P_b = \frac{1}{\sqrt{2\pi}} \int_{\frac{\sqrt{2E_b} \cos \gamma}{\sqrt{N_0}}}^{\infty} \exp\left(-z^2/2\right) dy_1 = Q\left(\sqrt{\frac{2E_b}{N_0}} \cos \gamma\right)$$

$$\text{Thay } E_b/N_0 = 10^{9,6/10} = 9,12; \gamma = 25^\circ \rightarrow \cos 25^\circ = 0,8; \sqrt{\frac{2E_b}{N_0}} \cos \gamma = \sqrt{2 \times 9,12 \times 0,8} = 3,417.$$

Tra bảng trong phụ lục ta được:  $P_b=3.10^{-4}$

b) Tra bảng trong phụ lục cho xác suất lỗi bit  $10^{-3}$  ta được:

$$\sqrt{\frac{2E_b}{N_0}} = \sqrt{2 \times 9,12} \cdot \cos \gamma = 3,1 \Rightarrow \cos \gamma = 0,726 \Rightarrow \gamma = 43,46^\circ$$

### Bài 16:

Cho hai máy thu hệ thống truyền dẫn nhất quán 16-QAM và QPSK với các tham số sau: công suất thu trung bình  $P_{avr}=10^{-5}W$ ;  $R_b=5000bps$ ;  $N_0=10^{-10}WHZ^{-1}$ .

a) Tìm xác suất lỗi trong hai hệ thống

b) Tìm và băng thông Nyquist của hai hệ thống khi cho hệ số dốc  $\alpha=0,2$

#### Hướng dẫn giải

a) Tốc độ ký hiệu cho 16-QAM và QPSK được xác định như sau:

$$R_{s-16QAM} = R_b \log_2(16) = 20000 \text{ bps}$$

$$R_{s-QPSK} = R_b \log_2(4) = 10000 \text{ bps}$$

Năng lượng trung bình cho 16-QAM và QPSK được xác định như sau:

$$E_{avr-16QAM} = P/R_s = 1,6.10^{-6}/20000 = 0,8.10^{-10}$$

$$E_{avr-QPSK} = P/R_s = 1,6.10^{-6}/10000 = 1,6.10^{-10}$$

Tỷ số tín hiệu trên tạp âm cho hai hệ thống như sau:

$$2 E_{avr-16QAM}/N_0 = 2 \times 0,8.10^{-10}/10^{-10} = 1,6$$

$$2 E_{avr-QPSK}/N_0 = 2 \times 1,6.10^{-10}/10^{-10} = 3,2$$

Tra bảng trong phụ lục ta được:

$$P_{b-16QAM} = 5,48.10^{-2}$$

$$P_{b-QPSK} = 7.10^{-4}$$

b) Băng thông Nyquist của hai hệ thống được xác định như sau:

$$B_{N-16QAM} = (1+\alpha)R_{s-16QAM} = 1,2 \times 20000 \text{ bps} = 24000 \text{ bps}$$

$$B_{N-QPSK} = (1+\alpha)R_{s-QPSK} = 1,2 \times 10000 \text{ bps} = 12000 \text{ bps}$$

### Bài 17:

Để hệ thống 16QAM đạt được xác suất lỗi giống như hệ thống QPSK ta cần tăng công suất cho hệ thống 16QAM lên bao nhiêu lần.

#### Hướng dẫn giải

Để đạt được xác suất lỗi giống như QPSK ta cần đạt được tỷ số tín hiệu trên tạp âm như sau:

$$2E_{avr-16QAM}/N_0 = 2P/(R_{s-16QAM} N_0) = 3,2$$

$$\rightarrow P = 1,6 \times R_{s-16QAM} N_0 = 1,6 \times 20000 \times 10^{-10} = 3,2 \times 10^{-6}$$

Như vậy ta cần phát công suất cho 16QAM gấp hai lần công suất cho QPSK để hai hệ thống có cùng xác suất lỗi.

**Bài 18:**

Một hệ thống BPSK nhất quán hoạt động liên tục có tốc độ lỗi trung bình là 100 lỗi/ngày. Tốc độ dữ liệu là 1000 bit/s, mật độ phổ công suất một phía của tạp âm là  $N_0 = 10^{-10}$  W/Hz

- Tìm xác suất lỗi bit trung bình nếu hệ thống là ergodic.
- Nếu công suất trung bình của tín hiệu thu là  $10^{-6}$  W thì xác suất lỗi bit có giống như tìm được trong câu a?

**Hướng dẫn giải**

a). Tổng số bit tách sóng trong 1 ngày  $1000 \text{ bits/s} \times 86400 \text{ s/ngày} = 8,64 \cdot 10^7$

$$\rightarrow P_B = \frac{100}{8,64 \cdot 10^7} = 1,16 \cdot 10^{-6}$$

b). Ta có 
$$P_B = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) = Q\left(\sqrt{\frac{2PT_b}{10^{-10}}}\right)$$

trong đó  $P = 10^{-6} \text{ W}$  và  $T_b = \frac{1}{1000} \text{ s}$

$$\rightarrow P_B = Q\left(\sqrt{\frac{2 \times 10^{-6}}{1000 \times 10^{-10}}}\right) = Q(\sqrt{20}) = Q(4,47) \rightarrow P_B = 3,911 \times 10^{-6}$$

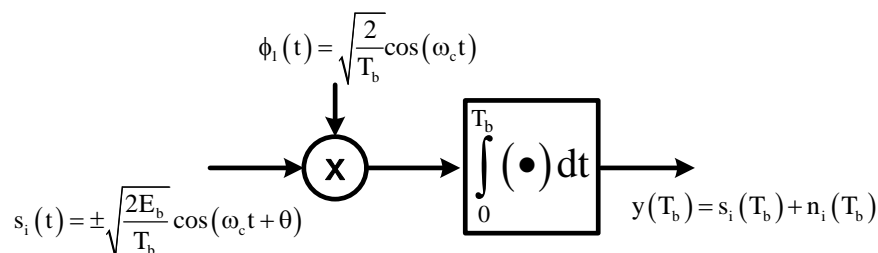
**Bài 19:**

Xét hệ thống BPSK với 2 sóng  $s_1(t) = \cos \omega_c t$  và  $s_2(t) = -\cos \omega_c t$ . Giả thiết rằng  $E_b/N_0 = 9,6 \text{ dB}$ , xác suất lỗi bit là  $10^{-5}$  và xem như hệ thống hoàn toàn được đồng bộ. Do khôi phục sóng mang bằng vòng khóa pha (PLL) nên tín hiệu nhận được là được biểu diễn là:  $\cos(\omega_c + \theta)$  và  $-\cos(\omega_c + \theta)$ .

- Tính xác suất lỗi khi  $\theta = 25^\circ$
- Tính lỗi pha khi xác suất lỗi bit là  $10^{-3}$

**Hướng dẫn giải**

Khi cho tín hiệu thu  $s_i(t) = \pm \sqrt{\frac{2E_b}{T_b}} \cos \omega_c(t + \theta)$  qua bộ tương quan, đầu ra của bộ tương quan ở thời điểm  $T_b$  là:  $y(T_b) = s_i(T_b) + n_i(T_b)$  với  $n_i(t)$  là tạp âm.



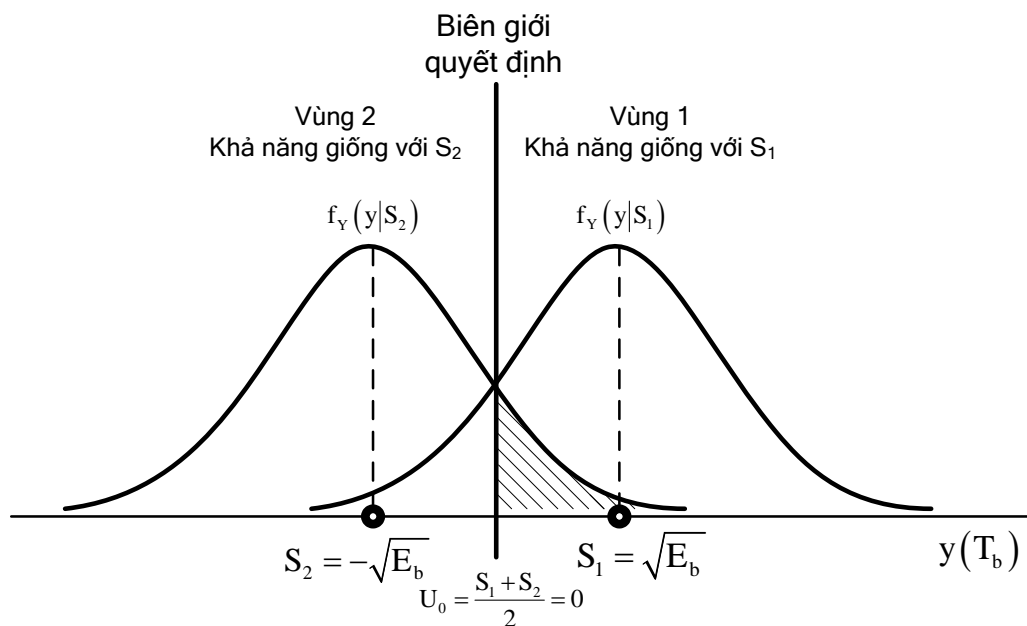
Ta tính được:

$$\begin{aligned}
 s_1(T_b) &= \frac{2}{T_b} \sqrt{E_b} \int_0^{T_b} (\cos(\omega_c t) \cos(\omega_c t + \theta)) dt \\
 &= \frac{2}{T_b} \sqrt{E_b} \int_0^{T_b} \frac{1}{2} (\cos\theta + \cos(2\omega_c t + \theta)) dt \\
 &= \sqrt{E_b} \cos\theta
 \end{aligned}$$

Tương tự,

$$s_2(T_b) = -\sqrt{E_b} \cos\theta$$

Khi  $\theta = 0 \rightarrow \cos\theta = 1 \rightarrow s_1 = \sqrt{E_b}$  và  $s_2 = -\sqrt{E_b}$ . Hàm mật độ xác suất có điều kiện:



Ta có

$$\sigma_0^2 = \frac{N_0}{2} \rightarrow p_B = Q\left(\frac{s_1 + s_2}{2\sigma_0}\right) = Q\left(\frac{\sqrt{E_b} + \sqrt{E_b}}{2\sqrt{N_0/2}}\right) = Q\left(\frac{2\sqrt{E_b}}{2\sqrt{N_0/2}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

Khi  $\theta \neq 0$ , đầu ra của bộ tương quan là:  $s_1 = \sqrt{E_b} \cos\theta$  và  $s_2 = -\sqrt{E_b} \cos\theta$ .

$$\rightarrow P_b = Q\left(\sqrt{\frac{2E_b}{N_0}} \cos\theta\right)$$

a) Với  $\frac{E_b}{N_0} = 9,6\text{dB} = 9,12$  và  $\cos\theta = \cos 25^\circ = 0,9063$

$$\rightarrow P_b = Q(\sqrt{18,24} \times 0,9063) = Q(3,87)$$

Khi  $x > 3$ , ta có thể sử dụng công thức tính xấp xỉ:

$$Q(x) = \frac{1}{x\sqrt{2\pi}} \exp\left(\frac{-x^2}{2}\right) \rightarrow Q(3,87) = \frac{1}{3,87\sqrt{2\pi}} \exp\left(\frac{-(3,87)^2}{2}\right) = 5,8 \times 10^{-5}$$

b). Sử dụng công thức tính xấp xỉ trên ta có:

$$P_b = Q(x) = \frac{1}{x\sqrt{2\pi}} \exp\left(\frac{-x^2}{2}\right) = 10^{-3} \Leftrightarrow x = 3,11525$$

$$\Leftrightarrow \sqrt{\frac{2E_b}{N_0}} \cos \theta = 3,11525$$

$$\Leftrightarrow \theta \cong 43^\circ$$

### Bài 20:

Tìm ngưỡng quyết định tối ưu (xác suất lỗi nhỏ nhất)  $u_0$  cho việc tách sóng BPSK hai tín hiệu đồng xác suất  $s_1(t) = \sqrt{2E_b/T_b} \cos \omega_c t$  và  $s_2(t) = \sqrt{E_b/2T_b} \cos(\omega_c t + \pi)$  trong kênh AWGN, sử dụng máy thu tương quan. Giả sử rằng hàm cơ sở là  $\phi_1(t) = \sqrt{2/T_b} \cos \omega_c t$ .

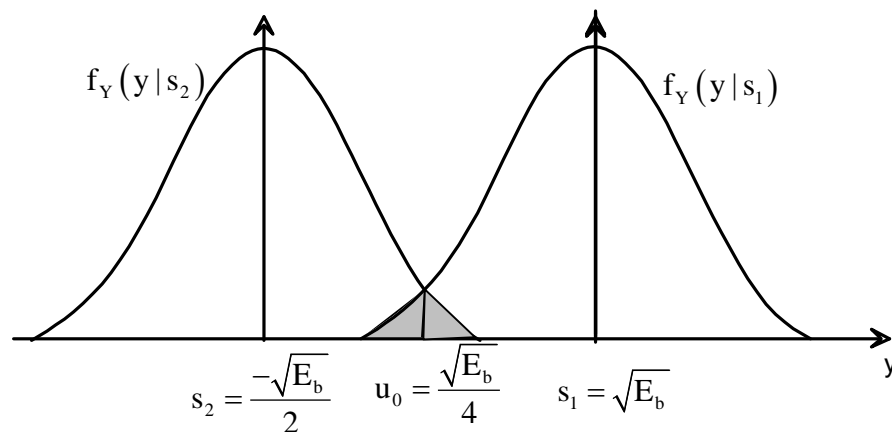
#### Hướng dẫn giải

Biểu diễn các tín hiệu  $s_1(t)$  và  $s_2(t)$  qua hàm cơ sở  $\phi_1(t) = \sqrt{2/T_b} \cos \omega_c t$  ta được:

$$s_1(t) = s_{11}\phi_1(t) = \sqrt{E_b}\phi_1(t)$$

$$s_2(t) = s_{21}\phi_1(t) = -\frac{1}{2}\sqrt{E_b}\phi_1(t)$$

Biểu diễn các hàm mật độ xác suất có điều kiện



Vậy ngưỡng quyết định tối ưu là:

$$u_0 = \frac{s_1 + s_2}{2} = \frac{\sqrt{E_b} + \left(-\frac{1}{2}\sqrt{E_b}\right)}{2} = \frac{\sqrt{E_b}}{4}$$



## Bài 21

Một hệ thống có sử dụng bộ lọc thích hợp để tách sóng các tín hiệu BPSK có xác suất bằng nhau  $s_1(t) = \sqrt{2E_b/T_b} \cos \omega_c t$  và  $s_2(t) = \sqrt{2E_b/T_b} \cos(\omega_c t + \pi)$  trong điều kiện tạp âm Gauss,  $E_b/N_0 = 6,8 \text{ dB}$ . Giả sử  $E\{y(T_b)\} = \pm\sqrt{E_b}$

a) Tìm xác suất lỗi bit nhỏ nhất.

b) Tìm xác suất lỗi bit khi ngưỡng quyết định  $u = 0,1\sqrt{E_b}$

c) Biết ngưỡng  $u_0 = 0,1\sqrt{E_b}$  là tối ưu cho một tập các xác suất tiên nghiệm  $P(s_1)$  và  $P(s_2)$ . Tính các giá trị xác suất tiên nghiệm này

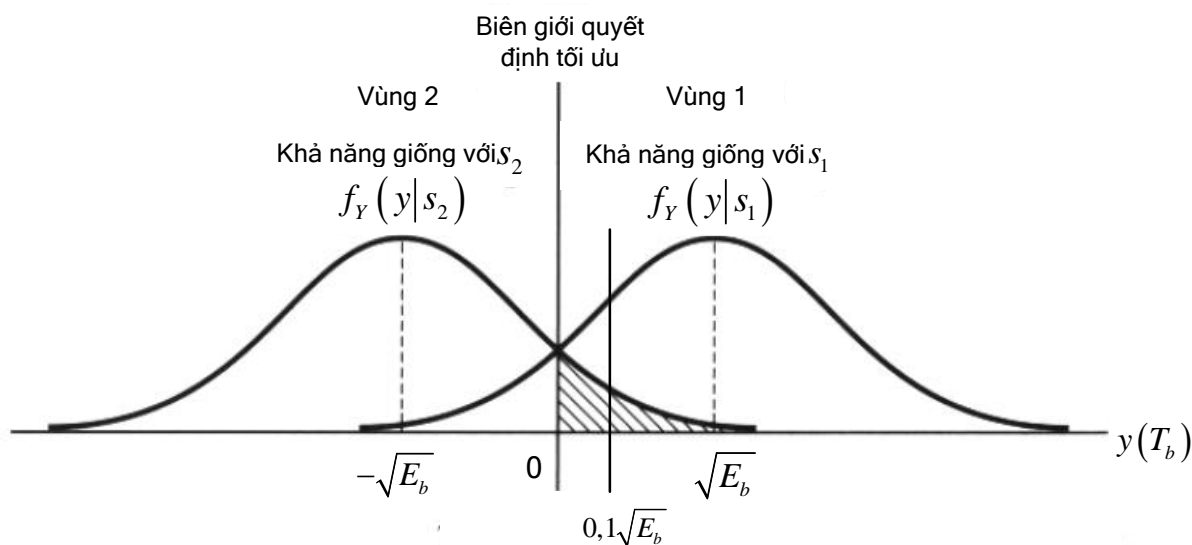
### Hướng dẫn giải

a) Ta có:

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

$$\text{Có } E_b/N_0 = 6,8 \text{ dB} = 4,786 \rightarrow P_b = Q(\sqrt{2 \times 4,786}) = Q(3,09) = 10^{-3}$$

b) Với ngưỡng quyết định là  $u = 0,1\sqrt{E_b}$  ta có hình vẽ sau:



Do các xác suất phát các tín hiệu ban đầu là như nhau nên xác suất lỗi bit được tính như sau:

$$\begin{aligned}
 P_b &= \frac{1}{2} \int_{-\infty}^{0,1\sqrt{E_b}} f_Y(y|s_1) dy + \frac{1}{2} \int_{0,1\sqrt{E_b}}^{\infty} f_Y(y|s_2) dy \\
 &= \frac{1}{2} \int_{-\infty}^{0,1\sqrt{E_b}} \frac{1}{\sigma_0\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-\sqrt{E_b}}{\sigma_0}\right)^2\right] dy \\
 &\quad + \frac{1}{2} \int_{0,1\sqrt{E_b}}^{\infty} \frac{1}{\sigma_0\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y+\sqrt{E_b}}{\sigma_0}\right)^2\right] dy
 \end{aligned}$$

Đặt  $z_1 = \frac{y-\sqrt{E_b}}{\sigma_0} \rightarrow dy = \sigma_0 dz_1$

$z_2 = \frac{y+\sqrt{E_b}}{\sigma_0} \rightarrow dy = \sigma_0 dz_2$

$$\rightarrow P_b = \frac{1}{2} \int_{-\infty}^{-0,9\sqrt{E_b}/\sigma_0} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z_1^2}{2}\right) dz_1 + \frac{1}{2} \int_{1,1\sqrt{E_b}/\sigma_0}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z_2^2}{2}\right) dz_2$$

Với hàm Gauss đối xứng ta có  $\int_{-\infty}^{-x} \exp\left(-\frac{z^2}{2}\right) dz = \int_x^{\infty} \exp\left(-\frac{z^2}{2}\right) dz$

$$\rightarrow P_b = \frac{1}{2} Q\left(\frac{0,9\sqrt{E_b}}{\sigma_0}\right) + \frac{1}{2} Q\left(\frac{1,1\sqrt{E_b}}{\sigma_0}\right)$$

Ta có

$$\sigma_0^2 = \frac{N_0}{2} \rightarrow P_b = \frac{1}{2} Q\left(0,9\sqrt{\frac{2E_b}{N_0}}\right) + \frac{1}{2} Q\left(1,1\sqrt{\frac{2E_b}{N_0}}\right)$$

Ta có  $\frac{E_b}{N_0} = 6,8 \text{ dB}$

$$\begin{aligned}
 \rightarrow P_b &= \frac{1}{2} Q(0,9 \times 3,09) + \frac{1}{2} Q(1,1 \times 3,09) = \frac{1}{2} Q(2,78) + \frac{1}{2} Q(3,4) \\
 &= 1,4 \times 10^{-3}
 \end{aligned}$$

c) Gọi  $H_1$  và  $H_2$  lần lượt là các sự kiện quyết định  $s_1$  và  $s_2$  đã được phát, ta có

$$\begin{aligned}
 \frac{f_Y(y|s_1)}{f_Y(y|s_2)} & \underset{H_2}{\overset{H_1}{>}} \frac{P(s_2)}{P(s_1)} \\
 \Rightarrow \frac{y(s_{11}-s_{21})}{\sigma_0^2} & \underset{H_2}{\overset{H_1}{>}} \ln \left[ \frac{P(s_2)}{P(s_1)} \right] \\
 \Rightarrow \frac{y(2\sqrt{E_b})}{N_0/2} & \underset{H_2}{\overset{H_1}{>}} \ln \left[ \frac{P(s_2)}{P(s_1)} \right] \\
 \Rightarrow y & \underset{H_2}{\overset{H_1}{>}} \frac{N_0/2}{2\sqrt{E_b}} \ln \left[ \frac{P(s_2)}{P(s_1)} \right] = u_0
 \end{aligned}$$

Do  $u_0 = 0,1\sqrt{E}$

$$\begin{aligned}
 \text{nên } \frac{N_0}{4\sqrt{E_b}} \ln \left[ \frac{P(s_1)}{P(s_2)} \right] &= 0,1\sqrt{E_b} \\
 \Rightarrow \ln \left[ \frac{P(s_2)}{P(s_1)} \right] &= \frac{0,4E_b}{N_0} = 0,4 \times 4,786 = 1,914 \\
 \Rightarrow \frac{P(s_2)}{1-P(s_2)} &= \exp(1,914) = 6,782 \\
 \Rightarrow P(s_2) = 0,87 &\Rightarrow P(s_1) = 0,13
 \end{aligned}$$

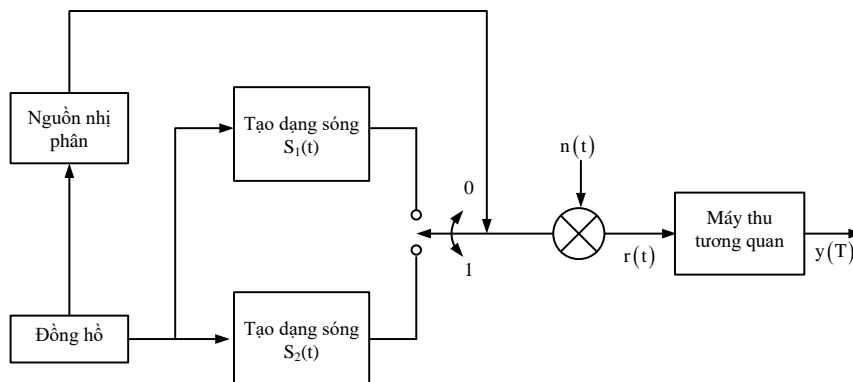
## Bài 22

Trong hình vẽ dưới đây, một nguồn nhị phân phát tín hiệu 0 và 1 với xác suất bằng nhau điều khiển một chuyển mạch vị trí hoạt động trong môi trường kênh AWGN, tốc độ bit là  $1/T$ . Tập âm có mật độ phổ hai phía là  $N_0/2$ . Tập tín hiệu đối cực  $s_1(t)$  và  $s_2(t)$  có năng lượng  $E$ . Xác suất lỗi của chuyển mạch là  $p$  và không phụ thuộc thời gian. Giả sử  $E\{Y(T)\} = \pm\sqrt{E}$

a) Vẽ đồ thị hàm xác suất có điều kiện  $p(y|s_1)$  và  $p(y|s_2)$

b) Hệ thống nào trong hai hệ thống sau đây tốt hơn?

$$p = 0,1 \quad \text{và} \quad \frac{E_b}{N_0} = \infty \quad \text{hay} \quad p = 0 \quad \text{và} \quad \frac{E_b}{N_0} = 7\text{dB}$$



### Hướng dẫn giải

a) Ta có:

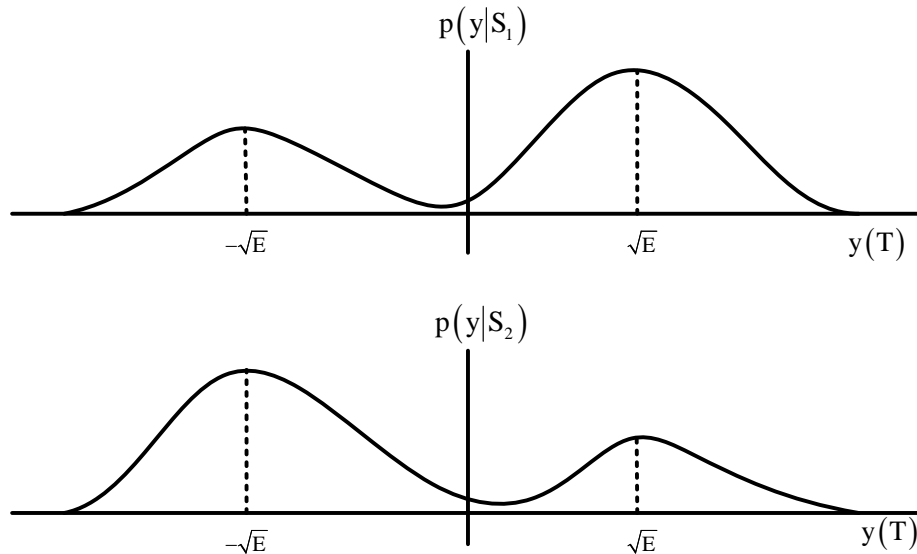
$$p(y|s_1) = p.p'(y|s_2) + (1-p).p'(y|s_1)$$

$$p(y|s_2) = p.p'(y|s_1) + (1-p).p'(y|s_2)$$

trong đó  $p$  là xác suất sai của chuyển mạch, hàm  $p'$  là phân bố xác suất của  $y$  nếu chỉ xét đến ảnh hưởng của tập âm. Các hàm mật độ xác suất có điều kiện có dạng như sau:

b) Với  $p = 0,1$  và  $\frac{E_b}{N_0} = \infty$  thì  $P_b = 0,1$

$$\text{Với } p = 0 \quad \text{và} \quad \frac{E_b}{N_0} = 7\text{dB} \quad \text{ta có: } P_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) = Q(3,167) = 8.10^{-4} \rightarrow \text{tốt hơn.}$$



**Bài 23:**

- Xét hệ thống 16-PSK với xác suất lỗi ký hiệu  $P_e = 10^{-5}$ , sử dụng mã Gray. Hỏi xác suất lỗi bit xấp xỉ bằng bao nhiêu?
- Lặp lại câu (a) khi hệ thống là 16-PSK trực giao

**Hướng dẫn giải**

- Ta có  $k = \log_2 M = \log_2 16 = 4$

$$P_b \approx \frac{P_e}{k} = \frac{10^{-5}}{4} = 2.5 \times 10^{-6}$$

- $P_b = \frac{2^{k-1}}{2^k - 1} P_e = \frac{2^3}{2^4 - 1} 10^{-5} = 5,3 \times 10^{-6}$

**Bài 24:**

Hệ thống có tốc độ dữ liệu là 100 kbps truyền qua kênh Gauss sử dụng điều chế M-PSK được tách sóng nhất quán yêu cầu xác suất lỗi  $P_b = 10^{-3}$ . Băng thông của hệ thống là 50 kHz. Hàm truyền đạt tần số của hệ thống là hàm cos tăng với hệ số dốc  $\alpha = 1$  và sử dụng mã Gray.

- Tính  $E_s/N_0$
- Tính  $E_b/N_0$

**Hướng dẫn giải**

- Với  $\alpha = 1$  và không có nhiễu ISI, ta có:

$$B_N = (1 + \alpha)R_s$$

$$\Rightarrow 50 \text{ kHz} = 2R_s \Rightarrow R_s = 25 \text{ ksym/s}$$

$$k = \log_2 M = \frac{R}{R_s} = \frac{100 \text{ kbps}}{25 \text{ ksym/s}} = 4 \Rightarrow M = 16$$

Khi sử dụng mã Gray ta có:

$$P_b \approx \frac{P_e}{\log_2 M} \Rightarrow P_e = P_b \log_2 M = 4.10^{-3}$$

$$\rightarrow P_e = 2Q\left[\left(\sqrt{\frac{2E_s}{N_0}}\right) \sin\left(\frac{\pi}{M}\right)\right] = 4.10^{-3} \rightarrow Q(x) = 2 \times 10^{-3}$$

$$\Leftrightarrow x = \left(\sqrt{\frac{2E_s}{N_0}}\right) \sin\left(\frac{\pi}{M}\right) = 2,88$$

$$\Leftrightarrow \frac{E_s}{N_0} = 108,9 \quad \text{hay} \quad 20,4\text{dB}$$

b).  $\frac{E_B}{N_0} = \frac{108,9}{k} = \frac{108,9}{4} = 27,2 \quad \text{hay} \quad 14,3\text{dB}$

### Bài 25:

Một hệ thống M-PSK nhất quán hoạt động qua kênh AWGN với  $E_b/N_0 = 10\text{dB}$ . Tính xác suất lỗi ký hiệu với  $M = 8$  và biết các ký hiệu có xác suất bằng nhau.

#### Hướng dẫn giải

Ta có

$$E_b/N_0 = 10\text{dB} \quad \text{hay} \quad E_b/N_0 = 10$$

$$k = \log_2 M = \log_2 8 = 3$$

$$\Rightarrow \frac{E_s}{N_0} = k \frac{E_b}{N_0} = 3.10 = 30 \quad \text{hay} \quad 14,77\text{dB}$$

Xác suất lỗi ký hiệu của hệ thống là:

$$\begin{aligned} P_e &= 2Q\left[\left(\sqrt{\frac{2E_s}{N_0}}\right) \sin\left(\frac{\pi}{\sqrt{2}M}\right)\right] = 2Q\left[\sqrt{60} \sin\left(\frac{\pi}{8\sqrt{2}}\right)\right] \\ &= 2Q\left[\sqrt{60} \times 0,2741\right] = 2Q(2,123) \\ P_e &= 2 \times 0,169 = 3,38.10^{-2} \end{aligned}$$

### Bài 26:

Một bộ giải điều chế/tách sóng BPSK có lỗi đồng bộ với độ lệch thời gian  $pT_b$ ,  $0 \leq p \leq 1$ . Nếu các tín hiệu đồng xác suất và đồng bộ về pha cũng như tần số.

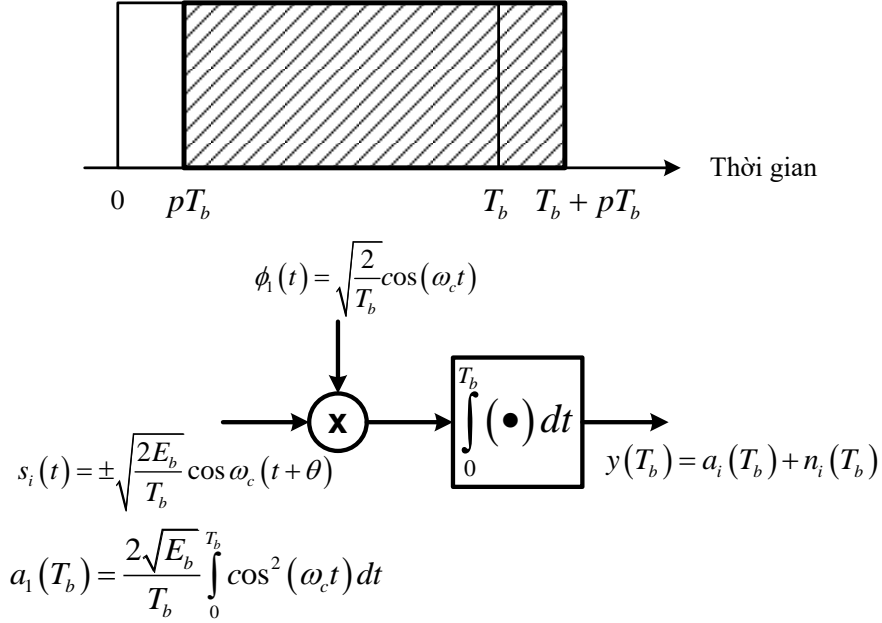
a) Biểu diễn của xác suất lỗi bit  $P_b$  theo  $p$

b) Nếu  $E_b/N_0 = 9,6\text{ dB}$  và  $p = 0,2$ , tính  $P_b$  gây ra do sự sai lệch thời gian.

c)  $E_b/N_0$  phải được bổ sung thêm bao nhiêu (theo dB) để có được xác suất lỗi với trường hợp  $p = 0$  bằng với xác suất lỗi khi  $E_b/N_0 = 9,6\text{ dB}$  và  $p = 0$ .

#### Hướng dẫn giải

a).



Theo thứ tự dạng sóng nhận được, giả sử rằng sau  $s_1(t)$  ta thu được

$s_2(t) = -\sqrt{\frac{2E_b}{N_0}} \cos \omega_c t$ . Do bộ tách sóng bị trễ đi một lượng  $pT_b$  nên ta có:

$$a_1(T_b) = \frac{2\sqrt{E_b}}{T_b} \left[ \int_0^{T_b} \cos^2(\omega_c t) dt + \int_{pT_b}^{T_b + pT_b} (-\cos^2(\omega_c t)) dt \right]$$

$$= \frac{\sqrt{E_b}}{T_b} [T_b - pT_b - (T_b + pT_b - T_b)] = \sqrt{E_b} (1 - 2p)$$

Nếu  $s_2(t)$  được phát, sau đó là  $s_1(t)$ , tính tương tự ta sẽ có  $a_2(T_b) = -\sqrt{E_b} (1 - 2p)$

Giả sử rằng trong  $\frac{1}{2}$  khoảng thời gian  $T_b$ ,  $s_1(t)$  phát và sau đó là  $s_2(t)$ ; và trong  $\frac{1}{2}$  khoảng thời gian còn lại  $s_1(t)$  phát và sau đó cũng là  $s_1(t)$  thì xác suất lỗi bit sẽ là:

$$P_b = \frac{1}{2} Q\left(\sqrt{\frac{2E_b}{N_0}}\right) + \frac{1}{2} Q\left(\sqrt{\frac{2E_b}{N_0}} (1 - 2p)\right)$$

b). Khi  $p = 0$  và  $E_b/N_0 = 9,6$  dB hay 9,12 thì:

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) = Q(4,27) = 10^{-5}$$

Khi  $p = 0,2$  thì

$$P_b = \frac{1}{2} Q\left(\sqrt{\frac{2E_b}{N_0}}\right) + \frac{1}{2} Q\left(\sqrt{\frac{2E_b}{N_0}} (1 - 2p)\right)$$

$$= \frac{1}{2} \times 10^{-5} + \frac{1}{2} Q(4,27 \times 0,6)$$

$$= 2,6 \cdot 10^{-3}$$

c).

$$P_b = \frac{1}{2} Q\left(\sqrt{\frac{2E_b}{N_0}}\right) + \frac{1}{2} Q\left(\sqrt{\frac{2E_b}{N_0}} (1 - 2p)\right) = 10^{-5}$$

Với  $p = 0$  ta sẽ tính được  $E_b/N_0 = 23,56$  hay 13,7dB

Vậy cần tăng  $E_b/N_0$  thêm 4,1dB để có được xác suất lỗi bit với trường hợp  $p = 0$  là  $P_B = 10^{-5}$

**Bài 27:**

Bộ giải điều chế/tách sóng BPSK có lỗi đồng bộ với độ lệch thời gian  $pT_b$ , ( $0 \leq p \leq 1$ ). Lỗi ước tính pha là  $\theta$ . Tín hiệu được phát với xác suất bằng nhau và đồng bộ về mặt tần số.

a) Biểu diễn của xác suất lỗi bit  $P_b$  theo  $p$  và  $\theta$

b) Tính  $P_b$  gây ra bởi sai lệch thời gian và pha biết  $E_b/N_0 = 9,6$  dB và  $p = 0,2$ ,  $\theta = 25^\circ$ .

c)  $E_b/N_0$  phải được bổ sung thêm bao nhiêu (theo dB) để có được xác suất lỗi với trường hợp  $p = 0$  và  $\phi = 0$  bằng với trường hợp  $E_b/N_0 = 9,6$  dB và  $p = 0$ ,  $\theta = 0$

**Hướng dẫn giải**

a). Xét trường hợp tín hiệu vào là  $s_1(t)$ . Khi có lỗi ước lượng pha  $\theta$  và độ lệch thời gian  $pT_b$ , đầu ra là:

$$a_1(T_b) = \frac{2}{T_b} \sqrt{E_b} \int_{pT_b}^{T_b+pT_b} \cos(\omega_c t) \cos(\omega_c t + \theta) dt$$

Nếu sau tín hiệu đang xét cũng là  $s_1(t)$ , thì

$$\begin{aligned} a_1(T_b) &= \frac{2}{T_b} \sqrt{E_b} \int_{pT_b}^{T_b+pT_b} \cos(\omega_c t) \cos(\omega_c t + \theta) dt \\ &= \frac{2}{T_b} \sqrt{E_b} \int_0^{T_b} \cos(\omega_c t) \cos(\omega_c t + \theta) dt \\ &= \sqrt{E_b} \cos \theta \end{aligned}$$

Nếu sau tín hiệu đang xét là  $s_2(t)$ , thì

$$\begin{aligned} a_1(T_b) &= \frac{2}{T_b} \sqrt{E_b} \int_{pT_b}^{T_b+pT_b} \cos \omega_c t \cdot \cos(\omega_c t + \theta) dt \\ &= \frac{2}{T_b} \sqrt{E_b} \int_{pT_b}^{T_b} \cos(\omega_c t) \cos(\omega_c t + \theta) dt + \frac{2}{T_b} \sqrt{E_b} \int_{T_b}^{T_b+pT_b} -[\cos(\omega_c t) \cdot \cos(\omega_c t + \theta)] dt \\ &= \sqrt{E_b} (1 - 2p) \cos \theta \end{aligned}$$

Cho rằng xác suất của mỗi trường hợp trên đều là  $1/2$ . Tính tương tự cho  $s_2$  ta có:

$$P_b = \frac{1}{2} Q \left( \sqrt{\frac{2E_b}{N_0}} \cos \theta \right) + \frac{1}{2} Q \left( \sqrt{\frac{2E_b}{N_0}} (1 - 2p) \cos \theta \right)$$

b). Thay  $E_b/N_0 = 9,6$  dB,  $p = 0,2$  và  $\theta = 25^\circ$  vào công thức trên ta được:

$$\begin{aligned} P_b &= \frac{1}{2} Q(4,27 \times 0,906) + \frac{1}{2} Q(4,27 \times 0,906 \times 0,6) \\ &= \frac{1}{2} Q(3,869) + \frac{1}{2} Q(2,321) = 5,1 \times 10^{-3} \end{aligned}$$



c). Khi  $p = 0$  và  $\theta = 0$  thì  $P_b = 10^{-5}$ . Theo công thức hàm xác suất lỗi ở phần (a) ta có:

$$P_b = 10^{-5} = \frac{1}{2} Q\left(\sqrt{\frac{2E_b}{N_0}} 0,906\right) + \frac{1}{2} Q\left(\sqrt{\frac{2E_b}{N_0}} 0,544\right)$$

$$\rightarrow 2 \times 10^{-5} \cong Q\left(\sqrt{\frac{2E_b}{N_0}} 0,544\right) \cong \frac{1}{x\sqrt{2\pi}} e^{-x^2/2} \Leftrightarrow x = \left(\sqrt{\frac{2E_b}{N_0}} 0,544\right) = 4,119$$

$$\rightarrow \frac{E_b}{N_0} = 28,66 = 14,6dB \text{ nghĩa là tăng } 5dB \text{ so với trường hợp (b)}$$

### Bài 28:

Cho luồng số đầu vào bộ điều chế QPSK  $b(t)$  có tốc độ  $R_b = 1\text{Mb/s}$ . Các bit chỉ số lẻ và chẵn tương ứng là  $b_1(t)$  và  $b_2(t)$ . Hãy xác định.

- Tốc độ ký hiệu  $R_s$  là bao nhiêu?
- Độ rộng băng thông của kênh truyền  $B_C$  là bao nhiêu? nếu chỉ phát đi búp chính của phổ tín hiệu điều chế QPSK.
- Xác suất lỗi ký hiệu bằng bao nhiêu? nếu công suất của sóng mang là  $4\text{mW}$  và  $N_0 = 10^{-9} \text{ W/Hz}$

### Hướng dẫn giải

- Với điều chế QPSK, thì luồng bit  $b(t)$  đầu vào được phân thành hai luồng  $b_1(t)$  tương ứng với các bit chỉ số lẻ của  $b(t)$  và luồng  $b_2(t)$  tương ứng với các bit chỉ số chẵn của  $b(t)$ . Vì vậy độ lâu của ký hiệu  $T_s = 2T_b$  (tốc độ ký hiệu bằng một nửa tốc độ bit đầu vào  $b(t)$ ). Kết quả tốc độ ký hiệu  $R_s = \frac{1}{T_s} = \frac{1}{2 \times T_b} = \frac{1}{2} \times R_b = \frac{1}{2} \times 10^6 = 500 \text{ K symbol/s}$ .
- Phổ tín hiệu QPSK có độ rộng băng búp chính là  $\frac{2}{T_s} = 2 \times R_s = 2 \times 500 = 1\text{MHz}$ . Trong đó  $T_s$  là độ lâu của ký hiệu và  $R_s$  là tốc độ ký hiệu, khi này tín hiệu QPSK chiếm độ rộng băng thông của kênh truyền là  $1\text{MHz}$  (phải cấp phát độ rộng băng tần là  $B_C = 1\text{MHz}$ )
- Với điều chế QPSK, xác suất lỗi ký hiệu được cho bởi.

$$P_e = 2Q\left(\sqrt{\frac{E_s}{N_0}}\right) - Q^2\left(\sqrt{\frac{E_s}{N_0}}\right)$$

$$= 2Q\left(\sqrt{\frac{2E_b}{N_0}}\right) - Q^2\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

Do  $Q^2\left(\sqrt{\frac{2E_b}{N_0}}\right) \ll Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$

nên  $P_e \approx 2 \times Q\left(\sqrt{\frac{2E_b}{N_0}}\right) = 2 \times Q\left(\sqrt{\frac{P \times T_s}{N_0}}\right)$  thay giá trị các tham số

$$P_e \approx 2 \times Q \left( \sqrt{\frac{P \times T_s}{N_0}} \right) = 2 \times Q \left( \sqrt{\frac{\underbrace{(4 \times 10^{-3} \text{ W})}_P \times \underbrace{(500 \times 10^3)^{-1} \text{ s}}_{T_s}}{\underbrace{10^{-9} \text{ W/Hz}}_{N_0}}} \right)$$

$$= 2 \times Q(\sqrt{8}) = 2 \times 2.37 \times 10^{-3} = 4.74 \times 10^{-3}$$

### Bài 29:

Cho độ rộng băng thông của kênh truyền **B<sub>C</sub> = 4000 Hz**. Tìm tốc độ dữ liệu đầu vào của các sơ đồ điều chế sau: a) QPSK; b) 64-QAM; f) 32-PSK sao cho chiếm dụng hết độ rộng băng thông của kênh này.

#### Hướng dẫn giải

Quan hệ giữa tốc độ và độ rộng băng thông đối với các sơ đồ điều chế được cho bởi:  $R = \frac{1}{2} \times B_C \times \log_2 M$ . Vì vậy, để chiếm dụng hết độ rộng băng thông của kênh  $B_C = 4000\text{Hz}$  thì tốc độ bit đầu vào các bộ điều chế là:

$$R_{\text{QPSK}} = \frac{1}{2} \times B_C \times \log_2 M = \frac{1}{2} \times 4000 \times \log_2 4 = 4 \text{ Kb/s}$$

$$R_{\text{64-QAM}} = \frac{1}{2} \times B_C \times \log_2 M = \frac{1}{2} \times 4000 \times \log_2 64 = 12 \text{ Kb/s}$$

$$R_{\text{32-PSK}} = \frac{1}{2} \times B_C \times \log_2 M = \frac{1}{2} \times 4000 \times \log_2 32 = 10 \text{ Kb/s}$$

### Bài 30:

Tìm độ rộng băng tần truyền dẫn cần thiết để đạt được tốc độ bit là 100 Kbit/s đối với các sơ đồ điều chế sau: a) 64 QAM; b) 16-PSK; c) BPSK; h) 256-QAM;

#### Hướng dẫn giải

Hiệu quả sử dụng độ rộng băng tần đối với các sơ đồ điều chế này là:  $\frac{1}{2} \times \log_2 M$ . Vì vậy, độ rộng băng tần truyền dẫn cần thiết để đạt được tốc độ bit 100 Kbit/s đối với các sơ đồ điều chế này là:

Loại điều chế	Độ rộng băng tần, KHz; $B_C = R_b / \rho$	Hiệu quả sử dụng băng tần $\rho$
M = 64 QAM	33,3	$\frac{1}{2} \times \log_2 M$
M = 16-PSK	50	$\frac{1}{2} \times \log_2 M$
BPSK	200	$\frac{1}{2} \times \log_2 M$
M = 256-QAM	25	$\frac{1}{2} \times \log_2 M$

### Bài 31:

So sánh M-PSK với  $M=2,4,8,16,32$  và  $64$  trên cơ sở  $E_b/N_0$  được yêu cầu đối với xác suất lỗi bit là  $10^{-5}$ ;  $10^{-6}$  và  $10^{-7}$ .

#### Hướng dẫn giải

Xác suất lỗi bit đối với BPSK và các thành phần vuông pha nhau của QPSK là.

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

Xác suất lỗi ký hiệu đối với M-PSK được lấy giới hạn trên bởi

$$P_s < 2Q\left[\sqrt{\frac{2E_s}{N_0}} \sin\left(\frac{\pi}{M}\right)\right]$$

Trong đó giới hạn là chặt chẽ khi  $M > 4$  và các tỉ số tín hiệu trên tạp âm tương đối lớn. Từ các công thức phân hiệu năng xác suất lỗi và  $E_s = (\log_2 M) \times E_b$ . Ta có thể liên hệ bit với xác suất lỗi ký hiệu và bit và năng lượng ký hiệu. Giới hạn xác suất lỗi bit đối với  $M = 8$  đến  $M = 64$  trở thành.

$$P_b < \frac{2}{\log_2 M} Q\left[\sqrt{2 \times (\log_2 M) \times \left(\frac{E_b}{N_0}\right)} \times \sin\left(\frac{\pi}{M}\right)\right]$$

Các phương trình này có thể được thực hiện bằng Matlab. Chẳng hạn dùng hàm fzero, hay được tính toán thông qua phép xấp xỉ.

$$Q(u) \approx \frac{\exp\left(-\frac{u^2}{2}\right)}{\sqrt{\pi} \times u}, \quad u \gg 1$$

**So sánh các xác suất lỗi bit đối với các sơ đồ điều chế khác nhau.**

Sơ đồ điều chế	$E_b/N_0$ (dB) được yêu cầu để xác suất lỗi bit $P_b =$		
	$P_b = 10^{-5}$	$P_b = 10^{-6}$	$P_b = 10^{-7}$
BPSK; QPSK	9,6	10,5	11,3
8-PSK	13	14	14,8
16-PSK	17,4	18,4	19,3
32-PSK	22,3	23,4	24,2
64-PSK	27,5	28,5	29,4

### Bài 32:

Cho độ rộng băng thông kênh truyền  $B_C = 4$  KHz; tốc độ dữ liệu  $R_b = 8$  Kbit/s. Nếu  $N_0 = 10^{-9}$  W/Hz. Hãy chọn sơ đồ điều chế sao cho cung cấp tốc độ dữ liệu này tại BER là  $P_b = 10^{-6}$ .

#### Hướng dẫn giải

Từ đầu bài ta có hiệu quả sử dụng băng tần là  $8/4 = 2 \text{ bit/s/Hz}$ . Đối với M-PSK, ta có  $\frac{1}{2} \log_2 M = 2 \Rightarrow M=16$ . Sử dụng kết quả ở bài tập trên ta có

Đối với 16-PSK: để đạt được xác suất lỗi bit  $P_b = 10^{-6}$  thì  $E_b/N_0 = 18,4 \text{ dB}$ .

Đối với 16-QAM: để đạt được  $P_b = 10^{-6}$  cần có  $E_b/N_0 = 15 \text{ dB}$ .

Để tính công suất sóng mang trung bình, cần phải tính năng lượng ký hiệu trung bình.

$$E_s = N_0 \times (\log_2 M) \times \left( \frac{E_b}{N_0} \right)_{\text{Yêu cầu}}$$

$$\Leftrightarrow E_s (\text{dB-jules}) = 10 \times (\log_{10} N_0) + 10 \times [\log_{10} (\log_2 M)] + 10 \times \left( \log_{10} \left( \frac{E_b}{N_0} \right)_{\text{Yêu cầu}} \right)$$

Hoặc, đối với 16-QAM, thì

$$\begin{aligned} E_s (\text{dB-jules}) &= 10 \log_{10} 10^{-9} + 10 \log_{10} 4 + 15 \\ &= -90 + 6 + 15 \\ &= -69 \text{ dB-J} \end{aligned}$$

Quan hệ giữa công suất trung bình và năng lượng trung bình của sóng mang

$$P_C = \frac{E_s}{T_s} = E_s \times R_s$$

Tốc độ ký hiệu  $R_s = 1/T_s$  được cho bởi

$$R_s = \frac{R_b}{\log_2 M} = \frac{8000}{4} = 2000 \text{ ký hiệu/giây}$$

Vì vậy công suất sóng mang trung bình là.

$$\begin{aligned} P_C (\text{dBW}) &= 10 \log_{10} (E_s \times R_s) && \text{dBW} \\ &= -69 + 10 \log_{10} (2000) && \text{dBW} \\ &= -36 && \text{dBW} \\ \Rightarrow P_C (\text{W}) &= 2,51 \times 10^{-4} && \text{W} \end{aligned}$$

### Bài 33:

- Tìm tốc độ bit đầu vào các sơ đồ điều chế dưới đây sao cho chiếm dụng hết độ rộng băng thông của kênh  $B_C = 4 \text{ KHz}$ : 1) BPSK; 2) 4-PSK; 3) 8-PSK; 4) 16-PSK; 5) 16-QAM;
- Nếu  $N_0 = 10^{-8} \text{ W/Hz}$ , tìm công suất sóng mang trong mỗi trường hợp để xác suất lỗi bit  $P_b = 10^{-6}$ ;

### Hướng dẫn giải

- a) Hiệu quả sử dụng độ rộng băng tần đối với các sơ đồ điều chế này là  $\frac{1}{2} \times \log_2 M$  (bit/s/Hz) với điều chế M-PSK và M-QAM ta có.  $R_b = \frac{1}{2} \times B_C \times \log_2 M$ .  
 $= 0,5 \times 4000 \times \log_2 M$ . Các tốc độ dữ liệu được phép đối với các kỹ thuật điều chế này là:

Các sơ đồ điều chế		Tốc độ dữ liệu, K bit/s
1	BPSK;	2
2	4-PSK;	4

3	8-PSK;	6
4	16-PSK;	8
5	16-QAM;	8

**b) Tính công suất sóng mang**

Công suất sóng mang được tính bởi

$$P_C = \frac{A^2}{2} = \frac{E_b}{T_b} = E_b \times R_b$$

Đặt  $Z_0$  là giá trị của  $E_b/N_0$  được yêu cầu để đảm bảo  $P_b = 10^{-6}$  trong mỗi trường hợp. Ta có

$$P_C = E_b \times R_b = \frac{E_b}{N_0} \times N_0 \times R_b = Z_0 \times N_0 \times R_b$$

Hay

$$\begin{aligned} P_{c,dBW} &= Z_{0,dB} + N_{0,dB-Hz} + R_{b,dB-Hz} \\ &= Z_{0,dB} - 80 + R_{b,dB-Hz} \end{aligned}$$

Sử dụng phương trình này và các kết quả ở phần a) ta được các kết quả:

*Các công suất được yêu cầu đối với các sơ đồ điều chế khác nhau*

Sơ đồ điều chế	$Z_0$ dB	$R_b$ dB-s <sup>-1</sup>	$P_{c,dBW}$	$P_{c,mW}$
BPSK;	10,5	33	-36,5	0,229
4-PSK;	10,5	36	-33,5	0,457
8-PSK;	13,6	37,8	-28,6	1,380
16-PSK;	18,5	39	-22,5	5,623
16-QAM;	14,5	39	-26,5	2,239

**Bài 34:**

- a) Dữ liệu phát ở tốc độ  $R_b = 5Kb/s$ . Đối với các sơ đồ điều chế được xét ở bài tập 33. Xác định độ rộng băng thông của kênh tương ứng với các sơ đồ điều chế.  
b) Nếu  $N_0 = 10^{-8}$  W/Hz, tìm công suất sóng mang thu để xác suất lỗi  $P_b = 10^{-6}$ .

**Hướng dẫn giải**

- a) Từ bảng trên ta có được các công thức sau đây để xác định độ rộng băng theo M và tốc độ dữ liệu.

Đối với M-PSK và M-QAM:  $B_C = \frac{2}{\log_2 M} \times R_b$

Đối với M-FSK nhất quán:  $B_C = \frac{M+3}{2 \times \log_2 M} \times R_b$

Đối với M-FSK không nhất quán:  $B_C = \frac{2 \times M}{\log_2 M} \times R_b$

Sử dụng các phương trình này, ta được các độ rộng băng tần kênh được cho ở bảng sau.

Các sơ đồ điều chế	Độ rộng băng thông, KHz
BPSK	10
4-PSK	5
8-PSK	3,3
16-PSK	2,5
16-QAM	2,5
BFSK nhất quán	12,5
4-FSK nhất quán	8,75
8-FSK nhất quán	9,17
BFSK không nhất quán	20,0
4-FSK không nhất quán	20,0
8-FSK không nhất quán	26,7

- b) Sử dụng phương trình được khai triển ở phần b của bài 33, ta tìm được các kết quả được cho ở bảng sau.

$$P_{c,dBW} = Z_{0,dB} - 80 + R_{b,dB-Hz}$$

$$= Z_{0,dB} - 43$$

Các sơ đồ điều chế	$P_{c,dBW}$	$P_{c,mW}$
BPSK	-32,4	0,575
4-PSK	-32,4	0,575
8-PSK	-29,4	1,148
16-PSK	-24,5	3,550
16-QAM	-28,5	1,410
BFSK nhất quán	-29,4	1,148
4-FSK nhất quán	-32,2	0,603
8-FSK nhất quán	-34,0	0,398
BFSK không nhất quán	-28,8	1,318
4-FSK không nhất quán	-31,8	0,661
8-FSK không nhất quán	-33,0	0,501

### Bài 35:

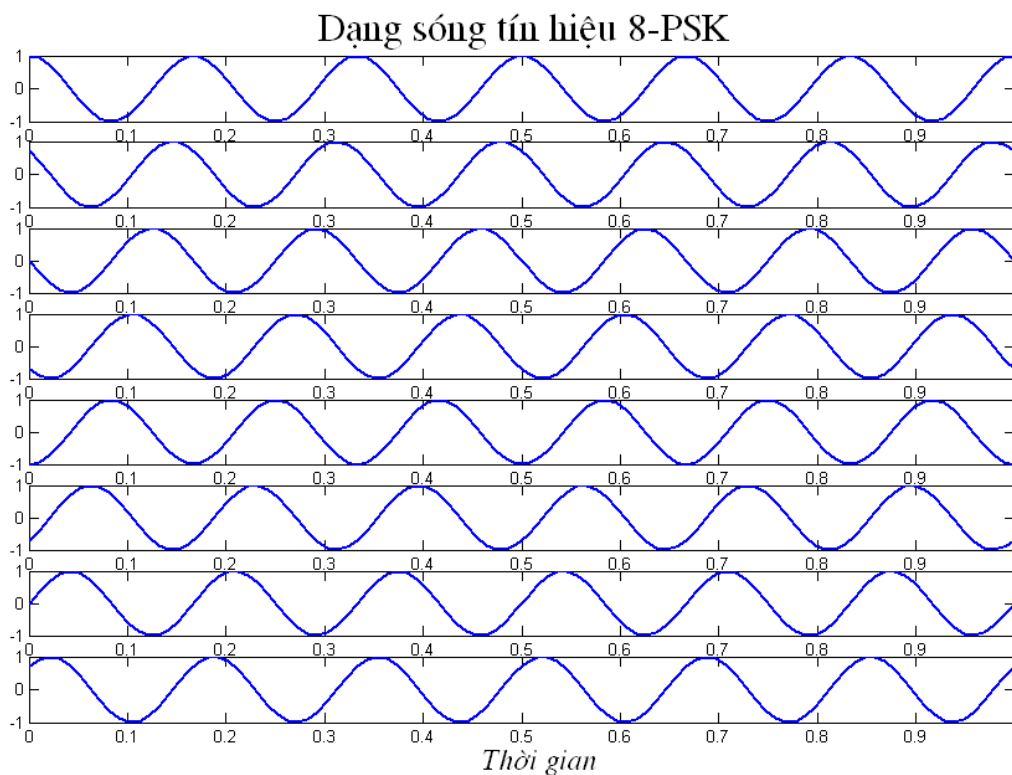
Hãy đọc mã chương trình mô phỏng **NVD\_8PSK\_Waveform.m** để tạo dạng sóng cho tín hiệu 8PSK, chạy chương trình mô phỏng, thay đổi các tham số đầu vào của chương trình, phân tích kết quả mô phỏng:

### Hướng dẫn giải

Tín hiệu điều chế được xác định bởi:

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos[2\pi f_c t + \theta(t) + \theta]$$

**trong đó:**  $E = nE_b$ ;  $T = nT_b$ ;  $n = \log_2 M$ ;  $\theta(t) = 2\pi i / M$ ,  $i = 0, 1, 2, \dots, M-1$ ;  $\theta_0$  là góc pha ban đầu.



```
function y = NVD_8PSK_Waveform
    T = 1;
    M = 8;
    Es = T/2;
    fc = 6/T; % carrier frequency
    N = 100; % number of samples
    delta_T = T/(N-1);
    t = 0:delta_T:T;
    u0 = sqrt(2*Es/T)*cos(2*pi*fc*t);
    u1 = sqrt(2*Es/T)*cos(2*pi*fc*t+2*pi/M);
    u2 = sqrt(2*Es/T)*cos(2*pi*fc*t+4*pi/M);
    u3 = sqrt(2*Es/T)*cos(2*pi*fc*t+6*pi/M);
    u4 = sqrt(2*Es/T)*cos(2*pi*fc*t+8*pi/M);
    u5 = sqrt(2*Es/T)*cos(2*pi*fc*t+10*pi/M);
    u6 = sqrt(2*Es/T)*cos(2*pi*fc*t+12*pi/M);
    u7 = sqrt(2*Es/T)*cos(2*pi*fc*t+14*pi/M);
    % plotting commands follow
    subplot(8,1,1);
```

```
plot(t,u0);
subplot(8,1,2);
plot(t,u1);
subplot(8,1,3);
plot(t,u2);
subplot(8,1,4);
plot(t,u3);
subplot(8,1,5);
plot(t,u4);
subplot(8,1,6);
plot(t,u5);
subplot(8,1,7);
plot(t,u6);
subplot(8,1,8);
plot(t,u7);
```

### Bài 35:

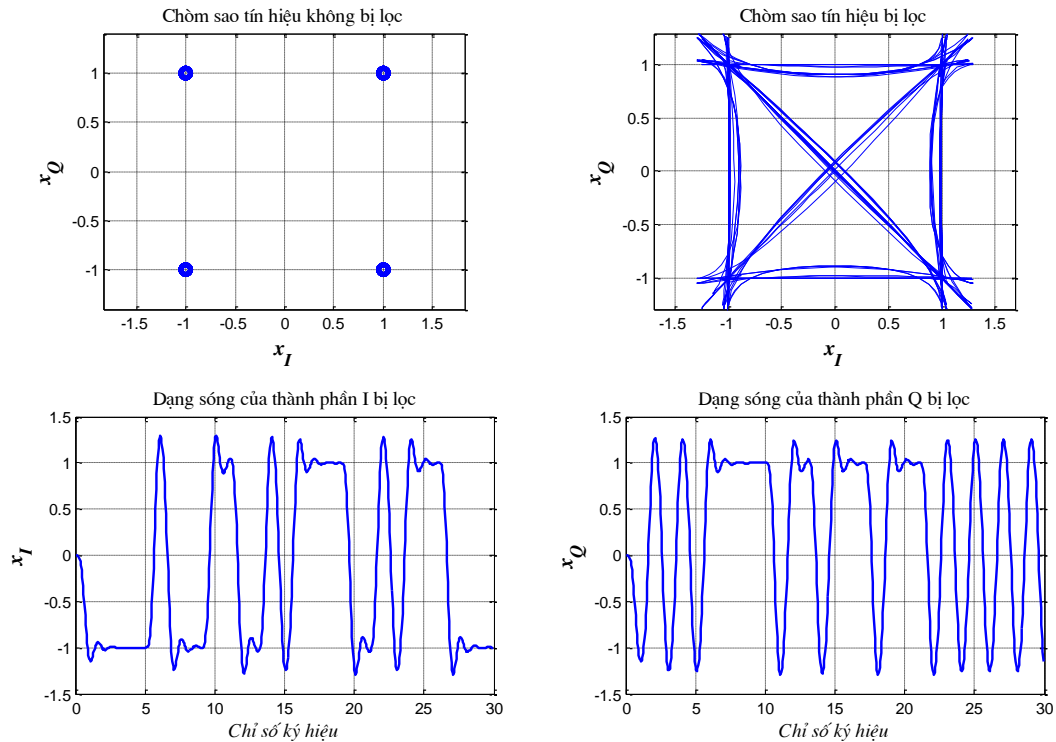
Hãy đọc mã chương trình mô phỏng **NVD4\_qamsim.m** cho QPSK, vẽ mô hình mô phỏng, chạy chương trình mô phỏng, thay đổi các tham số đầu vào của chương trình, phân tích kết quả mô phỏng:

#### Hướng dẫn giải

Để tạo ra một tín hiệu QPSK, các ký hiệu số liệu thành phần đồng pha Q và vuông pha Q được tạo ra bằng cách lấy 2 ký hiệu nhị phân mỗi lần. Theo đó, mỗi ký hiệu số liệu bao gồm 1 trong các cặp nhị phân: 00, 01, 10, 11; số mức trên trục I và trục Q được gán một cách độc lập, chúng đều bằng 2. Bộ lọc định dạng xung được chọn là bộ lọc Butterworth bậc sáu. Độ rộng băng của bộ lọc được dùng trong mô phỏng được ký hiệu là ***bw***, bằng tốc độ ký hiệu. Nhập các tham số tương ứng theo bảng cho chương trình và chạy kết quả được cho ở hình dưới đây.

Các tham số đầu vào chương trình mô phỏng <b>NVD4_qamsim.m</b> cho QPSK				
Số mức trên trục I	Số mức trên trục Q	Số ký hiệu được mô phỏng	Số mẫu trên ký hiệu	Độ rộng băng thông bộ lọc
2	2	100	20	0,1
<b>Lưu ý:</b> Bằng cách chọn các giá trị khác cho tập các tham số này cho phép khảo sát các ảnh hưởng của chúng lên kết quả mô phỏng. Kết quả mô phỏng cho trường hợp các giá trị này được cho ở hình dưới đây				





```
% file NVD4_qamsim
clc;
clear all;
close all;
=====
% setup parameter
=====
levelx      = input('So muc tren thanh phan I > ');
levely      = input('So muc tren thanh phan Q > ');
m           = input('So ky hieu > ');
n           = input('So mau tren ky hieu > ');
bw          = input('Do rong bang thong bo loc, 0<bw<1 > ');
=====
% Main Program
=====
[xd,xq]      = qam(levelx,levely,m,n);
[b,a]        = butter(6,bw);           % determine filter coefficients
yd           = filter(b,a,xd);         % filter Inphase coefficient
yq           = filter(b,a,xq);         % filter Quadrature coefficient
=====
% Process & Display results
=====
h4_5 = figure
set(h4_5,'name','H4.4 & H4.5: NVD');
subplot(2,2,1)                        % first pane
plot(xd,xq,'o')                       % unfiltered scatterplot
a      = 1.4;
maxd    = max(xd);
maxq    = max(xq);
mind    = min(xd);
minq    = min(xq);
axis([a*mind a*maxd a*minq a*maxq])
axis equal
xlabel('x_I','fontname','.Vntime','fontsize',14);
ylabel('x_Q','fontname','.Vntime','fontsize',14);
title('Chòm sao tín hiệu không bị lọc','fontname','.Vntime','fontsize',12)
```

```

grid on;
%=====
subplot(2,2,2)                                % second pane
plot(yd,yq)                                    % filtered scatterplot
axis equal;
xlabel('x_I','fontname','.Vntime','fontsize',14);
ylabel('x_Q','fontname','.Vntime','fontsize',14);
title('Chòm sao t́yn híu b́p lăc','fontname','.Vntime','fontsize',12);
grid on;
%=====
sym      = 30;                                % number of symbols in time plot
nsym     = (0:sym*n)/n;                       % x axis vector for time plots
subplot(2,2,3)                                % third pane
plot(nsym(1:sym*n),yd(1:sym*n))               % filtered direct component
xlabel('Chø sè ký híu','fontname','.Vntime','fontsize',12);
ylabel('x_I','fontname','.Vntime','fontsize',14);
title('Đ́ng săng cĩa thụn̄h ph́c̄n I b́p lăc','fontname','.Vntime','fontsize',12);
grid on;
%=====
subplot(2,2,4)                                % fourth pane
plot(nsym(1:sym*n),yq(1:sym*n))               % filtered quadrature component
xlabel('Chø sè ký híu','fontname','.Vntime','fontsize',12);
ylabel('x_Q','fontname','.Vntime','fontsize',14);
title('Đ́ng săng cĩa thụn̄h ph́c̄n Q b́p lăc','fontname','.Vntime','fontsize',12);
grid on;

```

### Bài 36:

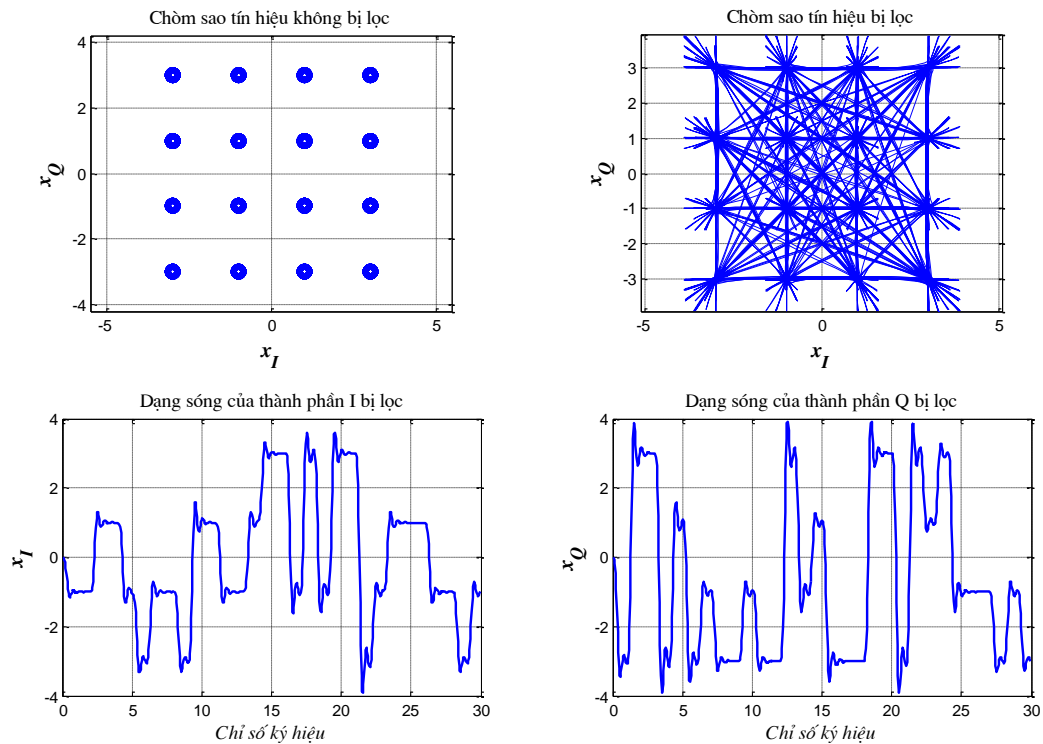
Hãy đọc mã chương trình mô phỏng **NVD4\_qamsim.m** cho 16-QAM, vẽ mô hình mô phỏng, chạy chương trình mô phỏng, thay đổi các tham số đầu vào chương trình mô phỏng và phân tích kết quả mô phỏng

### Hướng dẫn giải

Kết quả mô phỏng biểu đồ sao tín hiệu được minh hoạ hình dưới đây cùng với dãy nhị phân tương ứng cho mỗi ký hiệu 16-QAM. Cũng giống như trường hợp QPSK, việc sắp xếp các ký hiệu nhị phân thành ký hiệu 16-QAM là tùy ý nhưng việc sắp xếp này thường được xác định sao cho nhận được các kết quả mã Gray, lưu ý rằng kết quả mô phỏng là sắp xếp mã Gray.

Đối với 16-QAM, các mức trên trục I và trục Q đều là bốn mức. Bộ lọc định dạng xung là bộ lọc Butterworth bậc 6. Trong đó, băng thông của bộ lọc gấp 2 lần tốc độ ký hiệu. Tham số đầu vào của chương trình Matlab cho trường hợp này theo

<i>Các tham số đầu vào chương trình mô phỏng NVD4_qamsim.m cho 16-QAM</i>				
Số mức trên trục I	Số mức trên trục Q	Số ký hiệu được mô phỏng	Số mẫu trên ký hiệu	Độ rộng băng thông bộ lọc
4	4	500	20	0,2
<b>Lưu ý:</b> Chọn các giá trị khác cho tập các tham số này cho phép khảo sát các ảnh hưởng của chúng lên kết quả mô phỏng. Kết quả mô phỏng trong trường hợp cụ thể này được cho ở hình dưới đây				



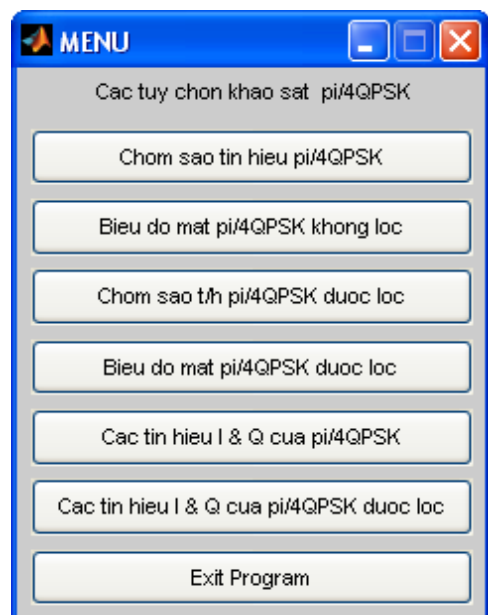
### Bài 37:

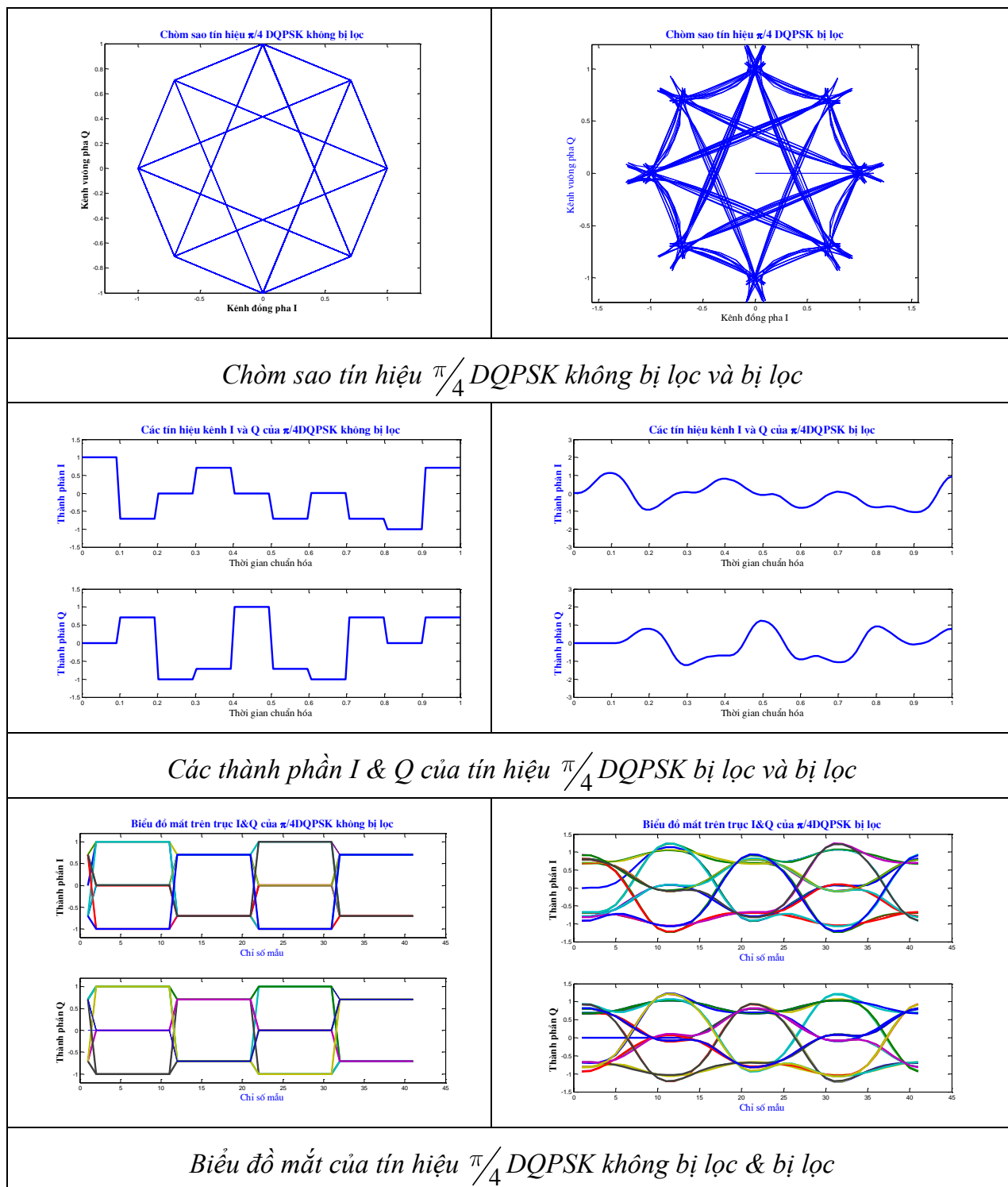
Hãy đọc mã chương trình mô phỏng **NVD8\_pi4sim** cho hệ thống QPSK vi sai  $\pi/4$ , vẽ mô hình mô phỏng, thực hiện mô phỏng, thay đổi các tham số mô phỏng và phân tích kết quả mô phỏng.

### Hướng dẫn giải

Khi chạy chương trình ở file **NVD8\_pi4sim**. Chạy chương trình **NVD\_pi4sim** tại dấu nhắc cửa sổ lệnh Matlab. Từ menu này chọn một trong 7 lựa chọn sau:

1. Chòm sao tín hiệu  $\pi/4$  DQPSK không bị lọc
2. Biểu đồ mắt tín hiệu  $\pi/4$  DQPSK không bị lọc
3. Chòm sao tín hiệu  $\pi/4$  DQPSK bị lọc
4. Biểu đồ mắt tín hiệu  $\pi/4$  DQPSK bị lọc
5. Các tín hiệu vuông pha và đồng pha không bị lọc
6. Các tín hiệu vuông pha và đồng pha bị lọc
7. Thoát chương trình mô phỏng





Lưu ý rằng, khi ta nghiên cứu mã chương trình mô phỏng, ở đó minh họa nhiều chương trình con về hỗ trợ biểu diễn kết quả mô phỏng. Mã chương trình được sử dụng để tạo các đồ thị khác các chương trình mô phỏng khác. Ta minh họa 3 kết quả đáng quan tâm. Kết quả mô phỏng minh họa biểu đồ tán xạ (chòm sao tín hiệu), các tín hiệu kênh vuông pha và kênh đồng pha, và các biểu đồ mắt trên kênh vuông pha và đồng pha. Lưu ý rằng, bằng cách mở rộng trục tín hiệu ba chiều trong không gian (I,Q,t) thấy rõ mối quan hệ giữa các hình. Để được rõ hơn về mức độ ảnh hưởng của tham số hệ thống lên các kết quả quan sát, ta thay đổi các tham số bộ lọc và các tham số khác như số ký hiệu mô phỏng, lấy mẫu trên ký hiệu.

```
% File: NVD8_pi4_sim.m
clc;
clear all;
close all;
%=====
% Setup parameters
%=====
m      = 200;          % number of symbols
bits   = 2*m;          % number of bits
sps    = 10;           % samples per symbol
iphase = 0;            % initial phase
order  = 5;            % filter order
bw      = 0.1;         % normalized filter bandwidth
%=====
% initialize vectors
%=====
data    = zeros(1,bits);
d       = zeros(1,m);
q       = zeros(1,m);
dd      = zeros(1,m);
qq      = zeros(1,m);
theta   = zeros(1,m);
thetaout = zeros(1,sps*m);
%=====
% set direct and quadrature bit streams
%=====
data    = round(rand(1,bits));
dd      = data(1:2:bits-1);
qq      = data(2:2:bits);
%=====
% main programs
%=====
theta(1)      = iphase;          % set initial phase
thetaout(1:sps) = theta(1)*ones(1,sps);
for k=2:m
    if dd(k) == 1
        phi_k = (2*qq(k)-1)*pi/4;
    else
        phi_k = (2*qq(k)-1)*3*pi/4;
    end
    theta(k) = phi_k + theta(k-1);
    for i=1:sps
        j = (k-1)*sps+i;
        thetaout(j) = theta(k);
    end
end
d = cos(thetaout);
q = sin(thetaout);
[b,a] = butter(order,bw);
df = filter(b,a,d);
qf = filter(b,a,q);

%=====
% postprocessor for plotting
%=====
kk = 0;          % set exit counter
while kk == 0    % test exit counter
k = menu('Cac tuy chon khao sat pi/4QPSK',...
    'Chom sao tin hieu pi/4QPSK',...
    'Bieu do mat pi/4QPSK khong loc',...
    'Chom sao t/h pi/4QPSK duoc loc',...
    'Bieu do mat pi/4QPSK duoc loc',...
    'Cac tin hieu I & Q cua pi/4QPSK',...
    'Cac tin hieu I & Q cua pi/4QPSK duoc loc',...
    'Exit');
```

```

'Exit Program');
if k == 1
    sigcon(d,q);          % plot unfiltered signal con.
    pause;
elseif k == 2
    dqeye(d,q,4*sps);    % plot unfiltered eye diagram
    pause;
elseif k == 3
    sigcon(df,qf);        % plot filtered signal con.
    pause;
elseif k == 4
    dqeye(df,qf,4*sps);  % plot filtered eye diagram
    pause;
elseif k == 5
    numbsym = 10;         % number of symbols plotted
    dt      = d(1:numbsym*sps); % truncate d vector
    qt      = q(1:numbsym*sps); % truncate q vector
    dqplot(dt,qt);        % plot truncated d and q signals
    pause;
elseif k == 6
    numbsym = 10;         % number of symbols to be plotted
    dft      = df(1:numbsym*sps); % truncate df to desired value
    qft      = qf(1:numbsym*sps); % truncate qf to desired value
    dqplot(dft,qft);      % plot truncated signals
    pause;
elseif k == 7
    kk      = 1;          % set exit counter to exit value
end
end

function []=sigcon(x,y)
% File: sigcon.m
plot(x,y)
axis('square');
axis('equal');
xlabel('Kanh @ang pha I','fontname','.vntime','fontsize',12);
ylabel('Kanh vuong pha Q ','fontname','.vntime','color','b','fontsize',12);
title('Cham sao tyn hiou
\pi/4DQPSK','fontname','.vntime','color','b','fontsize',16)

function [] = dqeye(xd,xq,m)
% File: dqeye.m
lx = length(xd);          % samples in data segment
kcol = floor(lx/m);       % number of columns
xda = [0,xd]; xqa = [0,xq]; % append zeros
for j = 1:kcol            % column index
    for i = 1:(m+1)        % row index
        kk = (j-1)*m+i;   % sample index
        y1(i,j) = xda(kk);
        y2(i,j) = xqa(kk);
    end
end
subplot(211)              % direct channel
plot(y1);
title('Biou @a mat tran troc I & Q cua
\pi/4DQPSK','fontname','.vntimeH','color','b','fontsize',14);
xlabel('Chø se meu','fontname','.vntime','color','b','fontsize',12);
ylabel('Thunh phun I ','fontname','.vntime','color','b','fontsize',12);

subplot(212)              % quadrature channel
plot(y2);
xlabel('Chø se meu','fontname','.vntime','color','b','fontsize',12);
ylabel('Thunh phun Q ','fontname','.vntime','color','b','fontsize',12);
subplot(111)              % restore

```

```
function [] = dqplot(xd,xq)
% File: dqplot.m
lx      = length(xd);
t       = 0:lx-1;
nt      = t/(lx-1);
nxd     = xd(1,1:lx);
nxq     = xq(1,1:lx);

subplot(211)
plot(nt,nxd);
a       = axis;
axis([a(1) a(2) 1.5*a(3) 1.5*a(4)]);
title('Tỷ lệ nhiễu kênh I vụ Q của \pi/4DQPSK','fontname','.vntime','color','b','fontsize',16);
xlabel('Thời gian chuẩn hóa','fontname','.vntime','fontsize',12);
ylabel('Thụnh ph@n I','fontname','.vntime','color','b','fontsize',12);
subplot(212)
plot(nt,nxq);
a       = axis;
axis([a(1) a(2) 1.5*a(3) 1.5*a(4)]);
xlabel('Thời gian chuẩn hóa','fontname','.vntime','fontsize',12);
ylabel('Thụnh ph@n Q','fontname','.vntime','color','b','fontsize',12);
subplot(111)

function y_out = delay(y_in,n_delay)

% y_out = delay(y_in,n_delay); shifts the vector y_in to the right
% by n_delay elements and fills in the vacated elements with 0's
%

N      = length(y_in);
y_out  = zeros(size(y_in));

y_out(n_delay+1:N) = y_in(1:N-n_delay);

function output = hold_in(in,k)
output = [];
L      = length(in);
b      = ones(k,1)*in;
output = reshape(b,1,L*k);
```

### Bài 38:

Hãy đọc mã chương trình mô phỏng **NVD8\_ofdmsimtx.m** và **NVD8\_ofdmsimrx.m** thể nguyên lý hoạt động của hệ thống OFDM, vẽ mô hình mô phỏng, thực hiện mô phỏng, thay đổi các tham số mô phỏng và phân tích kết quả mô phỏng.

### Hướng dẫn giải

Tín hiệu phát được mô tả bởi:

$$s(t) = \text{Re} \left\{ e^{j2\pi f_c t} \sum_{m=0}^{\infty} \sum_{\ell=0}^{N_F-1} \sum_{k=K_{\min}}^{K_{\max}} C_{m,\ell,k} \psi_{m,\ell,k}(t) \right\}$$

trong đó

$$\psi_{m,\ell,k}(t) = \begin{cases} e^{j2\pi \frac{k'}{T_U}(t - \Delta - \ell T_S - N_F \cdot m T_S)} & , \quad (\ell + N_F \cdot m)T_S \leq t \leq (\ell + N_F \cdot m + 1)T_S \\ 0, & \text{khác} \end{cases}$$

- $k$  là chỉ số sóng mang  
 $\ell$  là chỉ số ký hiệu OFDM  
 $m$  là số khung truyền dẫn  
 $K$  là số sóng mang được truyền  
 $T_S$  là khoảng thời gian của ký hiệu  
 $T_U$  là nghịch đảo của khoảng cách giữa các sóng mang.  
 $f_c$  là tần số trung tâm của sóng mang vô tuyến RF.  
 $k'$  là chỉ số sóng mang tương ứng với tần số trung tâm nghĩa là  
 $k' = k - \frac{K_{max} + K_{min}}{2}$   
 $C_{m,\ell,k}$  là ký hiệu phức cho sóng mang  $k$  của ký hiệu dữ liệu thứ  $\ell + 1$  trong khung thứ  $m$ .

Nếu xét biểu thức trên trong khoảng thời gian từ  $t=0$  đến  $t=T_S$  (tức là khảo sát trong một chu kỳ ký hiệu) thì nó trở thành

$$s(t) = \text{Re} \left\{ e^{j2\pi f_c t} \sum_{k=K_{min}}^{K_{max}} C_{0,0,k} e^{j2\pi k'(t-\Delta)/T_U} \right\}$$

với

$$k' = k - \frac{K_{max} + K_{min}}{2}$$

Nếu thực hiện điều chế 4-QAM để chuyển tín hiệu bằng tần cơ sở thành tín hiệu thông băng thì tín hiệu phát  $s(t)$  tương ứng với các thành phần đồng pha  $m_I(t)$  và vuông pha  $m_Q(t)$  là

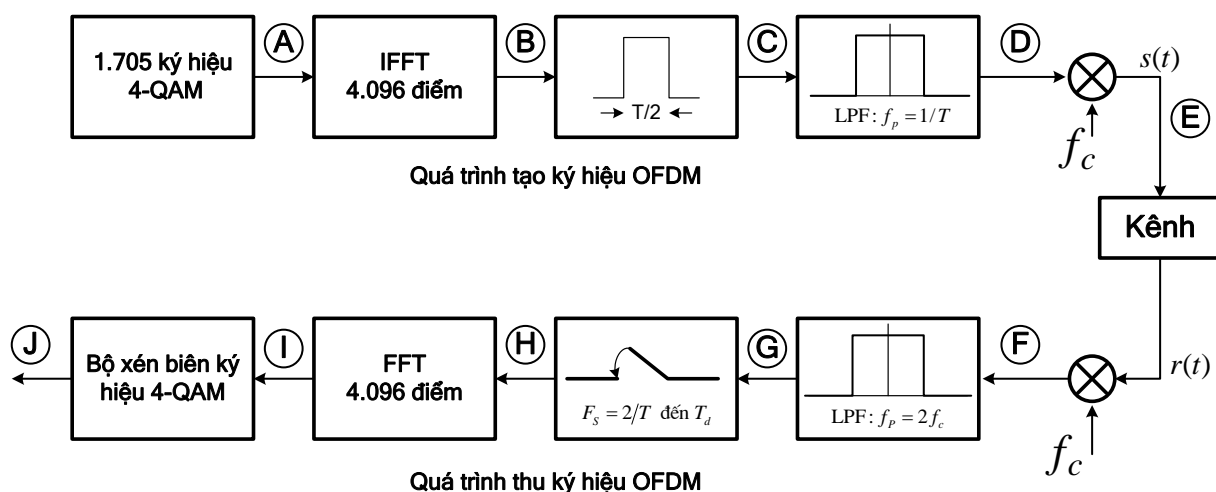
$$s(t) = m_I(t) \cdot \cos(2\pi f_c t) + m_Q(t) \cdot \sin(2\pi f_c t)$$

Khi này  $s(t)$  được khai triển như sau

$$\begin{aligned}
 s(t) = & \sum_{k=K_{min}}^{K_{max}} \text{Re}(C_{0,0,k}) \cdot \cos \left[ 2\pi \left( \frac{k - \frac{K_{max} + K_{min}}{2}}{T_U} + f_c \right) t - \frac{\Delta}{T_U} \right] \\
 & - \sum_{k=K_{min}}^{K_{max}} \text{Im}(C_{0,0,k}) \cdot \sin \left[ 2\pi \left( \frac{k - \frac{K_{max} + K_{min}}{2}}{T_U} + f_c \right) t - \frac{\Delta}{T_U} \right]
 \end{aligned}$$

Ta mô phỏng tín hiệu OFDM có cấu trúc khung và đa khung. Theo đó, mỗi khung có độ lâu  $T_F$  gồm  $N_F$  ký hiệu OFDM, mỗi đa khung gồm  $M_F$  khung ( $T_{MF} = M_F \times T_F$ ). Mỗi ký hiệu được tạo thành bởi một tập  $K$  (từ  $K_{min}$  đến  $K_{max}$ ) sóng mang và được truyền trong khoảng thời gian  $T_S$ ;  $T_S = T_U + \Delta$ . Các tham số cụ thể được cho ở bảng dưới đây. Các kết quả mô phỏng tương ứng với các điểm được đánh nhãn trên sơ đồ mô phỏng.



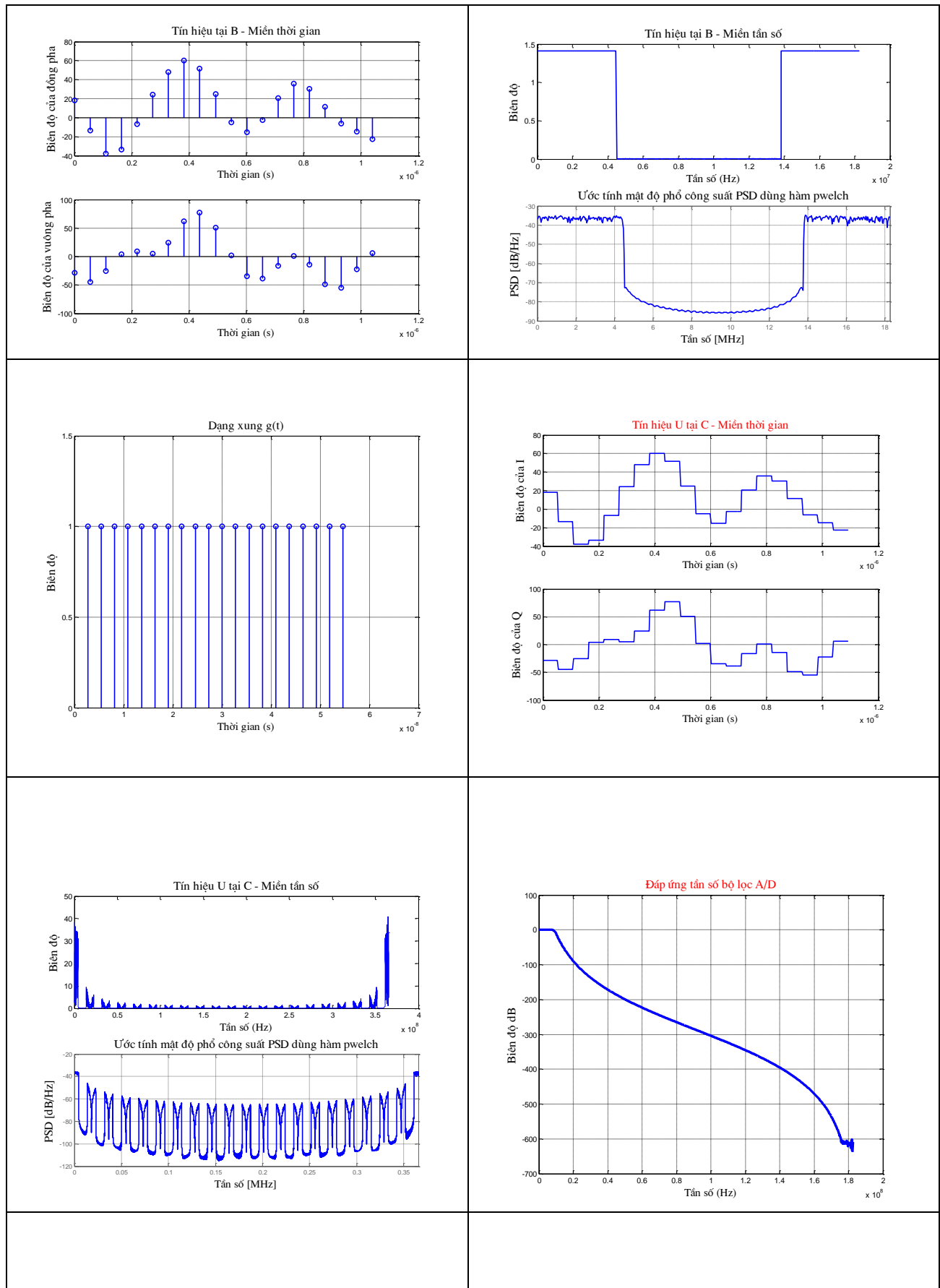


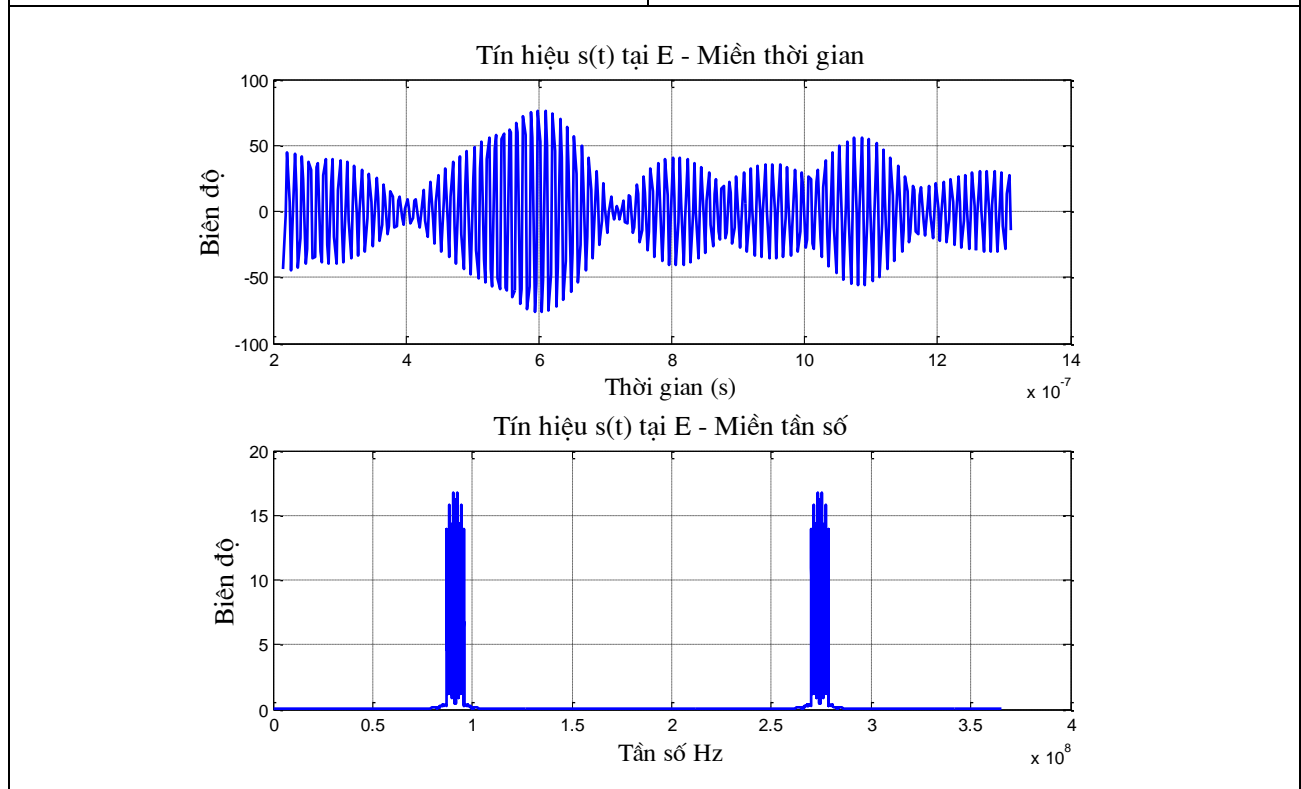
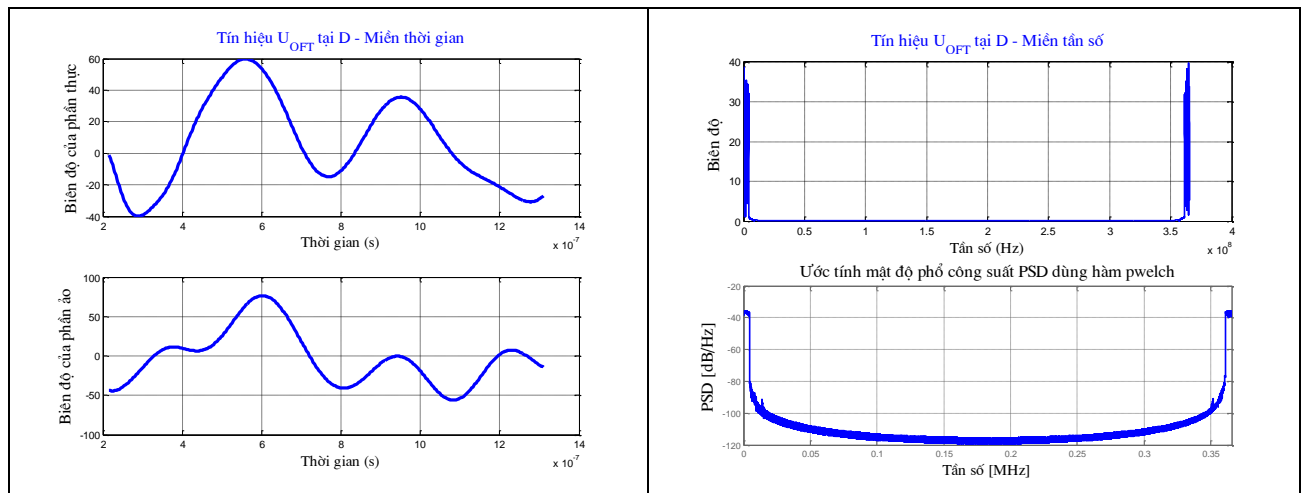
Các tham số và các giá trị của chúng trong mô phỏng OFDM

Tham số	Giá trị			
Chu kỳ cơ bản $T$	$7/64 \mu s$			
Số sóng mang $K$ ( $K_{min} \leq K \leq K_{max}$ )	1705 ( $K_{min} = 0; K_{max} = 1704$ )			
Khoảng thời gian ký hiệu hữu hiệu $T_U$	$224 \mu s$			
Khoảng cách các sóng mang con $1/T_U$	4464 Hz			
Khoảng cách $K_{min}$ và $K_{max}(K-1)/T_U$	7,61 MHz			
Khoảng thời gian bảo vệ được phép $\Delta/T_U$	1/4	1/8	1/16	1/32
Khoảng thời gian phần $T_U$	$2048 \times T = 224 \mu s$			
Khoảng thời gian bảo vệ $\Delta$	$512 \times T = 56 \mu s$	$256 \times T = 28 \mu s$	$128 \times T = 14 \mu s$	$512 \times T = 56 \mu s$
Khoảng thời gian ký hiệu $T_s = \Delta + T_U$	$2.560 \times T = 280 \mu s$	$2.304 \times T = 252 \mu s$	$1.176 \times T = 238 \mu s$	$2.112 \times T = 231 \mu s$

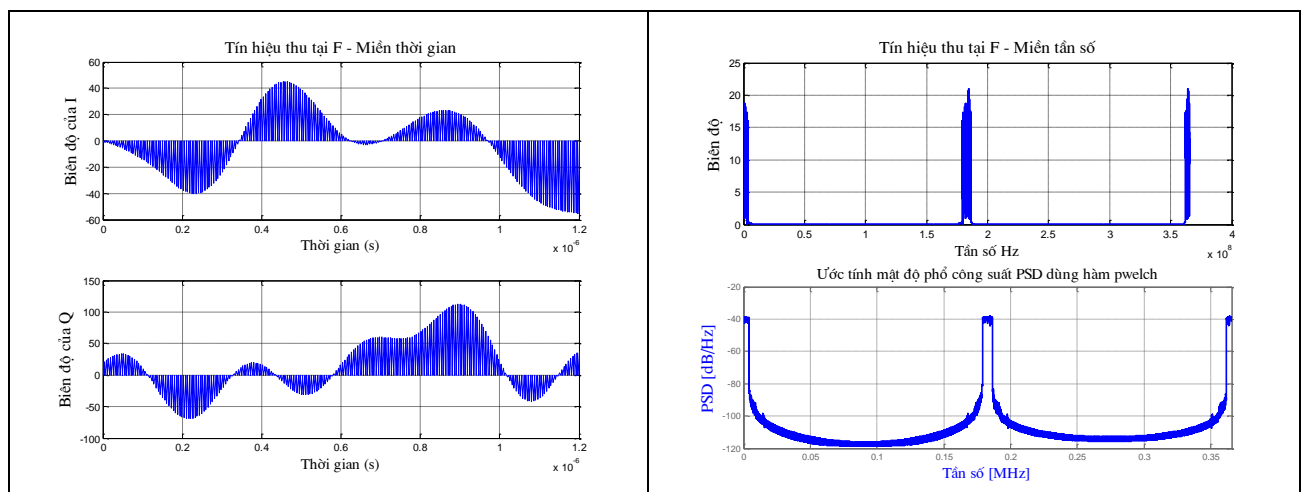
Chương trình mô phỏng tín hiệu OFDM được cho ở **NVD8\_ofdmsimtx.m** và **NVD8\_ofdmsimrx.m**, mô phỏng tín hiệu OFDM trong miền thời gian và tần số. Trong mã chương trình trình bày cách dùng các chương trình xử lý để trực quan hóa tín hiệu và hệ thống trong mô phỏng. Chương trình này rất hữu hiệu để tìm hiểu cách thiết kế và thực hiện mô phỏng hệ thống OFDM đơn giản. Từ chương trình này cho phép nâng cấp để tạo chương trình mô phỏng hệ thống phức tạp hơn. Sử dụng các chương trình thường trình hậu xử lý để trình bày các tín hiệu tại các điểm khác nhau trong sơ mô phỏng như sau:

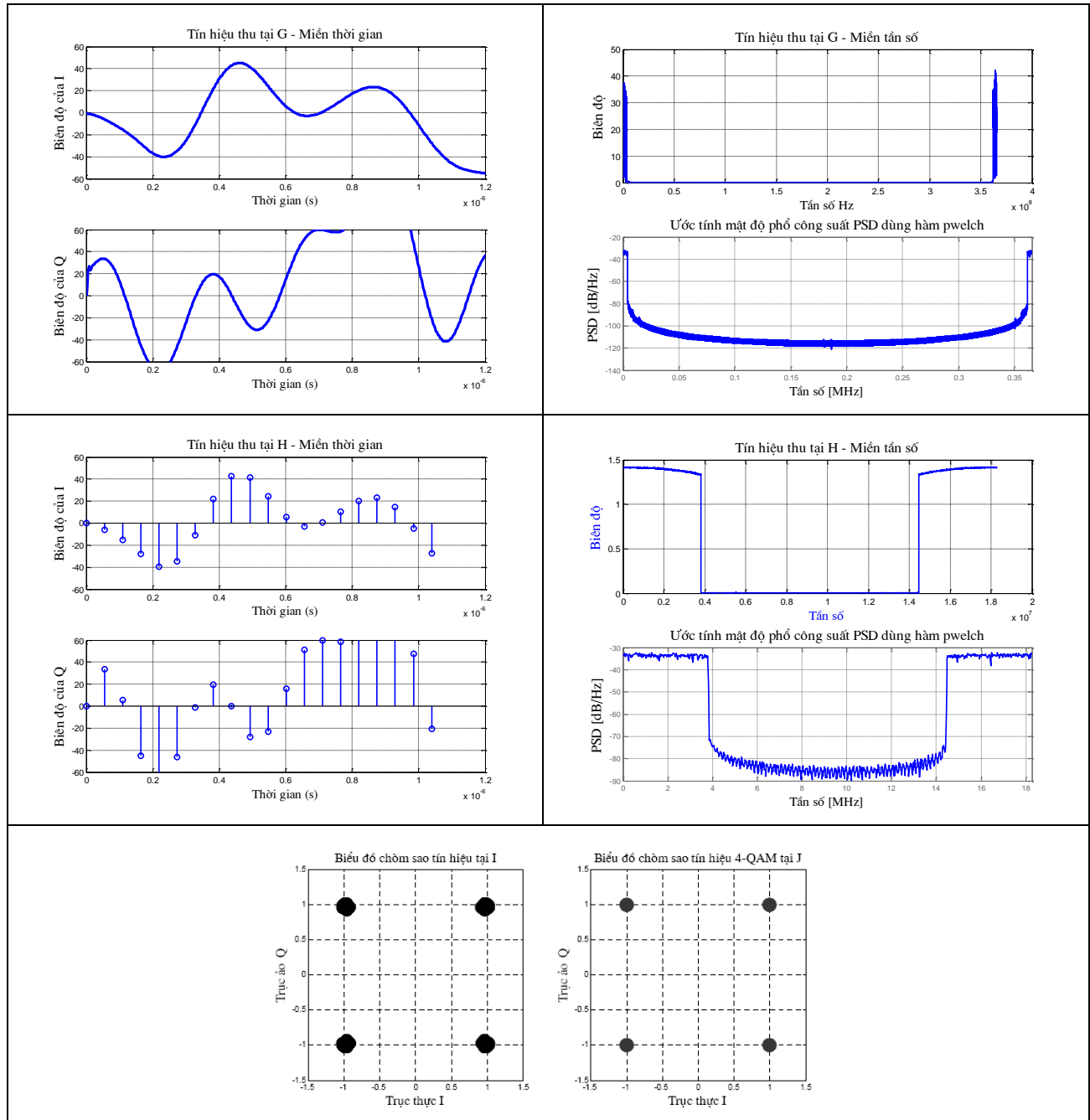
✓ Kết quả mô phỏng phần phát





## Kết quả mô phỏng phần thu





**% file: NVD8\_ofdmsimtx.m**

% The available bandwidth is 8 MHz

```
clc;
clear all;
close all;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Simulation Parameters
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
FS      = 4096/2;           %IFFT/FFT length <=> FFT_size
T        = 0.1094e-6;        %baseband elementary period (7/64)e-6
% T      = 4.8828e-008;      %baseband elementary period (7/64)e-6
Tu       = T*(FS/2);         %useful OFDM symbol period
G        = 1/4;              %choice of 1/4, 1/8, 1/16, and 1/32
delta    = G*Tu;             %guard band duration
Ts       = delta+Tu;         %total OFDM symbol period
Kmax     = 1000;             %number of subcarriers
```

```
Kmin    =    0;
q        =    10;                %carrier period to elementary period ratio
fc       =    q*1/T;            %carrier frequency <=> 90MHz
Rs       =    4*fc;             %simulation period
t        =    0:1/Rs:Tu;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%Data generator (A) %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

M        =    Kmax+1;
rand('state',0);
a        =    -1+2*round(rand(M,1)).' + i*(-1+2*round(rand(M,1))).';
A        =    length(a);
info     =    zeros(FS,1);
info(1:(A/2)) = [ a(1:(A/2)).' ];                %Zero padding
info((FS-(A/2)-1):FS) = [ a((A/2)+1:A).'];
% info_Mon = info;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%Subcarriers generation (B) %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
carriers =    FS.*ifft(info,FS);
tt       =    0:T/2:Tu;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%=====
h1 = figure(1);
set(h1,'color','g');
set(h1,'Name','H8_OFDMTx.1: NVD');
subplot(211);
stem(tt(1:20),real(carriers(1:20)));
xlabel('Thời gian (s)','FontName','.VnTime','FontSize',12)
ylabel('Biên độ của sóng pha','FontName','.VnTime','FontSize',14)
title('Tỷ lệ hiệu suất B - Miền thời gian','FontName','.VnTime','FontSize',14);
grid on
subplot(212);
stem(tt(1:20),imag(carriers(1:20)));
xlabel('Thời gian (s)','FontName','.VnTime','FontSize',12)
ylabel('Biên độ của sóng vuong pha','FontName','.VnTime','FontSize',14)
grid on
%=====
h2 = figure(2);
set(h2,'color','g');
set(h2,'Name','H8_OFDMTx.2: NVD');
f    = (2/T)*(1:(FS))/(FS);
subplot(211);
plot(f,abs(fft(carriers,FS))/FS);
xlabel('Tần số (Hz)','FontName','.VnTime','FontSize',12)
ylabel('Biên độ','FontName','.VnTime','FontSize',14)
title('Tỷ lệ hiệu suất B - Miền tần số','FontName','.VnTime','FontSize',14);
subplot(212);
pwelch(carriers,[],[],[],2/T);
xlabel('Tần số [MHz]','FontName','.VnTime','FontSize',12)
ylabel('PSD [dB/Hz]','FontName','.VnTime','FontSize',14)
title('Đặc tính mật độ phổ công suất PSD dùng hàm pwelch','FontName','.VnTime','FontSize',14);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%D/A simulation %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
L        =    length(carriers);
chips    =    [ carriers.';zeros((2*q)-1,L)];
p        =    1/Rs:1/Rs:T/2;
```

```
g = ones(length(p),1); %pulse shape
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%=====
h3 = figure(3);
set(h3,'color','g');
set(h3,'Name','H8_OFDMTx.3: NVD');
stem(p,g);
xlabel('Thời gian (s)','FontName','.VnTime','FontSize',12)
ylabel('Biên độ','FontName','.VnTime','FontSize',14)
% axis([ ]);
title('Định xung g(t)','FontName','.VnTime','FontSize',14);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dummy = conv(g,chips(:));
u = [dummy(1:length(t))]; % (C)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%=====
h4 = figure(4);
set(h4,'color','g');
set(h4,'Name','H8_OFDMTx.4: NVD');
subplot(211);
plot(t(1:400),real(u(1:400)));
xlabel('Thời gian (s)','FontName','.VnTime','FontSize',12)
ylabel('Biên độ của I','FontName','.VnTime','FontSize',14)
title('Tỷ lệ hiệu U thực I - Miền thời gian','FontName','.VnTime','color','r','FontSize',14);
subplot(212);
plot(t(1:400),imag(u(1:400)));
xlabel('Thời gian (s)','FontName','.VnTime','FontSize',12)
ylabel('Biên độ của Q','FontName','.VnTime','FontSize',14)

%=====
h5 = figure(5);
set(h5,'color','g');
set(h5,'Name','H8_OFDMTx.5: NVD');
ff = (Rs)*(1:(q*FS))/(q*FS);
subplot(211);
plot(ff,abs(fft(u,q*FS))/FS);
xlabel('Tần số (Hz)','FontName','.VnTime','FontSize',12)
ylabel('Biên độ','FontName','.VnTime','FontSize',14)
title('Tỷ lệ hiệu U thực I - Miền tần số','FontName','.VnTime','FontSize',14);
subplot(212);
pwelch(u,[],[],[],Rs);
xlabel('Tần số [MHz]','FontName','.VnTime','FontSize',12)
ylabel('PSD [dB/Hz]','FontName','.VnTime','FontSize',14)
title('Đặc tính mật độ phổ công suất PSD dùng hàm pwelch','FontName','.VnTime','FontSize',14);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%reconstruction filter
[b,a] = butter(13,1/20);
[H,F] = FREQZ(b,a,FS,Rs);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%=====
h6 = figure(6);
set(h6,'color','g');
set(h6,'Name','H8_OFDMTx.6: NVD');
plot(F,20*log10(abs(H)));
xlabel('Tần số (Hz)','FontName','.VnTime','FontSize',12)
ylabel('Biên độ dB','FontName','.VnTime','FontSize',14)
```

```
title('$p$ ợng t$ן s$ẻ b$ẻ l$ẻ  
A/D','FontName','.VnTime','color','r','FontSize',14);  
  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%baseband signal (D)  
uoft = filter(b,a,u);  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
h7 = figure(7);  
set(h7,'color','g');  
set(h7,'Name','H8_OFDMTx.7: NVD');  
  
subplot(211);  
plot(t(80:480),real(uoft(80:480)));  
xlabel('Th$ẻi gian (s)','FontName','.VnTime','FontSize',12)  
ylabel('Bi$ẻn $ẻc c$ẻn ph$ẻn th$ẻc','FontName','.VnTime','FontSize',14)  
title('T$ẻn hi$ẻu U_O_F_T t$ẻi D - Mi$ẻn th$ẻi  
gian','FontName','.VnTime','color','b','FontSize',14);  
  
subplot(212);  
plot(t(80:480),imag(uoft(80:480)));  
xlabel('Th$ẻi gian (s)','FontName','.VnTime','FontSize',12)  
ylabel('Bi$ẻn $ẻc c$ẻn ph$ẻn $ẻo','FontName','.VnTime','FontSize',14)  
  
%=====   
h8 = figure(8);  
set(h8,'color','g');  
set(h8,'Name','H8_OFDMTx.8: NVD');  
  
subplot(211);  
plot(ff,abs(fft(uoft,q*FS))/FS);  
xlabel('T$ẻn s$ẻ (Hz)','FontName','.VnTime','FontSize',12)  
ylabel('Bi$ẻn $ẻc ','FontName','.VnTime','FontSize',14)  
title('T$ẻn hi$ẻu U_O_F_T t$ẻi D - Mi$ẻn t$ẻn  
s$ẻ','FontName','.VnTime','color','b','FontSize',14);  
subplot(212);  
pwelch(uoft,[],[],[],Rs);  
xlabel('T$ẻn s$ẻ [MHz]','FontName','.VnTime','FontSize',12)  
ylabel('PSD [dB/Hz]','FontName','.VnTime','FontSize',14)  
title('I$ẻc t$ẻn m$ẻt $ẻc ph$ẻ c$ẻng su$ẻt PSD d$ẻng h$ẻm  
pwelch','FontName','.VnTime','FontSize',14);  
  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%Upconverter  
s_tilde = (uoft.*exp(1i*2*pi*fc*t));  
s = real(s_tilde); %passband signal (E)  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
%=====   
h9 = figure(9);  
set(h9,'color','g');  
set(h9,'Name','H8_OFDMTx.9: NVD');  
  
subplot(211)  
plot(t(80:480),s(80:480));  
xlabel('Th$ẻi gian (s)','FontName','.VnTime','FontSize',12)  
ylabel('Bi$ẻn $ẻc','FontName','.VnTime','FontSize',14)  
title('T$ẻn hi$ẻu s(t) t$ẻi E - Mi$ẻn th$ẻi  
gian','FontName','.VnTime','FontSize',14);  
  
subplot(212)  
plot(ff,abs(fft((real(uoft).)*cos(2*pi*fc*t)),q*FS))/FS);  
xlabel('T$ẻn s$ẻ Hz','FontName','.VnTime','FontSize',12)  
ylabel('Bi$ẻn $ẻc','FontName','.VnTime','FontSize',14)
```

```
title('Tỷ lệ hiệu suất (t) tại E - Miền tần số','FontName','.VnTime','FontSize',14);  
clc;
```

### % function NVD8\_ofdmsimrx

```
% function NVD8_ofdmsimrx
```

```
% Reception section simulation for OFDM system
```

```
clc;
```

```
clear all;
```

```
close all;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%% Simulation Parameters%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
Tu      = 224e-6;           % useful OFDM symbol period  
T        = Tu/2048;         % baseband elementary period  
G        = 1/4;             % choice of 1/4, 1/8, 1/16, and 1/32  
delta    = G*Tu;           % guard band duration  
Ts        = delta+Tu;       % total OFDM symbol period  
Kmax     = 1705;            % number of subcarriers  
Kmin     = 0;               %  
FS       = 4096;           % IFFT/FFT length  
q        = 10;             % carrier period to elementary period ratio  
fc       = q*1/T;          % carrier frequency  
Rs       = 4*fc;           % simulation period  
t        = 0:1/Rs:Tu;      %  
tt       = 0:T/2:Tu;       %
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%Data generator%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
sM       = 2;  
[x,y]    = meshgrid((-sM+1):2:(sM-1), (-sM+1):2:(sM-1));  
alphabet = x(:) + 1i*y(:);  
N        = Kmax+1;  
rand('state',0);  
a        = -1+2*round(rand(N,1)).' + i*(-1+2*round(rand(N,1))).';  
A        = length(a);
```

```
info      = zeros(FS,1);  
info(1:(A/2)) = [ a(1:(A/2)).' ];  
info((FS-(A/2)-1):FS) = [ a((A/2)+1:A).' ];  
carriers  = FS.*ifft(info,FS);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%Upconverter%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
L        = length(carriers);  
chips    = [ carriers.'; zeros((2*q)-1,L) ];  
p        = 1/Rs:1/Rs:T/2;  
g        = ones(length(p),1);  
dummy    = conv(g,chips(:));  
u        = [dummy; zeros(46,1)];  
[b,aa]   = butter(13,1/20);  
uoft     = filter(b,aa,u);  
delay    = 64;              %Reconstruction filter delay  
s_tilde  = (uoft(delay+(1:length(t))).').*exp(1i*2*pi*fc*t);  
  
s        = real(s_tilde);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%% OFDM RECEPTION
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%% Downconversion
```



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r_tilde      =      exp(-1i*2*pi*fc*t).*s; %(F)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%=====
h1 = figure(1);
set(h1,'color','g');
set(h1,'Name','H8_OFDMRx.1: NVD');
subplot(211);
plot(t,real(r_tilde));
axis([0e-7 12e-7 -60 60]);
xlabel('Thời gian (s)','FontName','.VnTime','FontSize',12)
ylabel('Biên độ của I','FontName','.VnTime','FontSize',14)
title('Tỷ lệ thu hồi F - Miền thời gian','FontName','.VnTime','FontSize',14);
grid on;
subplot(212);
plot(t,imag(r_tilde));
axis([0e-7 12e-7 -100 150]);
xlabel('Thời gian (s)','FontName','.VnTime','FontSize',12)
ylabel('Biên độ của Q','FontName','.VnTime','FontSize',14)
grid on;
%=====
h2 = figure(2);
set(h2,'color','g');
set(h2,'Name','H8_OFDMRx.2: NVD');
ff=(Rs)*(1:(q*FS))/(q*FS);
subplot(211);
plot(ff,abs(fft(r_tilde,q*FS))/FS);
xlabel('Tần số Hz','FontName','.VnTime','FontSize',12)
ylabel('Biên độ','FontName','.VnTime','FontSize',14)
title('Tỷ lệ thu hồi F - Miền tần số','FontName','.VnTime','FontSize',14);
grid on;

subplot(212);
pwelch(r_tilde,[],[],[],Rs);
xlabel('tần số [MHz]','FontName','.VnTime','color','b','FontSize',12)
ylabel('PSD [dB/Hz]','FontName','.VnTime','color','b','FontSize',14)
title('Đặc tính mật độ phổ công suất PSD dạng hòm pwelch','FontName','.VnTime','FontSize',14);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%% Carrier suppression %%%%%%%%%
[B,AA] = butter(3,1/2);
r_info = 2*filter(B,AA,r_tilde); %Baseband signal continuous-time (G)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%=====
h3 = figure(3);
set(h3,'color','w');
set(h3,'Name','H8_OFDMRx.3: NVD');
subplot(211);
plot(t,real(r_info));
xlabel('Thời gian (s)','FontName','.VnTime','FontSize',12)
ylabel('Biên độ của I','FontName','.VnTime','FontSize',14)
title('Tỷ lệ thu hồi G - Miền thời gian','FontName','.VnTime','FontSize',14);
axis([0 12e-7 -60 60]);
grid on;
subplot(212);
plot(t,imag(r_info));
xlabel('Thời gian (s)','FontName','.VnTime','FontSize',12)
ylabel('Biên độ của Q','FontName','.VnTime','FontSize',14)
axis([0 12e-7 -60 60]);
```

```
grid on;

%=====
h4 = figure(4);
f=(2/T)*(1:(FS))/(FS);
set(h4,'color','w');
set(h4,'Name','H8_OFDMRx.4: NVD');
subplot(211);
plot(ff,abs(fft(r_info,q*FS))/FS);
xlabel('Tần số Hz','FontName','.VnTime','FontSize',12)
ylabel('Biên độ','FontName','.VnTime','FontSize',14)
title('Tỷ lệ thu thập G - Miền tần số','FontName','.VnTime','FontSize',14);
grid on;

subplot(212);
pwelch(r_info,[],[],[],Rs);
xlabel('tần số [MHz]','FontName','.VnTime','FontSize',12)
ylabel('PSD [dB/Hz]','FontName','.VnTime','FontSize',14)
title('Đặc tính mật độ phổ công suất PSD dùng hàm
pwelch','FontName','.VnTime','FontSize',14);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%Sampling %%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Baseband signal, discrete time
r_data = real(r_info(1:(2*q):length(t)))...
+ 1i*imag(r_info(1:(2*q):length(t))); % (H); 1i*

%=====
h5 = figure(5);
set(h5,'color','g');
set(h5,'Name','H8_OFDMRx.5: NVD');
subplot(211);
stem(tt(1:20),(real(r_data(1:20))));
xlabel('Thời gian (s)','FontName','.VnTime','FontSize',12)
ylabel('Biên độ của I','FontName','.VnTime','FontSize',14)
title('Tỷ lệ thu thập H - Miền thời
gian','FontName','.VnTime','FontSize',14);
axis([0 12e-7 -60 60]);
grid on;
subplot(212);
stem(tt(1:20),(imag(r_data(1:20))));
xlabel('Thời gian (s)','FontName','.VnTime','FontSize',8)
ylabel('Biên độ của Q','FontName','.VnTime','FontSize',12)
axis([0 12e-7 -60 60]);
grid on;

%=====
h6 = figure(6);
set(h6,'color','g');
set(h6,'Name','H8_OFDMRx.6: NVD');
f=(2/T)*(1:(FS))/(FS);
subplot(211);
plot(f,abs(fft(r_data,FS))/FS);
xlabel('Tần số','FontName','.VnTime','color','b','FontSize',12)
ylabel('Biên độ','FontName','.VnTime','color','b','FontSize',14)
title('Tỷ lệ thu thập H - Miền tần số','FontName','.VnTime','FontSize',14);
grid on;
subplot(212);
pwelch(r_data,[],[],[],2/T);
xlabel('tần số [MHz]','FontName','.VnTime','FontSize',12)
ylabel('PSD [dB/Hz]','FontName','.VnTime','FontSize',14)
title('Đặc tính mật độ phổ công suất PSD dùng hàm
pwelch','FontName','.VnTime','FontSize',14);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% FFT %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
info_2N      =    (1/FS).*fft(r_data,FS); % (I)
info_h       =    [info_2N(1:A/2) info_2N((FS-(A/2)-1):FS)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% Slicing %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k=1:N,
    a_hat(k)=alphabet((info_h(k)-alphabet)==min(info_h(k)-alphabet)); % (J)
end;

%=====
h7 = figure(7)
set(h7,'color','g');
set(h7,'Name','H8_OFDMRx.7: NVD');
subplot(121)
plot(info_h((1:A)),'.k');
xlabel('Trởc thùc I','FontName','.VnTime','FontSize',12)
ylabel('Trởc thùc Q','FontName','.VnTime','FontSize',14)
title('Biểu đồ chòm sao tỖn hiệu tỖi I','FontName','.VnTime','FontSize',14);
axis square;
axis equal;
grid on;
axis([-1.5 1.5 -1.5 1.5]);

subplot(122)
plot(a_hat((1:A)), 'or');
xlabel('Trởc thùc I','FontName','.VnTime','FontSize',12)
ylabel('Trởc thùc Q','FontName','.VnTime','FontSize',14)
title('Biểu đồ chòm sao tỖn hiệu 4-QAM tỖi J','FontName','.VnTime','FontSize',14);
axis square;
axis equal;
grid on;
axis([-1.5 1.5 -1.5 1.5]);
```

### Bài 39:

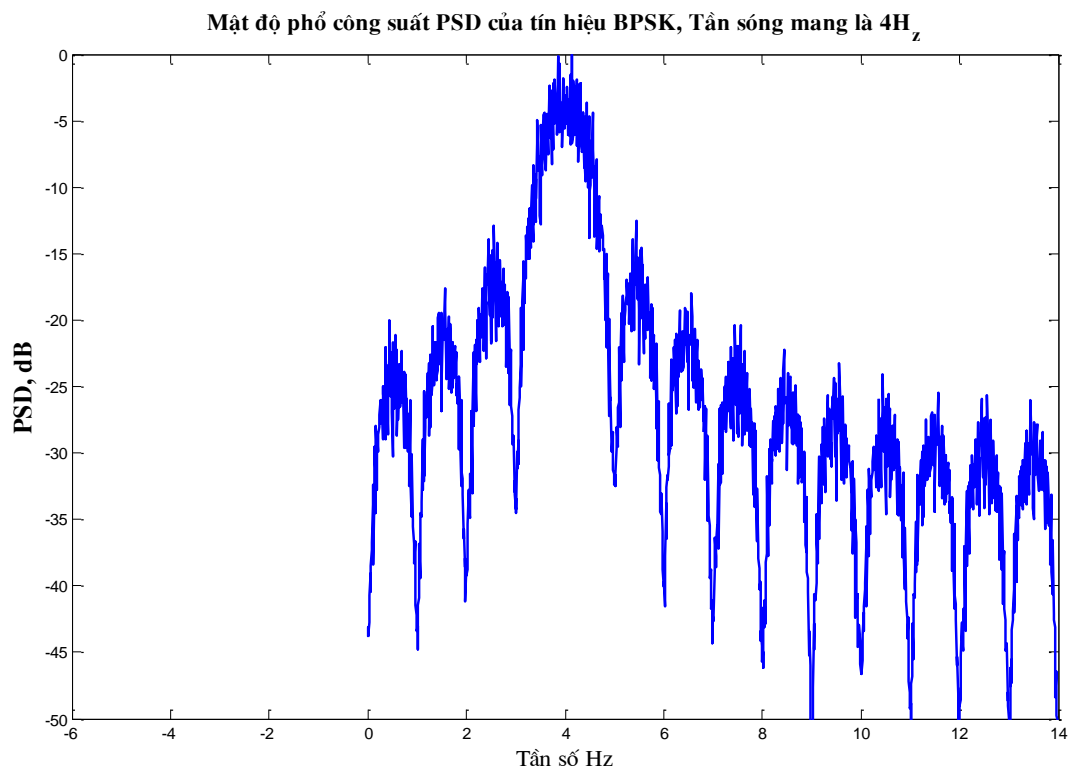
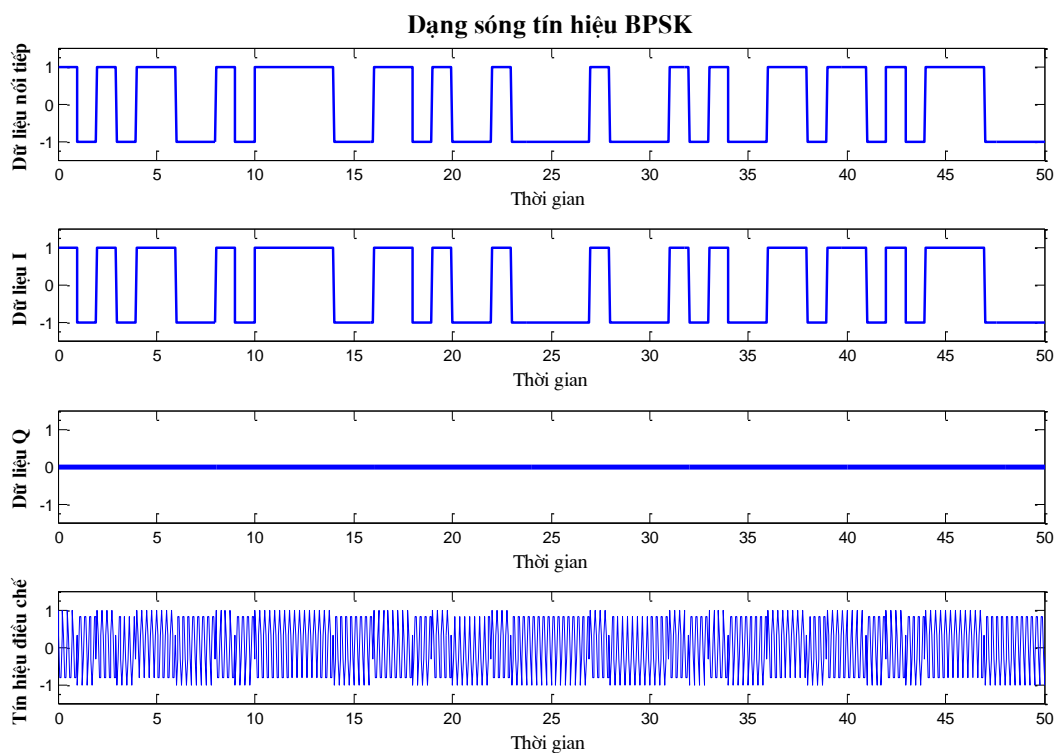
Hãy đọc mã chương trình mô phỏng **NVD8\_AD\_sim.m** thể hiện nguyên lý hoạt động của hệ thống BPSK, QPSK, OQPSK, MSK, 16QAM, vẽ mô hình mô phỏng, thực hiện mô phỏng, thay đổi các tham số mô phỏng và phân tích kết quả mô phỏng.

### Hướng dẫn giải

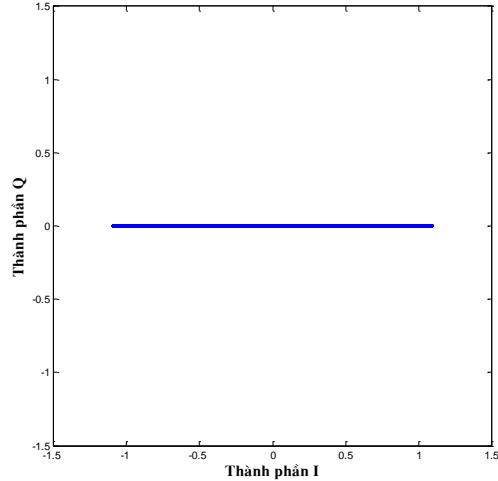
Chương trình mô phỏng tín hiệu và hệ thống BPSK, QPSK, OQPSK, MSK, 16QAM được cho ở **NVD8\_AD\_sim.m** thực hiện mô phỏng tín hiệu trong miền thời gian và tần số như: dạng sóng, biểu đồ mắt, biểu đồ pha (chòm sao), đường bao, mật độ phổ công suất PSD. Dưới đây là một số kết quả mô phỏng điển hình cho các hệ thống này. Để được tường minh, ta nên nghiên cứu kỹ mã chương trình Matlab, xác định các tham số đầu vào cho từng hệ thống, vẽ lưu đồ mô phỏng. Đây là một chương trình tổng hợp có tính liên kết cao. Trong đó, thể hiện rõ cách lấy dữ liệu từ các điểm của hệ thống để xử lý hiển thị, trực quan hóa nguyên lý hoạt động của các hệ thống này. Đặc biệt có ý nghĩa người mới lập trình Matlab và muốn tìm hiểu nguyên lý hoạt động các hệ thống truyền thông này. Dưới đây là một số kết quả điển hình khi chạy chương trình mô phỏng. Để hiểu rõ nguyên lý hoạt động các hệ thống này, ta nghiên cứu kỹ mã chương trình Matlab này, vẽ lại mô hình mô phỏng, công thức hóa cho từng đoạn chương trình và chạy chương trình mô phỏng cho

tập các tham số đặc trưng khác nhau cho từng mô hình đó. Có thể nói đây là một bài tập rất tốt.

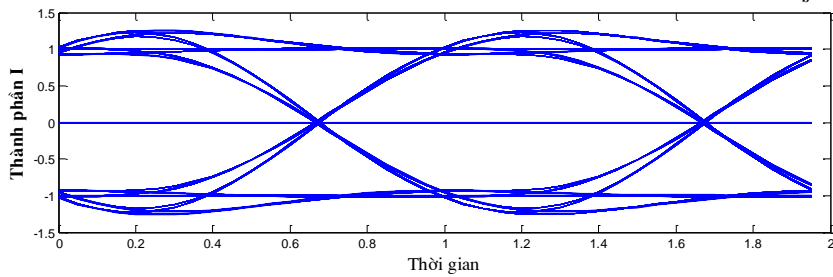
### Hệ thống BPSK:



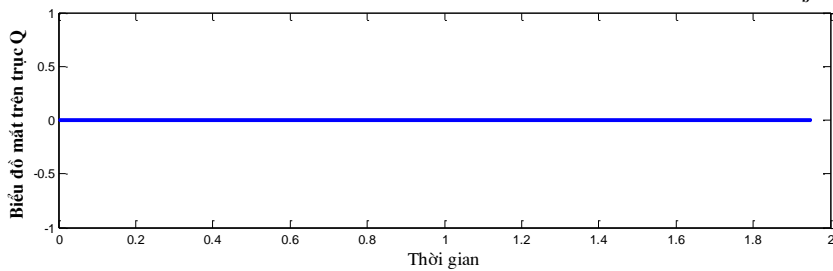
Biểu đồ pha tín hiệu BPSK đã lọc bởi bộ lọc Butterworth bậc 2;  $BW = 0,7R_b$



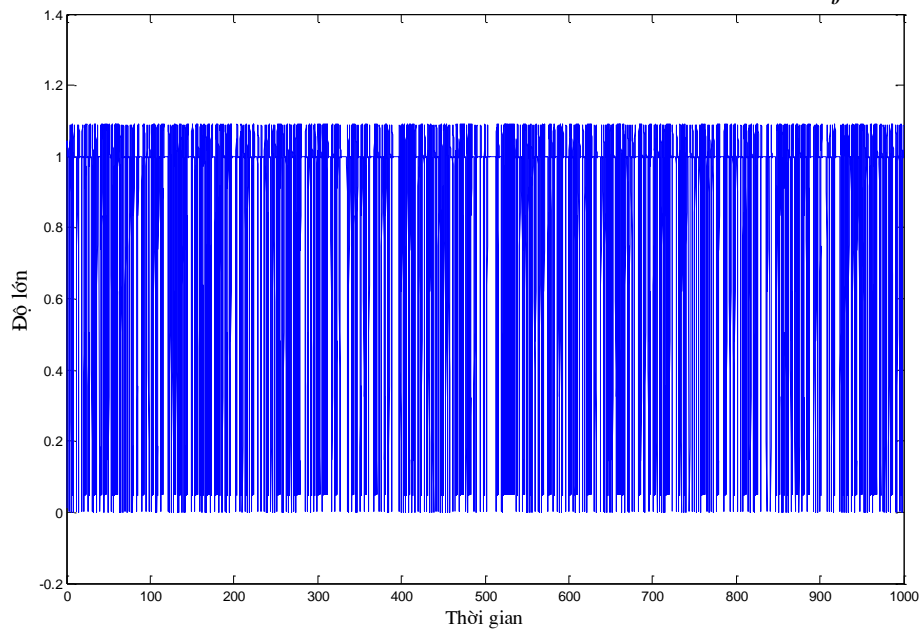
Biểu đồ mắt tín hiệu BPSK đã lọc bởi bộ lọc Butterworth bậc 4 độ rộng băng  $BW = 0,7R_b$



Biểu đồ mắt tín hiệu BPSK đã lọc bởi bộ lọc Butterworth bậc 4 độ rộng băng  $BW = 0,7R_b$

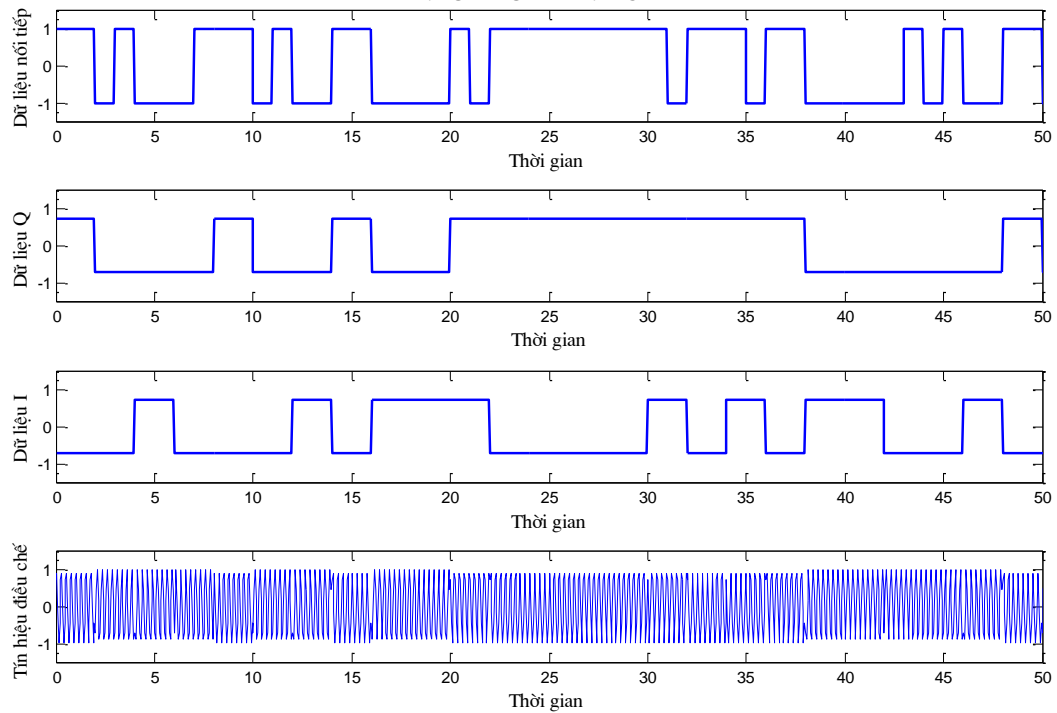


Đồ thị bao tín hiệu BPSK đã lọc bởi bộ lọc Butterworth bậc 2 có  $BW = 0,7R_b$

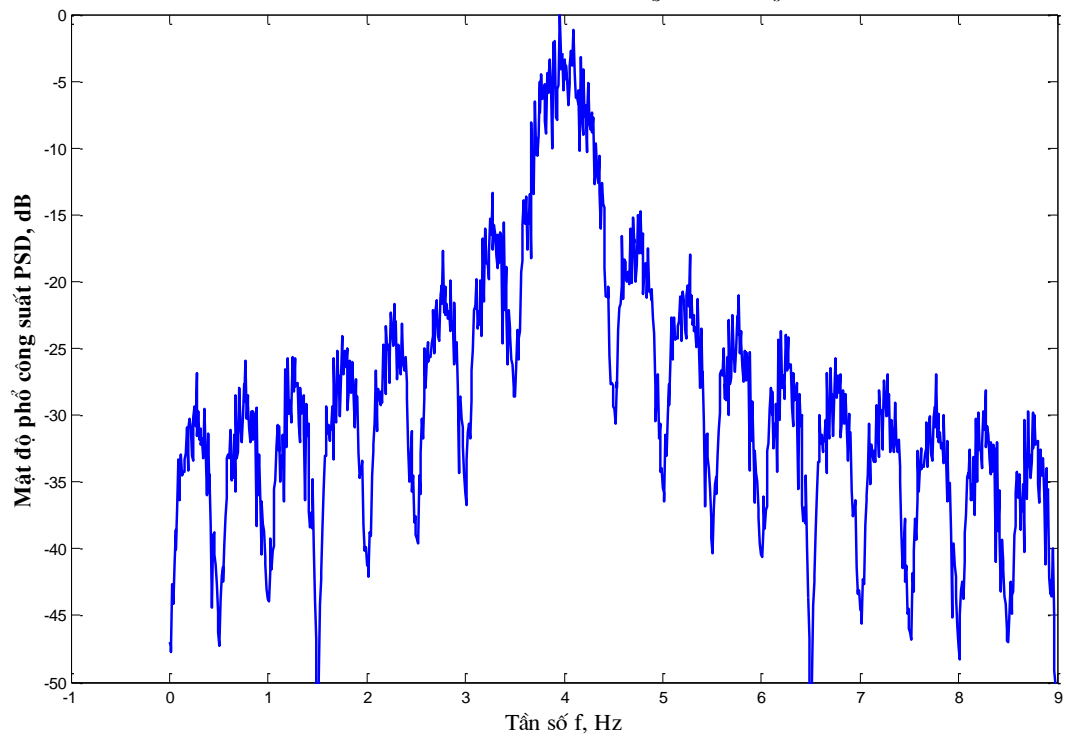


## Hệ thống QPSK

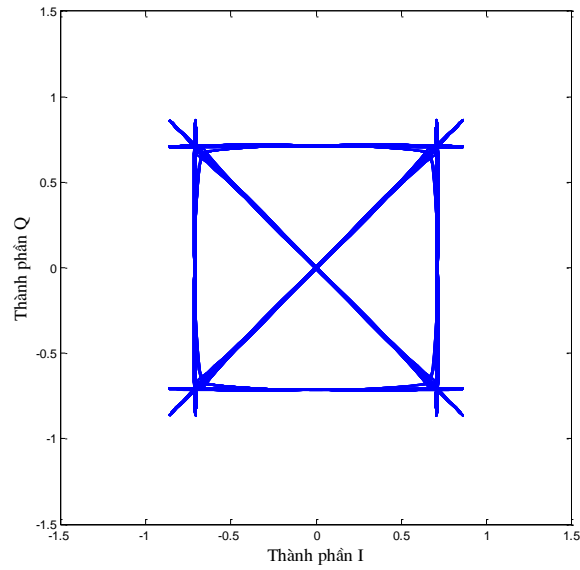
Dạng sóng tín hiệu QPSK



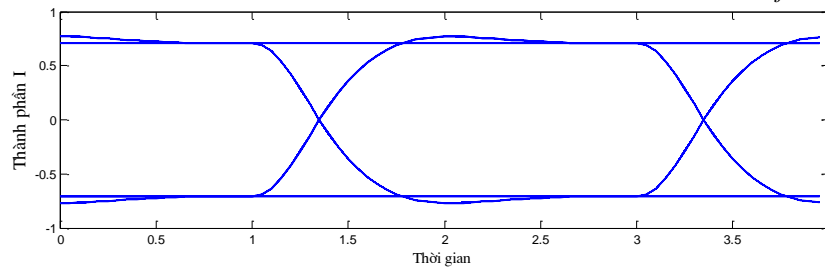
Phổ tín hiệu QPSK; Tần số trung tâm = 4Hz;  $R_b = 1$  bps;  $R_s = 0,5$ ; No ave = 7



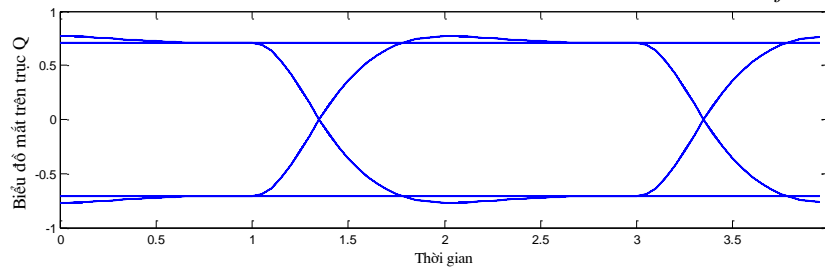
Biểu đồ pha tín hiệu QPSK ; được lọc bởi bộ lọc Butterworth bậc 4;  $BW=0,7R_b$



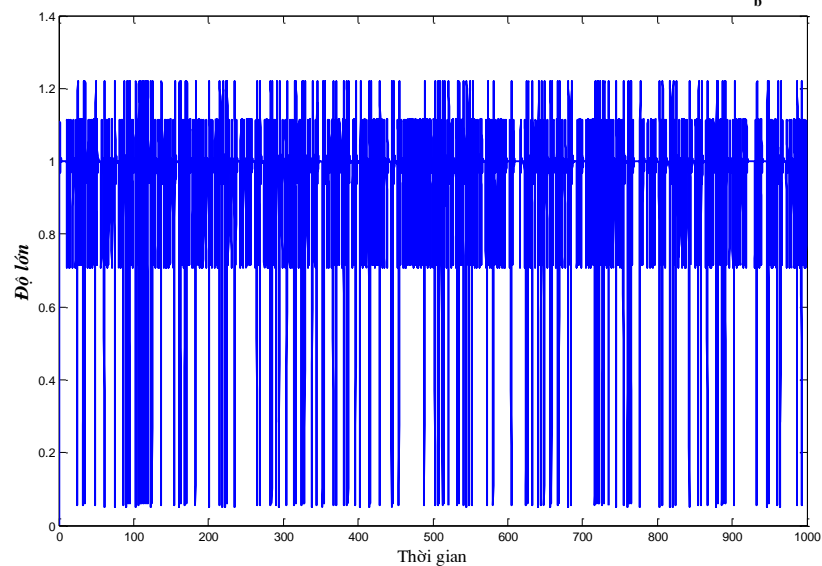
Biểu đồ mắt tín hiệu QPSK được lọc bởi bộ lọc Butterworth bậc 2 có độ rộng băng  $BW=0,7R_b$



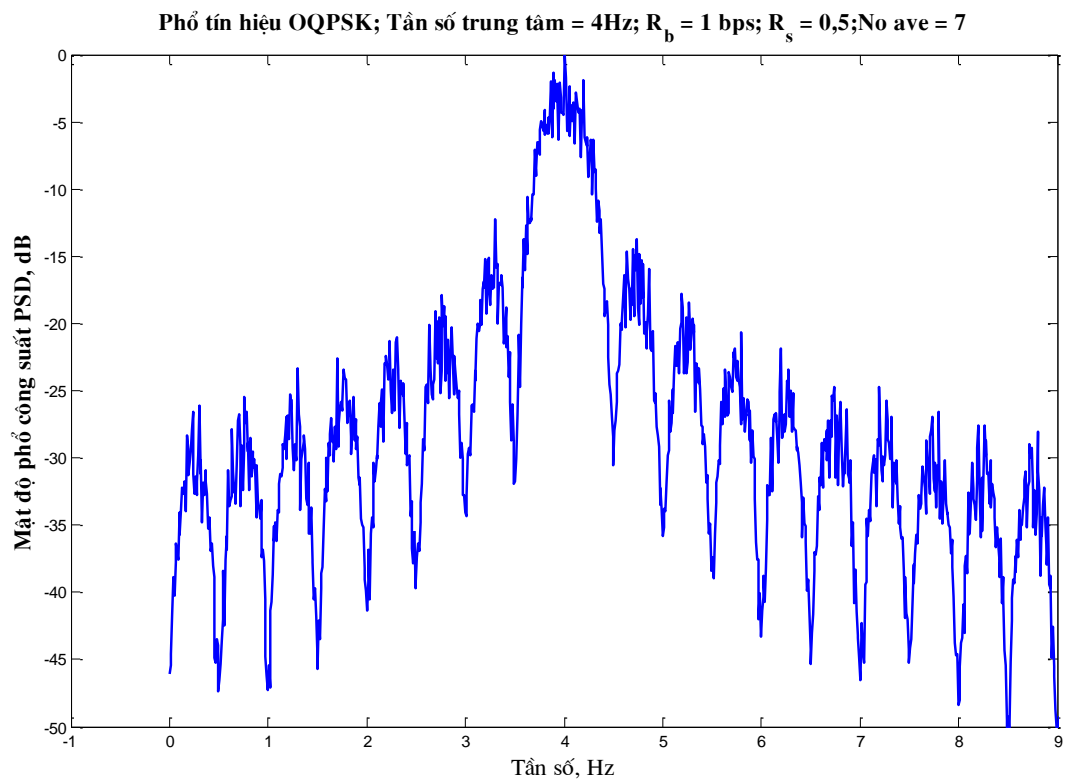
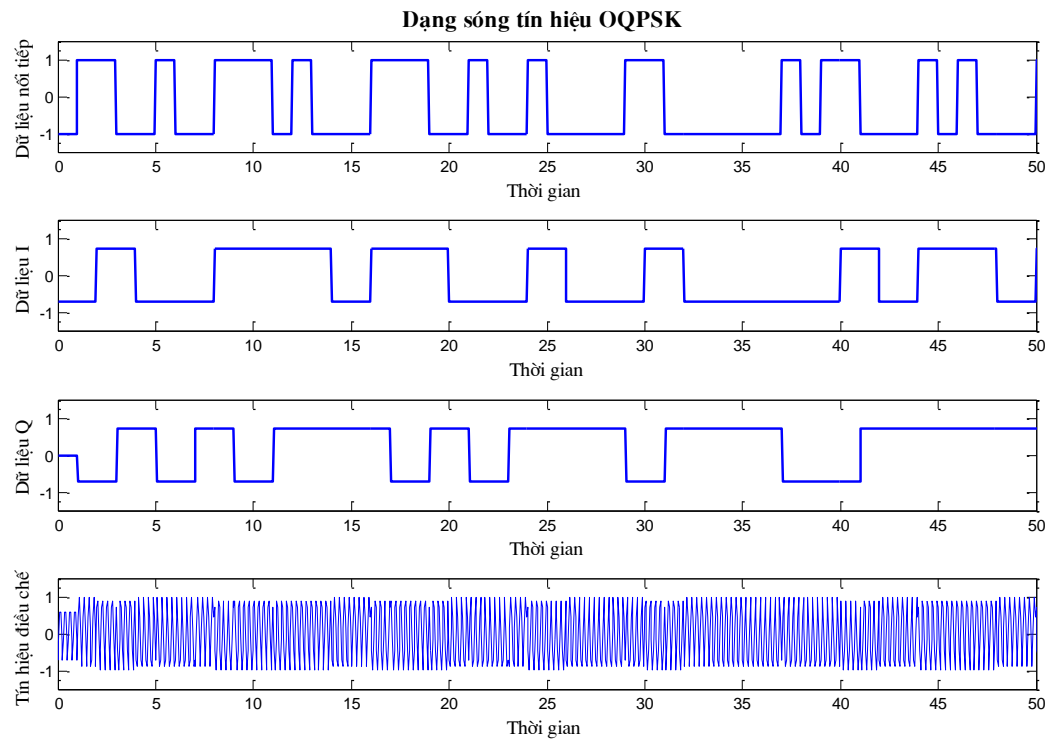
Biểu đồ mắt tín hiệu QPSK được lọc bởi bộ lọc Butterworth bậc 2 có độ rộng băng  $BW=0,7R_b$



Đồ thị bao tín hiệu QPSK được lọc bởi bộ lọc Butterworth bậc 4 có  $BW=0,7R_b$

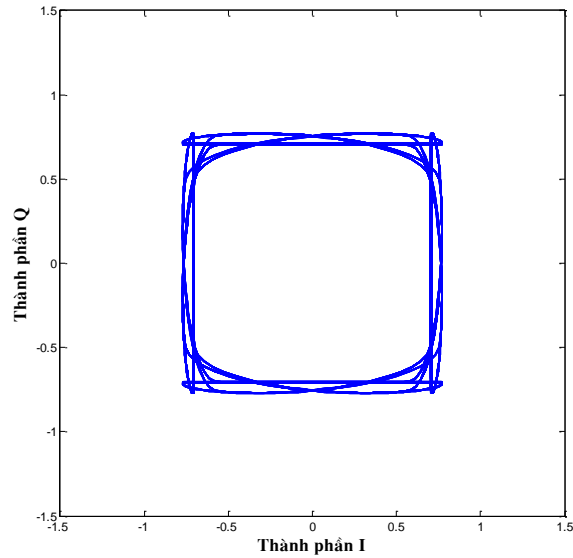


## Hệ thống OQPSK

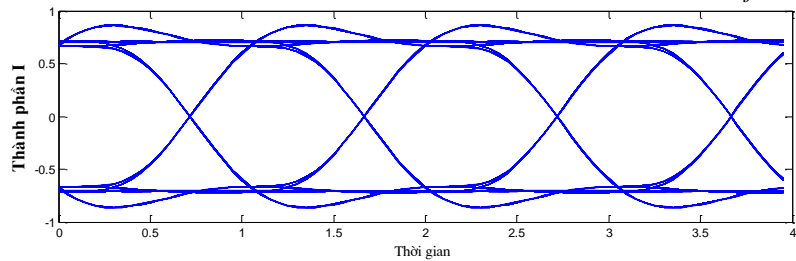




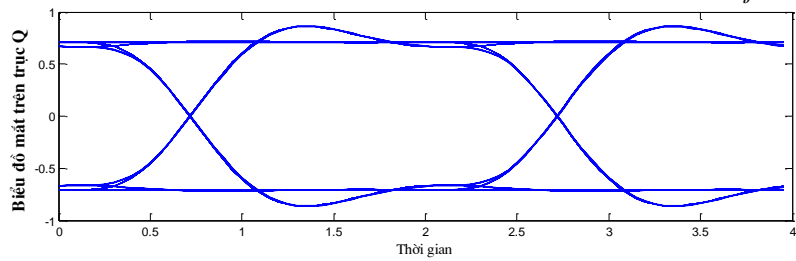
Biểu đồ pha tín hiệu QPSK được lọc bởi bộ lọc Butterworth bậc 2;  $BW = 0,4R_b$



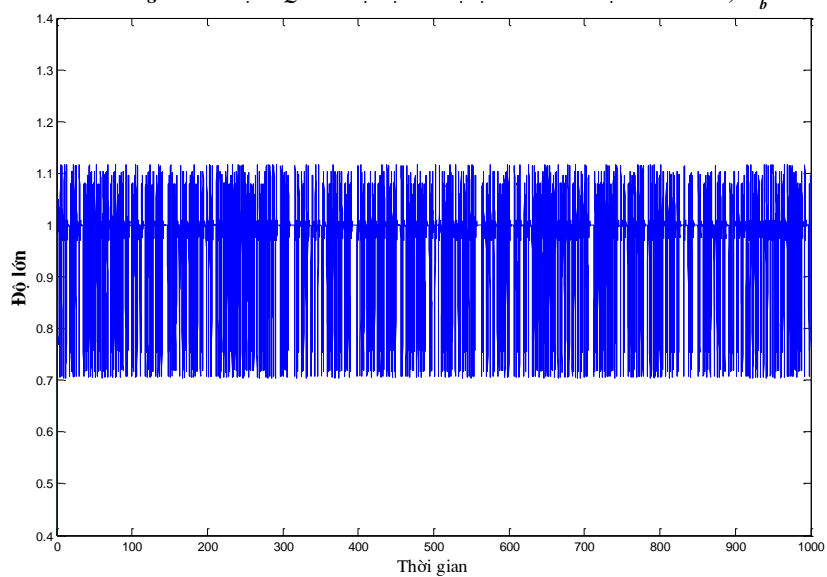
Biểu đồ mắt tín hiệu QPSK được lọc bởi bộ lọc butterworth bậc 4 có độ rộng băng  $BW = 0,7R_b$



Biểu đồ mắt tín hiệu QPSK được lọc bởi bộ lọc butterworth bậc 4 có độ rộng băng  $BW = 0,7R_b$

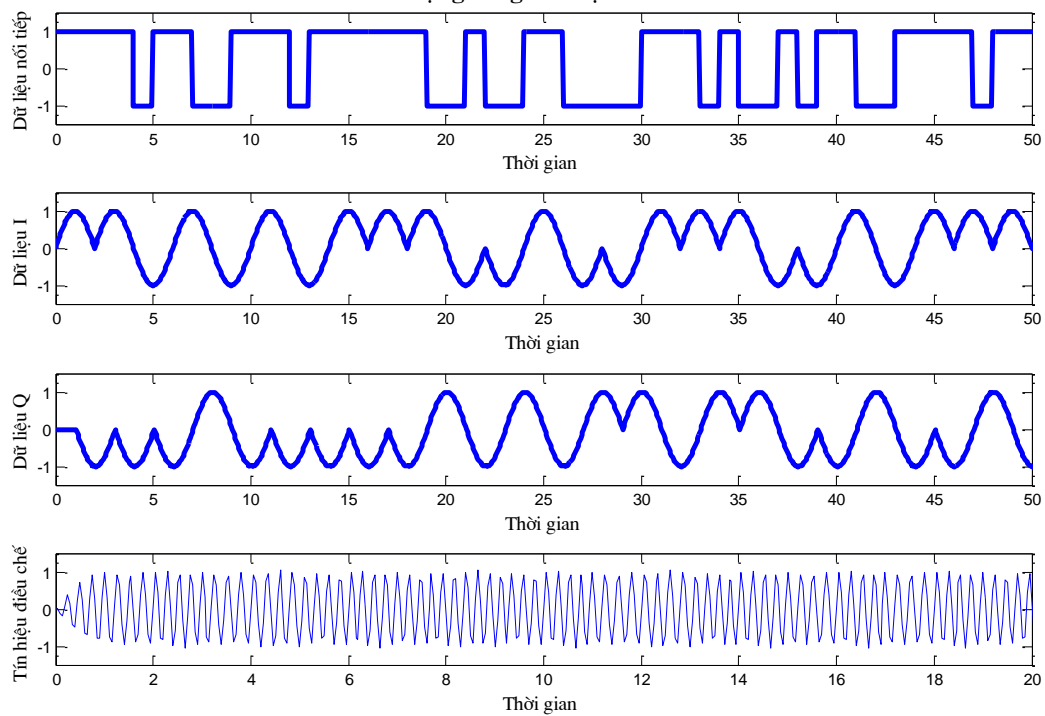


Đồng bộ tín hiệu QPSK được lọc bởi bộ lọc Butterworth bậc 4 có  $BW = 0,7R_b$

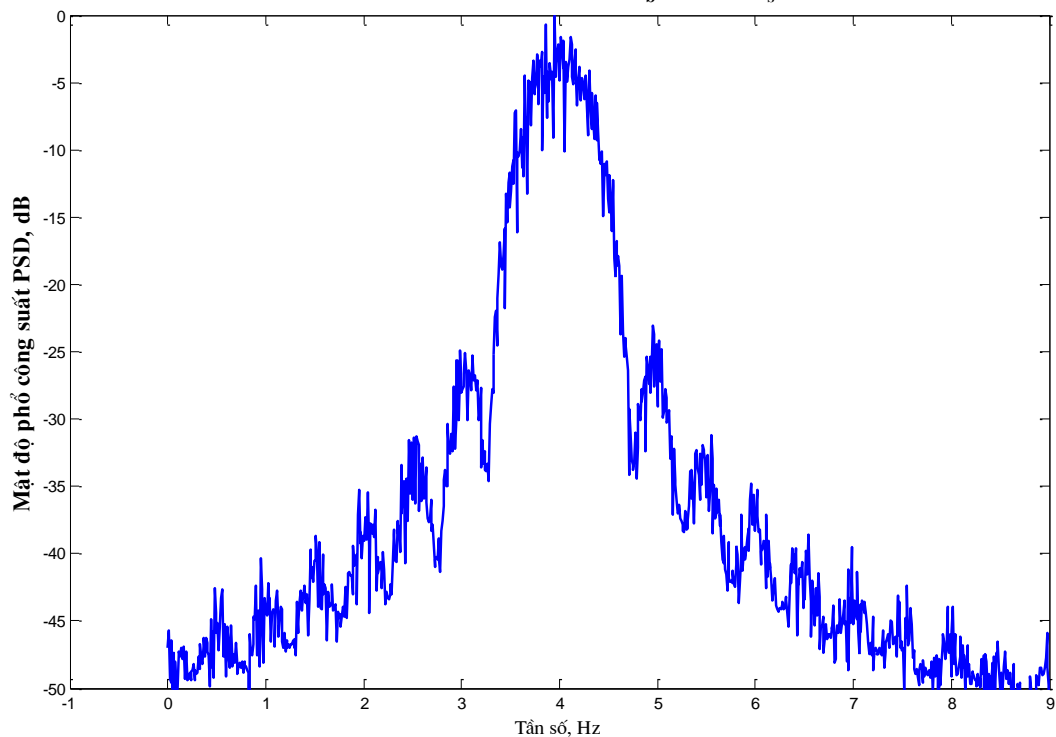


## Hệ thống MSK

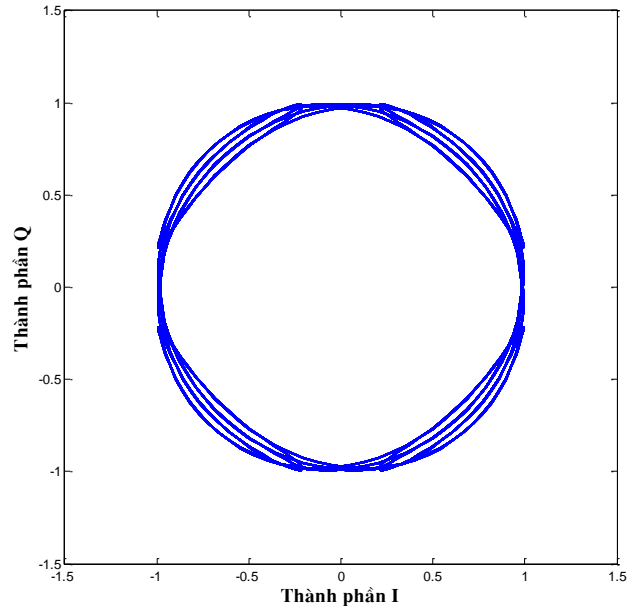
Dạng sóng tín hiệu MSK



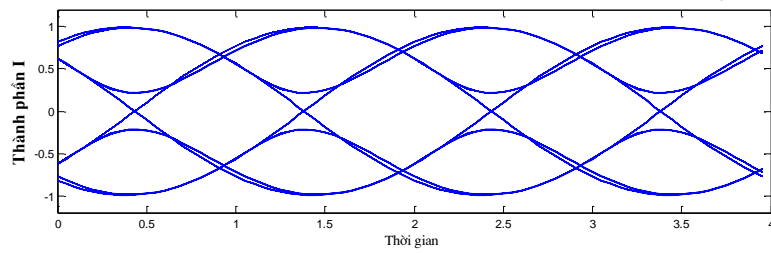
Phổ tín hiệu MSK; Tần số trung tâm = 4Hz;  $R_b = 1$  bps;  $R_s = 0,5$ ; No ave = 7



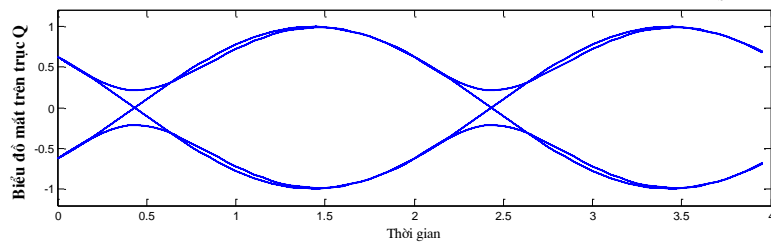
Biểu đồ pha tín hiệu MSK đ-ọc lọc bởi bộ lọc Butterworth bậc 2;  $BW=0,7R_b$



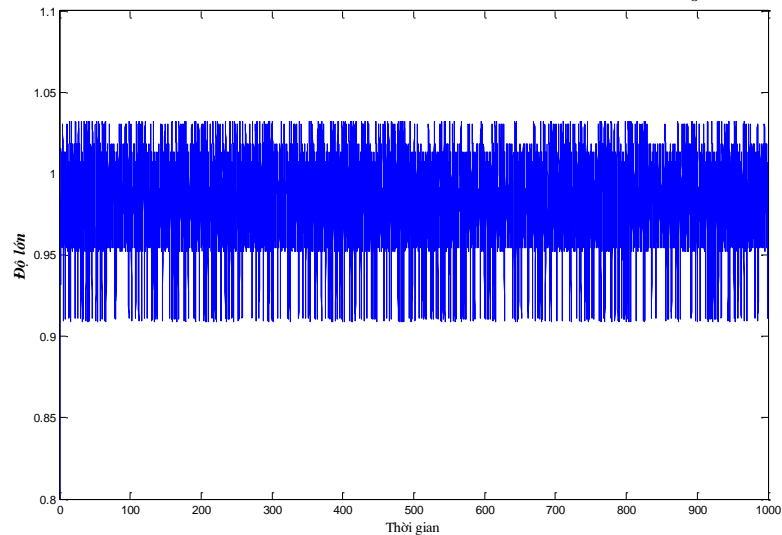
Biểu đồ mắt tín hiệu MSK đ-ọc lọc bởi bộ lọc Butterworth bậc 2 có  $BW=0,7R_b$



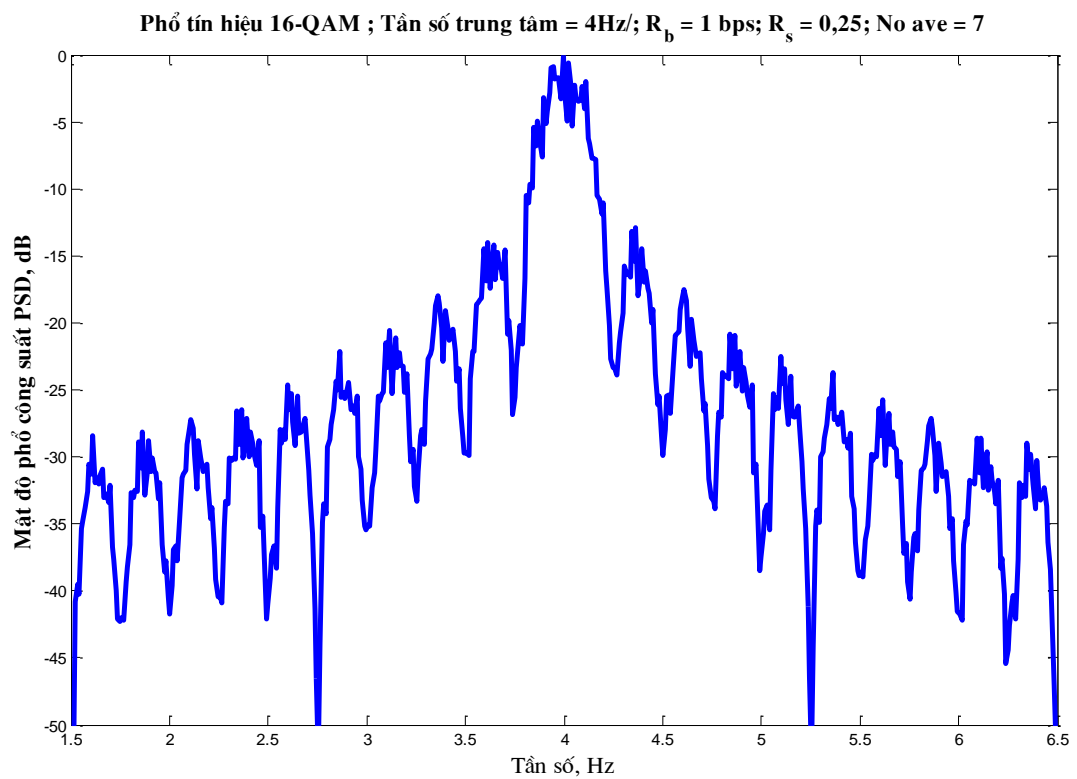
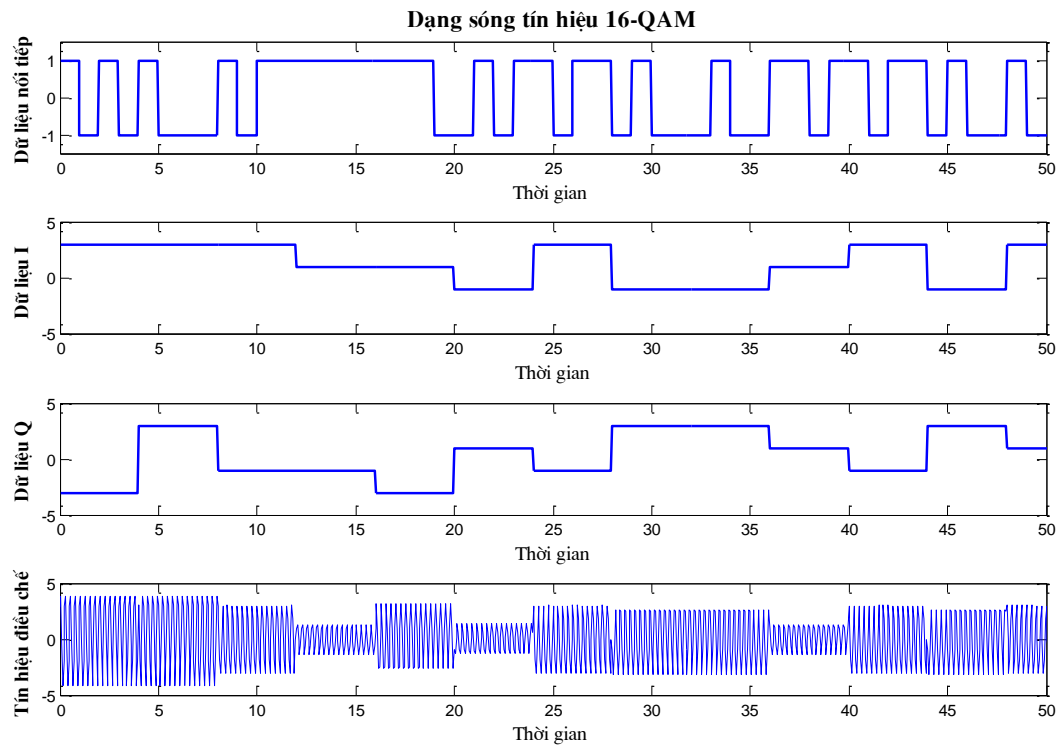
Biểu đồ mắt tín hiệu MSK đ-ọc lọc bởi bộ lọc Butterworth bậc 2 có  $BW=0,7R_b$



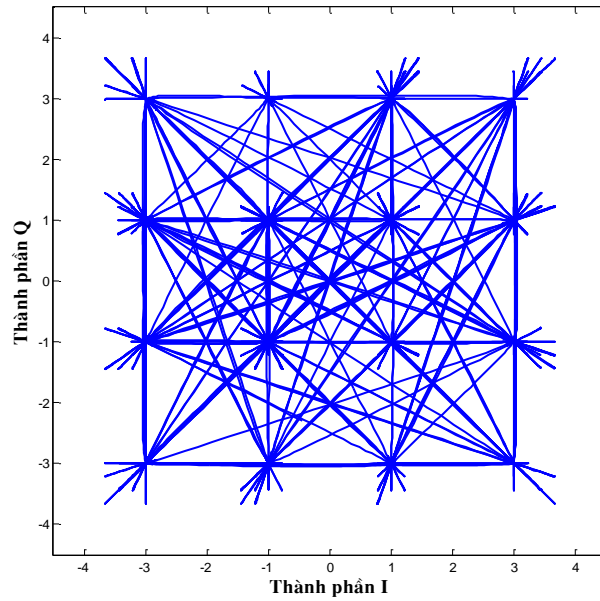
Đ-ờng bao tín hiệu MSK đ-ọc lọc bởi bộ lọc Butterworth bậc 2 có  $BW=0,7R_b$



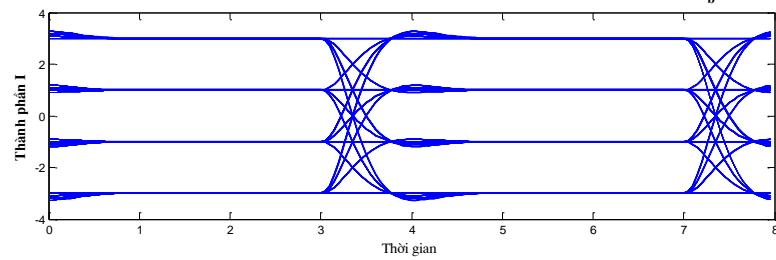
## Hệ thống 16-QAM



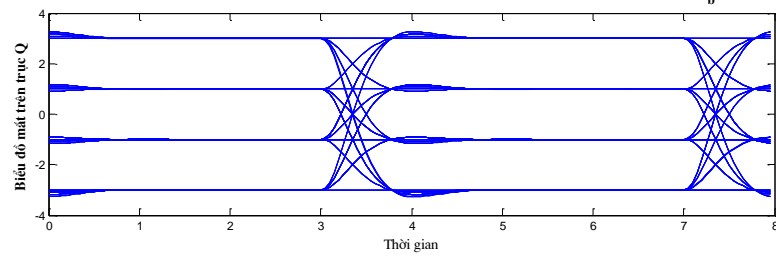
**Biểu đồ pha tín hiệu 16-QAM đ-ọc lọc bởi bộ lọc Butterworth bậc 4 ;  $BW=0,4R_b$**



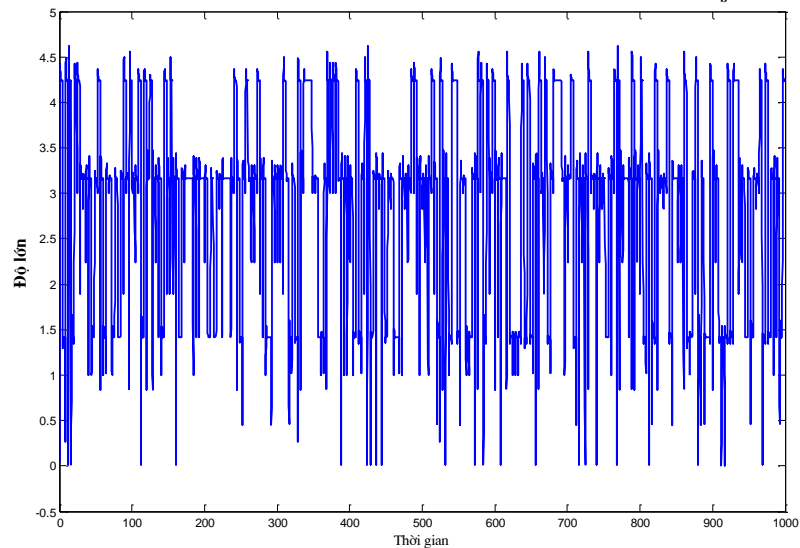
**Biểu đồ mắt tín hiệu 16-QAM đ-ọc lọc bởi bộ lọc Butterworth bậc 2 có  $BW=0,7R_b$**



**Biểu đồ mắt tín hiệu 16-QAM đ-ọc lọc bởi bộ lọc Butterworth bậc 2 có  $BW=0,7R_b$**



**Đ-ồ ờng bao tín hiệu 16-QAM đ-ọc lọc bởi bộ lọc Butterworth bậc 2 có  $BW=0,7R_b$**



```
% file: NVD8_AD_sim
clc;
clear all;
close all;

mod_type = input('      Nhap: 0 cho BPSK, 1 cho QPSK, 2 cho OQPSK, 3 cho MSK
& 4 cho 16 QAM = ');
display('      Luu y rang: Neu ve bieu do pha, bieu do mat & duong bao thi
Nhap 0 cho OFDM = ');
ofdm_mod = 0;      % input('  Nhap 1 cho OFDM = ');
ph_error = 0;      % input('  Nhap 1 cho loi pha giai dieu che (0) = ');
if ofdm_mod==1
    N_tones = input('  Nhap so tones cho OFDM (5) =      ')
end

samp_bit      = 20;%input('Nhap so mau/bit de MP (20) =      ');
f0            = 4;%input('Nhap tan so song mang Hz (4) =      ');
N_bits        = 1000;%input('Nhap tong so bit MP (1000) =      ');
I_plot        = input('      Khao sat:  1= Dang song; 2 = Pho; 3= Bieu do
mat; 4 = Duong bao; 5 = Bieu do pha =      ');

if I_plot ==2
    N_ave = 7;      % input('Nhap so pho de lay trung binh uoc tinh PSD (9)= ');
end
data          = 0.5*(sign(rand(1,N_bits))-0.5)+1);
T_bit         = 1;
del_t         = T_bit/samp_bit;
s              = 2*data - 1;
L              = length(s);
psi            = zeros(1,L);

if ph_error ==1
    type_ph_error = input('Nhap 1 cho loi pha tinh; 2 cho loi pha gaussian =
');
    if type_ph_error==1
        psi_rad = input('  Nhap loi pha radian =      ');
        psi = psi_rad*ones(1,L);
    elseif type_ph_error==2
        var_psi_rad2 = input('Nhap phuong sai loi pha =      ');
        psi = sqrt(var_psi_rad2)*randn(1,L);
    end
end

t = 0:del_t:L*T_bit - del_t;
%%=====
% Loop to repeat for phasor digrams
%%=====
clf
if mod_type~=4
    tp = 0: del_t:L*T_bit - del_t;
    tpp = 0:del_t:2*T_bit - del_t;
elseif mod_type==4
    tp = 0: del_t:(L*T_bit - del_t); % ? (L)
end
sigr          = [];
sigi          = [];
sig           = [];
s_t           = [];
xc            = [];
sigs          = [];
sig           = [];
y_sig         = [];
```

```
s_t      = s(ones(samp_bit,1),:); % Buld array whose columns are samp_bit long
s_t      = s_t(:)'; % Convert matrix where bit samples occupy columns to vector
if mod_type ~=4 & mod_type~=0
    T_sym  = 2*T_bit;
    %=====
    % Form array whose columns are the bits sampled samp_bit times
    sigs    = s(ones(2*samp_bit,1),:);
    sigr    = sigs(:,1:2:L); % Odd-indexed columns = Quadrature channel symbols
    sigi    = sigs(:,2:2:L); % Even-indexed columns = Inphase channel symbols
    if mod_type ==3
        AA   = sin(pi*ttp/(2*T_bit));
        CC   = (AA(ones(floor(L/2),1),:))';
        sigr  = CC.*sigr; %Weight symbols with half sine
        sigi  = CC.*sigi; %Weight symbols with half sine
    end
    sigr    = (sigr(:))'; % Concatenate columns into row matrix
    sigi    = (sigi(:))'; % Concatenate columns into row matrix
    if mod_type == 1
        sig  = sqrt(0.5)*(sigr - i*sigi); % Form complex baseband for QPSK
    elseif mod_type ==2
        sig  = sqrt(0.5)*(sigr - i*delay(sigi,samp_bit +1)); % OQPSK BB
    elseif mod_type ==3
        sig  = sigr - i* delay(sigi,samp_bit +1); % MSK complex baseband
    end
elseif mod_type ==4
    T_sym  = 4*T_bit;
    for l=1:4:L-3
        if s(l)==-1&s(l+1)==-1
            ampr=-3;
        elseif s(l)==-1&s(l+1)==1
            ampr=-1;
        elseif s(l)==1&s(l+1)==1
            ampr=1;
        elseif s(l)==1&s(l+1)==-1
            ampr=3;
        end
        if s(l+2)==-1& s(l+3)==-1
            ampi = -3;
        elseif s(l+2)==-1&s(l+3)==1
            ampi = -1;
        elseif s(l+2)==1&s(l+3)==1
            ampi = 1;
        elseif s(l+2)==1&s(l+3)==-1
            ampi = 3;
        end
        sigr = [sigr hold_in(ampr,4*samp_bit)];
        sigi = [sigi hold_in(ampi,4*samp_bit)];
        sig  = sigr - i*sigi;
    end
elseif mod_type ==0
    sigr    = s_t;
    sigi    = zeros(size(t));
    sig     = sigr - i*sigi;
    T_sym   = T_bit;
end
%=====
if ofdm_mod ==1
    MM      = [];
    sigp    =[];
    N_sig   =length(sig);
    N_cols  = fix(N_sig/N_tones);
    T_sym_ofdm = N_tones*T_sym;
    f_sym_ofdm = 1/T_sym_ofdm;
    f_step   = 0.5*f_sym_ofdm;
```

```
MM          = [0:N_tones-1]*f_step;
sigp        = sig(1:N_tones*N_cols);
sig         = sigp;
N_sig       = length(sig);
end
%=====
N_sig       = length(sig);
freq_array  = [];
sig_pr      = [];
if ofdm_mod ==1
    sig_pr = reshape(sig,N_tones,N_cols);
    for n=1:N_cols
        tn          =(n-1)*del_t;
        exp_fr      = exp(j*2*pi*MM*tn);
        freq_array  = [freq_array, exp_fr'];
    end
    tp          = 0:del_t:(N_cols-1)*del_t;
    sig         = sum(sig_pr.*freq_array);
end
%=====
xc          = sig.*exp(j*2*pi*f0*tp);
if ofdm_mod ==1
    f_s      = N_tones*f_sym_ofdm;
    f_ss     = f_sym_ofdm;
else
    f_s      = 1/T_sym;
    f_ss     = 1/T_sym;
end
%=====
if I_plot ==1
    subplot(4,1,1);
    plot(t,s_t);
    axis([0 50 -1.5 1.5]);
    xlabel('Thời gian ','fontname','.Vntime','fontsize',12);
    ylabel('Độ liêu nêi tiÕp','fontname','.Vntime','fontsize',12);
    if mod_type ==0 & ofdm_mod~=1
        title('Đing sãng tÝn hiÕu BPSK','fontname','.Vntime','fontsize',12);
        x_m = 1.5;
    elseif mod_type ==1 & ofdm_mod~=1
        title('Đing sãng tÝn hiÕu QPSK','fontname','.Vntime','fontsize',12);
        x_m = 1.5;
    elseif mod_type ==2 & ofdm_mod~=1
        title('Đing sãng tÝn hiÕu OQPSK ','fontname','.Vntime','fontsize',12);
        x_m = 1.5;
    elseif mod_type ==3 & ofdm_mod~=1
        title('Đing sãng tÝn hiÕu MSK','fontname','.Vntime','fontsize',12);
        x_m = 1.5;
    elseif mod_type ==4 & ofdm_mod~=1
        title('Đing sãng tÝn hiÕu 16-QAM','fontname','.Vntime','fontsize',12);
        x_m = 5;
    end
    if mod_type ==0 & ofdm_mod==1
        title(['Đing sãng tÝn hiÕu OFDM/BPSK; TÇn sè trung tøm ./tones =
',num2str(f0),...
        'Hz/',num2str(N_tones),'; R_b = 1 bps; R_s =
',num2str(f_ss)],...
        'fontname','.Vntime','fontsize',12)
        x_m=5;
    elseif mod_type ==1 & ofdm_mod==1
        title(['Đing sãng tÝn hiÕu OFDM/QPSK; TÇn sè trung tøm ./tones =
',num2str(f0),...
        'Hz/',num2str(N_tones),'; R_b = 1 bps; R_s =
',num2str(f_ss)],...
        'fontname','.Vntime','fontsize',12)
```



```

        x_m=5;
        elseif mod_type ==2 & ofdm_mod==1
            title(['Đ'ng s'ng t'yn hi'ou OFDM/OQPSK; T'cn s' trung t@m ./tones =
',num2str(f0),...
                'Hz/',num2str(N_tones),'; R_b = 1 bps; R_s =
',num2str(f_ss)],...
                'fontname','.Vntime','fontsize',12)
            x_m=5;
        elseif mod_type ==3 & ofdm_mod==1
            title(['Đ'ng s'ng t'yn hi'ou OFDM/MSK; T'cn s' trung t@m ./tones =
',num2str(f0),...
                'Hz/',num2str(N_tones),'; R_b = 1 bps; R_s =
',num2str(f_ss)],...
                'fontname','.Vntime','fontsize',12)
            x_m=5;
        elseif mod_type ==4 & ofdm_mod==1
            title(['Đ'ng s'ng t'yn hi'ou OFDM/16-QAM; T'cn s' trung t@m ./tones =
',num2str(f0),...
                'Hz/',num2str(N_tones),'; R_b = 1 bps; R_s =
',num2str(f_ss)],...
                'fontname','.Vntime','fontsize',12)
            x_m=7;
        end
        %% Dang song=====
        subplot(4,1,2);
        plot(tp,real(sig));
        axis([0 50 -x_m x_m]);
        xlabel('Th'ei gian','fontname','.Vntime','fontsize',12);
        ylabel('D÷ li'Nu I ','fontname','.Vntime','fontsize',12);
        subplot(4,1,3);
        plot(tp,imag(sig));
        axis([0 50 -x_m x_m]);
        xlabel('Th'ei gian','fontname','.Vntime','fontsize',12);
        ylabel('D÷ li'ou Q','fontname','.Vntime','fontsize',12);
        subplot(4,1,4);
        plot(tp,real(xc));
        axis([0 50 -x_m x_m]);
        xlabel('Th'ei gian','fontname','.Vntime','fontsize',12);
        ylabel('T'yn hi'ou @i'ou ch' ','fontname','.Vntime','fontsize',12);
    elseif I_plot==2
        %% Mat do pho cong suat PSD=====
        L_xc = length(xc);
        LL = fix(L_xc/N_ave);
        LLlog = fix(log2(LL));
        LL = 2^LLlog;
        fs=1/del_t;
        f_sym = 1/T_sym;
        del_f = fs/LL;
        % next paper
        f = 0:del_f:fs - del_f;
        xcp = [];
        xcp = xc(1:LL*N_ave);
        sig = [];
        xc = xcp;
        L_xc = length(xc);
        xc_pr = reshape(xc,LL,N_ave);
        Xc = (fft(xc_pr))';
        size(Xc) % note Xc & xc
        X_est = sum(abs(Xc));
        psd = 20*log10(X_est/max(X_est));
        plot(f,psd),...
            axis([f0-10*f_s f0+10*f_s -50 0]),...
            xlabel('T'cn s', Hz','fontname','.Vntime','fontsize',12),...

```

```
ylabel('Mật độ phổ công suất PSD, dB','fontname','.Vntime','fontsize',12),...
if mod_type ==1& ofdm_mod~=1
    title(['Phân tích hệ thống QPSK; Tần số trung tâm = ',num2str(f0),...
        'Hz; R_b = 1 bps; R_s = ',num2str(f_ss),' ; No ave = ',num2str(N_ave)],...
        'fontname','.Vntime','fontsize',12);
elseif mod_type ==2& ofdm_mod~=1
    title(['Phân tích hệ thống OQPSK; Tần số trung tâm = ',num2str(f0),...
        'Hz; R_b = 1 bps; R_s = ',num2str(f_ss),' ; No ave = ',num2str(N_ave)],...
        'fontname','.Vntime','fontsize',12);

elseif mod_type ==3& ofdm_mod~=1
    title(['Phân tích hệ thống MSK; Tần số trung tâm = ',num2str(f0),...
        'Hz; R_b = 1 bps; R_s = ',num2str(f_ss),' ; No ave = ',num2str(N_ave)],...
        'fontname','.Vntime','fontsize',12);

elseif mod_type ==4& ofdm_mod~=1
    title(['Phân tích hệ thống 16-QAM ; Tần số trung tâm = ',num2str(f0),...
        'Hz; R_b = 1 bps; R_s = ',num2str(f_ss),' ; No ave = ',num2str(N_ave)],...
        'fontname','.Vntime','fontsize',12);
    %%%%%%%%%

elseif mod_type ==1& ofdm_mod==1
    title(['Phân tích hệ thống OFDM/QPSK ; Tần số trung tâm = ',num2str(f0),...
        'Hz; ',num2str(N_tones),' R_b =1 bps; R_s = ',num2str(f_ss),...
        'No ave = ',num2str(N_ave)], 'fontname','.Vntime','fontsize',12);
elseif mod_type ==2& ofdm_mod==1
    title(['Phân tích hệ thống OFDM/OQPSK; Tần số trung tâm = ',num2str(f0),...
        'Hz; ',num2str(N_tones),' R_b =1 bps; R_s = ',num2str(f_ss),...
        'No ave = ',num2str(N_ave)], 'fontname','.Vntime','fontsize',12);

elseif mod_type ==3& ofdm_mod==1
    title(['Phân tích hệ thống OFDM/MSK; Tần số trung tâm = ',num2str(f0),...
        'Hz; ',num2str(N_tones),' R_b =1 bps; R_s = ',num2str(f_ss),...
        'No ave = ',num2str(N_ave)], 'fontname','.Vntime','fontsize',12);

elseif mod_type ==4& ofdm_mod==1
    title(['Phân tích hệ thống OFDM/16-QAM; Tần số trung tâm = ',num2str(f0),...
        'Hz; ',num2str(N_tones),' R_b =1 bps; R_s = ',num2str(f_ss),...
        'No ave = ',num2str(N_ave)], 'fontname','.Vntime','fontsize',12]);
end

elseif I_plot ==3|I_plot==4|I_plot==5
    I_filt = input('Nhập 1 để du lịch được lọc bởi Butterworth; Nhập 2 nếu không lọc: ');
    if I_filt ==1
        BW = input('Nhập độ rộng băng thông lọc: B= [0 đến 0.8]R_b: ');
        n_order = input('Nhập bậc lọc (1,2,3,4) = ');
        [num,den] = butter(n_order,2*BW/samp_bit);
        y_sig = [];
        zi = [];
        [y_sig,zf] = filter(num,den,sig,zi);
        zi=zf;
    end
    N_sig = length(sig);
    if I_plot==3
```

```
% Bieu do mat=====
%z_sig =[];
z_sig = zeros(size(y_sig));
if I_filt ==1
    z_sig(1:N_sig - 5*samp_bit) = y_sig(5*samp_bit:N_sig-1);
elseif I_filt ==2
    z_sig(1:N_sig - 5*samp_bit) = sig(5*samp_bit:N_sig-1);
end

N_sym = 2*T_sym/del_t;
tt= 0:del_t:2*T_sym - del_t;
for n=2:fix((N_sig-5*samp_bit)/N_sym)
    real_eye =[];
    imag_eye =[];
    real_eye(1:N_sym) = real(z_sig((n-1)*N_sym - 0.5*N_sym + 1:
n*N_sym - 0.5*N_sym));
    imag_eye(1:N_sym) = imag(z_sig((n-1)*N_sym - 0.5*N_sym + 1:
n*N_sym - 0.5*N_sym));
    %Thanh phan I =====
    subplot(2,1,1),
    plot(tt,real_eye),
    xlabel('Thêi gian','fontname','.Vntime','fontsize',12),
    ylabel('Biêu đồ m³t tr¹n trôc I
','fontname','.Vntime','fontsize',12);
    if mod_type ==0
        title(['Biêu đồ m³t tÝn hiÖu BPSK ®-íc l¹c bëi bé l¹c
Butterworth bËc ',num2str(n_order),' ®é réng b`ng BW =',...
num2str(BW),'R_b'],'fontname','.Vntime','fontsize',12);
    elseif mod_type ==1&I_filt==1
        title(['Biêu đồ m³t tÝn hiÖu QPSK ®-íc l¹c bëi bé l¹c
Butterworth bËc',num2str(n_order),' c¹ ®é réng b`ng BW=','...
num2str(BW),'R_b'],'fontname','.Vntime','fontsize',12);
    elseif mod_type ==2&I_filt==1
        title(['Biêu đồ m³t tÝn hiÖu OQPSK ®-íc l¹c bëi bé l¹c
butterwort bËc ',num2str(n_order),'c¹ ®é réng b`ng BW=','...
num2str(BW),'R_b'],'fontname','.Vntime','fontsize',12);
    elseif mod_type ==3&I_filt==1
        title(['Biêu đồ m³t tÝn hiÖu MSK ®-íc l¹c bëi bé l¹c Butterwort
bËc ',num2str(n_order),' c¹ BW=','...
num2str(BW),'R_b'],'fontname','.Vntime','fontsize',12);
    elseif mod_type ==4&I_filt==1
        title(['Biêu đồ m³t tÝn hiÖu 16-QAM ®-íc l¹c bëi bé l¹c
Butterworth bËc ',num2str(n_order),'c¹ BW=','...
num2str(BW),'R_b'],'fontname','.Vntime','fontsize',12);
    elseif mod_type ==0&I_filt==2
        title('Biêu đồ m³t tÝn hiÖu BPSK kh«ng ®-íc
l¹c','fontname','.Vntime','fontsize',12);
    elseif mod_type ==1&I_filt==2
        title('Biêu đồ m³t tÝn hiÖu QPSK kh«ng ®-íc l¹c
','fontname','.Vntime','fontsize',12);
    elseif mod_type ==2&I_filt==2
        title('Biêu đồ m³t tÝn hiÖu OQPSK kh«ng ®-íc l¹c
','fontname','.Vntime','fontsize',12);
    elseif mod_type ==3&I_filt==2
        title('Biêu đồ m³t tÝn hiÖu MSK kh«ng ®-íc l¹c
','fontname','.Vntime','fontsize',12);
    elseif mod_type ==4&I_filt==2
        title('Biêu đồ m³t tÝn hiÖu 16-QAM kh«ng ®-íc
l¹c','fontname','.Vntime','fontsize',12);
    end
    %Thanh phan Q =====
    subplot(2,1,2),
    plot(tt,imag_eye),
    xlabel('Thêi gian','fontname','.Vntime','fontsize',12),
```

```

        ylabel('Biểu đồ mật tần trục Q
','fontname','.Vntime','fontsize',12);
        if mod_type ==0 %%%%%%%%%%%%%%
            title(['Biểu đồ mật tần hiều BPSK @-íc lăc bẻi bẻ lăc
Butterworth bẻc ',num2str(n_order),' @ẻ rẻng bẻng BW=',...
                num2str(BW),'R_b'],'fontname','.Vntime','fontsize',12);
        elseif mod_type ==1&I_filt==1
            title(['Biểu đồ mật tần hiều QPSK @-íc lăc bẻi bẻ lăc
Butterworth bẻc ',num2str(n_order),' cả @ẻ rẻng bẻng BW=',...
                num2str(BW),'R_b'],'fontname','.Vntime','fontsize',12);
        elseif mod_type ==2&I_filt==1
            title(['Biểu đồ mật tần hiều QPSK @-íc lăc bẻi bẻ lăc
butterwort bẻc ',num2str(n_order),'cả @ẻ rẻng bẻng BW=',...
                num2str(BW),'R_b'],'fontname','.Vntime','fontsize',12);
        elseif mod_type ==3&I_filt==1
            title(['Biểu đồ mật tần hiều MSK @-íc lăc bẻi bẻ lăc Butterwort
bẻc ',num2str(n_order),' cả BW=',...
                num2str(BW),'R_b'],'fontname','.Vntime','fontsize',12);
        elseif mod_type ==4&I_filt==1
            title(['Biểu đồ mật tần hiều 16-QAM @-íc lăc bẻi bẻ lăc
Butterwort bẻc ',num2str(n_order),'cả BW=',...
                num2str(BW),'R_b'],'fontname','.Vntime','fontsize',12);
        elseif mod_type ==0&I_filt==2
            title('Biểu đồ mật tần hiều BPSK khẻng @-íc
lăc','fontname','.Vntime','fontsize',12);
        elseif mod_type ==1&I_filt==2
            title('Biểu đồ mật tần hiều QPSK khẻng @-íc lăc
','fontname','.Vntime','fontsize',12);
        elseif mod_type ==2&I_filt==2
            title('Biểu đồ mật tần hiều QPSK khẻng @-íc lăc
','fontname','.Vntime','fontsize',12);
        elseif mod_type ==3&I_filt==2
            title('Biểu đồ mật tần hiều MSK khẻng @-íc lăc
','fontname','.Vntime','fontsize',12);
        elseif mod_type ==4&I_filt==2
            title('Biểu đồ mật tần hiều 16-QAM khẻng @-íc
lăc','fontname','.Vntime','fontsize',12);
        end
        %===== Note
        if n==2
            hold on
        end
        subplot(2,1,1),
        plot(tt,imag_eye);
        xlabel('Thẻi gian ','fontname','.Vntime','fontsize',12);
        ylabel('Thẻnh phẻn Q_n','fontname','.Vntime','fontsize',12),...
        if n==2
            hold on
        end
    end
elseif I_plot==4
    % Duẻng bao =====
    subplot(2,1,1);
    plot(t,abs(y_sig));
    if mod_type ==0
        title(['Sẻng bao tần hiều BPSK @-íc lăc bẻi bẻ lăc Butterwort bẻc
',num2str(n_order),'cả BW= ',...
            num2str(BW),'R_b'],'fontname','.Vntime','fontsize',12);
    elseif mod_type ==1
        title(['Sẻng bao tần hiều QPSK @-íc lăc bẻi bẻ lăc Butterwort bẻc
',num2str(n_order),' cả BW= ',...
            num2str(BW),'R_b'],'fontname','.Vntime','fontsize',12);
    elseif mod_type ==2

```

```

        title(['$-êng bao tŷn hiŷu OQPSK ®-íc lăc bŷi bŷ lăc Butterwort bĚc
',num2str(n_order),' că BW= ',...
            num2str(BW),'R_b'],'fontname','.Vntime','fontsize',12);
    elseif mod_type ==3
        title(['$-êng bao tŷn hiŷu MSK ®-íc lăc bŷi bŷ lăc Butterwort bĚc
',num2str(n_order),' că BW= ',...
            num2str(BW),'R_b'],'fontname','.Vntime','fontsize',12);
    elseif mod_type ==4
        title(['$-êng bao tŷn hiŷu 16-QAM ®-íc lăc bŷi bŷ lăc Butterwort
bĚc ',num2str(n_order),'că BW= ',...
            num2str(BW),'R_b'],'fontname','.Vntime','fontsize',12);
    end
elseif I_plot==5
    % Bieu do pha =====
    z_sig=[];
    if I_filt==1
        z_sig(1:N_sig-5*samp_bit) = y_sig(5*samp_bit:N_sig-1);
    elseif I_filt==2
        z_sig(1:N_sig-5*samp_bit) = sig(5*samp_bit:N_sig-1);
    end
    plot(real(z_sig),imag(z_sig));
    axis square;
    xlabel('Thụnh phÇn I','fontname','.Vntime','fontsize',12);
    ylabel('Thụnh phÇn Q','fontname','.Vntime','fontsize',12);
    if mod_type~4
        axis([-1.5 1.5 -1.5 1.5]),...
    else
        axis([-4.5 4.5 -4.5 4.5]),...
    end
    if mod_type ==0 & I_filt==1
        title(['Biŷu ă pha tŷn hiŷu BPSK ®-íc lăc bŷi bŷ lăc Butterwort
bĚc ',num2str(n_order),' ; BW= ',num2str(BW),'R_b'],...
            'fontname','.Vntime','fontsize',12);
    elseif mod_type ==1 & I_filt==1
        title(['Biŷu ă pha tŷn hiŷu QPSK ; ®-íc lăc bŷi bŷ lăc Butterwort
bĚc ',num2str(n_order),' ; BW= ',num2str(BW),'R_b'],...
            'fontname','.Vntime','fontsize',12);
    elseif mod_type ==2 & I_filt==1
        title(['Biŷu ă pha tŷn hiŷu OQPSK ®-íc lăc bŷi bŷ lăc Butterwort
bĚc ',num2str(n_order),' ; BW= ',num2str(BW),'R_b'],...
            'fontname','.Vntime','fontsize',12);
    elseif mod_type ==3 & I_filt==1
        title(['Biŷu ă pha tŷn hiŷu MSK ®-íc lăc bŷi bŷ lăc Butterwort
bĚc; ',num2str(n_order),' ; BW= ',num2str(BW),'R_b'],...
            'fontname','.Vntime','fontsize',12);
    elseif mod_type ==4 & I_filt==1
        title(['Biŷu ă pha tŷn hiŷu 16-QAM ®-íc lăc bŷi bŷ lăc Butterwort
bĚc ',num2str(n_order),' ; BW= ',num2str(BW),'R_b'],...
            'fontname','.Vntime','fontsize',12);
    elseif mod_type ==0 & I_filt==2
        title('Biŷu ă pha tŷn hiŷu BPSK, khăng ®-íc lăc
','fontname','.Vntime','fontsize',12);
    elseif mod_type ==1 & I_filt==2
        title('Biŷu ă pha tŷn hiŷu QPSK, khăng ®-íc lăc
','fontname','.Vntime','fontsize',12);
    elseif mod_type ==2 & I_filt==2
        title('Biŷu ă pha tŷn hiŷu OQPSK, khăng ®-íc
lăc','fontname','.Vntime','fontsize',12);
    elseif mod_type ==3 & I_filt==2
        title('Biŷu ă pha tŷn hiŷu MSK, khăng ®-íc lăc MSK
','fontname','.Vntime','fontsize',12);
    elseif mod_type ==4 & I_filt==2
        title('Biŷu ă pha tŷn hiŷu 16_QAM, khăng ®-íc lăc
','fontname','.Vntime','fontsize',12);

```

```

end
end
end

function output = hold_in(in,k)

output = [];
L = length(in);
b = ones(k,1)*in;
output = reshape(b,1,L*k);

function y=q(x)

y = 0.5*erfc(x/sqrt(2));

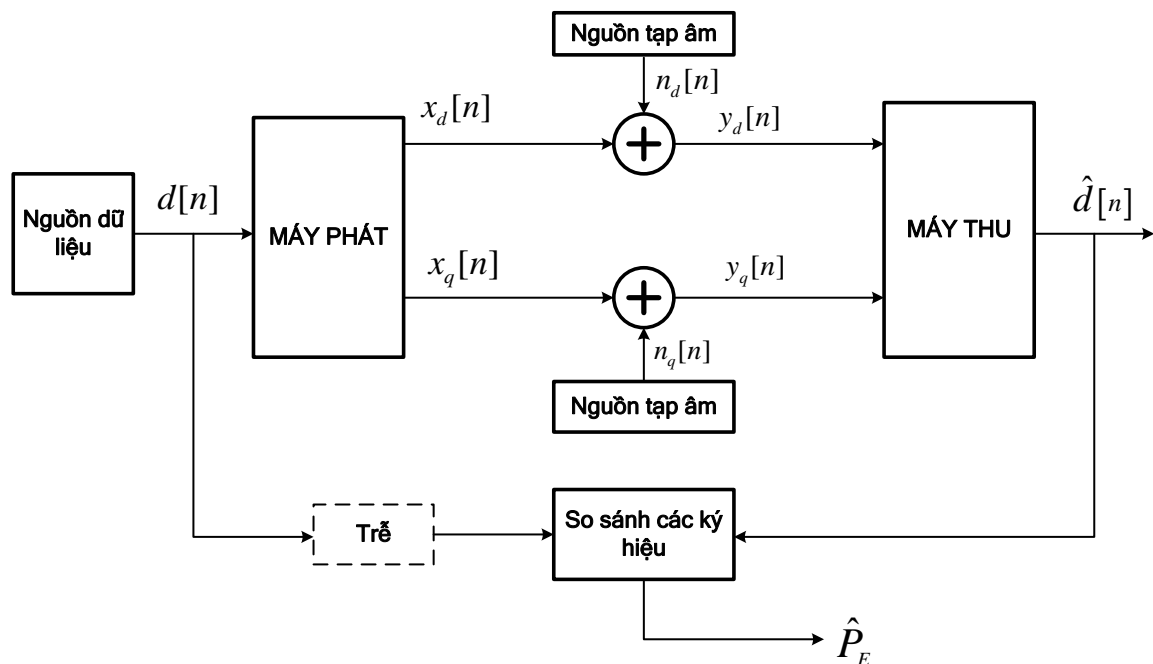
```

#### Bài 40:

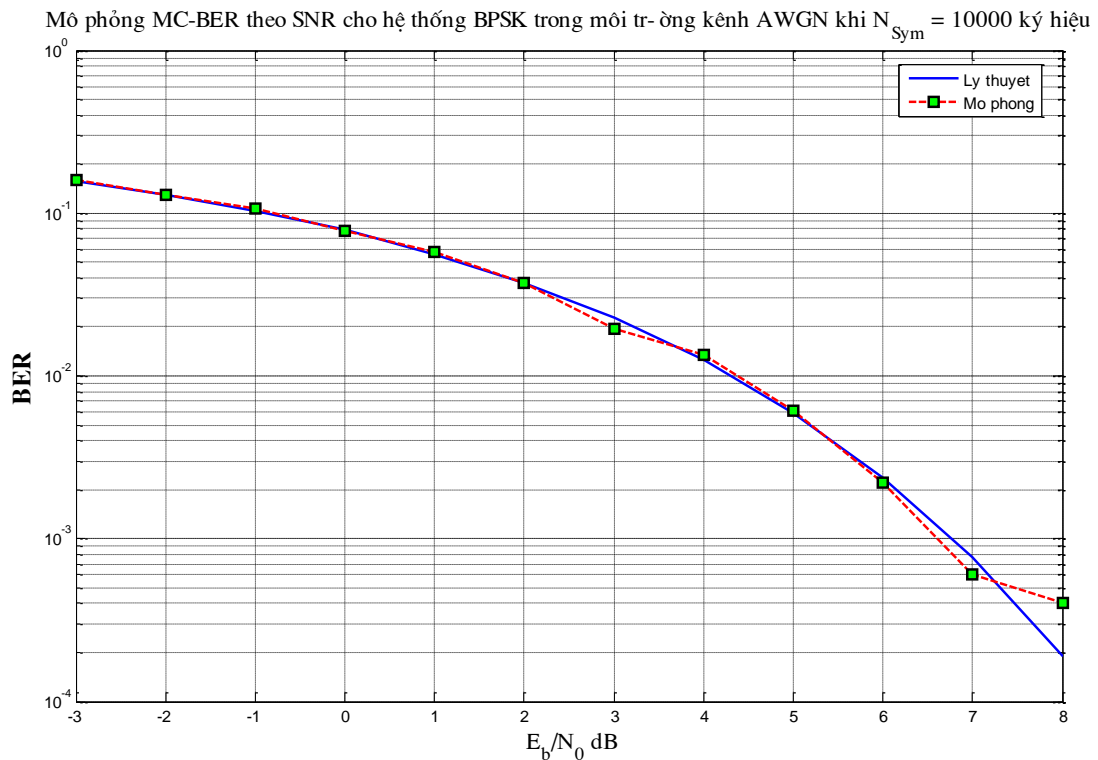
Hãy đọc mã chương trình Matlab được cho bởi **NVD9\_MCBPSK.m** dưới đây, vẽ mô hình mô phỏng, phân tích mô hình mô phỏng, chạy chương trình mô phỏng, phân tích kết quả mô phỏng.

#### Hướng dẫn giải

Mô hình mô phỏng trong các điều kiện giả định sau: Không thực hiện định dạng xung ở máy phát; Kênh là kênh AWGN; Các ký hiệu dữ liệu tại đầu ra của nguồn là độc lập nhau và đồng xác suất; Hệ thống không lọc và không có giao thoa giữa các ký hiệu ISI. Các giả định này làm đơn giản hệ thống và chương trình mô phỏng.



Chạy chương trình Matlab,  $N=1000$  ký hiệu cho mỗi giá trị của SNR, kết quả được minh họa ở hình dưới đây. Lưu ý rằng, độ tin cậy của mô phỏng BER giảm khi SNR tăng lên do thực tế đếm được ít lỗi hơn. Điều này gợi ý rằng có thể có quan hệ giữa số ký hiệu được mô phỏng với SNR hoặc tiếp tục thực hiện mô phỏng đến khi số lỗi đếm được là giống nhau tại mỗi giá trị của SNR.



```
% File: NVD9_MCBPSK.m
% File: NVD9_MCBPSK.m
clc;
clear all;
close all;
snrdB_min = -3; snrdB_max = 8;          % SNR (in dB) limits
snrdB      = snrdB_min:1:snrdB_max;
Nsymbols   = input('Nhap so ky hieu > ');
snr         = 10.^(snrdB/10);           % convert from dB
h           = waitbarqa(0, 'Mô phỏng BER cho tổng số, trên của SNR');
len_snr     = length(snrdB);

for j=1:len_snr                          % increment SNR
    waitbarqa(j/len_snr)
    sigma    = sqrt(1/(2*snr(j)));        % noise standard deviation
    error_count = 0;
    for k=1:Nsymbols                     % simulation loop begins
        d     = round(rand(1));          % data
        x_d   = 2*d - 1;                 % transmitter output
        n_d   = sigma*randn(1);          % noise
        y_d   = x_d + n_d;               % receiver input
        if y_d > 0                         % test condition
            d_est = 1;                   % conditional data estimate
        else
            d_est = 0;                   % conditional data estimate
        end
        if (d_est ~= d)
            error_count = error_count + 1; % error counter
        end
    end
    errors(j) = error_count;              % store error count for plot
end
close(h)
ber_sim      = errors/Nsymbols;           % BER estimate
ber_theor    = erfc(sqrt(1*snr))/2;      % theoretical BER
%=====
```

```
h9_8 = figure(1);
set(h9_8,'color','c','name','H9.8:NVD');
semilogy(snrdB,ber_theor,snrdB,ber_sim,...
    '--rs','LineWidth',2,...
    'MarkerEdgeColor','k',...
    'MarkerFaceColor','g',...
    'MarkerSize',7);
axis([snrdB_min snrdB_max 0.0001 1])
title(['M« pháng MC-BER theo SNR cho h« th«ng BPSK trong m«i tr-«ng k«nh AWGN'
    khi N_S_y_m = ',...
    num2str(Nsymbols),' ký hi«u'],'FontName','.VnTime','FontSize',14);
xlabel('E_b/N_0 dB','FontName','.VnTime','FontSize',14);
ylabel('BER','FontName','.VnTime','FontSize',16);
legend('Ly thuyet','Mo phong',1);
grid on;
```

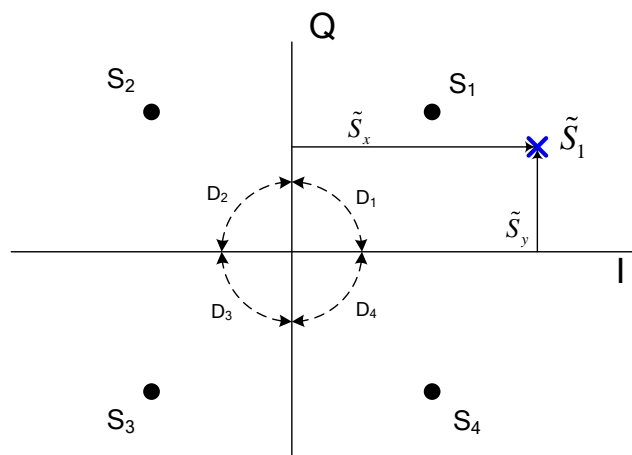
#### Bài 41:

Hãy đọc mã chương trình Matlab **NVD10\_QPSKSA.m** dưới đây, vẽ mô hình mô phỏng, phân tích mô hình mô phỏng, chạy chương trình mô phỏng, phân tích kết quả mô phỏng.

#### Hướng dẫn giải

Vì chòm sao tín hiệu QPSK có 4 điểm tín hiệu chứ không phải là hai, và vì không gian tín hiệu có hai chiều chứ không phải là một như trường hợp BPSK, trong đó phải thêm một chiều nữa để tạo kênh vuông pha.

Chòm sao tín hiệu được minh họa ở hình dưới đây. Các điểm tín hiệu phát  $S_i$ ,  $i=1,2,3,4$  tương ứng các vùng quyết định  $D_i$ ,  $i=1,2,3,4$ . Quyết định đúng ở máy thu nếu  $S_i$  được phát đi và tín hiệu thu rơi vào vùng quyết định  $D_i$ ; nếu ngược lại thì xảy ra lỗi. Cụ thể trong hình dưới đây, ta giả sử  $S_1$  được phát đi và tín hiệu thu không có tạp âm được ký hiệu là  $\tilde{S}_1$ . Hậu quả của ISI và méo làm cho  $\tilde{S}_1 \neq S_1$ . Nó là  $\tilde{S}_1$  chứ không phải là  $S_1$ . Mục đích mô phỏng này sẽ làm sáng tỏ các ảnh hưởng của ISI chứ không làm sáng tỏ các ảnh hưởng của tạp âm. Các thành phần đồng pha và vuông pha của  $\tilde{S}_1$  được ký hiệu là  $\tilde{S}_x$  và  $\tilde{S}_y$  trong đó  $\tilde{S}_x = \text{Re}(\tilde{S}_1)$  và  $\tilde{S}_y = \text{Im}(\tilde{S}_1)$ . Một khi xét đến tạp âm bằng cách cộng  $n_x$  và  $n_y$  với  $\tilde{S}_x$  và  $\tilde{S}_y$ , thì một quyết định đúng được thực hiện với điều kiện  $S_1$  đã được phát đi nếu nếu  $(\tilde{S}_x + n_x, \tilde{S}_y + n_y) \in D_1$ . Thực hiện quyết định sai nếu  $(\tilde{S}_x + n_x, \tilde{S}_y + n_y) \notin D_1$ .





Vấn đề ở chỗ, xác định các thành phần tạp âm  $n_x$  và  $n_y$  mà sẽ dẫn đến một lỗi cho điểm tín hiệu thu (không có tạp âm) trong không gian tín hiệu  $\tilde{S}_1$ . Vấn đề này rất giống với trường hợp BPSK nhưng có sự khác biệt căn bản là không gian tín hiệu. Nếu các thành phần tạp âm cộng đồng pha và vuông pha là không tương quan nhau và đều là phân bố Gauss. Theo đó, khi  $S_1$  được phát đi và thu được  $\tilde{S}_1$ , thì xảy ra một lỗi nếu

$$\Pr\{\text{lỗi}|S_1\} = \iint_{(\tilde{S}_x+n_x, \tilde{S}_y+n_y) \notin D_1} \frac{1}{2\pi\sigma_n\sigma_n} e^{-\left(\frac{(n_x-\tilde{S}_x)^2}{2\sigma_n^2} - \frac{(n_y-\tilde{S}_y)^2}{2\sigma_n^2}\right)} dn_x dn_y \quad (1)$$

trong đó  $n_x$  và  $n_y$  là các thành phần tạp âm đồng pha và vuông pha và  $\sigma_n$  thể hiện cho phương sai. Để đơn giản hóa ký hiệu, ta đặt

$$f_{N_x}(n_x|\tilde{S}_x, \sigma_n) = \frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{(n_x-\tilde{S}_x)^2}{2\sigma_n^2}} \quad (2)$$

$$f_{N_y}(n_y|\tilde{S}_y, \sigma_n) = \frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{(n_y-\tilde{S}_y)^2}{2\sigma_n^2}} \quad (3)$$

khi này (1) trở thành

$$\Pr\{\text{lỗi}|S_1\} = \iint_{(\tilde{S}_x+n_x, \tilde{S}_y+n_y) \notin D_1} f_{N_x}(n_x|\tilde{S}_x, \sigma_n) \cdot f_{N_y}(n_y|\tilde{S}_y, \sigma_n) dn_x dn_y \quad (4)$$

Xác suất lỗi này có thể được giới hạn bởi biểu thức

$$\begin{aligned} \Pr\{\text{lỗi}|S_1\} < \iint_{(\tilde{S}_x+n_x, \tilde{S}_y+n_y) \in (D_2 \cup D_3)} f_{N_x}(n_x|\tilde{S}_x, \sigma_n) \cdot f_{N_y}(n_y|\tilde{S}_y, \sigma_n) dn_x dn_y \\ + \iint_{(\tilde{S}_x+n_x, \tilde{S}_y+n_y) \in (D_3 \cup D_4)} f_{N_x}(n_x) \cdot f_{N_y}(n_y) dn_x dn_y \end{aligned} \quad (5)$$

trong đó xuất hiện ranh giới vì vùng quyết định  $D_3$  xảy ra hai lần trong (5). Từ định nghĩa về các vùng quyết định ta có thể viết

$$\begin{aligned} \Pr\{\text{lỗi}|S_1\} < \int_{-\infty}^0 f_{N_x}(n_x|\tilde{S}_x, \sigma_n) dx \int_{-\infty}^{\infty} f_{N_y}(n_y|\tilde{S}_y, \sigma_n) dy \\ + \int_{-\infty}^{\infty} f_{N_x}(n_x|\tilde{S}_x, \sigma_n) dn_x \int_{-\infty}^0 f_{N_y}(n_y|\tilde{S}_y, \sigma_n) dn_y \end{aligned} \quad (6)$$

Coi hai trong bốn tích phân trong (6) là bằng 1, nhận được

$$\Pr\{\text{lỗi}|S_1\} < \int_{-\infty}^0 f_{N_x}(n_x|\tilde{S}_x, \sigma_n) dn_x + \int_{-\infty}^0 f_{N_y}(n_y|\tilde{S}_y, \sigma_n) dn_y \quad (7)$$

Thế (2) và (3) vào biểu thức trên và dùng các định nghĩa về  $\tilde{S}_x$  và  $\tilde{S}_y$  nhận được giới hạn cho xác suất lỗi có điều kiện. Giới hạn xác suất lỗi có điều kiện này là

$$\Pr\{\text{lỗi}|S_1\} < Q\left(\frac{\text{Re}(\tilde{S}_1)}{\sigma_n}\right) + Q\left(\frac{\text{Im}(\tilde{S}_1)}{\sigma_n}\right) \quad (8)$$

trong đó, hàm  $Q(\cdot)$  là hàm Q Gauss. Do tính đối xứng xác suất có điều kiện là như nhau đối với bốn ký hiệu phát có thể có.

Giống như trường hợp PSK ta đã giả thiết  $S_k$  là ký hiệu phát thứ  $k$  trong chuỗi  $N$  ký hiệu được mô phỏng. Với mỗi giá trị của  $k$ ,  $1 \leq k \leq N$ , thì  $S_k$  sẽ là  $S_1$ ,  $S_2$ ,  $S_3$ , hoặc  $S_4$ . Ranh giới cho tỉ số lỗi ký hiệu có điều kiện là, từ (8)

$$\Pr\{\text{lỗi}|S_k\} < Q\left(\frac{\text{Re}(\tilde{S}_k)}{\sigma_n}\right) + Q\left(\frac{\text{Im}(\tilde{S}_k)}{\sigma_n}\right) \quad (9)$$

Tỉ số lỗi ký hiệu tổng đạt được bằng cách lấy trung bình xác suất lỗi ký hiệu có điều kiện trên toàn bộ chuỗi  $N$  ký hiệu được cho bởi

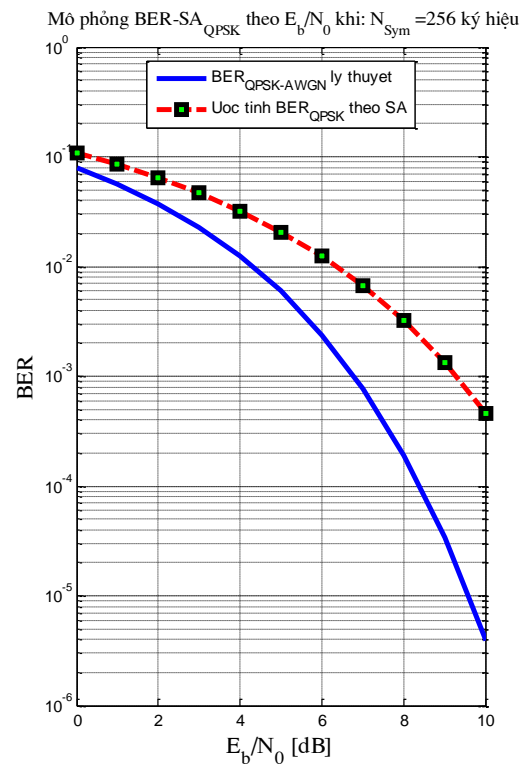
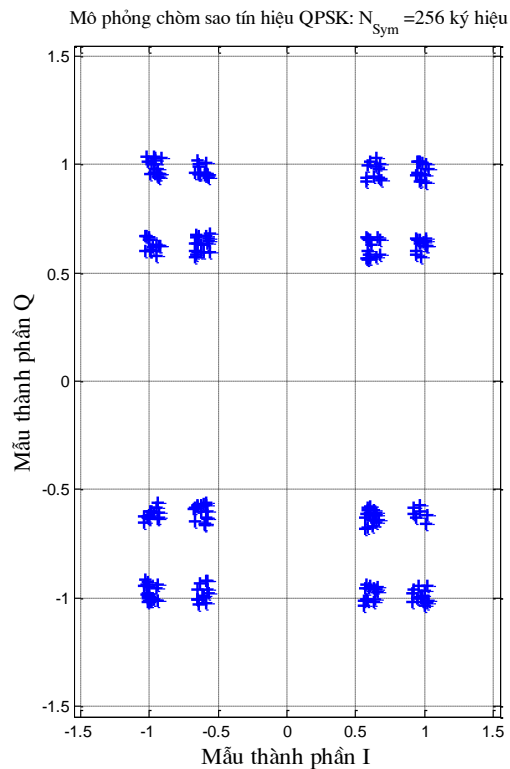
$$\Pr\{\text{lỗi}|S_1\} < \frac{1}{N} \sum_{k=1}^N \left[ Q\left(\frac{\text{Re}(\tilde{S}_k)}{\sigma_n}\right) + Q\left(\frac{\text{Im}(\tilde{S}_k)}{\sigma_n}\right) \right] \quad (10)$$

Tỉ số lỗi bit  $P_E = \frac{P_s}{2}$ . Lưu ý rằng, trong trường hợp PSK ta đạt được một nghiệm chính xác, trong khi đó với QPSK ta có một giới hạn (ranh giới). Kỹ thuật được dùng ở đây để phát triển bộ ước tính bán giải tích dễ dàng được mở rộng cho MPSK và QAM.

Bộ ước tính được triển khai ở đây sẽ được dùng thông suốt các phần còn lại của tài liệu nhằm ước lượng hiệu năng của một số hệ thống. Quy tụ lại sẽ là các ví dụ để minh họa ảnh hưởng của đa đường và fading trong hệ thống vô tuyến và ảnh hưởng của méo phi tuyến trong hệ thống truyền thông vệ tinh ghép kênh theo tần số.

Mã chương trình Matlab thực hiện mô phỏng bán giải tích hệ thống QPSK được cho ở file **NVD10\_QPSKSA.m** dưới đây. Thực hiện mô phỏng cho phép nghiên cứu các ảnh hưởng của ISI do lọc phát gây ra. Độ rộng băng tần được đặt bằng tốc độ ký hiệu (là một nửa tốc độ bit nghĩa là  $BW = \frac{1}{2}$ ). Vì chòm sao tín hiệu là đối xứng, nên tất cả các điểm tín hiệu thu bị quay đến cung phần tư thứ nhất như đã được đề cập ở trên.

Thực hiện mô phỏng, ta được chòm sao tín hiệu và BER được minh họa ở hình dưới đây. Lưu ý rằng, chòm sao tín hiệu thu không còn chứa 4 điểm như trường hợp QPSK lý tưởng nữa mà bây giờ nó gồm 16 điểm. Để làm rõ vấn đề này, ta giả sử rằng điểm tín hiệu thu trong cung phần tư thứ nhất trình bày cho bit dữ liệu 00 và do tính có nhớ của hệ thống gây ra ISI là hai ký hiệu (các ký hiệu phát hiện tại và ký hiệu phát trước đó). Hậu quả là tạo ra 4 điểm tín hiệu từ việc truyền cặp bit 00. Bốn điểm tín hiệu này tương ứng với 00|00, 00|01, 00|10, 00|11, trong đó thanh thẳng đứng phác họa cho ký hiệu hiện thời và các ký hiệu được phát trước đó. Cũng cần lưu ý rằng mỗi điểm trong số 4 điểm ở cung phần tư thứ nhất được bao gồm các điểm mà hơi bị tán xạ. Hiện tượng tán xạ này do hệ thống thể hiện tính chất nhớ với độ lâu lớn hơn hai ký hiệu, mặc dù ảnh hưởng của tính có nhớ này là nhỏ. Mặt phẳng phía bên phải của hình dưới đây minh họa cho BER của hệ thống khi có lọc phát. Kết quả mô phỏng cho kênh AWGN cũng được đưa ra nhằm so sánh và tham khảo. Thấy rõ BER tăng lên do ISI



### Kết quả mô phỏng QPSK trong điều kiện tính đến ISI

**% File: NVD10\_QPSKSA.m**

% File: NVD10\_QPSKSA.m

% H10.17

clc;

clear all;

close all;

=====

% Default parameters

=====

NN = 256; % number of symbols

tb = 0.5; % bit time

p0 = 1; % power

fs = 16; % samples/symbol

ebn0db = [0:1:10]; % Eb/N0 vector

[b,a] = butter(5,1/16); % transmitter filter parameters

=====

% Establish QPSK signals

=====

x = random\_binary(NN,fs)+i\*random\_binary(NN,fs); % QPSK signal

y1 = x; % save signal

y2a = y1\*sqrt(p0); % scale amplitude

=====

% Transmitter filter

=====

y2 = filter(b,a,y2a); % filtered signal

=====

% Matched filter

=====

b = ones(1,fs); b = b/fs; a = 1; % matched filter parameters

y = filter(b,a,y2); % matched filter output

```
% End of simulation
%=====
% Use the semianalytic BER estimator. The following sets
% up the semi analytic estimator. Find the maximum magnitude
% of the cross correlation and the corresponding lag.
%=====
[cor lags] = vxcorr(x,y);
cmax      = max(abs(cor));
nmax      = find(abs(cor)==cmax);
timelag   = lags(nmax);
theta     = angle(cor(nmax));
y         = y*exp(-i*theta);      % derotate
%=====
% Noise BW calibration
%=====
hh         = impz(b,a);           % receiver impulse response
nbw        = (fs/2)*sum(hh.^2);   % noise bandwidth
%=====
% Delay the input, and do BER estimation on the last 128 bits.
% Use middle sample. Make sure the index does not exceed number
% of input points. Eb should be computed at the receiver input.
%=====
index      = (10*fs+8:fs:(NN-10)*fs+8);
xx         = x(index);
yy         = y(index-timelag+1);
[n1 n2]    = size(y2); ny2=n1*n2;
eb         = tb*sum(sum(abs(y2).^2))/ny2;
eb         = eb/2;
[peideal,pesystem] = NVD10_qpsk_berest(xx,yy,ebn0db,eb,tb,nbw);

%=====
% Display for Results
%=====
h10_17 = figure
set(h10_17,'color','c','name','H10.17: NVD')
%=====
subplot(1,2,1)
yscale    = 1.5*max(real(yy));
plot(yy,'+')
xlabel('Mức thịnh phÇn I','fontname','.vnTime','FontSize',14);
ylabel('Mức thịnh phÇn Q','fontname','.vnTime','FontSize',14);
axis([-yscale yscale -yscale yscale]);
title([' M« phÇng chßm sao tÿn hiÖu QPSK: N_S_y_m =',num2str(NN),' ký hiÖu'],...
      'FontName','.VnTime','FontSize',12);
grid;
%=====
subplot(1,2,2)
semilogy(ebn0db,peideal,ebn0db,pesystem,...
         '--rs','LineWidth',2,...
         'MarkerEdgeColor','k',...
         'MarkerFaceColor','g',...
         'MarkerSize',7);
xlabel('E_b/N_0 [dB]','Fontname','.VnTime','FontSize',14);
ylabel('BER','Fontname','.VnTime','FontSize',14);
title([' M« phÇng BER-SA_Q_P_S_K theo E_b/N_0 khi: N_S_y_m =',num2str(NN),' ký hiÖu'],...
      'FontName','.VnTime','FontSize',12);
legend('BER_Q_P_S_K_-_A_W_G_N ly thuyet','Uoc tinh BER_Q_P_S_K theo SA',0);grid;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [x, bits] = random_binary(nbits,nsamples)
```

```
% File: random_binary.m
% This function generates a random binary waveform of length nbits
% sampled at a rate of nsamples/bit.

x      = zeros(1,nbits*nsamples);
bits   = round(rand(1,nbits));
for m=1:nbits
    for n=1:nsamples
        index      = (m-1)*nsamples + n;
        x(1,index) = (-1)^bits(m);
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [peideal,pesystem] = NVD10_qpsk_berest(xx,yy,ebn0db,eb,tb,nbw)
% File: NVD10_qpsk_berest.m
% ebn0db is an array of Eb/No values in db (specified at the
% receiver input); tb is the bit duration and nbw is the noise BW
% xx is the reference (ideal) input; yy is the distorted output;
%
[n1 n2]      = size(xx);
nx           = n1*n2;
[n3 n4]      = size(yy);
ny           = n3*n4;
[n5 n6]      = size(ebn0db);
neb          = n5*n6;
%=====
% For comparison purposes, set the noise BW of the ideal
% receiver (integrate and dump) to be equal to rs/2.
%=====
nbwideal     = 1/(2*tb*2);
h            = waitbarqa(0,'M« pháng BER-SA_Q_P_S_K cho tổng gi, trÞ của E_b/N_0');
for m=1:neb
    waitbarqa(m/length(neb));
    peideal(m) = 0.0; pesystem(m) = 0.0;           % initialize
    %-----
    % Find n0 and the variance of the noise.
    %-----
    string1 = ['Eb/No = ',num2str(ebn0db(m))];
    disp(string1)                                % track execution
    ebn0(m)  = 10^(ebn0db(m)/10);                % dB to linear
    n0       = eb/ebn0(m);                        % noise power
    sigma    = sqrt(n0*nbw*2);                    % variance
    sigma1   = sqrt(n0*nbwideal*2);                % variance of ideal
    %-----
    % Multiply the input constellation/signal by a scale factor so that
    % input constellation and the constellations/signal at the input to
    % receive filter have the same ave power a=sqrt(2*eb/(2*tb)).
    %-----
    b       = sqrt(2*eb/tb)/sqrt(sum(abs(xx).^2)/nx);
    for n=1:nx
        theta = angle(xx(n));
        if (theta<0)
            theta = theta+2*pi;
        end
        %-----
        % Rotate x and y to the first quadrant and compute BER.
        %-----
        xxx(n) = b*xx(n)*exp(-i*(theta-(pi/4)));
        yyy(n) = yy(n)*exp(-i*(theta-(pi/4)));
        d1     = real(xxx(n)); d2 = imag(xxx(n)); % reference
        d3     = real(yyy(n)); d4 = imag(yyy(n)); % system
        pe1    = q(d1/sigma1)+q(d2/sigma1);      % reference
    end
end
```

```

pe2          = q(d3/sigma)+q(d4/sigma); % system
peideal(m)   = peideal(m)+pe1;        % SER of reference
pesystem(m)  = pesystem(m)+pe2;       % SER of system
end
end
close(h);
peideal      = (1/2)*peideal./nx;      % convert to BER
pesystem     = (1/2)*pesystem./nx;     % convert to BER

```

#### Bài 42:

Hãy đọc mã chương trình Matlab **NVD14\_threeray.m** dưới đây, vẽ mô hình mô phỏng, phân tích mô hình mô phỏng, chạy chương trình mô phỏng, phân tích kết quả mô phỏng.

#### Hướng dẫn giải

Chương trình mô phỏng hiệu năng BER hệ thống QPSK hoạt động trong môi trường kênh đa đường 3 tia với AWGN và so sánh hiệu năng BER với chính hệ thống đó nhưng hoạt động trên kênh AWGN lý tưởng (không có đa đường). Trong đó giả thiết sau:

1. Kênh có 3 đường (tia) gồm có: 1 đường đi thẳng LOS không bị phản xạ và 2 thành phần Rayleigh. Các mức công suất thu liên quan tới mỗi đường và các trễ *chênh lệch* giữa ba đường là các tham số mô phỏng.
2. Phản xạ Rayleigh trong kênh chỉ ảnh hưởng lên *biên độ* của tín hiệu phát. Pha tức thời không bị ảnh hưởng.
3. Suy hao của mỗi thành phần đa đường là không đổi trên mỗi khoảng thời gian ký hiệu và có các giá trị độc lập trên các khoảng liên tiếp (không yêu cầu phải định dạng phổ Doppler).
4. Không sử dụng bộ lọc phát và mô hình máy thu là máy thu kết xuất và tích hợp lý tưởng.

Với các giả định trên thì, tín hiệu thu cho trong trường hợp này được viết là

$$\tilde{y}(t) = \underbrace{a_0 \tilde{x}(t)}_{\text{LOS}} + \underbrace{a_1 R_1 \tilde{x}(t)}_{\text{Rayleigh}} + \underbrace{a_2 R_2 \tilde{x}(t - \tau)}_{\text{Rayleigh bị trễ}}$$

trong đó  $R_1$  và  $R_2$  là hai biến ngẫu nhiên Rayleigh độc lập thể hiện cho suy hao của hai đường Rayleigh và  $\tau$  là trễ *trương đối* giữa hai thành phần Rayleigh. Biến đổi Fourier của  $\tilde{y}(t)$  là

$$\tilde{Y}(f) = a_0 \tilde{X}(f) + a_1 R_1 \tilde{X}(f) + a_2 R_2 \tilde{X}(f) e^{-j2\pi f \tau}$$

dẫn đến hàm truyền đạt kênh là

$$\tilde{H}(f) = \frac{\tilde{Y}(f)}{\tilde{X}(f)} = a_0 + a_1 R_1 + a_2 R_2 e^{-j2\pi f \tau}$$

Rõ ràng, nếu tích  $f\tau$  không thể bỏ qua so với độ rộng băng tần của tín hiệu, thì kênh là kênh chọn lọc tần số, dẫn đến trải rộng trễ và ISI. Các giá trị của  $a_0$ ,  $a_1$ ,  $a_2$  xác định các mức công suất tương đối  $P_0$ ,  $P_1$  và  $P_2$  của ba thành phần đa đường.

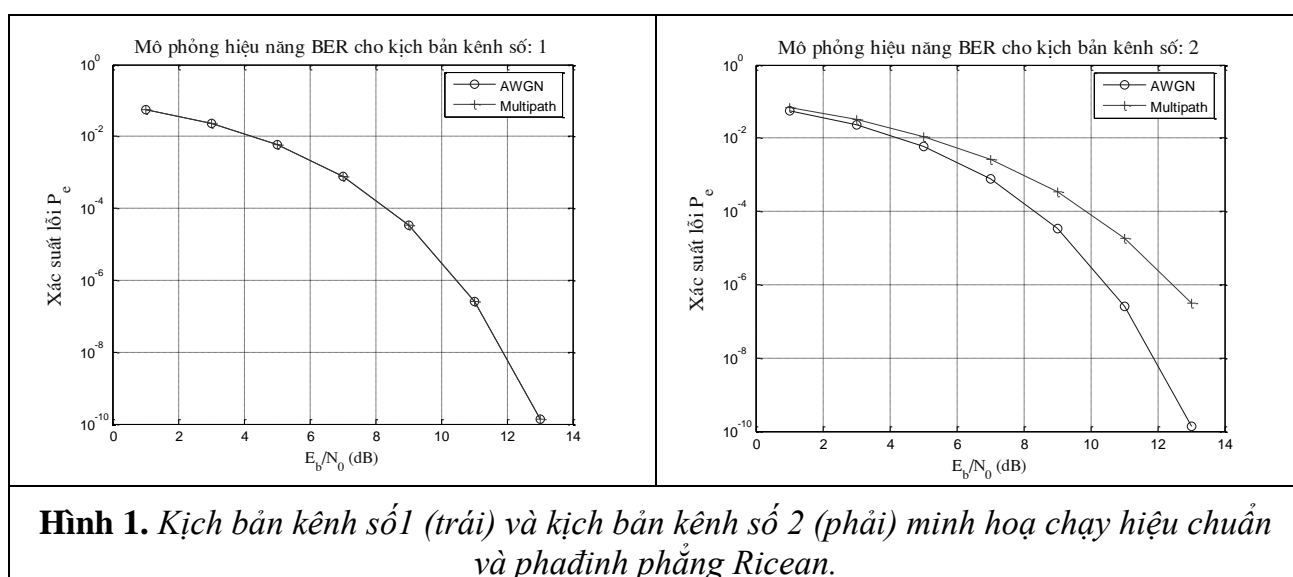
Mô phỏng được thực hiện cho một trong sáu tập giá trị tham số được cho ở bảng dưới đây. Với mỗi kịch bản kênh, BER được ước lượng dùng ước tính bán phân tích. Trong

bảng, trễ được biểu diễn theo chu kỳ lấy mẫu. Vì tần số lấy mẫu mô phỏng là 16 mẫu trên ký hiệu, nên  $\tau = 8$  tương đương với trễ là một nửa chu kỳ lấy mẫu.

Các kịch bản kênh để minh họa cho phasinh

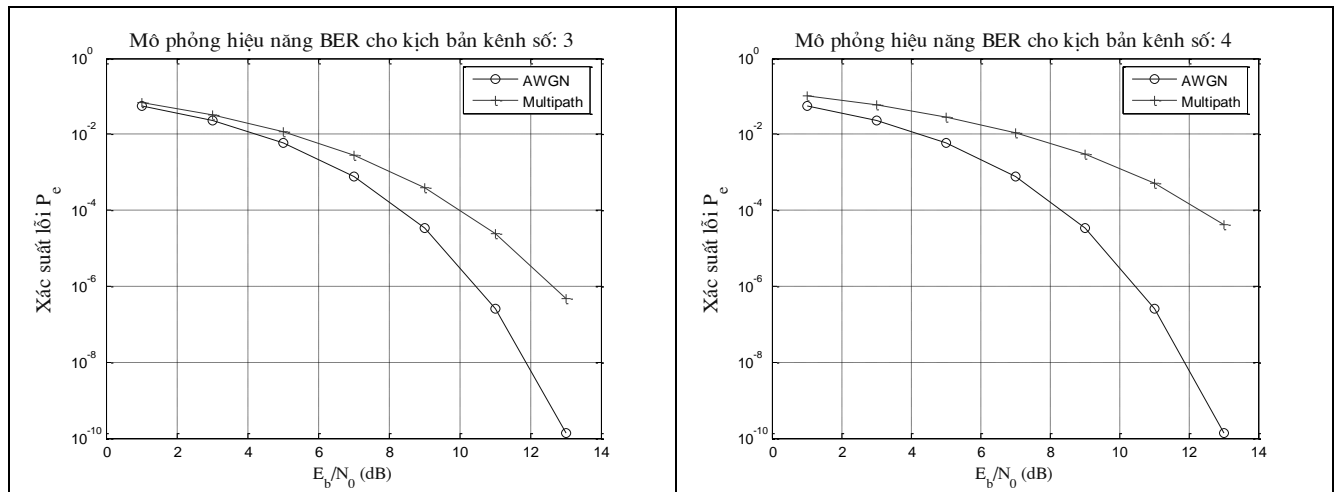
Kịch bản kênh số	$P_0$	$P_1$	$P_2$	$\tau$ (mẫu)	Chú thích
1	1,0	0	0	0	Dùng để phê chuẩn (kênh lý tưởng AWGN)
2	1,0	0,2	0	0	Phasinh phẳng Ricean
3	1,0	0	0,2	0	Phasinh phẳng Ricean
4	1,0	0	0,2	8	Phasinh chọn lọc tần số Ricean
5	0	1,0	0,2	0	Phasinh phẳng Rayleigh
6	0	1,0	0,2	8	Phasinh chọn lọc tần số Rayleigh

Kết quả mô phỏng đối với kịch bản kênh số 1 và 2 được minh họa ở hình 1. Trong kịch bản số 1, chỉ có một thành phần đi thẳng LOS mà không có đa đường, dẫn đến ước tính BER bán phân tích cho hệ thống QPSK hoạt động trong môi trường kênh AWGN. Việc mô phỏng này nhằm để kiểm tra phương pháp luận mô phỏng và tạo ra các *kết quả ranh giới* biểu diễn cho một hệ thống QPSK lý tưởng. Với mục đích so sánh, nên kết quả này được hiển thị cùng với kết quả BER của 5 kịch bản kênh còn lại. Bảng cho thấy, kịch bản 2 là kịch bản số 1 thêm một thành phần phasinh Rayleigh. Việc thêm vào này có thể đáp ứng với kênh phasinh Ricean. Vì  $\tau=0$ , nên kịch bản kênh số 2 là phasinh phẳng (không chọn lọc tần số). Thấy rõ, BER tăng lên so với kịch bản kênh số 1 (trường hợp kênh lý tưởng).



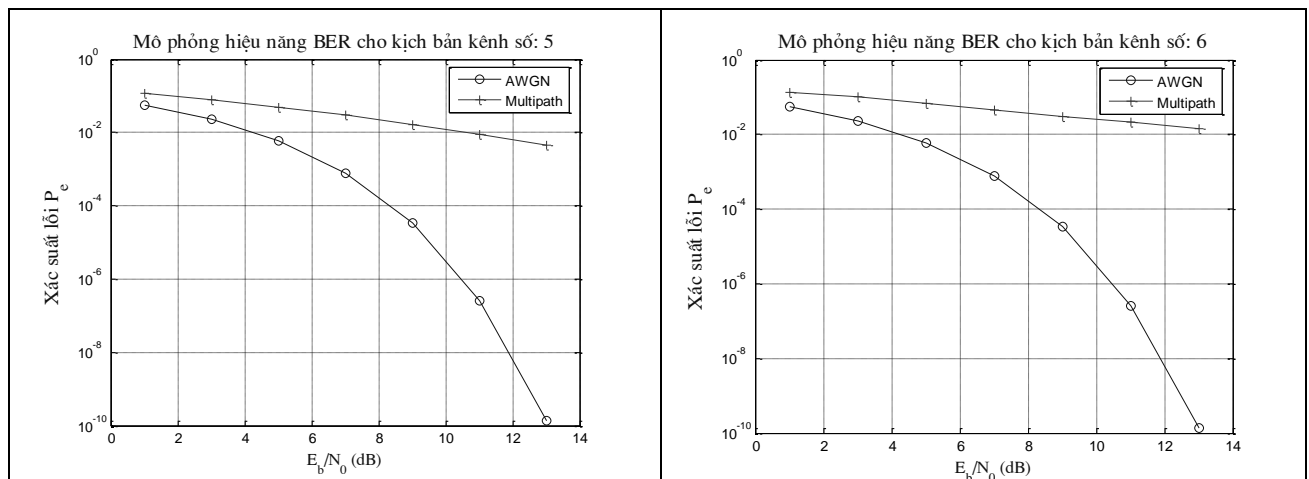
Kết quả mô phỏng cho kịch bản kênh số 3 và 4 được minh họa ở hình 3. Kịch bản kênh số 3 về cơ bản tương đương kịch bản kênh số 2. Sự khác nhau nhỏ là do quá trình

phản ánh khác với quá trình phản ánh được dùng ở kịch bản kênh số 2 do sự khởi tạo khác nhau của bộ tạo số ngẫu nhiên cơ bản. Kịch bản kênh số 4 giống với kịch bản kênh số 3 ngoại trừ là phản ánh bây giờ là chọn lọc tần số. Lưu ý hiệu năng hệ thống bị suy giảm hơn.



**Hình 3.** Lớp 3 (bảng bên trái) và lớp 4 (bảng bên phải) minh họa phản ánh nền Ricean và phản ánh chọn lọc tần số.

Kết quả mô phỏng cho kịch bản kênh số 5 và 6 được cho ở hình 4. Lưu ý rằng, cả hai kịch bản kênh này đều không có thành phần đi thẳng. Sự so sánh kết quả kịch bản số 5 với 4 kết quả trước cho thấy: kể cả với kịch bản phản ánh phẳng (bên trái), hiệu năng đều xấu hơn so với trường hợp có thành phần đi thẳng LOS. Kịch bản kênh số 6 tương tự với kịch bản kênh số 5 ngoại trừ là phản ánh bây giờ là chọn lọc tần số. Lưu ý rằng hiệu năng hệ thống bị suy thoái hơn. Các kênh Ricean và Rayleigh sẽ được khai thác sâu hơn trong các phần tiếp theo.



**Hình 4.** Kịch bản kênh số 5 (trái) và số 6 (phải) minh họa phản ánh chọn lọc tần số và phản ánh phẳng Rayleigh

```
% file: NVD14_threeray.m
clc;
clear all;
close all;
```



```
%=====
% Default parameters
NN = 256; % number of symbols
tb = 0.5; % bit time
fs = 16; % samples/symbol
ebn0db = [1:2:14]; % Eb/N0 vector

%=====
% Establish QPSK signals
x = NVD14_random_binary(NN,fs)+i*NVD14_random_binary(NN,fs);
%=====
% Input powers and delays
Scennario = input(' Chon kich ban kenh so 1-6 (theo bang 4.1): ');
if Scennario==1 % Validation: AWGN
    p0 = 1;
    p1 = 0;
    p2 = 0;
    delay = 0;
elseif Scennario==2 % Ricean flat fadinh
    p0 = 1;
    p1 = 0.2;
    p2 = 0;
    delay = 0;
elseif Scennario==3 % Ricean flat fadinh
    p0 = 1;
    p1 = 0;
    p2 = 0.2;
    delay = 0;
elseif Scennario==4 % Ricean frequency selective fadinh
    p0 = 1;
    p1 = 0;
    p2 = 0.2;
    delay = 8;
elseif Scennario==5 % Rayleigh flat fadinh
    p0 = 0;
    p1 = 1;
    p2 = 0.2;
    delay = 0;
else % Rayleigh frequency selective fadinh
    p0 = 0;
    p1 = 1;
    p2 = 0.2;
    delay = 8;
end

delay0 = 0; delay1 = 0; delay2 = delay;

%=====
% Set up the Complex Gaussian (Rayleigh) gains

gain1 = sqrt(p1)*abs(randn(1,NN) + i*randn(1,NN));
gain2 = sqrt(p2)*abs(randn(1,NN) + i*randn(1,NN));
for k = 1:NN
    for kk=1:fs
        index = (k-1)*fs+kk;
        ggain1(1,index) = gain1(1,k);
        ggain2(1,index) = gain2(1,k);
    end
end

y1 = x;
for k=1:delay2
    y2(1,k) = y1(1,k)*sqrt(p0);
end
```

```

for k=(delay2+1):(NN*fs)
    y2(1,k)= y1(1,k)*sqrt(p0) + y1(1,k-delay1)*ggain1(1,k)+ y1(1,k-
delay2)*ggain2(1,k);
end
%=====
% Matched filter
b = ones(1,fs); b = b/fs; a = 1;
y = filter(b,a,y2);
% End of simulation
%=====
%=====
% Use the semianalytic BER estimator. The following sets
% up the semi analytic estimator. Find the maximum magnitude
% of the cross correlation and the corresponding lag.
[cor lags] = NVD14_vxcorr(x,y);
[cmax nmax] = max(abs(cor));
timelag = lags(nmax);
theta = angle(cor(nmax));
y = y*exp(-i*theta); % derotate
%-----
% Noise BW calibration
%-----
hh = impz(b,a);
ts = 1/16;
nbw = (fs/2)*sum(hh.^2);
%-----
% Delay the input, and do BER estimation on the last 128 bits.
% Use middle sample. Make sure the index does not exceed number
% of input points. Eb should be computed at the receiver input.
%-----
index = (10*fs+8:fs:(NN-10)*fs+8);
xx = x(index);
yy = y(index-timelag+1);
[n1 n2] = size(y2); ny2=n1*n2;
eb = tb*sum(sum(abs(y2).^2))/ny2;
eb = eb/2;
[peideal,pesystem] = NVD14_qpsk_berest(xx,yy,ebn0db,eb,tb,nbw);
%=====
h14_2 = figure
set(h14_2,'name','H14.2 den H14.4: NVD')
semilogy(ebn0db,peideal,ebn0db,pesystem,...
    '--rs','LineWidth',2,...
    'MarkerEdgeColor','k',...
    'MarkerFaceColor','g',...
    'MarkerSize',7);
% semilogy(ebn0db,peideal,'-o',ebn0db,pesystem,'r+-');
xlabel('E_b/N_0 (dB)','fontname','.Vntime','color','b','fontsize',12);
ylabel('X_c suÊt lçi P_e','fontname','.Vntime','color','b','fontsize',14);
title(['M« pháng hiÖu n`ng BER cho kÐch b¶n k`nh sè: ',...
    num2str(Scennario),'; N_S_y_m =',num2str(NN),' ký hiÖu'],...
    'fontname','.Vntime','color','b','fontsize',16)
axis([0 14 10^(-10) 1]);
legend('AWGN','Da duong');
grid;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [x, bits] = NVD14_random_binary(nbits,nsamples)
% File: random_binary.m
% This function genrates a random binary waveform of length nbits
% sampled at a rate of nsamples/bit.
x = zeros(1,nbits*nsamples);
bits = round(rand(1,nbits));
for m=1:nbits
    for n=1:nsamples

```

```
        index      = (m-1)*nsamples + n;
        x(1,index)  = (-1)^bits(m);
    end
end
```

%%%

**function [c,lags] = NVD14\_vxcorr(a,b)**

% File: vxcorr.m

% This function calculates the unscaled cross-correlation of 2 vectors of the  
% same length. The output length(c) is length(a)+length(b)-1.

% It is a simplified function of xcorr function in matlabR12 using the  
% definition:  $c(m) = E[a(n+m) \cdot \text{conj}(b(n))] = E[a(n) \cdot \text{conj}(b(n-m))]$

%

a = a(:); % convert a to column vector

b = b(:); % convert b to column vector

M = length(a); % same as length(b)

maxlag = M-1; % maximum value of lag

lags = [-maxlag:maxlag]'; % vector of lags

A = fft(a,2^nextpow2(2\*M-1)); % fft of A

B = fft(b,2^nextpow2(2\*M-1)); % fft of B

c = ifft(A.\*conj(B)); % crosscorrelation

%

% Move negative lags before positive lags

%

c = [c(end-maxlag+1:end,1);c(1:maxlag+1,1)];

%

% Return row vector if a, b are row vectors

%

[nr nc]=size(a);

if(nr>nc)

c=c.';

lags=lags.';

end

% End of function file.

%%%

**function [peideal,pesystem] = NVD14\_qpsk\_berest(xx,yy,ebn0db,eb,tb,nbw)**

% File: qpsk\_berest.m

% ebn0db is an array of Eb/No values in db (specified at the

% receiver input); tb is the bit duration and nbw is the noise BW

% xx is the reference (ideal) input; yy is the distorted output;

[n1 n2] = size(xx);

nx = n1\*n2;

[n3 n4] = size(yy);

ny = n3\*n4;

[n5 n6] = size(ebn0db);

neb = n5\*n6;

%=====

% For comparison purposes, set the noise BW of the ideal

% receiver (integrate and dump) to be equal to rs/2.

nbwideal = 1/(2\*tb\*2);

for m=1:neb

% initialize

peideal(m) = 0.0;

pesystem(m) = 0.0;

%-----

% Find n0 and the variance of the noise.

string1 = ['Eb/No = ',num2str(ebn0db(m))];

disp(string1) % track execution

ebn0(m) = 10^(ebn0db(m)/10); % dB to linear

```

n0          = eb/ebn0(m);          % noise power
sigma       = sqrt(n0*nbw*2);      % variance
sigma1      = sqrt(n0*nbwideal*2); % variance of ideal
%-----
% Multiply the input constellation/signal by a scale factor so that
% input constellation and the constellations/signal at the input to
% receive filter have the same ave power a=sqrt(2*eb/(2*tb)).
b = sqrt(2*eb/tb)/sqrt(sum(abs(xx).^2)/nx);
for n=1:nx
    theta    = angle(xx(n));
    if (theta<0)
        theta = theta+2*pi;
    end
    %-----
    % Rotate x and y to the first quadrant and compute BER.
    %-----
    xxx(n)   = b*xx(n)*exp(-i*(theta-(pi/4)));
    yyy(n)   = yy(n)*exp(-i*(theta-(pi/4)));
    d1       = real(xxx(n)); d2 = imag(xxx(n)); % reference
    d3       = real(yyy(n)); d4 = imag(yyy(n)); % system
    pe1      = NVD14_q(d1/sigma1) + NVD14_q(d2/sigma1); % reference
    pe2      = NVD14_q(d3/sigma) + NVD14_q(d4/sigma); % system
    peideal(m) = peideal(m)+pe1; % SER of reference
    pesystem(m) = pesystem(m)+pe2; % SER of system
end
end
peideal      = (1/2)*peideal./nx; % convert to BER
pesystem     = (1/2)*pesystem./nx; % convert to BER

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function y=NVD14_q(x)
% File: NVD14_q.m
y = 0.5*erfc(x/sqrt(2));

```

### Bài 43:

Hãy đọc mã chương trình Matlab **NVD\_CS79.m** và chạy chương trình, để làm sáng tỏ nguyên lý hoạt động của vòng khóa pha PLL trong các mạch khôi phục sóng mang.

### Hướng dẫn giải

Khôi phục sóng mang gồm: Bộ dao động nội dùng vòng khoá pha PLL để thực hiện đồng bộ sóng mang sao cho pha sóng mang khôi phục được điều khiển để nằm trong trạng thái đồng bộ với tín hiệu sóng mang. Vòng khoá pha là một hệ thống điều khiển hồi tiếp phi tuyến, điều khiển pha bộ dao động nội thu.

Để đơn giản, ta xét hệ thống điều chế BPSK. Vòng khoá pha PLL được điều khiển bởi tín hiệu sin tại tần số sóng mang (hoặc bội số của nó). Để có được tín hiệu sin điều khiển vòng khoá pha PLL, thì tín hiệu điều chế DSB.

$$u(t) = A_c m(t) \cos(2\pi f_c t - \phi(t)) \quad (1)$$

trong đó  $m(t) = \pm 1$  lấy bình phương  $u(t)$  ta được.

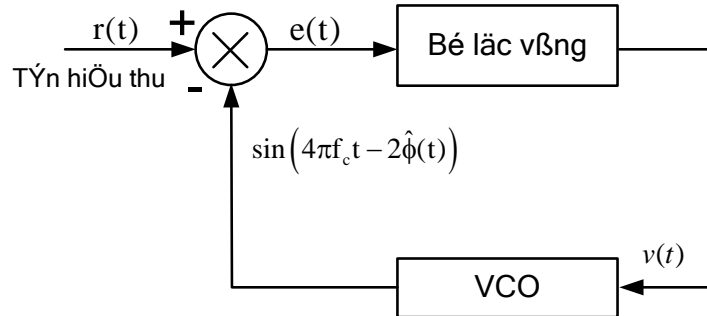
$$\begin{aligned}
 u^2(t) &= A_c^2 m^2(t) \cos^2(2\pi f_c t - \phi(t)) \\
 &= \frac{A_c^2}{2} m^2(t) + \frac{A_c^2}{2} m^2(t) \cos(4\pi f_c t - 2\phi(t)) \\
 &= \frac{A_c^2}{2} + \frac{A_c^2}{2} \cos(4\pi f_c t - 2\phi(t))
 \end{aligned} \tag{2}$$

Thấy rõ, tín hiệu này có thành phần tần số  $2f_c$ . Sở dĩ ta không xét trực tiếp  $u(t)$  là vì,  $m(t)$  thường là quá trình ngẫu nhiên có trung bình không ( $\pm 1$  đồng xác suất)  $\Rightarrow$  vì vậy không có công suất của  $u(t)$  tại  $f_c$ . Nếu cho tín hiệu  $u^2(t)$  qua bộ lọc thông băng được điều chỉnh đến  $2f_c$ , thì đầu ra bộ lọc sẽ là tín hiệu sin có tần số trung tâm  $2f_c$ , pha  $-2\phi(t)$  và biên độ  $\frac{1}{2}A_c^2 H(2f_c)$ . Bỏ qua suy hao, giả thiết biên độ bằng 1, nghĩa là tín hiệu đầu vào vòng khoá pha PLL là.

$$r(t) = \cos(4\pi f_c t - 2\phi(t)) \tag{3}$$

Vòng khoá pha gồm bộ nhân, bộ lọc vòng, bộ dao động điều khiển bằng điện áp VCO được cho ở hình 1. Nếu giả thiết đầu ra của VCO là  $\sin(4\pi f_c t - 2\hat{\phi}(t))$  thì tại đầu vào bộ lọc vòng ta có.

$$\begin{aligned}
 e(t) &= \cos[4\pi f_c t - 2\phi(t)] \times \sin[4\pi f_c t - 2\hat{\phi}(t)] \\
 &= \frac{1}{2} \sin[2\phi(t) - 2\hat{\phi}(t)] + \frac{1}{2} \sin[8\pi f_c t - 2\phi(t) - 2\hat{\phi}(t)]
 \end{aligned} \tag{4}$$



**Hình 1:** Vòng khoá pha PLL

Lưu ý rằng,  $e(t)$  chứa thành phần tần số thấp và tần số cao. Bộ lọc vòng khử thành phần tần số cao và đảm bảo  $\hat{\phi}(t)$  bám theo sự thay đổi của  $\phi(t)$ . Bộ lọc vòng đơn giản là bộ lọc thông thấp bậc 1 có hàm truyền đạt.

$$G(s) = \frac{1 + \tau_1 s}{1 + \tau_2 s} \tag{5}$$

trong đó:  $\tau_2 \gg \tau_1$ . Nếu ký hiệu đầu vào VCO là  $v(t)$ , thì đầu ra VCO sẽ là sin mà độ lệch tần tức thì của nó so với  $2f_c$  tỉ lệ với  $v(t)$ . Nhưng tần số tức thời đầu ra VCO là.

$$2f_c + \frac{1}{\pi} \frac{d}{dt} \hat{\phi}(t)$$

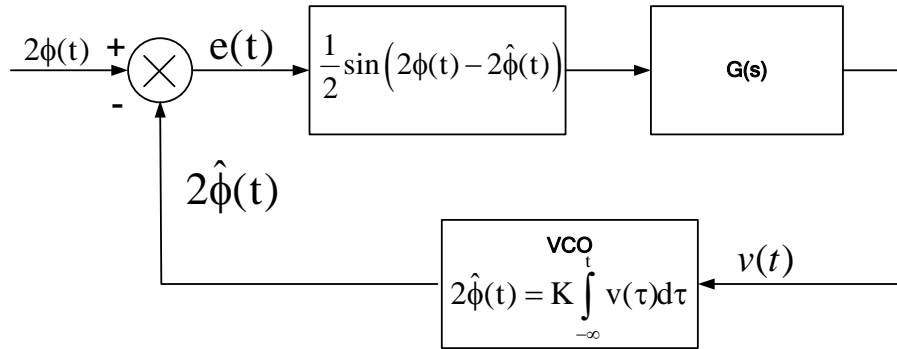
vì vậy.

$$\frac{d}{dt} \hat{\phi}(t) = \frac{K}{2} v(t) \quad (6)$$

tương đương.

$$2\hat{\phi}(t) = K \int_{-\infty}^t v(\tau) d\tau \quad (7)$$

trong đó: K là hằng số tỉ lệ. Sau khi khử các hài bậc 2 và bậc 4, vòng khóa pha PLL quy về hình 2.

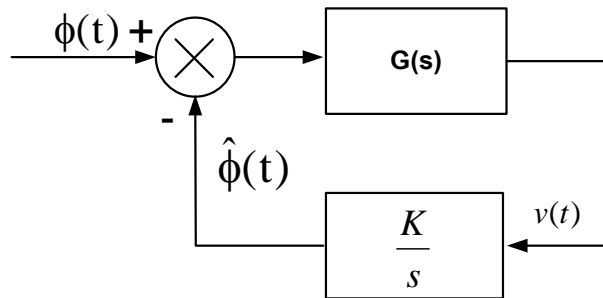


**Hình 2:** Vòng khóa pha PLL sau khi khử các thành phần tần số cao

Giả sử  $\hat{\phi}(t)$  bám theo  $\phi(t)$ , thì lệch pha  $2(\phi(t) - \hat{\phi}(t))$  là rất nhỏ  $\Rightarrow$  có thể lấy xấp xỉ.

$$\frac{1}{2} \sin[2\phi(t) - 2\hat{\phi}(t)] \approx \phi(t) - \hat{\phi}(t) \quad (8)$$

Bằng cách xấp xỉ hoá này, thành phần **phi tuyến** ở hình 2 được thay bởi thành phần **tuyến tính**, kết quả là mô hình vòng khóa pha **PLL tuyến tính** hình 3. Lưu ý rằng, mô hình này được biểu diễn trong miền s (Laplace), bộ tích phân được thay bằng chuyển đổi Laplace của nó  $1/s$ .



**Hình 3:** Mô hình tuyến tính hóa cho vòng khóa pha PLL

Mô hình hình là **hệ thống điều khiển tuyến tính** có hệ số khuếch đại thuận là  $G(s)$  và hệ số khuếch đại hồi tiếp là  $K/s \Rightarrow$  vì vậy hàm truyền đạt của hệ thống được cho bởi.

$$H(s) = \frac{\hat{\Phi}(s)}{\Phi(s)} = \frac{K \frac{G(s)}{s}}{1 + K \frac{G(s)}{s}} \quad (9)$$

với mô hình bậc 1 đối với  $G(s)$  được giả thiết ở trên là

$$G(s) = \frac{1 + \tau_1 s}{1 + \tau_2 s} \quad (10)$$

thì hàm truyền đạt  $H(s)$  được cho như sau.

$$H(s) = \frac{\hat{\Phi}(s)}{\Phi(s)} = \frac{K \frac{G(s)}{s}}{1 + K \frac{G(s)}{s}} = \frac{1 + \tau_1 s}{1 + \left( \tau_1 + \frac{1}{K} \right) s + \frac{\tau_2}{K} s^2} \quad (11)$$

Nếu đầu vào vòng khoá pha PLL là  $\Phi(s)$  thì **lỗi** sẽ là.

$$\begin{aligned} \Delta\Phi(s) &= \Phi(s) - \hat{\Phi}(s) \\ &= \Phi(s) - \Phi(s)H(s) \\ &= [1 - H(s)]\Phi(s) \\ &= \frac{(1 + \tau_2 s)s}{K + (1 + K\tau_1)s + \tau_2 s^2} \Phi(s) \end{aligned} \quad (12)$$

Giả thiết tại thời điểm nào đó  $\phi(t) \approx \hat{\phi}(t)$  sao cho  $\Delta\phi(t) \approx 0$ . Tại thời điểm này, một số thay đổi đột biến gây ra sự **nhảy** trong  $\phi(t)$  để có thể được mô hình hoá như là bước, nghĩa là  $\Phi(s) = K_1/s$ . Với thay đổi này, ta có

$$\begin{aligned} \Delta\Phi(s) &= \frac{(1 + \tau_2 s)s}{K + (1 + K\tau_1)s + \tau_2 s^2} \times \frac{K_1}{s} \\ &= \frac{K_1(1 + \tau_2 s)}{K + (1 + K\tau_1)s + \tau_2 s^2} \end{aligned} \quad (13)$$

Bằng cách dùng định lý biến đổi Laplace

$$\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} sF(s) \quad (14)$$

chỉ cần tất cả các điểm cực của  $sF(s)$  có các phần thực âm, thì ta kết luận được

$$\lim_{t \rightarrow \infty} \Delta\phi(t) = \lim_{s \rightarrow 0} s\Delta\Phi(s) = \lim_{s \rightarrow 0} \frac{K_1(1 + \tau_2 s)}{K + (1 + K\tau_1)s + \tau_2 s^2} = 0 \quad (15)$$

Nói cách khác, bộ lọc vòng bậc 1 làm cho vòng khoá pha PLL có thể bám các bước nhảy trong pha đầu vào.

Hàm truyền đạt (11) được viết lại dưới dạng chuẩn hoá

$$H(s) = \frac{\left( 2\zeta\omega_n - \frac{\omega_n^2}{K} \right) s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (16)$$

$$\text{ở đây: } \omega_n = \sqrt{\frac{K}{\tau_2}}; \zeta = \frac{\omega_n(\tau_1 + 1/K)}{2}$$

trong đó:  $\omega_n$  là tần số tự nhiên và  $\zeta$  là hệ số tắt dần (*damping factor*)

**Minh hoạ: [Vòng khoá pha PLL bậc 1]**

Giả sử, cho  $G(s) = \frac{1+0,01s}{1+s}$  và  $K=1$ , xác định và vẽ đáp ứng của PLL đối với sự thay đổi đột biến bằng 1 đối với pha đầu vào. ở đây  $\tau_1 = 0,01$  và  $\tau_2 = 1$  vì vậy  $\omega_n = 1$ ,  $\xi = 0,505$

Kết quả nhận được.  $H(s) = \frac{0,01s+1}{s^2+1,01s+1} \Rightarrow$  Đáp ứng với  $\phi(t) = u(t)$ , nghĩa là  $\Phi(s) = 1/s$  được cho bởi.  $\hat{\Phi}(s) = \frac{0,01s+1}{s^3+1,01s^2+s+1}$

Để xác định và vẽ được đáp ứng theo gian  $\hat{\phi}(t)$  đối với đầu vào là xung đơn vị  $u(t)$ , ta phải xác định đầu ra hệ thống có hàm truyền đạt  $H(s)$  đối với đầu vào  $u(t)$ . Thực hiện một cách đơn giản nhất bằng cách dùng các *kỹ thuật không gian trạng thái*. Dùng hàm Matlab **tf2ss.m**, nhận được mô hình không gian trạng thái hệ thống được mô tả bởi hàm truyền đạt của nó. Sau khi tìm được cách biểu diễn không gian trạng thái hệ thống đó, tìm được đáp ứng từng bước một.

Hàm **tf2ss.m** thực hiện lấy tử và mẫu số của hàm truyền đạt  $H(s)$  và trả lại các giá trị A, B, C, D, biểu diễn không gian trạng thái của nó dưới dạng.

$$\begin{cases} \frac{d}{dt} x(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$

Sự biểu diễn này có thể được xấp xỉ bởi

$$\begin{cases} x(t + \Delta t) = x(t) + Ax(t)\Delta t + Bu(t)\Delta t \\ y(t) = Cx(t) + Du(t) \end{cases}$$

Hoặc tương đương với.

$$\begin{cases} x(i+1) = x(i) + Ax(i)\Delta t + Bu(i)\Delta t \\ y(i) = Cx(i) + Du(i) \end{cases}$$

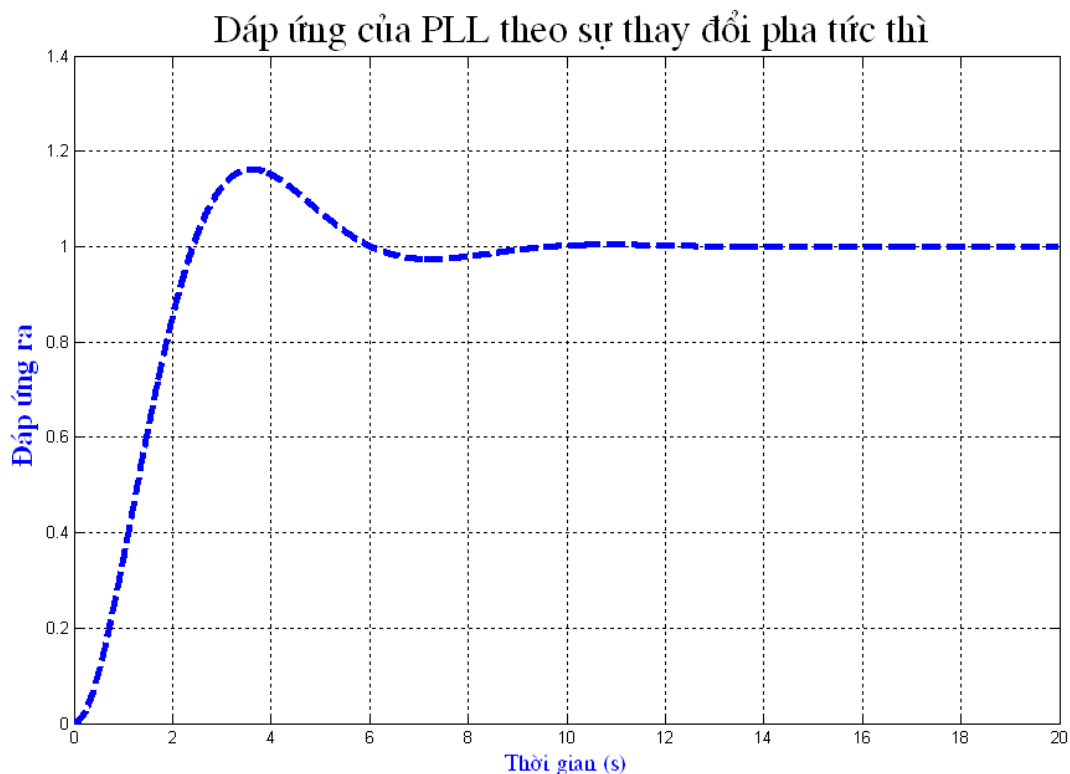
Trong bài toán này, nó phù hợp với việc chọn  $u(t)$  là hàm bước nhảy (step function) và các vector tử số & mẫu số của  $H(s)$  là  $[0,01 \ 1]$  và  $[1 \ 1,01 \ 1]$  tương ứng. Với cách chọn các vector tử số và mẫu số này, thì các tham số không gian trạng thái của hệ thống sẽ là.

$$A = \begin{bmatrix} -1,01 & -1 \\ 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad C = [0,01 \ 1] \quad D = 0$$

Vẽ đầu ra của PLL được cho ở hình 4 (kết quả chạy chương trình Matlab). Như được thấy ở hình 4, đầu ra của PLL cuối cùng bám theo đầu vào, tuy nhiên tốc độ bám của nó phụ thuộc vào các tham số của bộ lọc vòng và tham số K, các hằng số tỉ lệ VCO.

```
function y = NVD_PLL_1
num      = [0.01 1];
den      = [1 1.01 1];
[a,b,c,d] = tf2ss(num,den);
dt       = 0.01;
u        = ones(1,2000);
x        = zeros(2,2001);
for i=1:2000
    x(:,i+1) = x(:,i) + dt.*a*x(:,i) + dt.*b*u(i);
    y(i)     = c*x(:,i);
end
t         = [0:dt:20];
plot(t(1:2000),y)
```





**Hình 4:** Đáp ứng của PLL đối với sự thay đổi tức thì về pha.

#### Bài 44:

Hãy đọc mã chương trình Matlab **NVD\_CS710.m** và chạy chương trình để làm sáng tỏ nguyên lý hoạt động của bộ mở cổng sớm muộn trong việc khôi phục định thời kỳ hiệu.

#### Hướng dẫn giải

Máy thu tối ưu dùng các bộ lọc thích hợp và các bộ lấy mẫu tại đầu ra các bộ lọc thích hợp để quyết định tín hiệu thu. Trong mọi trường hợp, ta thường giả thiết, máy thu hoàn toàn biết trước thời điểm lấy mẫu và thực hiện lấy mẫu chính xác tại thời điểm đó. Các hệ thống mà đạt được sự đồng bộ giữa máy thu và máy phát được gọi là khôi phục định thời, đồng bộ đồng hồ hay đồng bộ ký hiệu.

Thực hiện đơn giản về đồng bộ đồng hồ là dùng cổng sớm muộn “**Early-Late Gate**”. Hoạt động của cổng sớm muộn này dựa trên cơ sở hệ thống truyền thông PAM, đầu ra của bộ lọc thích hợp là hàm tự tương quan của tín hiệu xung cơ bản được dùng trong hệ thống PAM đó (có thể bị dịch thời). Hàm tự tương quan được cực đại hoá tại thời điểm lấy mẫu tối ưu (giá trị hàm tự tương quan lớn nhất tại thời điểm lấy mẫu tối ưu) và có tính đối xứng (đối xứng qua giá trị cực đại). Nghĩa là, khi không có tạp âm, thì đầu ra của bộ lấy mẫu tại các thời điểm lấy mẫu  $T^+ = T + \delta$  và  $T^- = T - \delta$  có giá trị bằng nhau, nghĩa là.

$$y(T^+) = y(T^-)$$

Khi này, thấy rõ thời điểm lấy mẫu tối ưu  $t=T$  là thời điểm giữa của hai thời điểm lấy mẫu sớm và muộn, tức là

$$T = \frac{T^+ + T^-}{2}$$

Giả sử, không lấy mẫu tại thời điểm lấy mẫu tối ưu  $t=T$  nhưng lấy mẫu tại thời điểm  $T_1$ . Nếu lấy hai mẫu tại  $T^+ = T_1 + \delta$  và  $T^- = T_1 - \delta$ , thì các mẫu này không đối xứng đối nhau qua thời điểm lấy mẫu tối ưu  $t=T \Rightarrow$  vì vậy chúng không bằng nhau.

Hàm tự tương quan điển hình đối với các xung đến âm và dương và ba mẫu được cho ở hình 1 dưới đây. Ở đây

$$T^- = T - \delta_1$$

$$T^+ = T + \delta_2$$

Trong đó  $\delta_1 < \delta_2$  và từ hình vẽ cho thấy

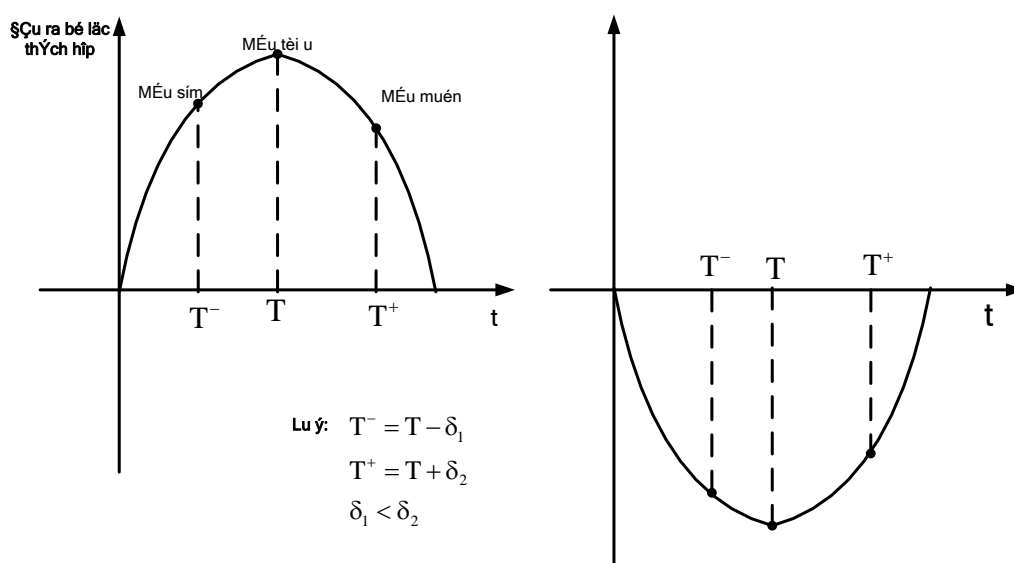
$$|y(T^-)| > |y(T^+)|$$

Trong trường hợp này:  $T < T_1 = \frac{T^+ + T^-}{2}$

**Kết luận:** Dựa vào tính chất đối xứng tín hiệu ra của bộ tương quan (lọc thích hợp) ta có:

- ✓ **Khi  $|y(T^-)| > |y(T^+)|$ , thì thời điểm lấy mẫu chính xác là thời điểm lấy mẫu sớm hơn thời điểm lấy mẫu được giả định  $\Rightarrow$  lấy mẫu nên được thực hiện sớm hơn.**
- ✓ **Khi  $|y(T^-)| < |y(T^+)|$ , thì thời điểm lấy mẫu chính xác là thời điểm lấy mẫu muộn hơn thời điểm lấy mẫu được giả định  $\Rightarrow$  thời điểm lấy mẫu nên được trễ đi.**
- ✓ **Khi  $|y(T^-)| = |y(T^+)|$ , thì thời điểm lấy mẫu là chính xác  $\Rightarrow$  không cần hiệu chỉnh.**

**Vì vậy:** Nguyên tắc hoạt động hệ thống đồng bộ định thời **mở cổng sớm-muộn:** Thực hiện lấy ba mẫu tín hiệu ra của bộ tương quan tại các thời điểm  $T_1$ ,  $T = T_1 - \delta$  và  $T^+ = T_1 + \delta$ , sau đó so sánh giá trị  $|y(T^-)|$  và  $|y(T^+)|$ , tùy thuộc vào các giá trị tương đối của chúng, tạo ra tín hiệu điều khiển để hiệu chuẩn thời điểm lấy mẫu.



**Hình 1:** Đầu ra bộ lọc thích hợp và các mẫu sớm muộn

**Minh họa: Đồng bộ đồng hồ**

Hệ thống truyền thông PAM cơ hai sử dụng dạng sóng dốc Cosine có hệ số dốc  $\alpha = 0,4$ . Tốc độ truyền dẫn hệ thống là 4800bit/s. Chương trình Matlab mô phỏng hoạt động **mở cổng sớm-muộn** cho hệ thống này được cho ở dưới đây. Vì tốc độ truyền dẫn là 4800bit/s, nên  $T=1/R_b=1/4800$ . Với hệ số dốc  $\alpha = 0,4$ , khai triển sóng dốc Cosine trở thành.

$$x(t) = \text{SinC}(4800t) \frac{\text{Cos}(4800 \times 0,4 \times \pi \times t)}{1 - 4 \times 0,16 \times 4800^2 \times t^2}$$

$$= \text{SinC}(4800t) \frac{\text{Cos}(1920\pi t)}{1 - 1,4746 \times 10^7 \times t^2}$$

Rõ ràng tín hiệu này được mở rộng từ  $-\infty$  đến  $+\infty$  được cho ở hình 6 (*kết quả chạy chương trình*). Từ hình vẽ ta thấy rằng, với  $\forall$  mục đích thực tế là đủ để ta xét trong khoảng  $|t| \leq 0,6 \times 10^{-3}$ , một cách xấp xỉ  $[-3T, 3T]$ . Việc xử lý cắt xén (Truncating) xung dốc Cosine này trong khoảng này và tính hàm tự tương quan mang lại kết quả dạng sóng được cho ở hình 7 (*kết quả chạy chương trình*).

Chương trình Matlab được viết để tính và vẽ hàm tự tương quan và dạng sóng dốc Cosine được thực hiện dưới đây (dạng tổng quát). đối với ví dụ cụ thể ta truyền tham biến cho hàm cụ thể. Độ dài của hàm tự tương quan là: 1201 giá trị cực đại (tức là: thời điểm lấy mẫu tối ưu) xuất hiện ở tại thành phần thứ 600.

Hãy kiểm tra cho hai trường hợp:

✓ Khi thời điểm lấy mẫu không đúng tại: 700.

✓ Khi thời điểm lấy mẫu không đúng tại: 500.

Trong cả hai trường hợp thì công "sớm muộn" thực hiện hiệu chỉnh thời điểm lấy mẫu về thời điểm lấy mẫu tối ưu 600.

```
function y = NVD_Clock_SYN
alpha      = 0.4;
T           = 1/4800;
t           = [-3*T:1.001*T/100:3*T];
x           = sinc(t./T).*(cos(pi*alpha*t./T)./(1-4*alpha^2*t.^2/T^2));

pause      % Press any key to see a plot of x(t).

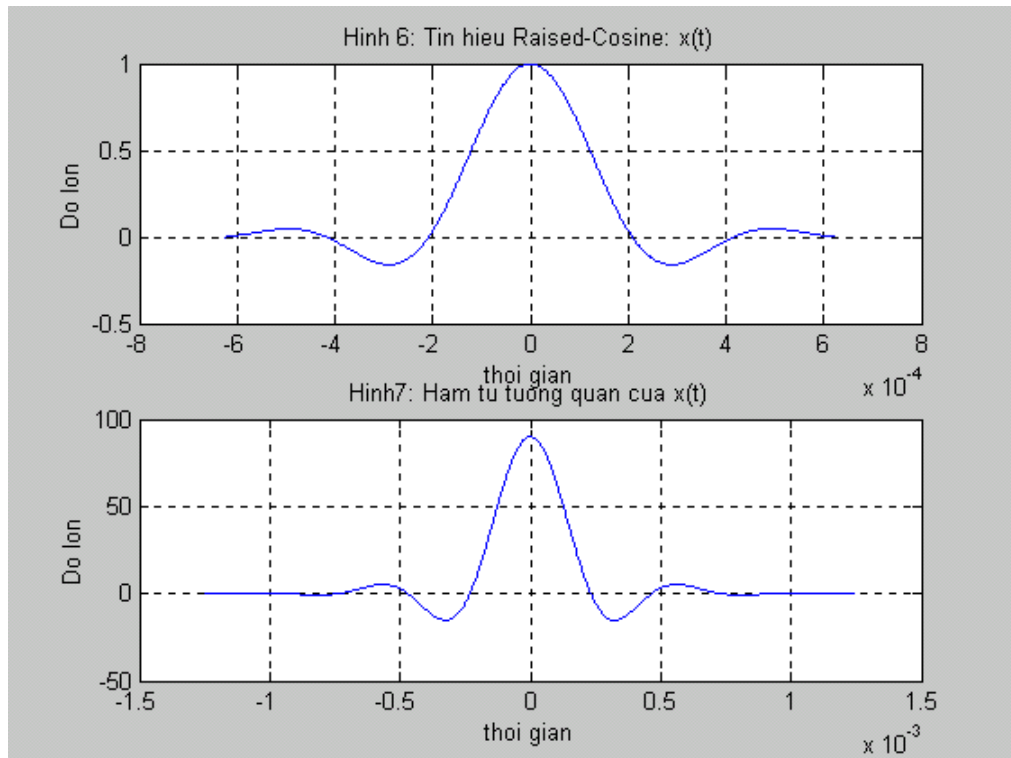
plot(t,x)

y          = xcorr(x);
ty         = [t-3*T,t(2:length(t))+3*T];

pause % Press any key to see a plot of the autocorrelation of x(t).
plot(ty,y);

d          = 60;          % Early and late advance and delay
ee         = 0.01;        % Precision
e          = 1;           % Step size
n          = 700;         % The incorrect sampling time
while abs(abs(y(n+d))-abs(y(n-d)))>=ee
    if abs(y(n+d))-abs(y(n-d))>0
        n = n+e;
    elseif abs(y(n+d))-abs(y(n-d))<0
        n = n-e;
    end
end
pause % Press any key to see the corrected sampling time
```

```
n
n = 500; % Another incorrect sampling time
while abs(abs(y(n+d))-abs(y(n-d)))>=ee
    if abs(y(n+d))-abs(y(n-d))>0
        n = n+e;
    elseif abs(y(n+d))-abs(y(n-d))<0
        n = n-e;
    end
end
end
pause % Press any key to see the corrected sampling time
n
```



Khi nhập các giá trị cho các thời điểm lấy mẫu sai (các giá trị 500 và 700) thì hàm thực hiện hiệu chỉnh về giá trị thời điểm lấy mẫu tối ưu đúng (600).