

## CHƯƠNG 4

### Bài 1:

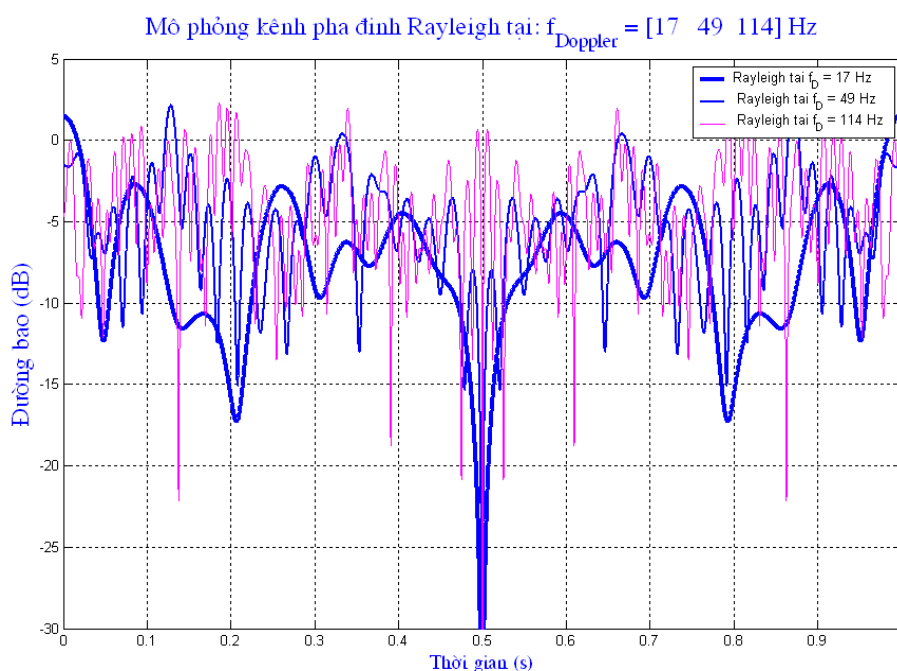
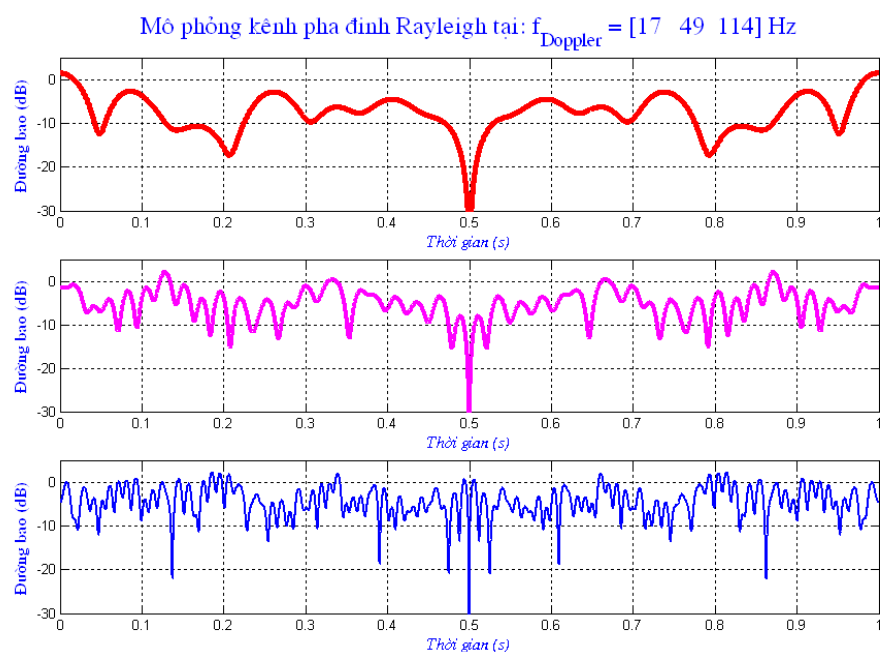
Hãy đọc mã chương trình mô phỏng **NVD\_Rayleigh\_Fading\_Sim** thực hiện mô phỏng kênh pha đình Reyleigh, phân tích và nhập tham số đầu vào, chạy chương trình, phân tích kết quả. Vẽ mô hình mô phỏng, vẽ lưu đồ thuật toán.

### Hướng dẫn giải:

Đọc kỹ phần phụ lục 4A, hiểu mô hình toán.

Chạy chương trình theo từng bước, thay đổi các tham số đầu vào cho chương trình, phân tích ảnh hưởng của các tham số đầu vào lên kết quả mô phỏng

Vẽ lưu đồ thuật toán.



```
function [h_t] = NVD_Rayleigh_Fading_Sim()
%=====
%=====
% Inputs
% f_D : [Hz] Doppler frequency
% t : simulation time interval length, time interval [0,t]
% f_s : [Hz] sampling frequency, set to 1000 if smaller.
% Outputs
% Ts : [Sec][1xN double] time instances for the Rayleigh signal
% z_dB : [dB] [1xN double] Rayleigh fading signal

%%%=====
%%%      Programmed by Nguyen Viet Dam PTIT
%%%      Date: 25-12-2007
%%%=====

B          = [1.4 5.6 11.2 22.4]*1e6;
FFTsize    = [128 512 1024 2048];
ScaleB1     = B./FFTsize;          % SubcarrierSpace=10.938e3 (fixed);
OfdmSize    = [128 512 1024 2048]+(1/8)*[128 512 1024 2048];
f_s         = ScaleB1(1);
%=====
fc          = 3.5e9;                % RF Carrier
v           = [5 15 35];           % Speed Km/h
f_D         = ceil((v/3.6)*(fc/3e8)); % Doppler Frequency Vector
t           = 1;                   % Time Simulation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Required parameters
if f_s < 1000,
    f_s      = 1000;
end
% [Hz} Min. required sampling rate
N           = ceil(t*f_s); % Number of samples
if mod(N,2) == 1,
    N        = N + 1;
end
Ts          = linspace(0,t,N);
% Use even number of samples
f           = linspace(-f_s,f_s,N);
% Generate I & Q complex Gaussian samples in frequency domain
Gfi_p       = randn(2,N/2);
Gfq_p       = randn(2,N/2);
CGfi_p      = Gfi_p(1,:)+ i*Gfi_p(2,:);
CGfq_p      = Gfq_p(1,:)+ i*Gfq_p(2,:);

CGfi        = [fliplr(CGfi_p) CGfi_p];
CGfq        = [fliplr(CGfq_p) CGfq_p];

% Generate fading spectrum for shaping Gaussian line spectra
P_r         = 1; % normalize average received envelope to 0dB
% S_r       = P_r/(4*pi)./(f_D*sqrt(1-(f/f_D).^2)); %Doppler spectra

for k=1:length(f_D)
    S_r(k,:) = P_r/(4*pi)./(f_D(k)*sqrt(1-(f/f_D(k)).^2)); %Doppler spectra
    % Set samples outside the Doppler frequency range to 0
    idx1     = find(f>f_D(k));
    idx2     = find(f<-f_D(k));
    S_r(k,idx1) = 0;
    S_r(k,idx2) = 0;
    % Generate r_I(t) and r+Q(t) using inverse FFT:
    r_I(k,:)  = N*ifft(CGfi.*sqrt(S_r(k,:)));
    r_Q(k,:)  = -i*N*ifft(CGfq.*sqrt(S_r(k,:)));

    % Finally, generate the Rayleigh distributed signal envelope
```

```
z(k,:) = sqrt(abs(r_I(k,:)).^2+abs(r_Q(k,:)).^2);
z_dB(k,:) = 20*log10(z(k,:));
z_dB(k,:) = z_dB(k,1:length(Ts)); % Return correct number of points
h_t(k,:) = r_I(k,:) + r_Q(k,:);
end

PLOT = 1;
if PLOT==1
    h1 = figure(1);
    set(h1,'color','g');
    set(h1,'Name','Simulation for Rayleigh fading Channel: Programmed by Nguyen Viet Dam PTIT');

    subplot(311)
    plot(Ts,z_dB(1,:),'LineWidth',[1.5],'color','r');
    xlabel('Thời gian (s)','FontName','.VnTime','color','b','FontSize',8);
    ylabel('$-êng \hspace{10cm} bao (dB)','FontName','.VnTime','color','b','FontSize',12);
    axis([min(Ts) max(Ts) -30 5]);
    title(['M« pháng kªnh pha ®inh Rayleigh tªi: f_D_o_p_p_l_e_r = ',num2str(f_D),' Hz',...
    ],'FontName','.VnTime','color','b','FontSize',14);
    grid on;

    subplot(312)
    plot(Ts,z_dB(2,:),'LineWidth',[1.5],'color','m');
    xlabel('Thời gian (s)','FontName','.VnTime','color','b','FontSize',8);
    ylabel('$-êng \hspace{10cm} bao (dB)','FontName','.VnTime','color','b','FontSize',12);
    axis([min(Ts) max(Ts) -30 5]);
    grid on;

    subplot(313)
    plot(Ts,z_dB(3,:),'LineWidth',[1.3],'color','b');
    xlabel('Thời gian (s)','FontName','.VnTime','color','b','FontSize',8);
    ylabel('$-êng \hspace{10cm} bao (dB)','FontName','.VnTime','color','b','FontSize',12);
    axis([min(Ts) max(Ts) -30 5]);
    grid on;
%=====
    h2 = figure(2);
    set(h2,'color','g');
    set(h2,'Name','Simulation for Rayleigh fading Channel: Programmed by Nguyen Viet Dam PTIT');
    plot(Ts,z_dB(1,:),'LineWidth',[2.5],'color','b'); hold on;
    plot(Ts,z_dB(2,:),'LineWidth',[1],'color','b'); hold on;
    plot(Ts,z_dB(3,:),'LineWidth',[1],'color','m'); hold on;
    xlabel('Thời gian (s)','FontName','.VnTime','color','b','FontSize',8);
    ylabel('$-êng \hspace{10cm} bao (dB)','FontName','.VnTime','color','b','FontSize',12);
    axis([min(Ts) max(Ts) -30 5]);
    title(['M« pháng kªnh pha ®inh Rayleigh tªi: f_D_o_p_p_l_e_r = ',num2str(f_D),' Hz',...
    ],'FontName','.VnTime','color','b','FontSize',14);
    legend('Rayleigh tại f_D = 17 Hz',' Rayleigh tại f_D = 49 Hz',' Rayleigh tại f_D = 114 Hz');
    grid on;
end
```

## CHƯƠNG 5

### Bài 1:

Hãy so sánh xác suất lỗi bản tin khi dùng và không dùng mã hóa kênh sửa lỗi. Nếu dùng điều chế BPSK, tạp âm Gauss,  $P_r/N_0 = 43,776$ , tốc độ dữ liệu  $R_b = 4800$ bps. Trường hợp dùng mã hóa kênh, giả thiết dùng mã  $C(15,11)$  có khả năng sửa được một lỗi trong khối 15bits.

#### Hướng dẫn giải:

Với BPSK,  $P_u = Q(\sqrt{2E_b/N_0})$  là xác suất lỗi ký hiệu kênh không được mã hóa và  $P_c = Q(\sqrt{2E_c/N_0})$  là xác suất lỗi ký hiệu kênh được mã hóa, trong đó  $E_b/N_0$  là tỷ số năng lượng bit trên mật độ phổ tập âm và  $E_c/N_0$  là tỷ số năng lượng bit mã trên mật độ phổ tập âm.

#### ❖ Trường hợp không dùng mã hóa kênh:

$$\begin{aligned} \frac{E_b}{N_0} &= \frac{P_r}{N_0} \left( \frac{1}{R} \right) = 9,12 \quad (9,6\text{dB}) \\ P_u &= Q\left(\sqrt{\frac{2E_b}{N_0}}\right) = Q(\sqrt{18,24}) = 1,02 \times 10^{-5} \end{aligned} \quad (1)$$

$Q(x)$  được tính xấp xỉ theo:

$$Q(x) \approx \frac{1}{x\sqrt{2\pi}} \exp\left(\frac{-x^2}{2}\right), \text{ với } x > 3$$

Xác suất lỗi khối bản tin không được mã hóa  $P_M^u$

$$\begin{aligned} P_M^u &= 1 - (1 - p_u)^k \\ &= 1 - \underbrace{(1 - p_u)^{11}}_{\substack{\text{xác suất mà toàn bộ} \\ \text{11 bit trong khối} \\ \text{không được mã hóa} \\ \text{là đúng}}} = \underbrace{1,12 \times 10^{-4}}_{\substack{\text{xác suất mà ít nhất} \\ \text{có 1 bit trong số} \\ \text{11 bit bị lỗi}}} \end{aligned} \quad (2)$$

#### ❖ Trường hợp sử dụng mã hóa kênh:

Tốc độ ký hiệu kênh hoặc tốc độ bit mã  $R_c$  gấp 15/11 lần tốc độ dữ liệu  $R_b$ :

$$\begin{aligned} R_c &= 4800 \times \frac{15}{11} \approx 6545 \quad \text{bps} \\ \frac{E_c}{N_0} &= \frac{P_r}{N_0} \left( \frac{1}{R} \right) = 6,69 \quad (8,3 \text{ dB}) \end{aligned}$$

Tỷ số  $\frac{E_c}{N_0}$  cho mỗi bit mã nhỏ hơn  $E_b/N_0$  cho mỗi bit dữ liệu do tốc độ bit kênh đã được tăng lên, nhưng công suất phát được giả thiết là không đổi:

$$P_c = Q\left(\sqrt{\frac{2E_c}{N_0}}\right) = Q(\sqrt{13,38}) = 1,36 \times 10^{-4} \quad (3)$$

Khi so sánh các kết quả (1) với (3), do đưa bit dư vào, nên xác suất lỗi bit kênh bị suy thoái (tần lên). Trong cùng một khoảng thời gian với cùng một công suất khả dụng phải phát hiện nhiều bit hơn, vì vậy cải thiện hiệu năng chất lượng do mã hóa kênh không thể hiện ở đây.

Tỉ lệ lỗi bản tin được mã hóa  $P_M^c$  được tính là:

$$P_M^c = \sum_{j=2}^{n=15} \binom{15}{j} (p_c)^j (1-p_c)^{15-j}$$

Tổng trên bắt đầu tại  $j = 2$ , vì mã sửa được một lỗi trong mỗi khối  $n = 15$  bits. Lấy xấp xỉ lấy kết quả tại giá trị của số hạng đầu tiên. Với giá trị của  $p_c$  được tính ở (3) thì

$$P_M^c \approx \binom{15}{2} (p_c)^2 (1-p_c)^{13} = 1,94 \times 10^{-6} \quad (4)$$

Từ các kết quả ở (2) và (4), cho thấy xác suất lỗi bản tin đã được cải thiện tới 58 lần (có được từ việc dùng mã sửa lỗi). Khi đưa thêm bit dư vào khối bản tin ban đầu (tốc độ lớn hơn), năng lượng ký hiệu trên kênh (bit mã hóa) nhỏ hơn; lỗi ở đầu ra bộ giải điều chế nhiều hơn. Lợi ích từ mã kênh (tại một giá trị tỷ số  $\frac{E_b}{N_0}$  thích hợp) sẽ bù lại và được tăng lên.

## Bài 2:

Một bản tin 3 bit được truyền trên hệ thống BPSK và tỷ số tín hiệu trên tạp âm thu được bằng 7 dB.

- Tính xác suất 2 bit mắc lỗi
- Bản tin được mã hóa sao cho từ mã tăng lên 5 bit. Xác suất 2 bit mắc lỗi bằng bao nhiêu. Giả thiết rằng công suất phát trong hai trường hợp a) và b) là như nhau.

### Hướng dẫn giải:

a) Xác suất 2 bit mắc lỗi trong trường hợp không dùng mã hóa kênh:

$$\frac{E_b}{N_0} = 10^{7/10} = 5 \Rightarrow \sqrt{\frac{2E_b}{N_0}} = \sqrt{2 \times 5} = 3,16.$$

Tra bảng hàm  $Q(x)$  trong phụ lục ta được  $P_b = 7,9 \cdot 10^{-4}$ .

Đối với ba bit mắc 2 lỗi ta được số tổ hợp lỗi hai bit lỗi trong 3 bit như sau:

$$\binom{3}{2} = \frac{3!}{2!(3-2)!} = 3.$$

Xác suất lỗi 2 bit như sau:

$$p_e = \binom{3}{2} (P_b)^2 (1-P_b)^{3-2} = 3 \left(7,9 \times 10^{-4}\right)^2 \left(1 - 7,9 \times 10^{-4}\right) \approx 6,4 \cdot 10^{-7}$$

trong đó  $P_b$ ,  $P_b^2$  và  $(1-P_b)^{3-2}$  là xác suất lỗi một bit, xác suất lỗi 2 bit và xác suất một bit không mắc lỗi.

b) Xác suất 2 bit mắc lỗi trong trường hợp dùng mã hóa kênh:

Tốc độ bit sau mã hóa tăng và bằng:  $R_{bc}=R_b/r=(n/k)R_b$ , trong đó  $R_b$  là tốc độ bit không mã hóa và  $r=3/5$  là tỷ lệ mã. Năng lượng bit cho trường hợp mã hoá bằng:

$$E_{bc} = \frac{P}{R_{bc}} = \frac{k}{n} \frac{P}{R_b} = rE_b$$

$$\sqrt{\frac{2E_{bc}}{N_0}} = \sqrt{\frac{2rE_b}{N_0}} = \sqrt{2 \frac{E_b}{N_0} \left(\frac{k}{n}\right)} = \sqrt{10 \times 3/5} = 2,45$$

Tra bảng trong phụ lục ta được xác suất lỗi bit khi mã hóa bằng:  $P_b^c = 7,14.10^{-3}$

### Bài 3:

Tìm xác suất lỗi bản tin cho hệ thống truyền dẫn trong bài 2 cho:

a) Trường hợp không mã hóa  $P_M^u$

b) Trường hợp mã hóa  $P_M^c$

### Hướng dẫn giải:

a) Xác suất lỗi bản tin là xác suất ít nhất có một trong số 3 bit mắc lỗi:

$$P_M^u = 1 - \underbrace{(1 - P_b)^3}_{\text{Xác suất cả 3 bit không bị lỗi}} = 1 - (1 - 7,9 \times 10^{-4})^3 = 2,37 \times 10^{-3}$$

b) Xác suất lỗi bản tin trong trường hợp dùng mã hóa kênh:

Vì khoảng cách Hamming cực tiểu  $d_{\min}=3$ , nên mã có thể sửa được một bit. Bản tin sẽ **chỉ** bị lỗi khi khi mắc lỗi 2,3,4,5 bit, vì thế:

$$P_M^c = \sum_{j=2}^n \binom{n}{j} (P_{bc})^j (1 - P_{bc})^{n-j}$$

$$= \sum_{j=2}^5 \binom{5}{j} (P_{bc})^j (1 - P_{bc})^{5-j}$$

Hay một cách gần đúng nếu ta chỉ xét thành phần thứ nhất:

$$P_M^c \approx \binom{5}{2} (P_{bc})^2 (1 - P_{bc})^3 = \binom{5}{2} (7,14.10^{-3})^2 (1 - 7,14.10^{-3})^3 = 4,99.10^{-4}$$

Kết quả cho thấy, nhờ mã hóa xác suất lỗi bản tin được cải thiện:

$$(2,37.10^{-3}) / (4,99.10^{-4}) = 4,75 \text{ lần}$$

### Bài 4 a:

Bảng thông cho hệ thống được mã hóa trong bài 2 tăng lên bao nhiêu lần so với hệ thống không mã hóa.

### Hướng dẫn giải:

Băng thông Nyquist cho hệ thống không mã hóa là:  $B_N = (1 + \alpha)R_b$ , trong đó  $\alpha$  là hệ số dắc bộ lọc.

Băng thông Nyquist cho hệ thống mã hóa là:  $B_N = (1 + \alpha)R_{bc} \times \frac{1}{r} = \frac{n}{k}(1 + \alpha)R_b$ , trong đó  $r$  là tỷ lệ mã. Vậy, băng thông hệ thống mã hóa tăng so với hệ thống không mã hóa là:

$$\frac{R_{bc}}{(R_{bc} \times r)} = \frac{1}{r} = \frac{n}{k} = \frac{5}{3} \text{ lần}$$

#### Bài 4 b:

Một hệ thống điều chế BPSK có tốc độ bit  $R_b = 4800\text{bps}$ . Tỷ số tín hiệu trên tạp âm thu:  $E_b/N_0 = 8\text{dB}$ .

- Tìm xác suất lỗi bit  $P_b$  và xác suất lỗi bản tin  $P_M^u$  cho hệ thống không mã hóa, trong đó bản tin dài 11 bit
- Tìm xác suất lỗi bit mã hóa  $P_b^c$  và xác suất lỗi bản tin được mã hóa  $P_M^c$  cho hệ thống dùng mã khối (15,11) sửa được lỗi đơn ( $t=1$ )

#### Hướng dẫn giải:

- Trường hợp hệ thống không dùng mã hóa kênh

$$\sqrt{\frac{2E_b}{N_0}} = \sqrt{2 \times 10^{0,8}} = 3,55 \Rightarrow P_b = 1,9 \cdot 10^{-4}$$

$$P_M^u = 1 - (1 - P_b)^{11} = 1 - (1 - 1,9 \cdot 10^{-4})^{11} = 2,1 \cdot 10^{-3}$$

- Trường hợp hệ thống dùng mã khối (15,11):

$$\sqrt{\frac{2E_b}{N_0} r} = \sqrt{2 \times 10^{0,8} \times \frac{11}{15}} = 3,04 \Rightarrow P_b^c = 1,18 \cdot 10^{-3}$$

$$P_M^c = \sum_{j=t+1}^{15} \binom{15}{j} (P_b^c)^j (1 - P_b^c)^{15-j}$$

$$\approx \binom{15}{2} (1,18 \cdot 10^{-3})^2 (1 - 1,18 \cdot 10^{-3})^{15-2} = 1,44 \cdot 10^{-4}$$

Hệ số cải thiện hiệu năng chất lượng là:  $\frac{P_M^u}{P_M^c} = \frac{2,1 \cdot 10^{-3}}{1,44 \cdot 10^{-4}} = 14,58 \text{ lần}$

#### Bài 5:

Thiết kế một mã chẵn lẻ đơn  $(n, k)$  để phát hiện các mẫu lỗi 1, 3, 5, 7 trong một khối dữ liệu. Xác định các giá trị của  $n$  và  $k$ , và tìm xác suất lỗi khối không thể phát hiện được nếu xác suất lỗi ký hiệu kênh là  $10^{-2}$ .

#### Hướng dẫn giải:

$$(n, k) = (8, 7)$$

$$\begin{aligned}
 P_{\text{NonDetect}} &= \sum_{j=1}^{\frac{n}{2} \text{ (khi } n \text{ chẵn)}} \binom{n}{2j} p^{2j} (1-p)^{n-2j} \\
 P_{\text{nd}} &= \underbrace{\binom{8}{2} p^2 (1-p)^6}_{j=1} + \underbrace{\binom{8}{4} p^4 (1-p)^4}_{j=3} + \underbrace{\binom{8}{6} p^6 (1-p)^2}_{j=5} + \underbrace{\binom{8}{8} p^8}_{j=7} \\
 &= 28(10^{-2})^2 (1-10^{-2})^6 + 70(10^{-2})^4 (1-10^{-2})^4 + 28(10^{-2})^6 (1-10^{-2})^2 + (10^{-2})^8 \\
 &= 2,6 \times 10^{-3}
 \end{aligned}$$

### Bài 6:

Tính xác suất lỗi bản tin cho một chuỗi dữ liệu 12 bit được mã hóa bởi mã khối tuyến tính (24,12). Giả thiết: (i) mã này sửa được các mẫu lỗi 1 bit và 2 bit và nó không sửa được các mẫu lỗi nhiều hơn 2 lỗi; (ii) xác suất lỗi ký hiệu kênh là  $10^{-3}$ .

#### Hướng dẫn giải:

$$\begin{aligned}
 P_M &= \sum_{j=t+1}^n \binom{n}{j} p^j (1-p)^{n-j} \\
 P_M &= \sum_{k=3}^{24} \binom{24}{k} p^k (1-p)^{24-k} \\
 &\cong \binom{24}{3} (10^{-3})^3 (1-10^{-3})^{21} = 1,98 \times 10^{-6}
 \end{aligned}$$

### Bài 7:

Mã khối tuyến tính (127,92) có khả năng sửa ba lỗi (t=3).

- Tìm xác suất lỗi bản tin cho khối dữ liệu 92 bit chưa được mã hóa nếu xác suất lỗi ký hiệu kênh là  $10^{-3}$
- Tìm xác suất lỗi bản tin khi sử dụng mã khối (127, 92) nếu xác suất lỗi ký hiệu kênh là  $10^{-3}$

#### Hướng dẫn giải:

- Xác suất lỗi bản tin cho khối dữ liệu 92 bit chưa được mã hóa nếu xác suất lỗi ký hiệu kênh là  $10^{-3}$ .

$$P_M^{\text{uncoded}} = 1 - (1 - 10^{-3})^{92} = 8,8 \times 10^{-2}$$

- Xác suất lỗi bản tin dùng mã khối (127,92) khi xác suất lỗi ký hiệu kênh là  $10^{-3}$

$$\begin{aligned}
 P_M^{\text{coded}} &= \sum_{k=4}^{127} \binom{127}{k} p^k (1-p)^{127-k} \\
 &\cong \binom{127}{4} (10^{-3})^4 (1-10^{-3})^{123} \\
 &= 9,14 \times 10^{-6}
 \end{aligned}$$



**Bài 8:**

Tính sự cải thiện về xác suất lỗi bản tin so với trường hợp không dùng mã khối tuyến tính (24,12) sửa được 2 lỗi khi dùng điều chế BPSK nhất quán và  $E_b/N_0$  thu là 10 dB.

**Hướng dẫn giải:**

$$P_M = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) = Q\sqrt{2 \times 10} = Q(4,47)$$

$$\cong \frac{1}{x\sqrt{2\pi}} e^{-x^2/2} = \frac{1}{4,47\sqrt{2\pi}} e^{-10} = 4,05 \times 10^{-6}$$

$$P_M^u = 1 - (1 - 4,05 \times 10^{-6})^{12} = 4,86 \times 10^{-5}$$

Với mã (24,12), tỉ lệ mã hóa là 1/2. Do đó, tốc độ dữ liệu là gấp đôi tốc độ dữ liệu khi không mã hóa kênh, hoặc  $E_c/N_0$  nhỏ hơn  $E_b/N_0$  là 3dB

$$E_c/N_0 = 7\text{dB} = \underbrace{10^{\frac{7}{10}}}_{\text{Note}} = 5,01$$

$$p_c = Q\left(\sqrt{\frac{2E_c}{N_0}}\right) = Q(\sqrt{2 \times 5,01}) = Q(3,16)$$

tra bảng hàm  $Q(x) \Rightarrow p_c = 0,0008$

$$P_M^c = \sum_{k=3}^{24} \binom{24}{k} p^k (1-p)^{24-k} \cong \binom{24}{3} (0,0008)^3 (1-0,0008)^{21}$$

$$P_M^c \cong 1,02 \times 10^{-6}$$

$$\text{Cải thiện hiệu năng chất lượng} = \frac{4,86 \times 10^{-5}}{1,02 \times 10^{-6}} = 47,6$$

**Bài 9:**

Cho khoảng cách nhỏ nhất đối với một mã khối tuyến tính là 11. Tìm khả năng phát hiện và sửa lỗi tối đa.

**Hướng dẫn giải:**

$$d_{\min} = 11$$

$$\text{Sửa lỗi: } t_{\text{cor}} = \frac{d_{\min} - 1}{2} = 5$$

$$\text{Phát hiện lỗi: } t_{\text{detect}} = d_{\min} - 1 = 10$$

**Bài 10:**

Cho một bộ tạo mã khối tuyến tính có ma trận tạo mã sau:

$$G = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

a) Tìm các từ mã

b) Tìm Syndrome trong trường hợp từ mã phát là "111" còn từ mã thu là "110"

### Hướng dẫn giải:

a) Tìm các từ mã

$$\mathbf{c} = \mathbf{mG} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

b) Tìm Syndrome trong trường hợp từ mã phát là "111" còn từ mã thu là "110"

$$\mathbf{H} = [\mathbf{I}_{n-k} : \mathbf{P}^T] = [\mathbf{I}_3 : \mathbf{P}^T]; \mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \mathbf{P}^T = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \rightarrow \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$\rightarrow \mathbf{H}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

$$\mathbf{s} = \mathbf{yH}^T = [0 \ 0 \ 0 \ 1 \ 1 \ 0] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} = [1 \ 1 \ 0]$$

### Bài 11:

Một bộ tạo mã vòng (7,4) có đa thức tạo mã  $g(x) = 1 + x + x^3$  và bản tin đầu vào 101. Tìm từ mã đầu ra.

### Hướng dẫn giải:

Đa thức bản tin có dạng sau:  $m(x)=1+x^2$ . Nhân đa thức bản tin với  $x^{7-4}=x^3$  ta được:  $x^3+x^5$ .

Chia tích trên cho đa thức tạo mã để nhận được phần dư  $b(x)$ :

$$\begin{array}{r} x^5 + x^3 \quad | \quad x^3 + x + 1 \\ \underline{x^3 \phantom{+ x^2} + x^2 + x + 1} \\ x^2 \phantom{+ x^3} \end{array}$$

$$\oplus$$

$$\begin{array}{r} x^5 + x^3 + x^2 \\ \underline{x^3 \phantom{+ x^2} + x^2 + x + 1} \\ x^2 \phantom{+ x^3} \end{array}$$

Vậy  $b(x)=x^2$ ;  $c(x) = b(x) + x^{n-k} m(x) = x^2 + x^3 + x^5 \rightarrow c=[001101]$

### Bài 12:

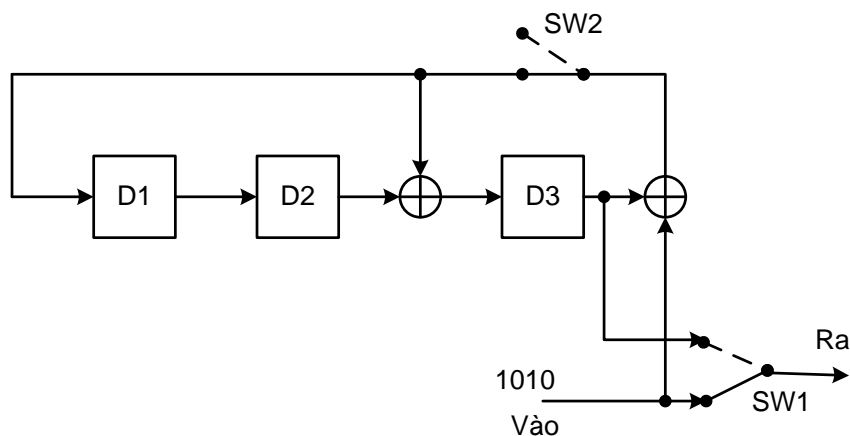
Một bộ tạo mã vòng có đa thức tạo mã  $g(x)=1+x^2+x^3$ .

a) Thiết kế sơ đồ bộ tạo mã

b) Kiểm tra hoạt động của nó với bản tin  $m=[1010]$ .

### Hướng dẫn giải:

a) Sơ đồ bộ mã hóa như sau



b) Nội dung ở thanh ghi dịch khi bản tin vào (10110)

Dịch	Bit vào	Nội dung thanh ghi ( $D_1D_2D_3$ )
		000 (trạng thái đầu)
1	1	101
2	0	111
3	1	011
4	1	010

### Bài 13:

Một bộ tạo mã xoắn với chuỗi tạo mã sau:

$$\mathbf{g_1} = (g_{0,1}, g_{1,1}, g_{2,1}) = (1, 0, 1)$$

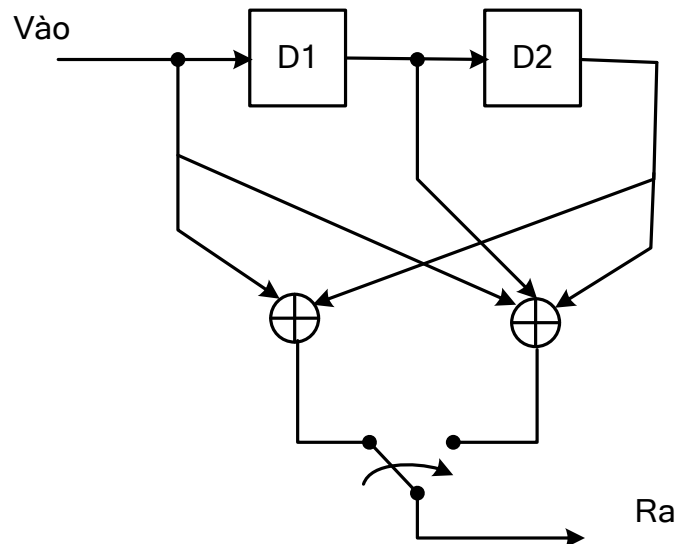
$$\mathbf{g}_2 = (g_{0,2}, g_{1,2}, g_{2,2}) = (1, 1, 1)$$

a) Thiết kế sơ đồ

b) Tính toán chuỗi đầu ra theo bảng khi cho chuỗi đầu vào  $\mathbf{m}=[101011]$ , trong đó bit ngoài cùng bên trái là bit vào bộ tạo mã đầu tiên

### Hướng dẫn giải:

a) Sơ đồ bộ tạo mã như sau:



b) Tính toán chuỗi mã đầu ra theo bảng

Chuỗi bản tin m	1 0 1 0 1 1
1/ Bổ sung hai bit đuôi: m'	0 0 1 0 1 0 1 1 0 0
2/ trể một bit: m'.x	0 0 1 0 1 0 1 1 0 0
3/ trể hai bit: m'.x <sup>2</sup>	0 0 1 0 1 0 1 1 0 0
$\mathbf{c}_1 = \mathbf{m}' + \mathbf{m}'x^2$	1 0 0 0 0 1 1 1
$\mathbf{c}_2 = \mathbf{m}' + \mathbf{m}'x + \mathbf{m}'x^2$	1 1 0 1 0 0 0 1
<b>C</b>	11 01 00 01 00 10 1011

### Bài 14:(Tiếp)

Tìm chuỗi mã đầu ra theo phương pháp chuỗi tạo mã.

### Hướng dẫn giải:

Ta chuỗi bit đầu ra được xác định

$$c_{\ell,j} = \sum_{p=0}^M g_{p,j} m_{\ell-p} \quad j=1,2; p=0,1,2; \ell=0,1,\dots,7; p \leq \ell$$

$$\mathbf{g}_1 = (g_{0,1}, g_{1,1}, g_{2,1}) = (1, 0, 1)$$

$$\mathbf{g}_2 = (g_{0,2}, g_{1,2}, g_{2,2}) = (1, 1, 1)$$

$$\mathbf{m}=[m_0, m_1, m_2, m_3, m_4, m_5] = [101011],$$

Đối với nhánh trên  $j=1$  ta được:

$$c_{0,1} = g_{0,1} m_0 = 1 \times 1 = 1$$

$$c_{1,1} = g_{0,1} m_1 + g_{1,1} m_0 = 1 \times 0 + 0 \times 1 = 0$$

$$\begin{aligned}c_{2,1} &= g_{0,1} m_2 + g_{1,1} m_1 + g_{2,1} m_0 = 1 \times 1 + 0 \times 0 + 1 \times 1 = 0 \\c_{3,1} &= g_{0,1} m_3 + g_{1,1} m_2 + g_{2,1} m_1 = 1 \times 0 + 0 \times 1 + 1 \times 0 = 0 \\c_{4,1} &= g_{0,1} m_4 + g_{1,1} m_3 + g_{2,1} m_2 = 1 \times 1 + 0 \times 0 + 1 \times 1 = 0 \\c_{5,1} &= g_{0,1} m_5 + g_{1,1} m_4 + g_{2,1} m_3 = 1 \times 1 + 0 \times 1 + 1 \times 0 = 1 \\c_{6,1} &= g_{0,1} m_6 + g_{1,1} m_5 + g_{2,1} m_4 = 1 \times 0 + 0 \times 1 + 1 \times 1 = 1 \\c_{7,1} &= g_{0,1} m_7 + g_{1,1} m_6 + g_{2,1} m_5 = 1 \times 0 + 0 \times 0 + 1 \times 1 = 1\end{aligned}$$

Vậy chuỗi bit ở đầu ra của nhánh trên là :

$$c_1 = (10000111)$$

Tương tự đối với nhánh dưới  $j=2$  ta được: ,

$$\begin{aligned}c_{0,2} &= g_{0,2} m_0 = 1 \times 1 = 1 \\c_{1,2} &= g_{0,2} m_1 + g_{1,2} m_0 = 1 \times 0 + 1 \times 1 = 1 \\c_{2,2} &= g_{0,2} m_2 + g_{1,2} m_1 + g_{2,2} m_0 = 1 \times 1 + 1 \times 0 + 1 \times 1 = 0 \\c_{3,2} &= g_{0,2} m_3 + g_{1,2} m_2 + g_{2,2} m_1 = 1 \times 0 + 1 \times 1 + 1 \times 0 = 1 \\c_{4,2} &= g_{0,2} m_4 + g_{1,2} m_3 + g_{2,2} m_2 = 1 \times 1 + 1 \times 0 + 1 \times 1 = 0 \\c_{5,2} &= g_{0,2} m_5 + g_{1,2} m_4 + g_{2,2} m_3 = 1 \times 1 + 1 \times 1 + 1 \times 0 = 0 \\c_{6,2} &= g_{0,2} m_6 + g_{1,2} m_5 + g_{2,2} m_4 = 1 \times 0 + 1 \times 1 + 1 \times 1 = 0 \\c_{7,2} &= g_{0,2} m_7 + g_{1,2} m_6 + g_{2,2} m_5 = 1 \times 0 + 1 \times 0 + 1 \times 1 = 1\end{aligned}$$

Vậy chuỗi bit ở đầu ra của nhánh dưới sẽ là:

$$c_2 = (11010001)$$

Chuỗi bit ở đầu ra của bộ lập mã là ghép chung của hai chuỗi  $c_1, c_2$  như sau:

$$c = (11\ 01\ 00\ 01\ 00\ 10\ 1011) \text{ có 16 bit.}$$

### Bài 15: (Tiếp)

Tìm chuỗi mã đầu ra theo phương pháp đa thức tạo mã.

#### Hướng dẫn giải:

Bằng cách sử dụng đa thức tạo mã, ta có thể viết:  $[101011]$ ,

$$m(x) = 1 + x^2 + x^4 + x^5$$

$$g_1(x) = 1 + x^2$$

$$g_2(x) = 1 + x + x^2$$

$$\begin{aligned}c_1(x) &= (1 + x^2 + x^4 + x^5) (1 + x^2) = 1 + x^2 + x^2 + x^4 + x^4 + x^6 + x^5 + x^7 \\&= 1 + x^5 + x^6 + x^7\end{aligned}$$

$$\text{hay: } c_1 = (c_{0,1}, c_{1,1}, c_{2,1}, c_{3,1}, c_{4,1}, c_{5,1}, c_{6,1}, c_{7,1}) = (1\ 0\ 0\ 0\ 01\ 1\ 1)$$

$$\begin{aligned}c_2(x) &= (1 + x^2 + x^4 + x^5) (1 + x + x^2) \\&= 1 + x + x^2 + x^2 + x^3 + x^4 + x^4 + x^5 + x^6 + x^5 + x^6 + x^7 \\&= 1 + x + x^3 + x^7\end{aligned}$$

$$\text{hay: } c_2 = (c_{0,2}, c_{1,2}, c_{2,2}, c_{3,2}, c_{4,2}, c_{5,2}, c_{6,2}, c_{7,2}) = (1\ 1\ 0\ 1\ 0\ 0\ 01)$$

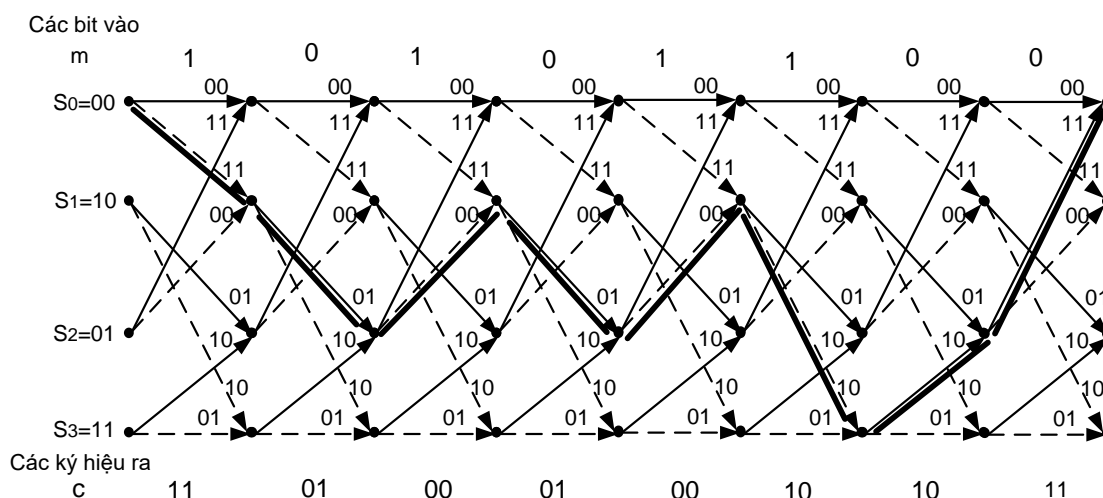
Cuối cùng ta được chuỗi đầu ra là ghép xen của hai chuỗi trên như sau:

$$c = (11\ 01\ 00\ 01\ 0010\ 1011)$$

### Bài 16: (Tiếp)

Tìm chuỗi mã đầu ra theo biểu đồ lưới

### Hướng dẫn giải:



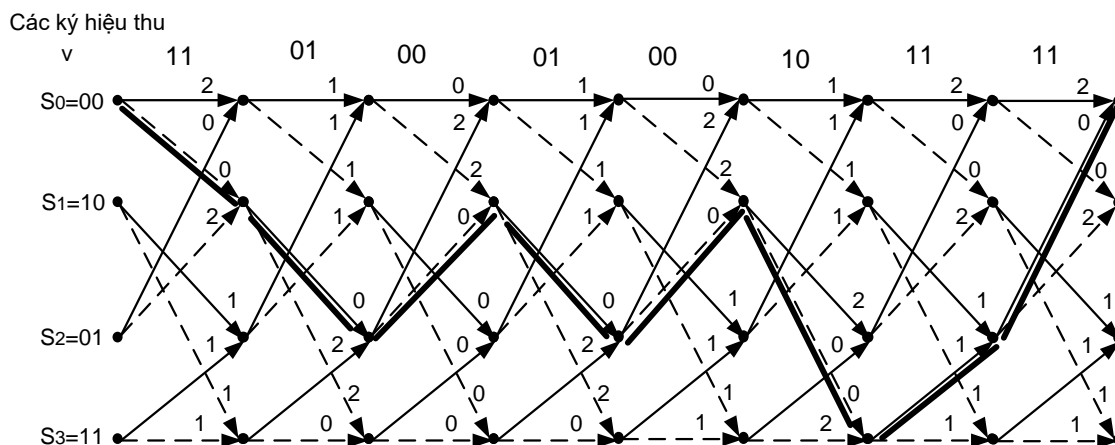
Từ biểu đồ lưới ta thấy đường dẫn chuỗi ký hiệu ra bắt đầu từ trạng thái  $s=(00)$  và kết thúc ở trạng thái này. Sở dĩ như vậy vì hai bit đuôi bằng "00".

### Bài 17: (Tiếp)

Khi chuỗi ký hiệu thu được bằng:  $v=[11 \ 01 \ 00 \ 01 \ 00 \ 10 \ 11 \ 11]$ . Tìm khoảng cách Hamming giữa chuỗi ký hiệu thu và ký hiệu phát.

### Hướng dẫn giải:

Ta biểu diễn khoảng cách Hamming của chuỗi ký hiệu thu cho từng nhánh trên biểu đồ lưới trên hình sau.

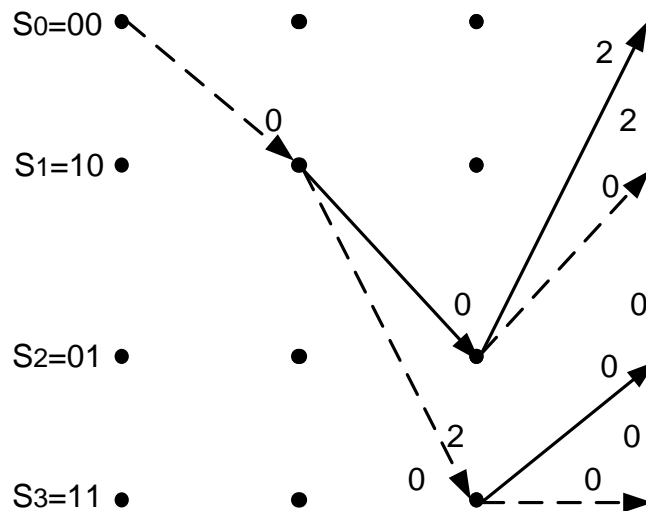


Từ hình trên ta thấy khoảng cách Hamming bằng 1.

### Bài 18: (Tiếp)

Biểu thị các đường dẫn sóng sót sau lần thứ nhất hội nhập các cặp đường dẫn.

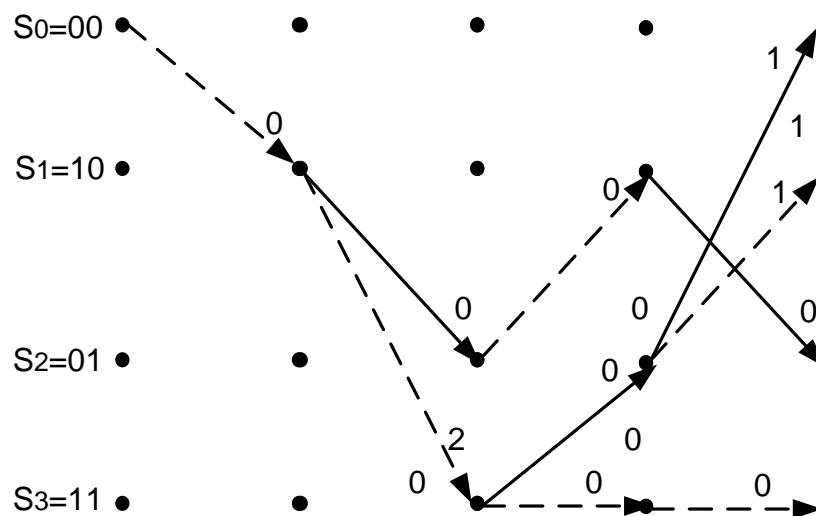
### Hướng dẫn giải:



**Bài 19: (Tiếp)**

Biểu thị các đường dẫn sóng sau lần thứ hai hội nhập các cặp đường dẫn.

**Hướng dẫn giải:**



**Bài 20:**

Một bộ tạo mã xoắn với các đa thức tạo mã sau:

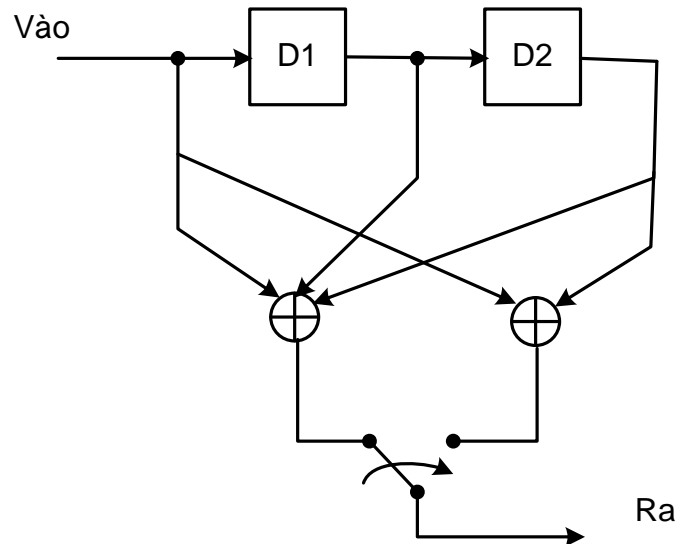
$$g_1(x) = 1 + x + x^2$$

$$g_2(x) = 1 + x^2$$

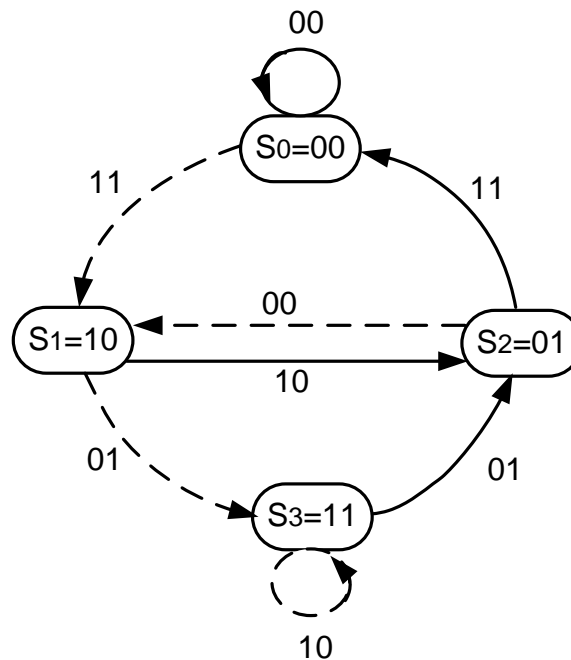
- Thiết kế sơ đồ
- Vẽ biểu đồ trạng thái
- Vẽ biểu đồ lưới

**Hướng dẫn giải:**

a)



b)



### Bài 21: (Tiếp)

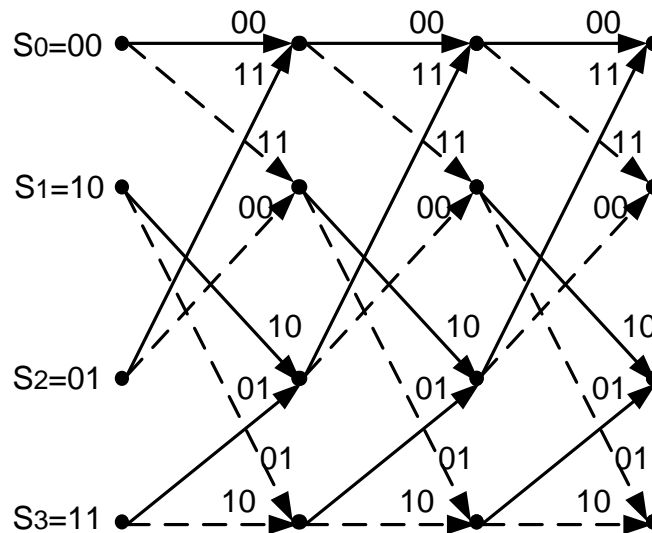
a) Vẽ biểu đồ lưới

b) Tìm chuỗi ký hiệu ra theo biểu đồ lưới khi chuỗi bit vào là:  $m=[101011]$ .

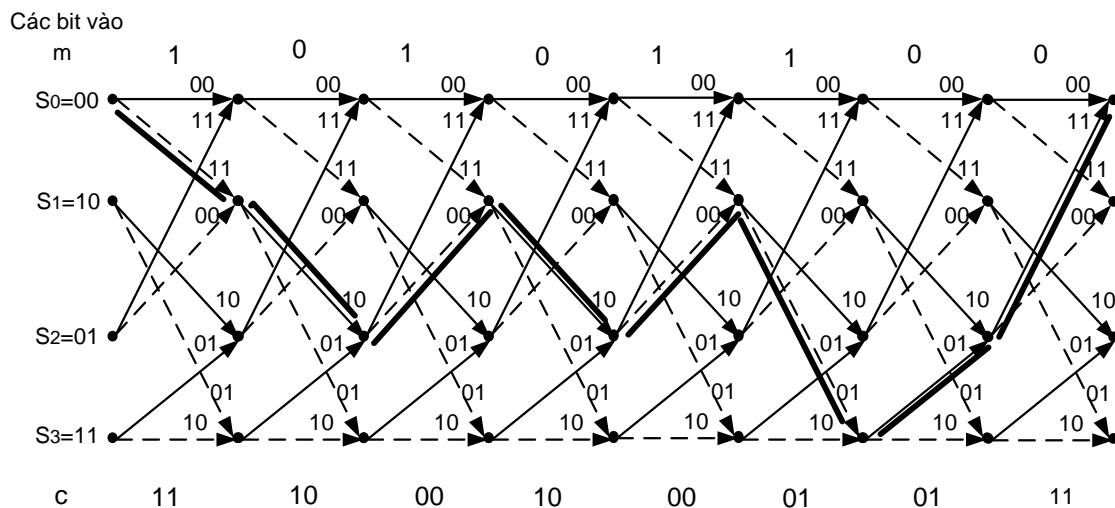
### Hướng dẫn giải:

a)





b)



## Bài 22:

Cho đa thức tạo mã sau:

$$g(x)=1+x^2$$

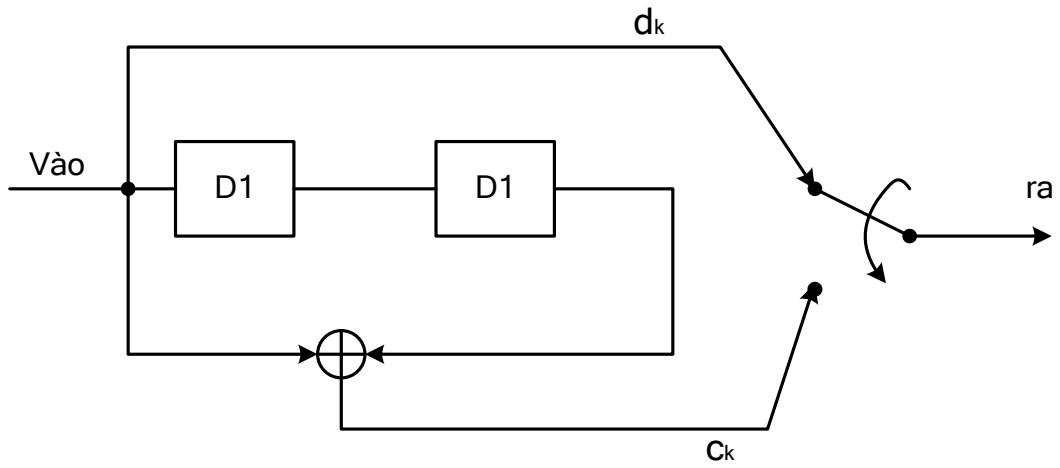
a) Thiết kế bộ mã hóa xoắn hệ thống SC

b) Vẽ biểu đồ lưới

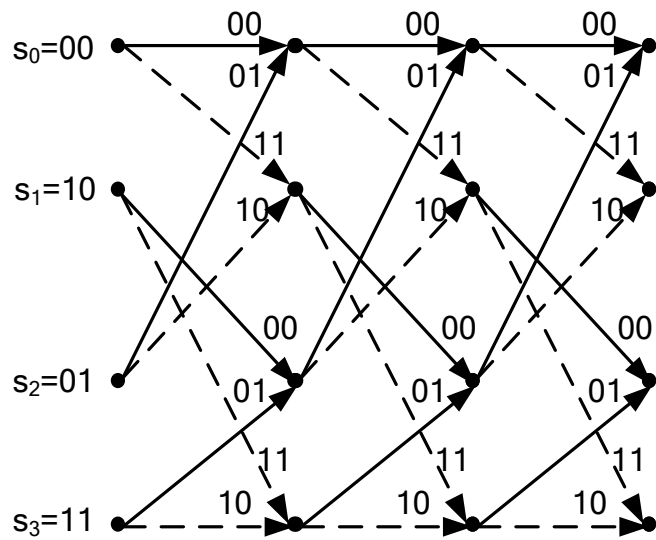
c) Tìm chuỗi ký hiệu ra khi chuỗi bit vào là  $m=[100]$ .

## Hướng dẫn giải:

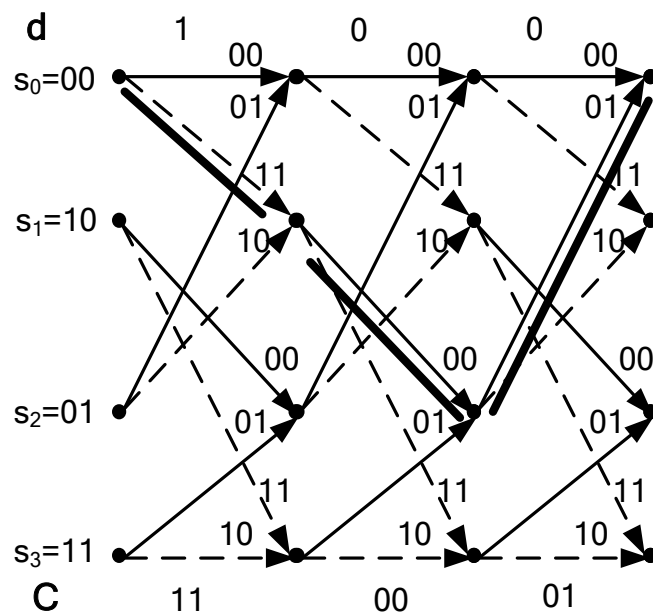
a)



b)



c)



Chuỗi ký hiệu ra là:  $C=[110001]$

**Bài 23:**

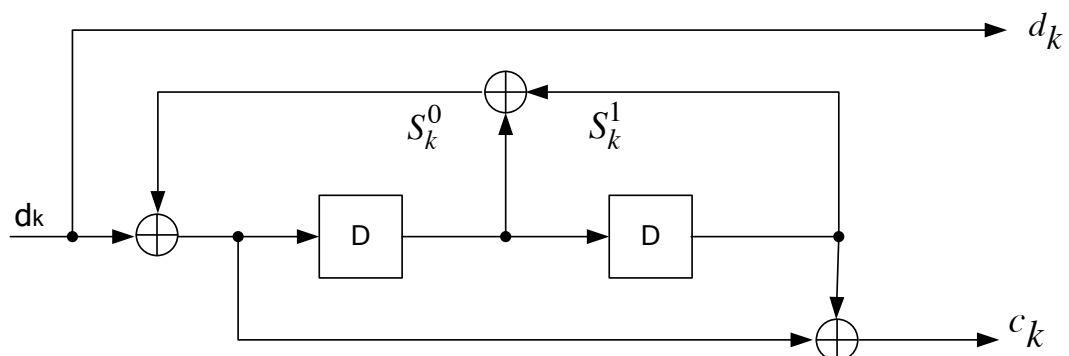
Cho đa thức tạo mã sau:

$$g(x) = \left( 1, \frac{1+x^2}{1+x+x^2} \right)$$

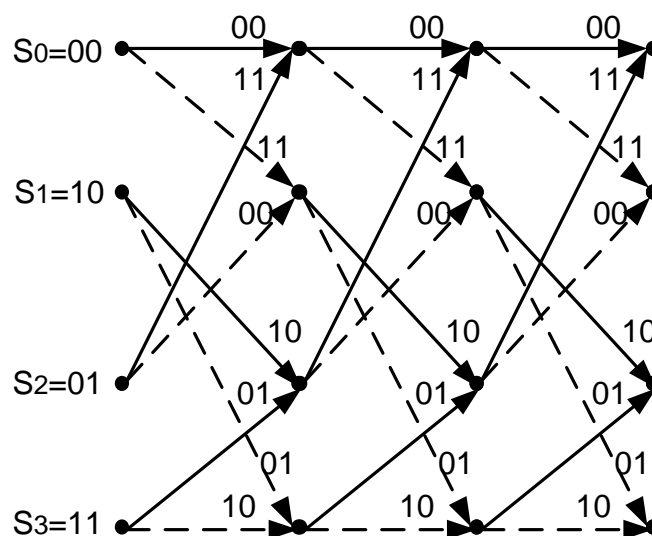
- Thiết kế bộ tạo mã xoắn hồi quy RSC
- Vẽ biểu đồ lưới

**Hướng dẫn giải:**

a)



b)



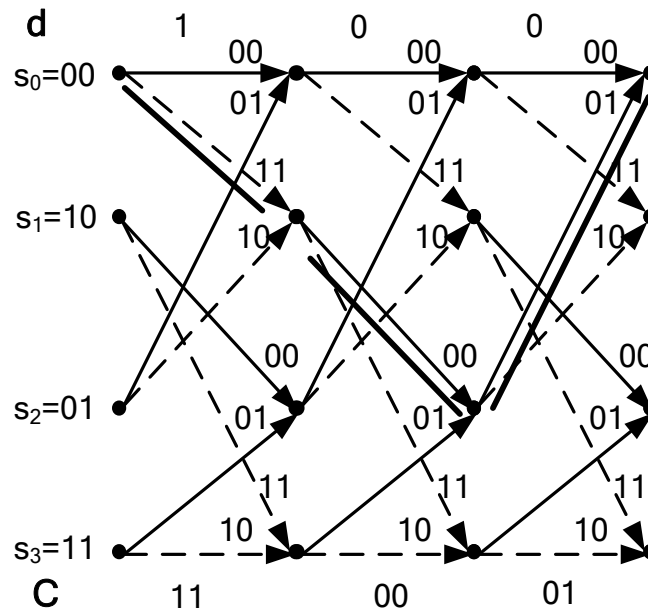
**Bài 24:(Tiếp)**

Chuỗi bit số liệu vào bộ mã hóa RSC là  $\mathbf{d}=[100]$

- Tìm chuỗi ký hiệu đầu ra của bộ mã hóa dựa trên biểu đồ lưới.
- Chuỗi ký hiệu lưỡng cực đưa lên điều chế.

**Hướng dẫn giải:**

a)



$$C=\{1,1,0,0,0,1\}$$

b) Chuỗi lưỡng cực được xác định theo công thức:

$$a_k=1-2d_k$$

$$b_k=1-2c_k$$

$$C=\{-1,-1,+1,+1,+1,-1\}$$

### Bài 25:(Tiếp)

Với giả thiết  $\sigma^2=1$  và chuỗi ký hiệu thu được tại đầu vào bộ giải mã MAP tại các thời điểm  $k=[0,1,2]$  như sau:  $R=[(-0,5;-0,2), (1,5;1,2),(0,8;-0,4)]$ . Giả thiết xác suất phát  $d_k=0$  và  $d_k=1$  là như nhau. Tính toán số đo nhánh cho

a)  $k=0$

b)  $k=1$

c)  $k=2$

a) ta có:

$$\delta_{k,i}(m) = \gamma_{k,i} \exp \left[ \frac{1}{\sigma^2} (x_k a_{k,i} + y_k b_{k,i}(m)) \right]. \text{ Vì xác suất phát } d_k=0 \text{ và } d_k=1 \text{ bằng nhau,}$$

nên  $\gamma_{k,i} = 1/2$ . Giả thiết  $\sigma^2=1$ .

- $R_1=(-0,5;-0,2)$

$$\delta_{1,0}(00) = \delta_{1,0}(10) = \frac{1}{2} \exp((-0,5 \times 1) + (-0,2 \times 1)) = 0,25$$

$$\delta_{1,1}(00) = \delta_{1,1}(10) = \frac{1}{2} \exp((-0,5 \times -1) + (-0,2 \times -1)) = 1$$

$$\delta_{1,0}(01) = \delta_{1,0}(11) = \frac{1}{2} \exp((-0,5 \times 1) + (-0,2 \times -1)) = 0,37$$

$$\delta_{1,1}(01) = \delta_{1,1}(11) = \frac{1}{2} \exp((-0,5 \times -1) + (-0,2 \times 1)) = 0,67$$

- $R_2=(1,5;1,2)$

$$\delta_{2,0}(00) = \delta_{2,0}(10) = \frac{1}{2} \exp((1,5 \times 1) + (1,2 \times 1)) = 7,4$$

$$\delta_{2,1}(00) = \delta_{2,1}(10) = \frac{1}{2} \exp((1,5 \times -1) + (1,2 \times -1)) = 0,03$$

$$\delta_{2,0}(01) = \delta_{2,0}(11) = \frac{1}{2} \exp((1,5 \times 1) + (1,2 \times -1)) = 0,67$$

$$\delta_{2,1}(01) = \delta_{2,1}(11) = \frac{1}{2} \exp((1,5 \times -1) + (1,2 \times 1)) = 0,37$$

- $R_3=(0,8;-0,4)$

$$\delta_{3,0}(00) = \delta_{3,0}(10) = \frac{1}{2} \exp((0,8 \times 1) + (-0,4 \times 1)) = 0,75$$

$$\delta_{3,1}(00) = \delta_{3,1}(10) = \frac{1}{2} \exp((0,8 \times -1) + (-0,4 \times -1)) = 0,34$$

$$\delta_{3,0}(01) = \delta_{3,0}(11) = \frac{1}{2} \exp((0,8 \times 1) + (-0,4 \times -1)) = 1,66$$

$$\delta_{3,1}(01) = \delta_{3,1}(11) = \frac{1}{2} \exp((0,8 \times -1) + (-0,4 \times 1)) = 0,15$$

### Bài 26:(Tiếp)

Tính toán số đo trạng thái thuận (Forward State Metric).

#### Hướng dẫn giải:

Ta có:

$$\alpha_k(m) = \sum_{j=0}^1 \alpha_{k-1}(b_j(m)) \delta_{k,j}(b_j(m))$$

Giả thiết bộ giải mã bắt đầu từ  $m=00$ ,

- $k=1$

$$\alpha_1(00) = 1 \text{ và } \alpha_1(m \neq 00) = 0 \text{ khi } m \neq 00$$

- $k=2$

$$m=00: \alpha_2(00) = \alpha_1(00)\delta_{1,0}(00) + \alpha_1(01)\delta_{1,0}(01) = \delta_{1,0}(00) = 0,25$$

$$m=10: \alpha_2(10) = \alpha_1(00)\delta_{1,1}(00) + \alpha_1(01)\delta_{1,1}(01) = \delta_{1,1}(00) = 1$$

$$m=01: \alpha_2(01) = \alpha_1(10)\delta_{1,0}(10) + \alpha_1(11)\delta_{1,0}(11) = 0$$

$$m=11: \alpha_2(11) = \alpha_1(10)\delta_{1,1}(10) + \alpha_1(11)\delta_{1,1}(11) = 0$$

- $k=3$

$$m=00: \alpha_3(00) = \alpha_2(00)\delta_{2,0}(00) + \alpha_2(01)\delta_{2,0}(01) = 1,85$$

$$m=10: \alpha_3(10) = \alpha_2(00)\delta_{2,1}(00) + \alpha_2(01)\delta_{2,1}(01) = 0,07$$

$$m=01: \alpha_3(01) = \alpha_2(10)\delta_{2,0}(10) + \alpha_2(11)\delta_{2,0}(11) = 7,4$$

$$m=11: \alpha_3(11) = \alpha_2(10)\delta_{2,1}(10) + \alpha_2(11)\delta_{2,1}(11) = 0,3$$

- $k=4$

$$m=00: \alpha_4(00) = \alpha_3(00)\delta_{3,0}(00) + \alpha_3(01)\delta_{3,0}(01) = 13,67$$

$$m=10: \alpha_4(10) = \alpha_3(00)\delta_{3,1}(00) + \alpha_3(01)\delta_{3,1}(01) = 1,74$$

$$m=01: \alpha_4(01) = \alpha_3(10)\delta_{3,0}(10) + \alpha_3(11)\delta_{3,0}(11) = 0,55$$

$$m=11: \alpha_4(11) = \alpha_3(10)\delta_{3,1}(10) + \alpha_3(11)\delta_{3,1}(11) = 0,07$$

**Bài 27: (Tiếp).**

Tính toán số đo trạng thái ngược (Backward State Metric).

**Hướng dẫn giải:**

Ta có:

$$\beta_k(m) = \sum_{j=0}^1 \delta_{k,j}(m) \beta_{k+1}(f(j, m))$$

- k=4

$$m=00 \beta_4(00)=1$$

$$m=10 \beta_4(10)=0$$

$$m=01 \beta_4(01)=0$$

$$m=11 \beta_4(11)=0$$

- k=3

$$\begin{aligned} m=00 \beta_3(00) &= \delta_{3,0}(00) \beta_4(00) + \delta_{3,1}(10) \beta_4(10) \\ &= 0,75 \times 1 + 0,34 \times 0 = 0,75 \end{aligned}$$

$$\begin{aligned} m=10 \beta_3(10) &= \delta_{3,0}(01) \beta_4(01) + \delta_{3,1}(10) \beta_4(11) \\ &= 0,75 \times 0 + 0,34 \times 0 = 0 \end{aligned}$$

$$\begin{aligned} m=01 \beta_3(01) &= \delta_{3,0}(00) \beta_4(00) + \delta_{3,1}(10) \beta_4(10) \\ &= 1,66 \times 1 + 0,15 \times 0 = 1,66 \end{aligned}$$

$$\begin{aligned} m=11 \beta_3(11) &= \delta_{3,0}(01) \beta_4(01) + \delta_{3,1}(11) \beta_4(11) \\ &= 1,66 \times 0 + 0,15 \times 0 = 0 \end{aligned}$$

- k=2

$$\begin{aligned} m=00 \beta_2(00) &= \delta_{2,0}(00) \beta_3(00) + \delta_{2,1}(10) \beta_3(10) \\ &= 7,4 \times 0,75 + 0,3 \times 0 = 5,55 \end{aligned}$$

$$\begin{aligned} m=10 \beta_2(10) &= \delta_{2,0}(01) \beta_3(01) + \delta_{2,1}(10) \beta_3(11) \\ &= 7,4 \times 1,66 + 0,67 \times 0 = 12,5 \end{aligned}$$

$$\begin{aligned} m=01 \beta_2(01) &= \delta_{2,0}(00) \beta_3(00) + \delta_{2,1}(10) \beta_3(10) \\ &= 0,67 \times 0,75 + 0,37 \times 0 = 0,5 \end{aligned}$$

$$\begin{aligned} m=11 \beta_2(11) &= \delta_{2,0}(01) \beta_3(01) + \delta_{2,1}(11) \beta_3(11) \\ &= 0,67 \times 1,66 + 0,37 \times 0 = 1,11 \end{aligned}$$

- k=1

$$\begin{aligned} m=00 \beta_1(00) &= \delta_{1,0}(00) \beta_2(00) + \delta_{1,1}(10) \beta_2(10) \\ &= 0,25 \times 5,55 + 12,28 \times 1 = 13,67 \end{aligned}$$

$$\begin{aligned} m=10 \beta_1(10) &= \delta_{1,0}(01) \beta_2(01) + \delta_{1,1}(10) \beta_2(11) \\ &= 0,25 \times 0,5 + 1 \times 1,11 = 1,24 \end{aligned}$$

$$\begin{aligned} m=01 \beta_1(01) &= \delta_{1,0}(00) \beta_2(00) + \delta_{1,1}(10) \beta_2(10) \\ &= 0,37 \times 5,55 + 0,67 \times 12,28 = 10,28 \end{aligned}$$

$$\begin{aligned} m=11 \beta_1(11) &= \delta_{1,0}(01) \beta_2(01) + \delta_{1,1}(11) \beta_2(11) \\ &= 0,5 \times 0,37 + 0,67 \times 1,11 = 0,93 \end{aligned}$$

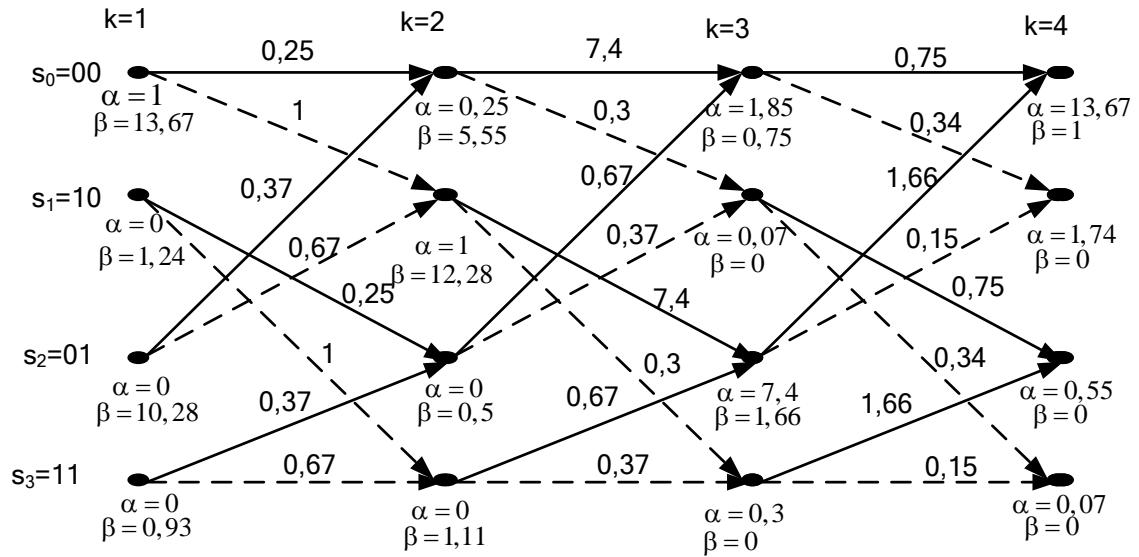
**Bài 28: (Tiếp)**

a) Điền các số đo tìm được trong các bài trước lên biểu đồ lưới

b) Tính  $L(\hat{d}_k)$  và tìm ước tính nhận được sau giải mã.

**Hướng dẫn giải:**

a)



b)

ta có:

$$L(\ddot{\mathbf{d}}_k) = \ln \left[ \frac{\sum_m \alpha_k(m) \delta_{k,0}(m) \beta_{k+1,0}(f(0,m))}{\sum_m \alpha_k(m) \delta_{k,1}(m) \beta_{k+1,1}(f(1,m))} \right]$$

• k=1

$$L(\ddot{\mathbf{d}}_1) = \ln \left[ \frac{1 \times 0,25 \times 5,55}{1 \times 1 \times 12,28} \right] = -2,18 \rightarrow \ddot{\mathbf{d}}_1 = 1$$

• k=2

$$L(\ddot{\mathbf{d}}_2) = \ln \left[ \frac{0,25 \times 7,2 \times 0,75 + 1 \times 7,4 \times 1,66}{0} \right] = \infty \rightarrow \ddot{\mathbf{d}}_2 = 0$$

• k=3

$$L(\ddot{\mathbf{d}}_2) = \ln \left[ \frac{1,85 \times 0,75 \times 1 + 7,4 \times 1,66 \times 1}{0} \right] = \infty \rightarrow \ddot{\mathbf{d}}_3 = 0$$

**Bài 29:**

Hãy đọc mã chương trình **NVD8\_nkchoose** thể hiện hiệu năng của mã khối tuyến tính bằng cách chạy chương trình, thay đổi các tham số đầu vào chương trình và phân tích kết quả.

**Hướng dẫn giải:**

Hệ thống BPSK hoạt động trong môi trường kênh AWGN. Khi này, xác suất lỗi bit không mã hoá là

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

Với mã khối (n,k), xác suất lỗi ký hiệu kênh là

$$P_b^c = Q\left(\sqrt{\frac{k}{n} \times 2 \frac{E_b}{N_0}}\right)$$

Ta xét hai mã khác nhau là: (i) mã Golay (23,12) trong đó  $n=23$ ,  $t=3$  và  $d=7$ ; (ii) mã Hamming (15,11) trong đó  $n=15$ ,  $t=1$  và  $d=3$ . Ta lưu ý rằng, ta phải ước lượng

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}$$

```
function out=NVD8_nkchoose(n,k)
% File: NVD8_nkchoose.m
% Tính n!/k!/(n-k)!
a = sum(log(1:n)); % ln của n!
b = sum(log(1:k)); % ln của k!
c = sum(log(1:(n-k))); % ln của (n-k)!

out = round(exp(a-b-c)); % Kết quả
```

Chương trình Matlab để tính toán các đường cong hiệu năng cho mã Hamming (15,11) và mã Golay (23,12) sửa ba lỗi được cho ở file **NVD8\_cersim** dưới đây. Cần lưu ý rằng, ta dùng kênh AWGN và điều chế BPSK.

```
% File: NVD8_cersim.m
zdB = 0:0.1:10; % véc tơ Eb/No đơn vị dB
z = 10.^(zdB/10); % Đổi đơn vị
ber1 = q(sqrt(2*z));
ber2 = q(sqrt(12*2*z/23)); % CSER cho mã Golay(23,12)
ber3 = q(sqrt(11*z*2/15)); % CSER cho mã Hamming(15,11)
berg = NVD8_cer2ber(2,23,7,3,ber2); % BER cho mã Golay
berh = NVD8_cer2ber(2,15,3,1,ber3); % BER cho mã Golay

%=====
% plot results
%=====
semilogy(zdB,ber1,zdB,berg,zdB,berh);
xlabel('E_b/N_o dB','fontname','.Vntime');
ylabel('Xác suất lỗi bit','fontname','.Vntime');
```

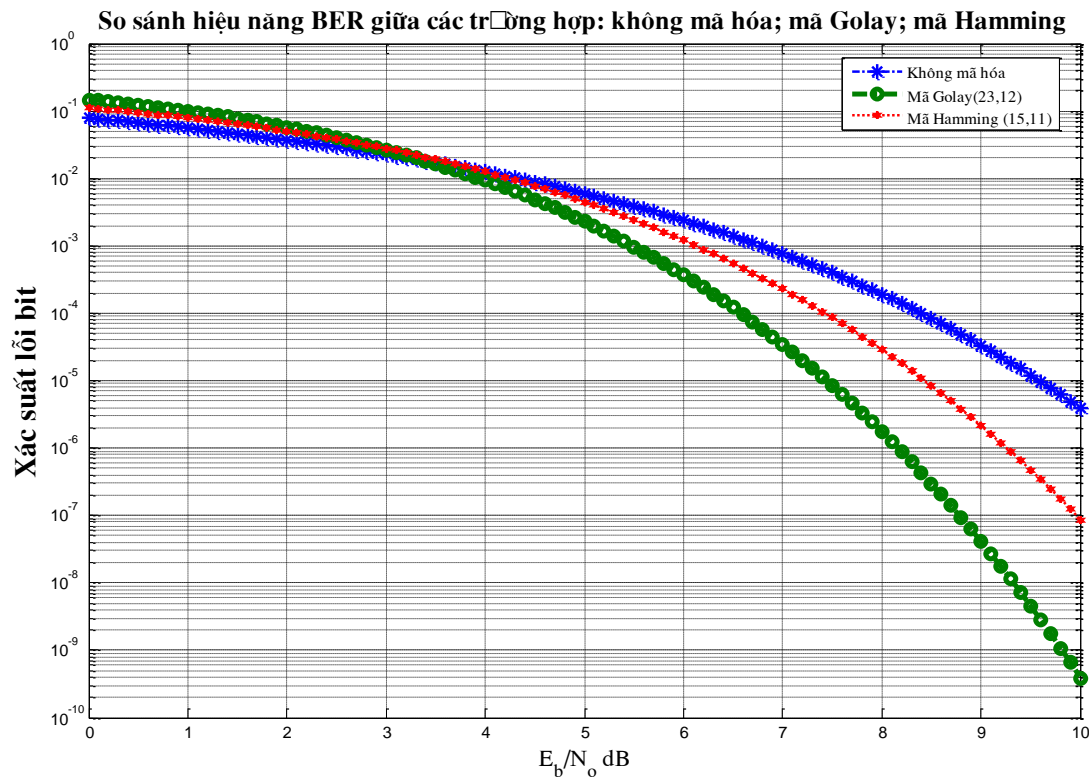
Ta lưu ý rằng, trong chương trình **NVD8\_cersim.m** ta đã dùng chương trình con **NVD8\_cer2ber** dưới đây để chuyển xác suất lỗi ký hiệu kênh thành xác suất lỗi bit giải mã (đọc thêm cuối chương 8 của tài liệu tham khảo 1).

```
function [ber] = NVD8_cer2ber(q,n,d,t,ps)
% File: NVD8_cer2ber.m
lnps = length(ps);
ber = zeros(1,lnps);
for k=1:lnps
    cer = ps(k);
    sum1 = 0; sum2 = 0;
    %=====
    % Vòng lặp thứ nhất ước lượng tổng thứ nhất
    %=====
    for i=(t+1):d
        term = nkchoose(n,i)*(cer^i)*((1-cer)^(n-i));
        sum1 = sum1+term;
    end
    %=====
    % Vòng lặp thứ hai ước lượng tổng thứ hai
    %=====
    for i=(d+1):n
        term = i*nkchoose(n,i)*(cer^i)*((1-cer)^(n-i));
        sum2 = sum2+term;
    end
    ber(k) = sum1+sum2;
end
```



```
% Tính BER
ber(k) = (q/(2*(q-1))) * ((d/n)*sum1 + (1/n)*sum2);
end
```

Kết quả của các tính toán này được minh họa ở hình



So sánh hiệu năng các mã khối Gray và Hamming

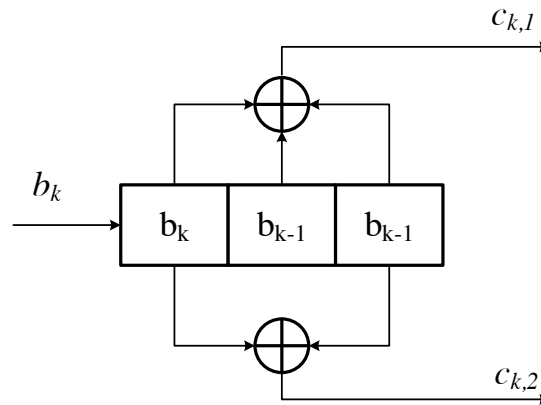
### Bài 30:

Hãy đọc mã chương trình **NVD8\_convcode.m** thể hiện hiệu năng của mã xoắn bằng cách chạy chương trình, thay đổi các tham số đầu vào cho chương trình và phân tích kết quả.

### Hướng dẫn giải:

Ta có giới hạn hiệu năng của mã xoắn là (xem thêm cuối chương 8 của tài liệu tham khảo 1)

$$P_E < \frac{1}{2} \frac{\partial}{\partial I} \left( \frac{D^5 I}{1 - 2DI} \right) \Big|_{I=1, D=e^{RE_b/N_0}} = \frac{1}{2} \frac{D^5}{(1 - 2D)^2} \Big|_{D=e^{RE_b/N_0}}$$

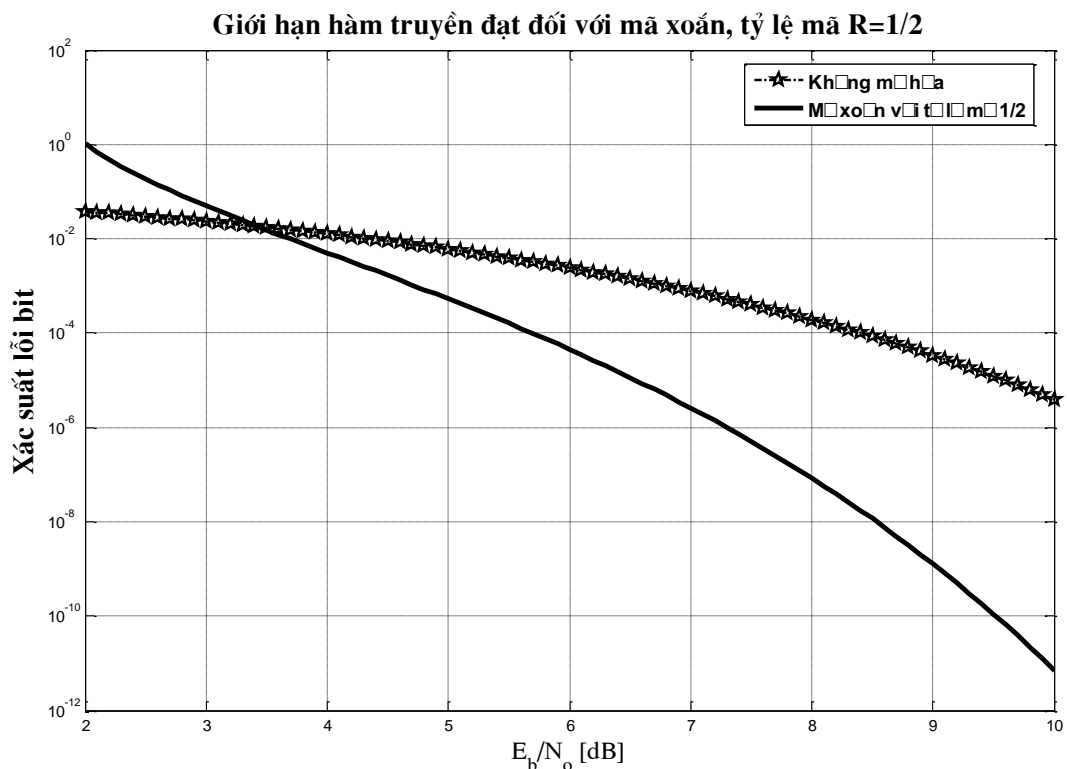


Bộ lập mã xoắn tỷ lệ 1/2

% File NVD8\_convcode.m

```
zdB = 2:0.1:10; % Tạo véc tơ Eb/No [dB]
z = 10.^(zdB/10); % Chuyển thành tuyến tính
puc = Q(sqrt(2*z)); % BER không mã hóa
W = exp(-z/2);
Num = W.^5;
Den = 1-4*W+4*W.*W;
ps = 0.5*Num./Den;
%=====
% Hiện thị kết quả
%=====
semilogy(zdB,puc,'-.',zdB,ps);
xlabel('E_b/N_o [dB]','fontname','.vntime');
ylabel('Xác suất lỗi bit','fontname','.vntime');
grid; legend('Không mã hóa','Mã hóa');

```



Giới hạn hàm truyền đạt đối với ví dụ mã xoắn tỷ lệ mã 1/2

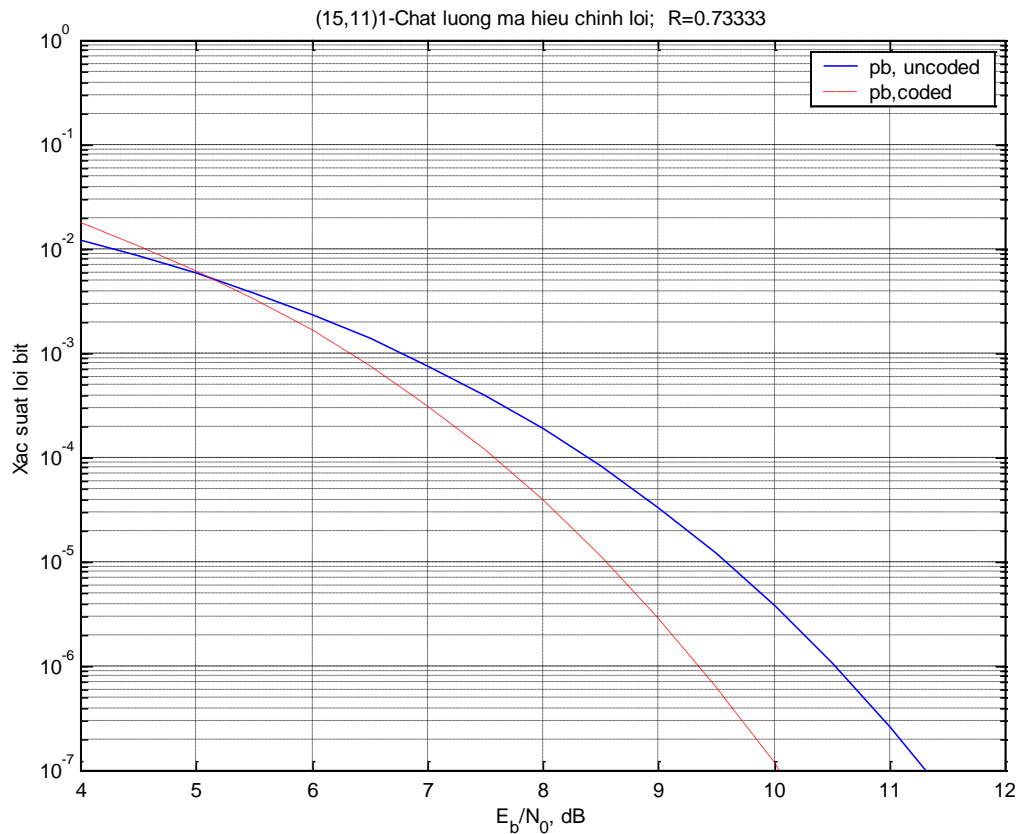
### Bài 31:

Hãy đọc mã chương trình **NVD\_DC649.m** thể hiện hiệu năng của mã khối tuyến tính BCH bằng cách chạy chương trình, thay đổi các tham số đầu vào cho chương trình và phân tích kết quả.

#### Hướng dẫn giải:

Tìm hiểu mã khối tuyến tính BCH, phân tích các tham số đặc trưng, chạy từng lệnh theo từng bước, phân tích so sánh kết quả tương ứng với các tập tham số đầu vào.

```
function y = NVD_DC649(Eb_over_N0_dB)
clf;
A=char('-', '--', ':');
%disp('Nhap n va k cho ma C(n,k)');
n=input('Nhap n cho ma C(n,k)=');
%n=7;
k=input('Nhap k cho ma C(n,k)=');
%k=4;
t= input('Nhap t (xem bang 6-9) doi voi ma BCH =');
B=pascal(n+1);
R=k/n;
Eb_over_N0_dB=4:.5:12;
Eb_over_N0=10.^(Eb_over_N0_dB/10);
pb_unc=0.5*erfc(sqrt(Eb_over_N0));
p=.5*erfc(sqrt(R*Eb_over_N0));
term1=[];
term2=[];
term3=[];
j=0;
for i=t+1:n
    j=j+1;
    term1(j)=min(k,i+t);
    term2(j)=B(i+1,n-i+1);
    term3(j,:)=(p.^i).*(1-p).^(n-i);
end
summand=(term1.*term2).*term3;
pb=(1/k)*summand;
semilogy(Eb_over_N0_dB,pb_unc,A(1,:));
xlabel('E_b/N_0, dB');
axis([4 12 10^(-7) 1]),...
ylabel('Xac suat loi bit'),...
title(['(', num2str(n), ', ', num2str(k), ') ', num2str(t), '-Chat luong ma hieu chinh loi; R=', num2str(R)]);
hold on;
grid on;
semilogy(Eb_over_N0_dB,pb,A(2,:));
legend('pb, uncoded', 'pb, coded', 1);
```



### Bài 32:

Hãy đọc mã chương trình **NVD\_CS88(k)** để khảo sát quá trình tạo mã khối tuyến tính, quá trình tính trọng lượng Hamming cực tiểu, chạy chương trình, thay đổi các tham số đầu vào cho chương trình và phân tích kết quả.

#### Hướng dẫn giải:

Xác định: Ma trận tạo mã và Ma trận bản tin

Quá trình (thuật toán) tạo ra bản tin m sắp xếp các bit thông tin.

Thuật toán nhân hai ma trận tạo ra Codeword đầu ra bộ mã hoá

Quá trình tìm trọng lượng cực tiểu của mã.

Các xác định số liệu vào ra của bộ lập mã.

```
function y = NVD_ChannelCoder1
k=input('Nhap do dai k cho ma C(n,k) (k=4) =');
% generate information message matrix U
for i=1:2^k,
% pause;
for j=k:-1:1,
if rem(i-1,2^(-j+k+1))>=2^(-j+k)
u(i,j)=1;
else
u(i,j)=0;
end
%disp(u(i,j));
end
%disp(u(i,j));
end
```

```
% Define G, the Generator matrix
%g=input('Define G & enter the Generator matrix=');
g=[1 0 0 1 1 1 0 1 1 1
    1 1 1 0 0 0 1 1 1 0
    0 1 1 0 1 1 0 1 0 1
    1 1 0 1 1 1 1 0 0 1];
% Generate Codewords
c=rem(u*g,2);
% Find the minimum distance
w_min=min(sum((c(2:2^k,:))'));
disp('          Ma tran ban tin U ');
disp(u);
disp('          Ma tran tu ma ra ');
disp(c);
disp('          Trong luong ma cuc tieu Wm')
disp(w_min);
% Prod(n1:n2) la tích số của các số từ n1 đến n2. nếu n1=1 thì là giai
thua
```

<u>Kết quả ma trận m được tạo ra</u>	<u>Ma trận Codeword đầu ra bộ mã hoá</u>
0000	0000000000
0001	1101111001
0010	0110110101
0011	1011001100
0100	1110001110
0101	0011110111
0110	1000111011
0111	0101000010
1000	1001110111
1001	0100001110
1010	1111000010
1011	0010111011
1100	0111111001
1101	1010000000
1110	0001001100
1111	1100110101
<u>Trong luong ma cuc tieu</u> Wm = 2	

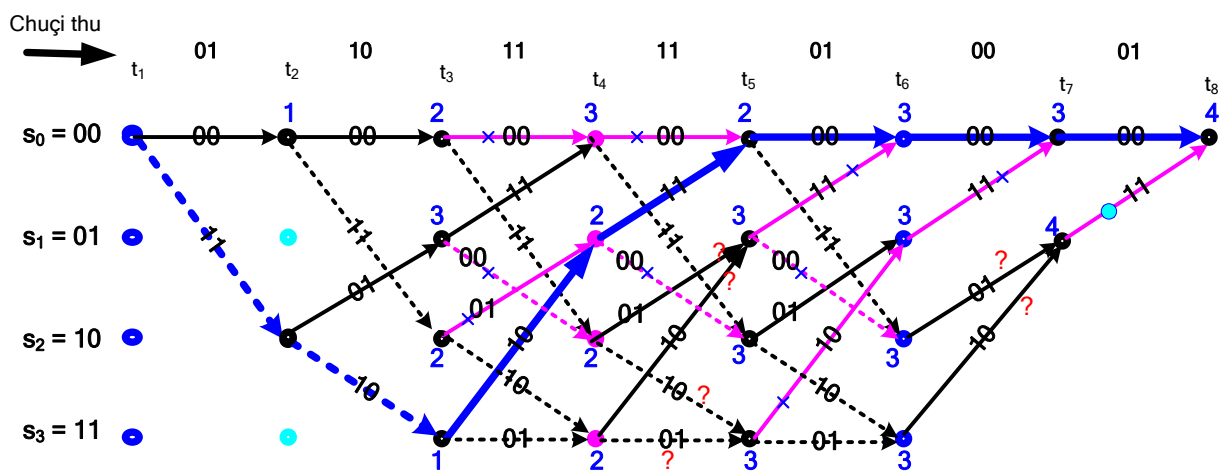
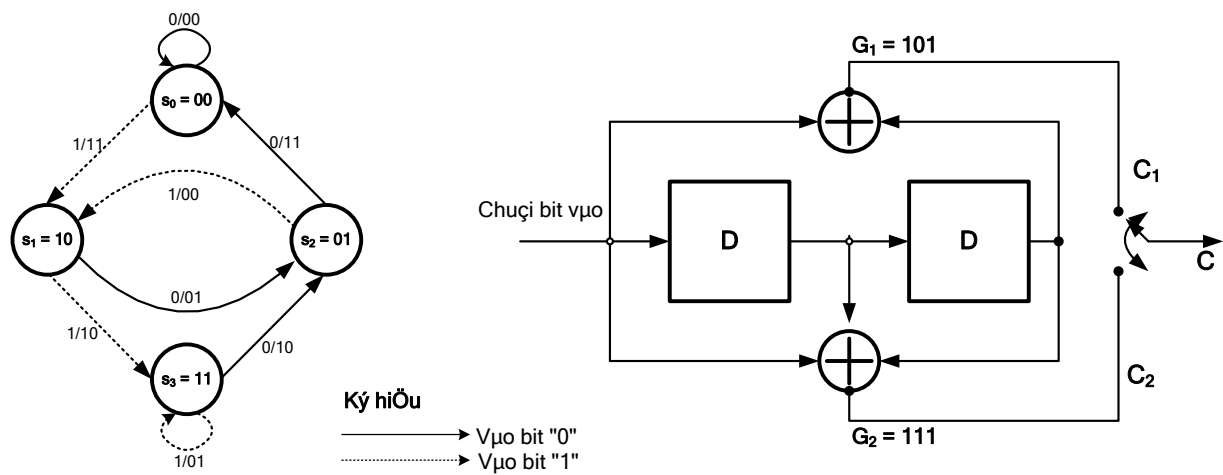
### Bài 33:

Hãy đọc mã chương trình **NVD\_conv\_encd.m** để khảo sát quá trình tạo mã xoắn, phân tích các tham số đặc trưng của mã, chạy chương trình, thay đổi các tham số đầu vào cho chương trình và phân tích kết quả.

### Hướng dẫn giải:

Hãy xác định chuỗi mã mã **C** tương ứng với bản tin đầu vào **m** và ma trận tạo mã **G** theo phương pháp biểu đồ lưới, biểu đồ trạng thái... Sau đó kiểm tra kết quả bằng cách chạy chương trình. Kiểm tra các kết quả cho tất cả các bài tập lập mã xoắn bằng cách chạy chương trình. Đặc biệt hãy so sánh với các trường hợp dưới đây.

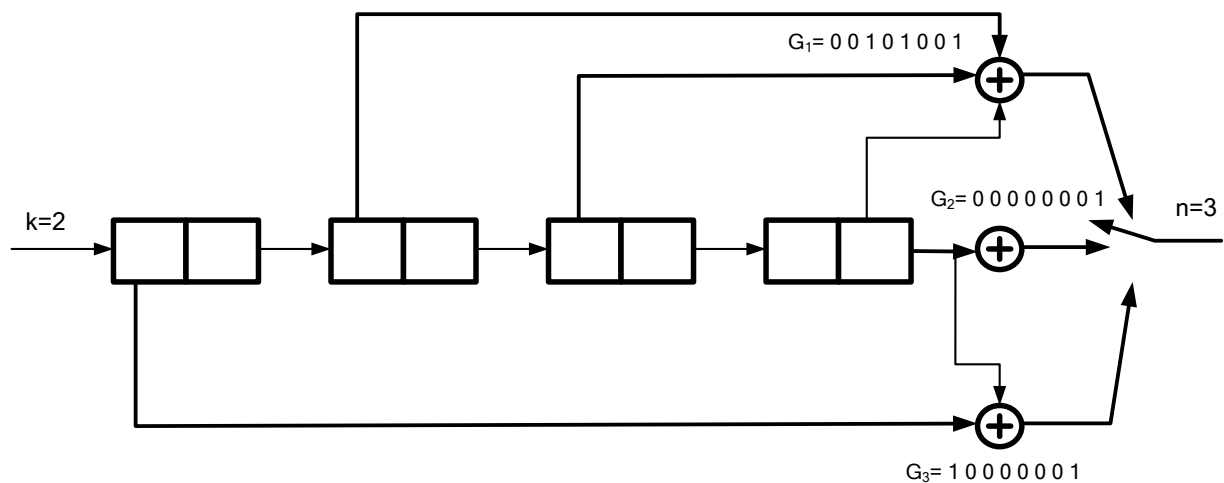
#### ❖ Trường hợp 1:



Hình PL3.3 Sơ đồ, sơ đồ trạng thái và sơ đồ lưới của bộ lập mã xoắn

### ❖ Trường hợp 2

Sơ đồ khối bộ lập mã xoắn có  $k=2$   $n=3$  và  $K=4$  được cho ở hình PL3.4



Hình PL3.4: Sơ đồ bộ lập mã xoắn với  $k=2, n=3, M=4, K=4$

Xác định đầu ra bộ mã hoa xoắn hình PL3.4 khi chuỗi tin vào là

$$\mathbf{m}=[1\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 1\ 1]$$

## Phân tích bài toán

Tỷ lệ mã  $r=2/3$ . Lưu ý rằng 3 bit ra bộ lập mã *không những* phụ thuộc vào 2 bit tin được nạp vào bộ lập mã *mà còn* phụ thuộc vào nội dung của ba tầng đầu tiên (6 bit) của bộ lập mã. Nội dung của tầng cuối cùng không ảnh hưởng đến đầu ra vì chúng bị rời khỏi bộ lập mã mỗi khi 2 bit tin được nạp vào bộ lập mã.

Ma trận tạo mã đặc trưng cho bộ lập mã được xác định bởi

$$G = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_{n-1} \\ g_n \end{bmatrix}_{n \times (Kk)}$$

$$\text{Thành phần thứ } i \text{ của } g_j = \begin{cases} 1, & \text{nếu thành phần thứ } i \text{ của bộ ghi dịch } d\text{-} \text{ợc nối} \\ & \text{với bộ kết hợp } t\text{-} \text{ong ứng với ký hiệu thứ } j \text{ của đầu ra} \\ 0, & \text{nếu khác} \end{cases} \quad \text{với}$$

$$\begin{cases} 1 \leq i \leq Kk \\ 1 \leq j \leq n \end{cases}$$

Theo đó cụ thể đối với hình PL3.4 nhận được

$$g_1 = [0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1]$$

$$g_2 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1]$$

$$g_3 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1]$$

Nói chung nó là ma trận  $n \times Kk \Rightarrow$  với hình PL3.4 nhận được.

$$G = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{3 \times 8}$$

## Độn các bit không

**Độn loại 1:** Do độ dài bản tin ban đầu bài cho là 17 không phải là bội số của  $k=2$ , nên độn thêm một bit 0 là đủ, kết quả độ dài chuỗi tin là  $L_m = 18$ .

$$m = \begin{bmatrix} \begin{array}{c} \leftarrow \text{Direction of message} \\ \underbrace{10011100110000111}_{17 \text{ bit; } k=2} \quad \underbrace{0}_{\text{Để độ dài bản tin là}} \\ \underbrace{\hspace{10em}}_{\text{Độ dài là số nguyên lần } k} \end{array} \end{bmatrix}$$

**Độn loại 2:** Để đưa bộ lập mã về trạng thái khởi đầu toàn không, phải độn  $(K-1)k$  số không. Vì vậy chuỗi bit vào bộ lập mã khi này là.

$$m = \left[ \begin{array}{ccccccc} \underbrace{000000}_{\text{Loại 2}} & \underbrace{1\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 1\ 1}_{\text{Độ dài bản tin ban đầu}} & \underbrace{0}_{\text{Loại 1}} & \underbrace{000000}_{\text{Loại 2}} \\ \text{Hai loại độn số 0: Loại 1 để độ dài là số nguyên lần } k \text{ (} k=2 \text{)} \\ \text{Loại 2 độn } (K-1)k \text{ số không} = 6 \text{ để đưa bộ lập mã} \\ \text{về trạng thái toàn không} \end{array} \right]$$

**Ma trận tạo mã G**

$$G = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$n=3, K=4, L_m=18$  độ dài chuỗi bit ra sẽ là.

$$\left( \frac{L_m}{k} + K - 1 \right) \times n = \left( \frac{18}{2} + 4 - 1 \right) \times 3 = 36$$

**Kết quả chạy chương trình Matlab**

output= **NVD\_conv\_encd** ([0 0 1 0 1 0 0 1;0 0 0 0 0 0 0 1;1 0 0 0 0 0 0 1],2,[1 0 0 1 1 0 0 1 1 0 0 0 0 1 1 1])

output = 0 0 0 0 0 1 1 0 1 1 1 1 1 0 1 0 1 1 1 0 0 1 1 0 1 0 0 1 0 0 1 1 1 1 1

### ❖ Trường hợp 3:

Tìm chuỗi bit đầu ra theo phương pháp chuỗi tạo mã và đa thức tạo mã cho hình 5.11a. Tìm các bit ra của bộ lập mã được cho ở hình 5.11a khi chuỗi bit vào có  $L=2$  và một bit đuôi (độn) như sau:

$$m^{(1)} = (110)$$

$$m^{(2)} = (010)$$

**Chuỗi tạo mã**

Ta viết lại các dữ liệu vào ở dạng sau

$$m^{(1)} = (110)$$

$$m^{(2)} = m^{(1)} + m^{(2)} = (100)$$

$$m^{(3)} = (010)$$

Vậy có thể biểu diễn các vector bản tin tại các thời điểm  $i-p=0,1,2$  như sau

$$m_0 = (110)$$

$$m_1 = (101)$$

$$m_2 = (000)$$



Trong trường hợp này có ba đầu vào nên  $q=1,2,3$ ; một bộ nhớ và đầu vào bộ lập mã được nối đến các nhánh cộng nên  $p=0,1$ ; ba đầu ra nên  $j=1,2,3$ .

Đối với đầu vào thứ nhất  $q=1$  có được đáp ứng của ba nhánh ra như sau:

$$g_1^{(1)} = (0,1)$$

$$g_2^{(1)} = (0,1)$$

$$g_3^{(1)} = (0,0)$$

Đối với đầu vào  $q=2$

$$g_1^{(2)} = (1,0)$$

$$g_2^{(2)} = (0,0)$$

$$g_3^{(2)} = (0,1)$$

Tương tự đối với đầu vào  $q=3$

$$g_1^{(3)} = (0,1)$$

$$g_2^{(3)} = (1,0)$$

$$g_3^{(3)} = (0,1)$$

Từ các đáp ứng trên viết các ma trận kết nối như sau:

$$G_0 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad G_1 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

**Ta được**

$$c_0 = m_0 \cdot G_0 = (101)$$

$$c_1 = m_1 \cdot G_0 + m_0 \cdot G_1 = (100)$$

$$c_2 = m_2 \cdot G_0 + m_1 \cdot G_1 = (011)$$

Ghép chung các bit ở ba thời điểm trên ta được các bit đầu ra:

$$c = (101 \quad 100 \quad 011)$$

### Đa thức tạo mã

Để tính toán đầu ra bằng đa thức tạo mã trước hết cần phải biểu diễn các bit ở đầu vào thứ  $q=1,2,3$  ở dạng các đa thức bản tin như sau:

$$m^{(1)}(x) = 1 + x$$

$$m^{(2)}(x) = m^{(1)}(x) + m^{(2)}(x) = 1$$

$$m^{(3)}(x) = m^{(2)}(x) = x$$

Đối với đầu ra  $j=1$  ta được các đa thức tạo mã như sau:

$$g_1^{(1)}(x) = x$$

$$g_1^{(2)}(x) = 1$$

$$g_1^{(3)}(x) = x$$

Đối với đầu ra  $j=2$  ta được các đa thức tạo mã như sau:

$$g_2^{(1)}(x) = x$$

$$g_2^{(2)}(x) = 0$$

$$g_2^{(3)}(x) = 1$$

Đối với đầu ra  $j=3$  ta được các đa thức tạo mã như sau:

$$g_3^{(1)}(x) = 0$$

$$g_3^{(2)}(x) = 1$$

$$g_3^{(3)}(x) = x$$

**Ta để tính chuỗi bit ra của nhánh  $j$**

Với  $j=1$  ta được

$$\begin{aligned} c_j(x) &= \underbrace{\sum_{q=1}^k m^{(q)}(x) \cdot g_j^{(q)}(x)}_{\substack{j=1; k=3 \\ \Downarrow}} \\ c_1(x) &= \sum_{q=1}^3 m^{(q)}(x) \cdot g_1^{(q)}(x) \\ &= m^{(1)}(x) \cdot g_1^{(1)}(x) + m^{(2)}(x) \cdot g_1^{(2)}(x) + m^{(3)}(x) \cdot g_1^{(3)}(x) \\ &= (1+x)x + 1 \cdot 1 + x \cdot x = 1 + x \end{aligned}$$

$\Rightarrow$  Vậy  $c_1=(110)$

Với  $j=2$  ta được

$$\begin{aligned} c_j(x) &= \underbrace{\sum_{q=1}^k m^{(q)}(x) \cdot g_j^{(q)}(x)}_{\substack{j=2; k=3 \\ \Downarrow}} \\ c_2(x) &= \sum_{q=1}^3 m^{(q)}(x) \cdot g_2^{(q)}(x) \\ &= m^{(1)}(x) \cdot g_2^{(1)}(x) + m^{(2)}(x) \cdot g_2^{(2)}(x) + m^{(3)}(x) \cdot g_2^{(3)}(x) \\ &= (1+x)x + 1 \cdot 0 + x \cdot 1 = x^2 \end{aligned}$$

$\Rightarrow$  Vậy  $c_2=(001)$

Với  $j=3$  ta được:

$$\begin{aligned} c_j(x) &= \underbrace{\sum_{q=1}^k m^{(q)}(x) \cdot g_j^{(q)}(x)}_{\substack{j=3; k=3 \\ \Downarrow}} \\ c_3(x) &= \sum_{q=1}^3 m^{(q)}(x) \cdot g_3^{(q)}(x) \\ &= m^{(1)}(x) \cdot g_3^{(1)}(x) + m^{(2)}(x) \cdot g_3^{(2)}(x) + m^{(3)}(x) \cdot g_3^{(3)}(x) \\ &= (1+x) \cdot 0 + 1 \cdot 1 + x \cdot x = 1 + x^2 \end{aligned}$$

$\Rightarrow$  Vậy  $c_3=(101)$

Ghép các bit ở đầu ra nói trên ta được chuỗi bit đầu ra bộ lập mã

**c=(101 101 011)**

```
function output = NVD_COV_Encoder(g,k0,input)

% cnv_encd(g,k0,input)
% determines the output sequence of a binary convolutional encoder
% g is the generator matrix of the convolutional code
% with n0 rows and l*k0 columns. Its rows are g1,g2,...,gn.
% k0 is the number of bits entering the encoder at each clock cycle.
% input is the binary input seq.

% Check to see if extra zero-padding is necessary.
if rem(length(input),k0) > 0
    input = [input,zeros(size(1:k0-rem(length(input),k0)))];
end
n =length(input)/k0;
% Check the size of matrix g.
if rem(size(g,2),k0) > 0
    error('Error, g is not of the right size.')
end
% Determine l and n0.
l = size(g,2)/k0;
n0 = size(g,1);
% add extra zeros
u = [zeros(size(1:(l-1)*k0)),input,zeros(size(1:(l-1)*k0))];

% Generate uu, a matrix whose columns are the contents of
% conv. encoder at various clock cycles.
u1 = u(1*k0:-1:1);
for i=1:n+1-2
    u1 = [u1,u((i+1)*k0:-1:i*k0+1)];
end
uu=reshape(u1,l*k0,n+1-1);
% Determine the output
output = reshape(rem(g*uu,2),1,n0*(l+n-1));
```

#### Bài 34:

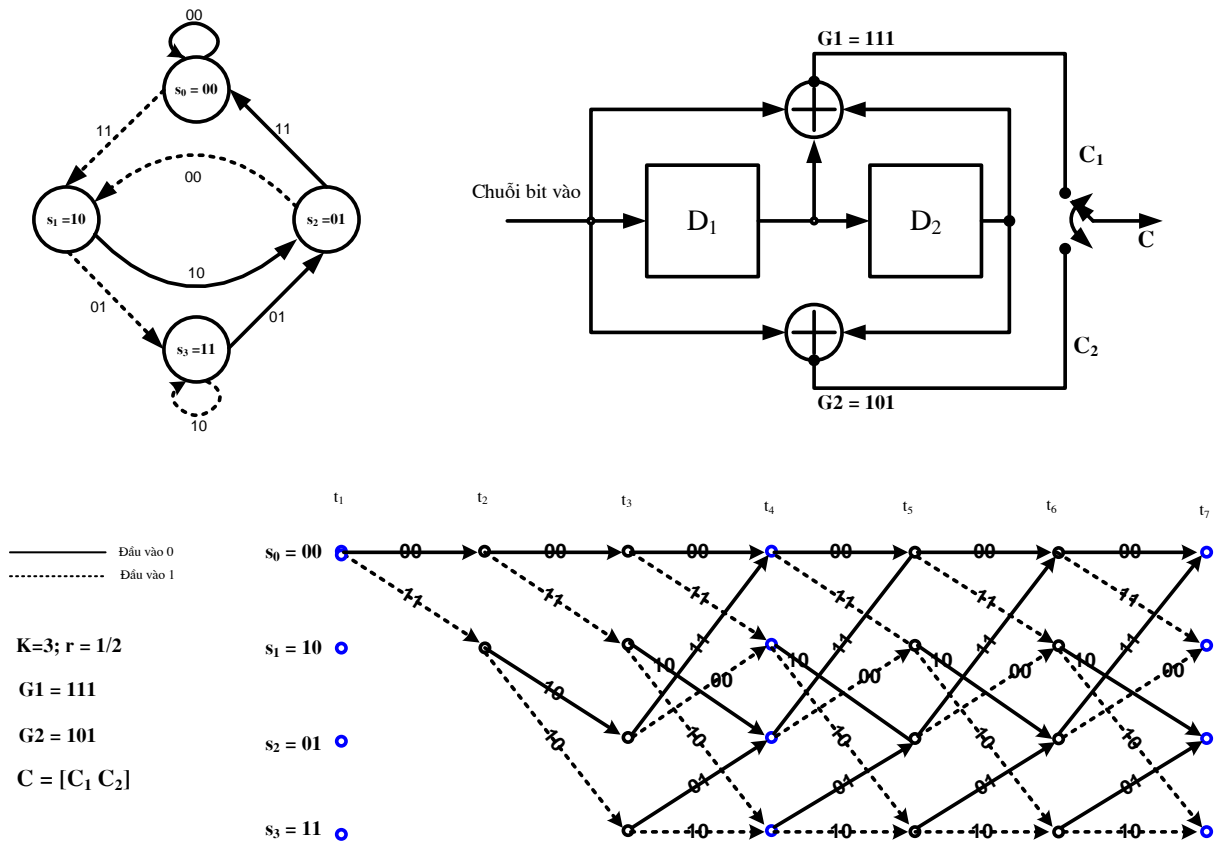
Hãy đọc mã chương trình **NVD\_COV\_Dencoder** để khảo sát quá trình giải mã xoắn quyết định cứng Viterbi, phân tích các tham số đặc trưng của mã, chạy chương trình, thay đổi các tham số đầu vào cho chương trình và phân tích kết quả. Đặc biệt vẽ lưu đồ giải thuật.

#### Hướng dẫn giải:

Xác định chuỗi mã thu V và ma trận tạo mã G, tham số k. Sau đó kiểm tra kết quả bằng cách chạy chương trình. Kiểm tra các kết quả cho tất cả các bài tập giải mã xoắn bằng cách chạy chương trình. Đặc biệt hãy so sánh với các trường hợp dưới đây.

##### ❖ Trường hợp 1:

Xét giải mã cho mã xoắn cho sơ đồ hình 1 dưới đây. Theo đó có  $k=1$ ,  $n = 2$  và  $L=3$  theo đó có 4 trạng thái có thể có được thể hiện bởi biểu đồ trạng thái và biểu đồ lưới tương ứng.



**Hình 1.** Sơ đồ và biểu đồ trạng thái, biểu đồ lưới

Chuỗi tin đầu vào bộ lập mã là

$$m = [1 \ 0 \ 1 \ 0 \ 1 \ 1]$$

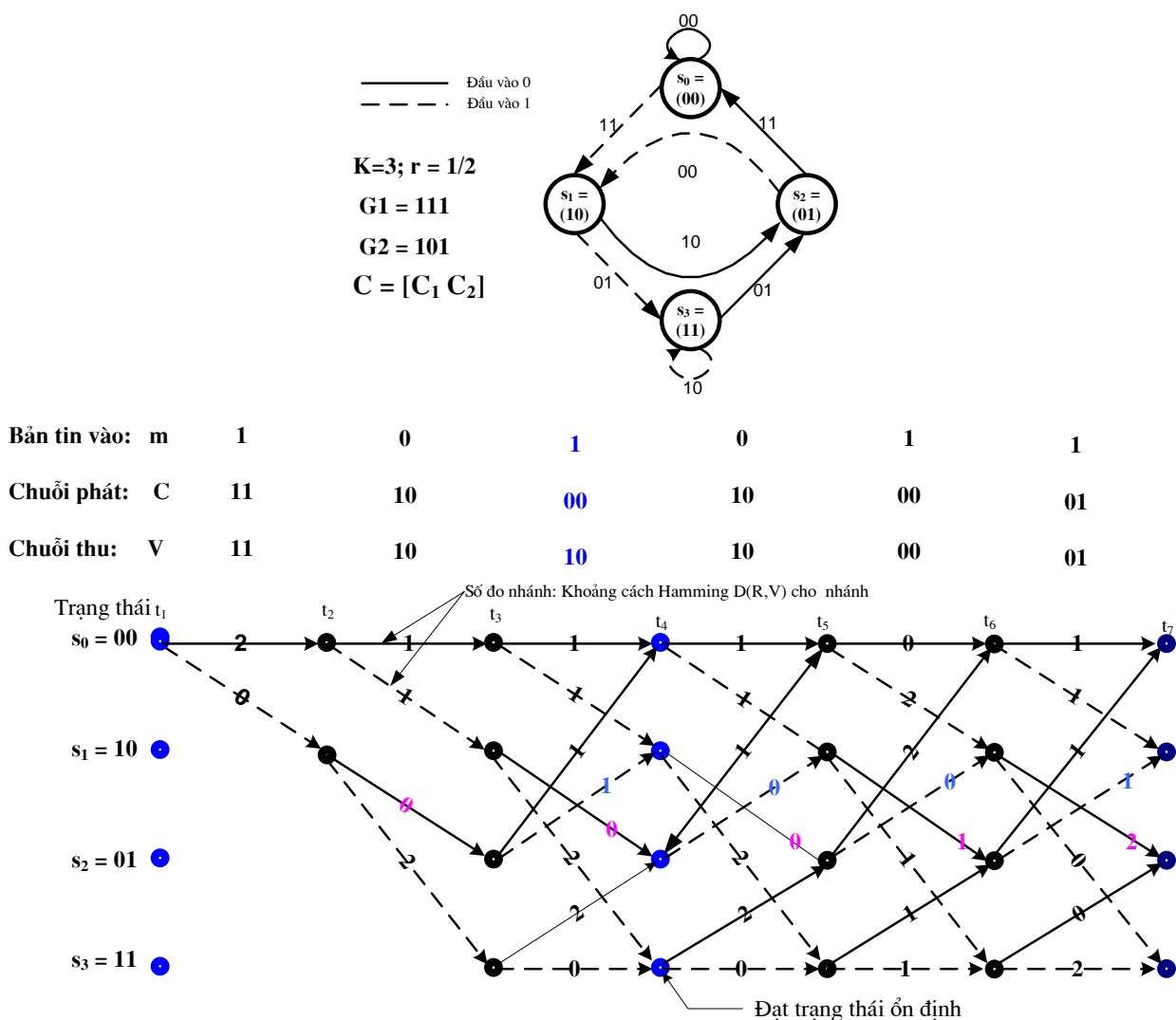
Chuỗi vector phát (từ mã phát) được cho bởi.

$$C = [1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1]$$

Giả sử chuỗi thu là

$$V = \left[ 1 \ 1 \ 1 \ 0 \ \underbrace{1}_{\text{Error}} \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \right]$$

Theo đó ta có biểu đồ lưới được cho ở hình PL3.6.



**Hình 2.** Biểu đồ lưới với các số đo nhánh được cho bởi khoảng cách Hamming

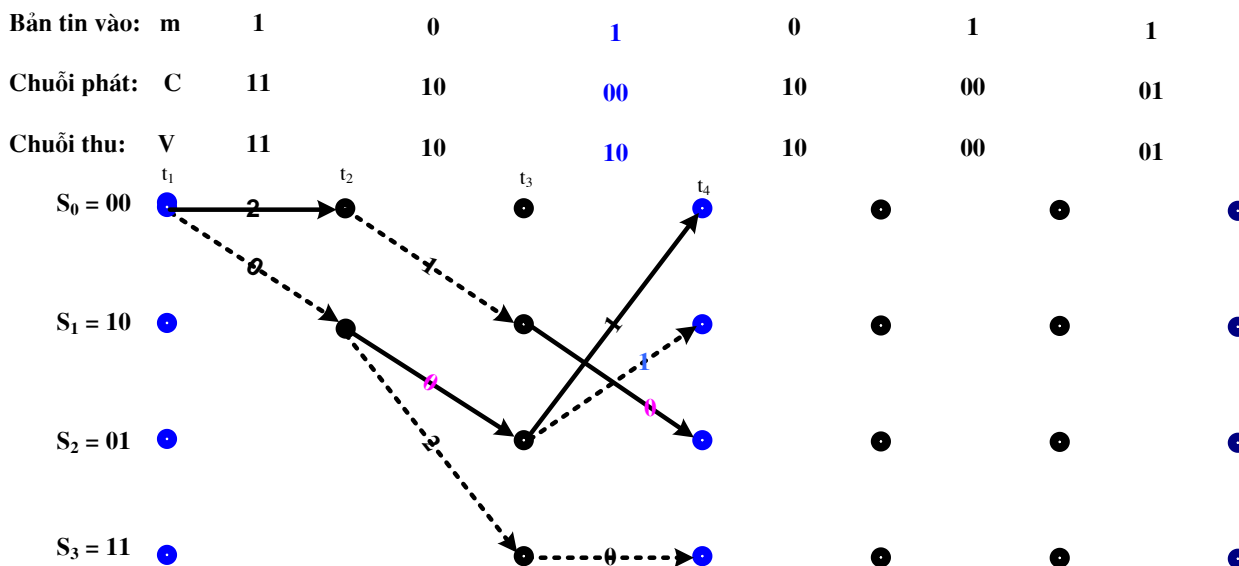
Với mỗi nhánh biểu đồ lưới ở hình 2 có một số đo nhánh được xác định bởi khoảng cách Hamming giữa  $n$  bit trên mỗi nhánh và  $n$  bit của chuỗi bit thu  $V$ . Ví dụ: hai bit thu đầu tiên là 11 tương ứng có khoảng cách Hamming là 2 và 0 so với 2 bit tương ứng với sự chuyển dịch trạng thái từ  $S_0 \Rightarrow S_0$  và  $S_0 \Rightarrow S_1$ . Tại thời điểm  $t = t_4$  đạt được trạng thái ổn định của lưới và trạng thái khởi đầu của bộ lập mã là  $S_0 = (00)$ . Nếu hai đường dẫn nào đó cùng hội nhập vào một nút trạng thái, thì đường dẫn có khoảng cách Hamming cực tiểu so với chuỗi bit thu tại thời điểm đó được giữ lại còn đường dẫn kia bị loại bỏ. Tại mỗi thời điểm  $t = t_i$  có  $2^{K-1}$  nút trạng thái, trong đó  $K$  là độ dài hạn chế của bộ lập mã. Quá trình giải mã được thực hiện như sau:

➤ **Tại thời điểm  $t = t_4$**

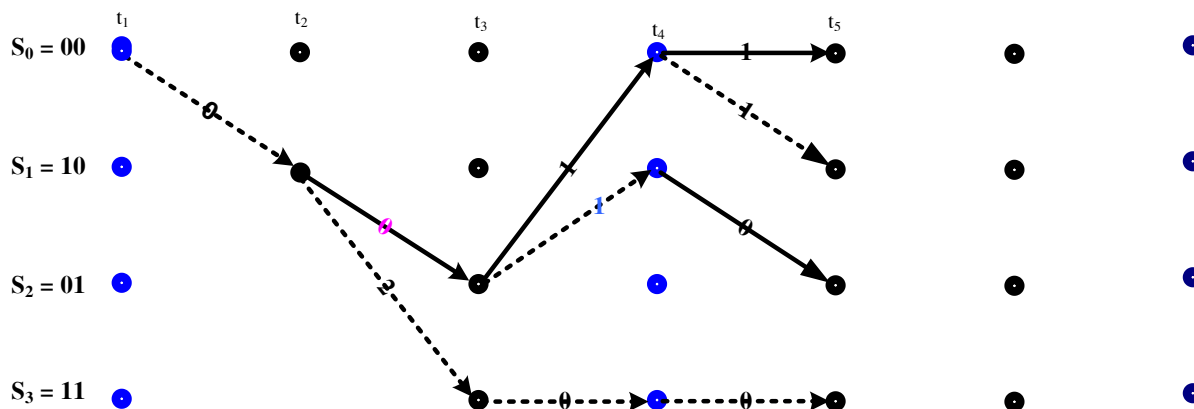
Các đường dẫn sống sót ở mỗi trạng thái tại thời điểm  $t = t_4$  được cho ở hình 3 (a). Đường dẫn sống sót ở trạng thái a tại thời điểm  $t = t_4$  là đường dẫn đi qua các trạng thái  $S_0 \Rightarrow S_1 \Rightarrow S_2 \Rightarrow S_0$  với tổng khoảng cách Hamming  $0+0+1=1$ , được chọn so với đường dẫn đi qua các trạng thái  $S_0 \Rightarrow S_0 \Rightarrow S_0 \Rightarrow S_0$  có tổng khoảng cách Hamming là  $2+1+1=4$ . Tại trạng thái b (nút thứ hai) tại thời điểm  $t = t_4$ , đường dẫn sống sót  $S_0 \Rightarrow S_1 \Rightarrow S_2 \Rightarrow S_1$  có tổng khoảng cách Hamming bằng 1 và loại bỏ đường dẫn  $S_0 \Rightarrow S_0 \Rightarrow S_0 \Rightarrow S_1$  có tổng khoảng cách Hamming là 4. Lưu ý rằng, khi này vẫn chưa quyết định được bit vào đầu tiên.

➤ **Tại thời điểm  $t=t_5$**

Các đường dẫn sống sót tại mỗi trạng thái được cho ở hình 4 (b) được xác định theo các khoảng cách Hamming mỗi khi chuyển dịch trạng thái. Do tất cả các đường dẫn sống sót đều có sự chuyển dịch trạng thái từ  $S_0 \Rightarrow S_1$  (từ  $t_1$  đến  $t_2$ ) nên quyết định bit đầu vào là 1  $\Rightarrow$  vì vậy khoảng thời gian trễ giải mã từ  $t_1$  đến  $t_5$  đặc trưng cho cặp sơ đồ lập mã và giải mã này.



a) Các đường dẫn sống sót tại thời điểm  $t = t_4$



b) Các đường dẫn sống sót tại thời điểm  $t = t_5$

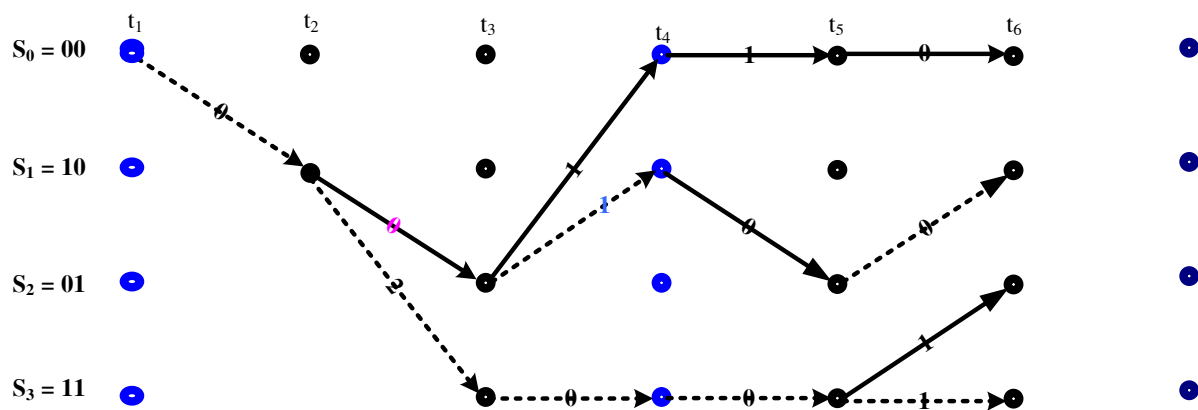
**Hình 4.** Các đường dẫn sống sót

➤ **Tại thời điểm  $t=t_6$**

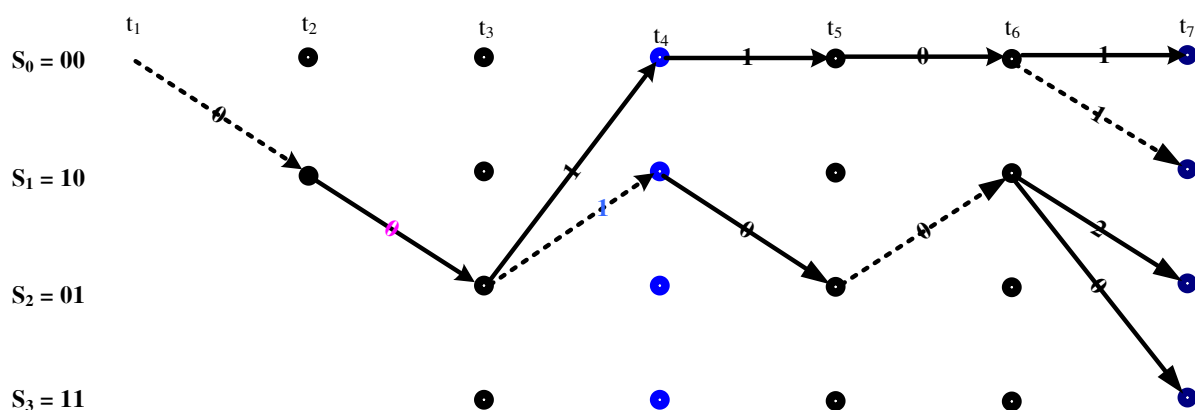
Các đường dẫn sống sót được cho ở hình 5 (c) tại nút trạng thái c và nút d đồng xác suất ( $P=0,5$ ) vì các khoảng cách Hamming đều bằng nhau.

➤ **Tại thời điểm  $t=t_7$**

Thực hiện tương tự nhận được các đường dẫn sống sót như được cho ở hình 6(d)



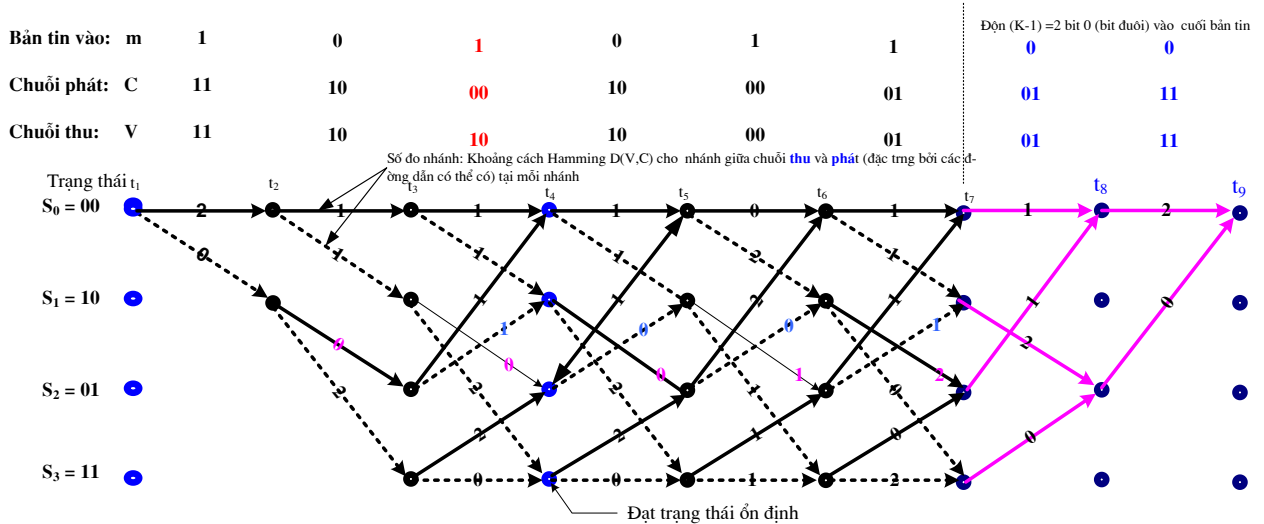
c) Các đường dẫn sóng sót tại thời điểm  $t = t_6$



d) Các đường dẫn sóng sót tại thời điểm  $t = t_7$

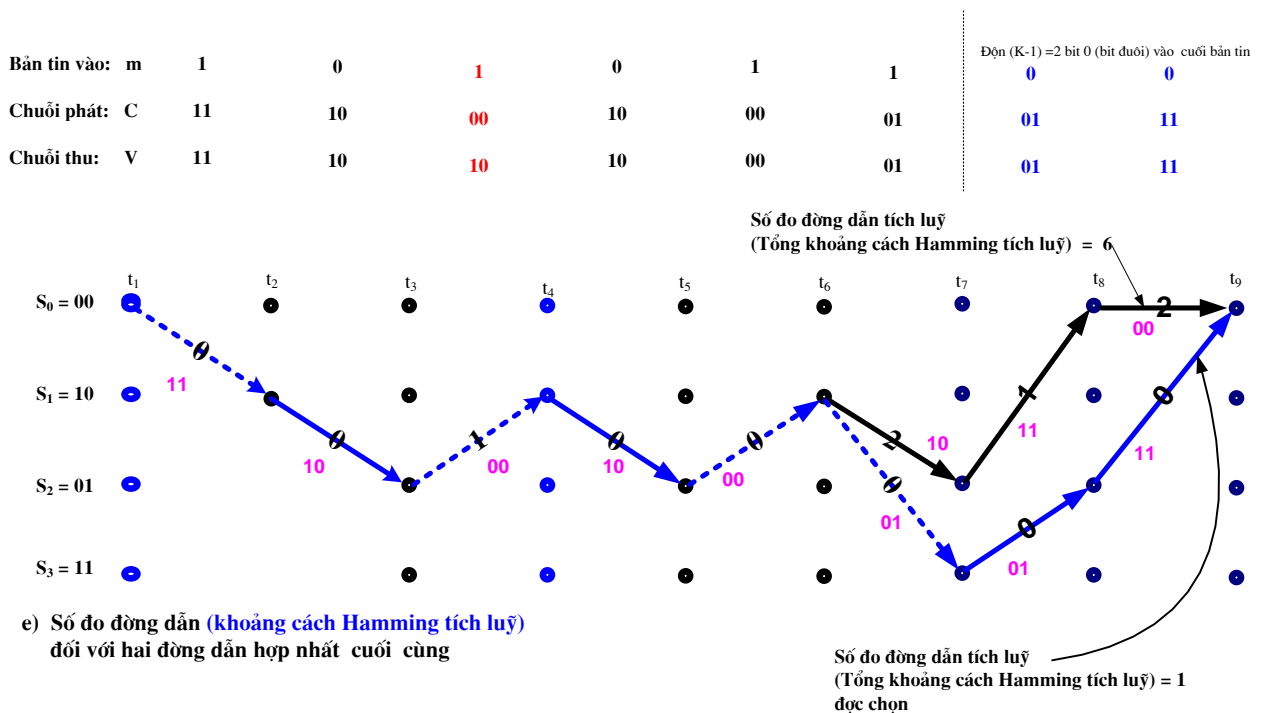
Hình 5. Các đường dẫn sóng sót

Thấy rõ, thuật toán Viterbi yêu cầu phải lưu trữ đường dẫn khoảng cách cực tiểu tại mỗi nút trạng thái cũng như số đo đường dẫn của nó, khoảng cách cực tiểu so với chuỗi thu. Hơn nữa, tại mỗi bước cần phải so sánh giữa hai đường dẫn cùng hội nhập vào một nút trạng thái, theo đó cần phải bốn lần so sánh (đối với ví dụ này). Để tránh dùng quá nhiều bộ nhớ ở phía phát thực hiện phân đoạn chuỗi bit và đưa thêm  $(K-1)k=2$  số 0 (các bit đuôi) vào các phân đoạn để đưa trạng thái cuối cùng về trạng thái  $S_0 = (00) \Rightarrow$  đường dẫn sóng sót cuối cùng (kết quả) là đường dẫn kết thúc tại node trạng thái toàn không  $S_0=(00)$ . Như vậy việc thêm  $(K-1)k$  số 0 vào cuối chuỗi bản tin dẫn đến đã ép các nút trạng thái có thể có về trạng thái toàn không. Cụ thể trong ví dụ này các trạng thái  $S_0, S_1, S_2, S_3$ , chuyển về trạng thái a như sau:  $S_0 \Rightarrow S_0 \Rightarrow S_0$ ;  $S_1 \Rightarrow S_2 \Rightarrow S_0$ ;  $S_2 \Rightarrow S_0 \Rightarrow S_0$ ;  $S_3 \Rightarrow S_2 \Rightarrow S_0$ ;



Hình 6. Biểu đồ lưới cho thấy sự hội tụ các trạng thái về trạng thái  $S_0 = (00)$ .

Như vậy đến thời điểm  $t_9$ , mới nhận được đường dẫn tối ưu. Kết quả ta được đường dẫn tối ưu là đường dẫn mà có tổng khoảng cách Hamming tại  $t_9$  trạng thái a nhỏ nhất.



Kết quả:  $C^{(m)} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$

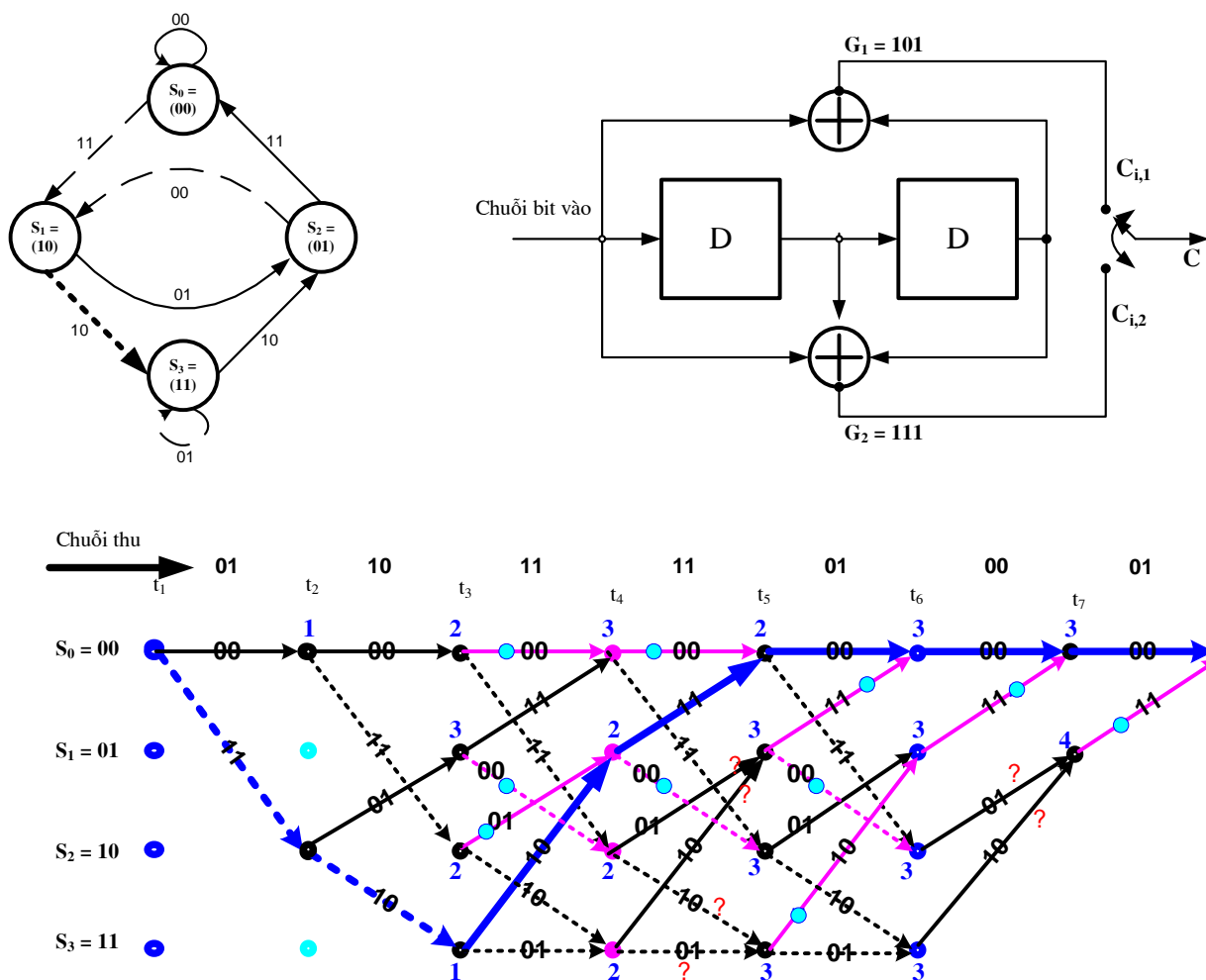
Hình 7. Đường dẫn tối ưu có khoảng cách Hamming nhỏ nhất được chọn.

## ❖ Trường hợp 2

Sơ đồ lập mã được cho ở hình 8. Hãy dùng phương pháp giải mã Viterbi theo quyết định cứng để giải mã chuỗi thu sau

$$V = 01101111010001$$





Hình 8. Sơ đồ khối, biểu đồ trạng thái và biểu đồ lưới

➤ **Các tham số đặc trưng**

✓  $M=3$ ;  $K=3$ ;  $n=2$ ;  $k=1$

✓ Độ dài chuỗi thu  $V = 14 \Rightarrow (L+K-1) = 7 \Rightarrow$  phải vẽ lưới có độ sâu là 7 được cho ở hình PL3.10  $\Rightarrow$  chuỗi bit tin phát là  $L=5$

Cũng cần lưu ý rằng do chuỗi bit tin đầu vào đã được đệm  $(K-1)k$  bit "0" = 2 bit 0, vì thế hai tầng cuối cùng của lưới chỉ cần vẽ các nhánh tương ứng với các đầu vào toàn bit không, nghĩa là độ dài thực tế của chuỗi tin vào là 5, sau khi đệm 2 bit 0 được tăng lên thành 7. Biểu đồ lưới cho trường hợp này được cho ở hình 8.

❖ **Chuỗi thu  $V$  cũng được cho ở hình 8**

➤ **Giải mã quyết định cứng**

- ✓ Lưu ý rằng trong quá trình vẽ lưới ở hai tầng (stage) cuối cùng, chỉ xét các đầu vào zeros tới bộ lập mã. (hai tầng cuối cùng, không có các đường nét đứt tương ứng với các đầu vào bit 1).
- ✓ Số đo trạng thái toàn không ban đầu được đặt là 0, số đo tầng kế tiếp được tính. Trong bước này thì chỉ có một nhánh (Branch) đi vào mỗi trạng thái (State), vì vậy sẽ không có sự so sánh và các số đo (metrics số đo là khoảng cách Hamming giữa một phần chuỗi thu và các nhánh của lưới) được cộng với số đo của trạng thái state trước đó.

- ✓ Tầng tiếp theo, không có sự so sánh.
- ✓ Trong tầng thứ 4, vì lần đầu tiên có hai nhánh đều đi vào mỗi trạng thái *state*, nghĩa phải thực hiện so sánh và chọn đường sống sót. Từ hai nhánh đi vào mỗi trạng thái *state*, nhánh tương ứng với số đo tích lũy tổng cuối cùng nhỏ nhất được giữ lại làm đường dẫn sống sót, đường còn lại bị xóa (được đánh dấu bởi một dấu x trên lưới).
- ✓ Nếu ở một tầng nào đó, hai đường dẫn cùng cho kết quả số đo bằng nhau, thì chúng đều có thể là đường dẫn sống sót, những trường hợp như vậy phải được đánh dấu bởi dấu hỏi '?' trong biểu đồ lưới.
- ✓ Thủ tục trên được diễn ra cho đến trạng thái toàn không cuối cùng của lưới; sau đó khởi đầu từ trạng thái đó (trạng thái toàn không), *di chuyển dịch dọc theo các đường dẫn sống sót tới trạng thái toàn không khởi đầu (Initial)*. Đường dẫn này được ký hiệu bằng một đường dẫn đậm thông qua lưới, là **đường dẫn tối ưu là đường dẫn có khoảng cách Hamming cực tiểu so với chuỗi thu V**.
- ✓ Chuỗi bit đầu vào tương ứng đường dẫn này là 1100000, trong đó hai bit cuối cùng là hai bit không mang thông tin nhưng được cộng thêm vào để đưa bộ lập mã trở về trạng thái toàn không. Vì vậy chuỗi bit thông tin là 11000.
- ✓ Từ mã tương ứng với đường dẫn được chọn là 11101011000000, là từ mã có khoảng cách Hamming = 4 so với chuỗi thu. Tất cả các đường dẫn khác thông qua lưới có khoảng cách Hamming so với chuỗi bit thu lớn hơn 4 đều không phải là đường dẫn tối ưu.

➤ **Giải mã quyết định mềm**

Quy trình tương tự với giải mã quyết định cứng song khoảng cách Euclid bình phương thay thế cho khoảng cách Hamming.

```
function [decoder_output] = NVD_COV_Dencoder(G,k,channel_output);
n = size(G,1);

% check the sizes
if rem(size(G,2),k) ~=0
    error('Size of G and k do not agree')
end
if rem(size(channel_output,2),n) ~=0
    error('channel output not of the right size')
end
L = size(G,2)/k;
number_of_states=2^((L-1)*k);

% Generate state transition matrix, output matrix, and input matrix.
for j=0:number_of_states-1
    for l=0:2^k-1
        [next_state,memory_contents]=nxt_stat(j,l,L,k);
        input(j+1,next_state+1)=l;
        branch_output=rem(memory_contents*G',2);
        nextstate(j+1,l+1)=next_state;
        output(j+1,l+1)=bin2dec(branch_output);
    end
end
state_metric=zeros(number_of_states,2);
depth_of_trellis=length(channel_output)/n;
channel_output_matrix=reshape(channel_output,n,depth_of_trellis);
survivor_state=zeros(number_of_states,depth_of_trellis+1);
% Start decoding of non-tail channel outputs.
```

```
for i=1:depth_of_trellis-L+1
    flag=zeros(1,number_of_states);
    if i <= L
        step=2^((L-i)*k);
    else
        step=1;
    end
    for j=0:step:number_of_states-1
        for l=0:2^k-1
            branch_metric=0;
            binary_output=dec2bin(output(j+1,l+1),n);
            for ll=1:n

branch_metric=branch_metric+metric(channel_output_matrix(ll,i),binary_output(
ll));
            end
            if((state_metric(nextstate(j+1,l+1)+1,2) > state_metric(j+1,1)...
+branch_metric) | flag(nextstate(j+1,l+1)+1)==0)
                state_metric(nextstate(j+1,l+1)+1,2) = state_metric(j+1,1)+branch_metric;
                survivor_state(nextstate(j+1,l+1)+1,i+1)=j;
                flag(nextstate(j+1,l+1)+1)=1;
            end
        end
    end
    state_metric=state_metric(:,2:-1:1);
end
% Start decoding of the tail channel-outputs.
for i=depth_of_trellis-L+2:depth_of_trellis
    flag=zeros(1,number_of_states);
    last_stop=number_of_states/(2^((i-depth_of_trellis+L-2)*k));
    for j=0:last_stop-1
        branch_metric=0;
        binary_output=dec2bin(output(j+1,1),n);
        for ll=1:n

branch_metric=branch_metric+metric(channel_output_matrix(ll,i),binary_output(
ll));
        end
        if((state_metric(nextstate(j+1,1)+1,2) > state_metric(j+1,1)...
+branch_metric) | flag(nextstate(j+1,1)+1)==0)
            state_metric(nextstate(j+1,1)+1,2) = state_metric(j+1,1)+branch_metric;
            survivor_state(nextstate(j+1,1)+1,i+1)=j;
            flag(nextstate(j+1,1)+1)=1;
        end
    end
    state_metric=state_metric(:,2:-1:1);
end

% Generate the decoder output from the optimal path.
state_sequence=zeros(1,depth_of_trellis+1);
state_sequence(1,depth_of_trellis)=survivor_state(1,depth_of_trellis+1);
for i=1:depth_of_trellis
    state_sequence(1,depth_of_trellis-
i+1)=survivor_state((state_sequence(1,depth_of_trellis+2-i)...
+1),depth_of_trellis-i+2);
end
decoder_output_matrix=zeros(k,depth_of_trellis-L+1);
for i=1:depth_of_trellis-L+1
    dec_output_dec=input(state_sequence(1,i)+1,state_sequence(1,i+1)+1);
    dec_output_bin=dec2bin(dec_output_dec,k);
    decoder_output_matrix(:,i)=dec_output_bin(k:-1:1)';
end
decoder_output=reshape(decoder_output_matrix,1,k*(depth_of_trellis-L+1));
cumulated_metric=state_metric(1,1);
```

```
function y=bin2deci(x)
l=length(x);
y=(l-1:-1:0);
y=2.^y;
y=x*y';
function y=deci2bin(x,l)
y = zeros(1,l);
i = 1;
while x>=0 & i<=l
    y(i)=rem(x,2);
    x=(x-y(i))/2;
    i=i+1;
end
y=y(1:-1:1);
```

## CHƯƠNG 6

### Bài 1:

Cho một xung chịu méo kênh tại đầu vào bộ cân bằng là

$$x(t) = \frac{1}{1 + (2t/T)^2}$$

trong đó  $1/T$  là tốc độ ký hiệu. Xung được lấy mẫu tại tốc độ  $2/T$  và được sửa bằng bộ cân bằng cường ép không.

Hãy xác định các hệ số của bộ cân bằng cường ép không 5 nhánh.

### Hướng dẫn giải

Theo (6.6.8), bộ cân bằng cường ép không phải thỏa mãn các phương trình

$$q(mT) = \sum_{n=-2}^2 c_n x\left(mT - \frac{nT}{2}\right) = \begin{cases} 1, & m=0 \\ 0, & m=\pm 1, \pm 2 \end{cases}$$

Ma trận  $\mathbf{X}$  có các phần tử  $x(mT - nT/2)$  được tính là

$$\mathbf{X} = \begin{bmatrix} \frac{1}{5} & \frac{1}{10} & \frac{1}{17} & \frac{1}{26} & \frac{1}{37} \\ 1 & \frac{1}{2} & \frac{1}{5} & \frac{1}{10} & \frac{1}{17} \\ \frac{1}{5} & \frac{1}{2} & 1 & \frac{1}{2} & \frac{1}{5} \\ \frac{1}{17} & \frac{1}{10} & \frac{1}{5} & \frac{1}{2} & 1 \\ \frac{1}{37} & \frac{1}{26} & \frac{1}{17} & \frac{1}{10} & \frac{1}{5} \end{bmatrix}$$

Véc-tơ hệ số  $\mathbf{c}$  và véc-tơ  $\mathbf{q}$  được cho như sau

$$\mathbf{c} = \begin{bmatrix} c_{-2} \\ c_{-1} \\ c_0 \\ c_1 \\ c_2 \end{bmatrix}; \quad \mathbf{q} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Vì vậy, tìm được các hệ số của bộ cân bằng bằng cách giải các phương trình tuyến tính  $\mathbf{Xc} = \mathbf{q}$  theo cách lấy nghịch đảo ma trận  $\mathbf{X}$

$$\mathbf{c}_{\text{opt}} = \mathbf{X}^{-1} \mathbf{q} = \begin{bmatrix} -2,2 \\ 4,9 \\ -3 \\ 4,9 \\ -2,2 \end{bmatrix}$$

Chương trình Matlab được cho ở file NVD\_CS610.m. Hãy chạy chương trình và phân tích kết quả.

```
% NVD_CS10.m
T=1;
Fs=2/T;
Ts=1/Fs;
c_opt=[-2.2 4.9 -3 4.9 -2.2];
t=-5*T:T/2:5*T;
x=1./(1+((2/T)*t).^2); % sampled pulse
equalized_x=filter(c_opt,1,[x 0 0]); % since there will be a delay of two
% samples at the output

% to take care of the delay
equalized_x=equalized_x(3:length(equalized_x));
% Now, let us downsample the equalizer output.
for i=1:2:length(equalized_x),
    downsampled_equalizer_output((i+1)/2)=equalized_x(i);
end;
% Plotting commands follow.
```

## Bài 2:

Cho xung bị méo kênh  $x(t)$  như trong bài tập 6.10, nhưng ở đây hãy thiết kế bộ cân bằng 5 nhánh dựa trên tiêu chí MMSE. Các ký hiệu thông tin có trung bình bằng 0, phương sai bằng 1 và không tương quan nhau, nghĩa là  $E(a_n)=0$ ,

$E(a_n a_m)=0$  ( $n \neq m$ ),  $E(|a_n|^2)=1$ . Tạp âm cộng  $v(t)$  có trung bình bằng 0 và hàm tự

tương quan là:  $\phi_{vv}(\tau)=\frac{N_0}{2}\delta(\tau)$

## Hướng dẫn giải

Bằng cách giải (6.6.16) với  $K=2$  và  $\tau=T/2$ , ta tìm được các hệ số nhánh của bộ cân bằng. Ma trận với các phần tử  $R_y(n-k)$  khi này chỉ là

$$\mathbf{R}_y = \mathbf{X}^T \mathbf{X} + \frac{N_0}{2} \mathbf{I}$$

trong đó:  $\mathbf{X}$  được xác định từ bài 1;  $\mathbf{I}$  là ma trận đơn vị. Véc-tơ chứa các phần tử  $\mathbf{R}_{ay}(k)$ :

$$\mathbf{R}_{ay} = \begin{bmatrix} 1 \\ \frac{1}{5} \\ \frac{1}{2} \\ 1 \\ \frac{1}{2} \\ \frac{1}{5} \end{bmatrix}$$

Nhận được các hệ số bộ cân bằng bằng cách giải (6.6.16):

$$\mathbf{c}_{\text{opt}} = \begin{bmatrix} 0,0956 \\ -0,7347 \\ 1,6761 \\ -0,7347 \\ 0,0956 \end{bmatrix}$$

Chương trình mô phỏng được cho ở **NVD\_CS611.m**. Hãy chạy chương trình và phân tích kết quả mô phỏng.

```
%NVD_CS611.m
T=1;
for n=-2:2,
    for k=-2:2,
        temp=0;
        for i=-2:2, temp=temp+(1/(1+(n-i)^2))*(1/(1+(k-i)^2)); end;
        X(k+3,n+3)=temp;
    end;
end;
N0=0.01; % assuming that N0=0.01
Ry=X+(N0/2)*eye(5);
Riy=[1/5 1/2 1 1/2 1/5].';
c_opt=inv(Ry)*Riy; % optimal tap coefficients
% find the equalized pulse...
t=-3:1/2:3;
x=1./(1+(2*t/T).^2); % sampled pulse
equalized_pulse=conv(x,c_opt);
% Decimate the pulse to get the samples at the symbol rate.
decimated_equalized_pulse=equalized_pulse(1:2:length(equalized_pulse));
% Plotting command follows.
```

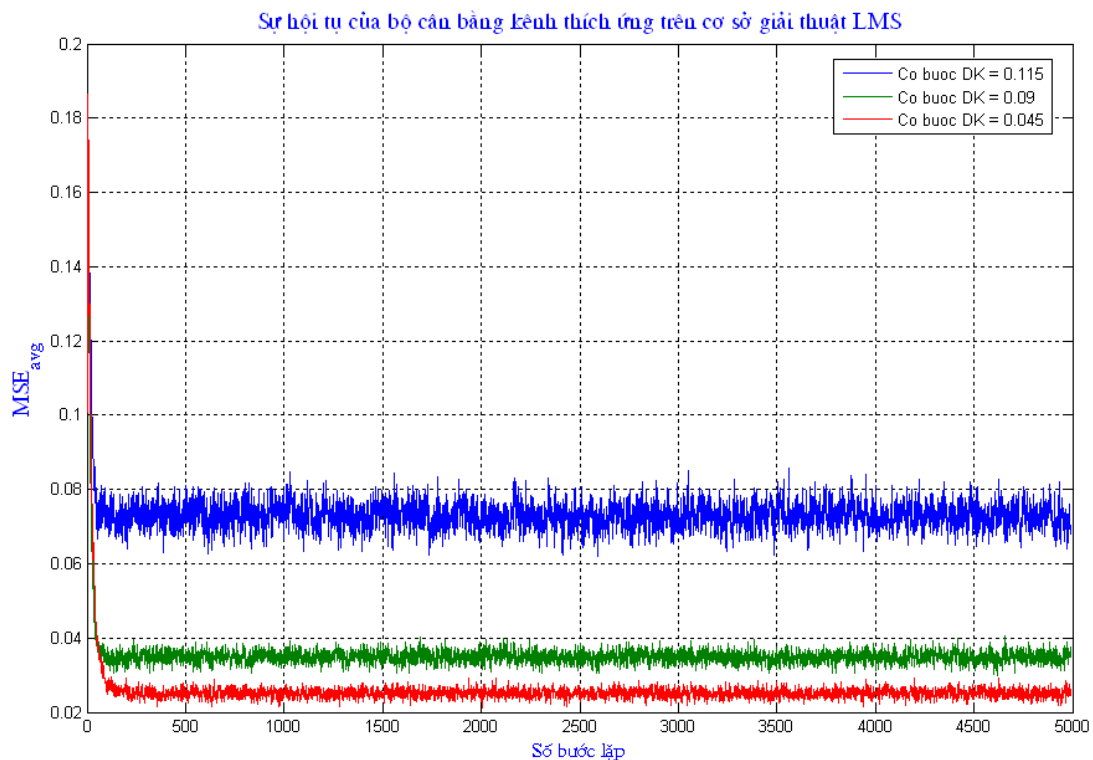
### Bài 3:

Hãy thực hiện một bộ cân bằng thích ứng dựa trên thuật toán LMS được cho trong (6.63). Số nhánh của kênh được chọn cho bộ cân bằng là  $2K + 1 = 11$ . Công suất tín hiệu thu cộng với tạp âm  $P_R$  được chuẩn hóa bằng 1. Đặc tuyến của kênh được cho bởi véc-tơ  $\mathbf{x}$  như sau

$$\mathbf{X} = [0,05; -0,063; 0,088; -0,126; -0,25; 0,9047; 0,25; 0; 0,126; 0,038; 0,088]$$

### Hướng dẫn giải

Các đặc tính hội tụ của thuật toán gradient ngẫu nhiên trong (6.63) được minh họa trên hình dưới đây. Các kết quả mô phỏng này được thực hiện đối với bộ cân bằng thích ứng 11 nhánh. Các kết quả mô phỏng về lỗi trung bình bình phương được lấy trung bình trên một số lần thực hiện. Từ kết quả mô phỏng cho thấy, khi  $\Delta$  giảm, sự hội tụ chậm hơn đôi chút nhưng đạt được MSE nhỏ hơn, cũng cho thấy rằng các hệ số nhánh được ước tính gần với  $\mathbf{c}_{\text{opt}}$  hơn. Chương trình mô phỏng được cho ở file **NVD\_CS612.m**.



*Các đặc tính hội tụ ban đầu của thuật toán LMS với các kích cỡ bước khác nhau*

#### **% function NVD\_CS612.m**

```
% y = filter(b,a,X) filters the data in vector X with
% the filter described by numerator coefficient vector b
% and denominator coefficient vector a.
% If a(1) is not equal to 1,
% filter normalizes the filter coefficients by a(1).
% If a(1) equals 0, filter returns an error.

clc;
clear all;
close all;
N = 5000; % length of the information sequence
K = 5;
actual_isi = [0.05 -0.063 0.088 -0.126 -0.25 0.9047 0.25 0 0.126 0.038 0.088];
sigma = 0.1; % 0.01
delta_vector = [0.118 0.07 0.036];
delta = delta_vector(3); % [0.115 0.09 0.045]
Num_of_realizations = 1000;
mse_av = zeros(1,N-2*K);
%=====
tic;
fpname = strcat(mfilename, '.txt');
fp=fopen(fpname,'a+');
fprintf(fp,'*****\n');
fprintf(fp,'+ + + Adaptive LMS Filter simulator is designed by Nguyen Viet
Dam - PTIT + + + \n');
fprintf(fp,'+ + + Please be patient. Wait a while to get the result. + + +
\n');
fprintf(fp,'*****\n\n');
h_1 = waitbar(0,'Sàng m« pháng ®,nh gi, hiÖu n`ng bé lác thÝch øng LMS');
%=====
```



```
for j=1:Num_of_realizations % Compute the average over a number of
realizations.
    % the information sequence
    for i=1:N
        if (rand<0.5)
            info(i)=-1;
        else
            info(i)=1;
        end
    end
    % the channel output
    y = filter(actual_isi,1,info); % Note
    Y_test = y;
    for i=1:2:N
        [noise(i) noise(i+1)] = NVD_gngauss(sigma);
    end
    y = y + noise;
    % Now the equalization part follows.
    estimated_c = [0 0 0 0 0 1 0 0 0 0]; % initial estimate of ISI
    for k=1:N-2*K
        y_k = y(k:k+2*K);
        z_k = estimated_c*y_k.';
        e_k = info(k)-z_k;
        estimated_c = estimated_c + delta*e_k*y_k;
        mse(k) = e_k^2;
    end
    mse_av = mse_av + mse;

    run_time(j) = toc; % note NVD
    waitbar(j/Num_of_realizations); % note NVD
end
close(h_1);

Thoi_gian_chay = toc/3600;
fprintf('***** The total time that used is : %3.4f hour *****\n',
Thoi_gian_chay);
fprintf(fp, '***** The total time that used is : %3.4f hour
*****\n',Thoi_gian_chay);
fclose(fp);

mse_av = mse_av/Num_of_realizations; % mean-squared error versus iterations

if delta==delta_vector(1)
    mse_av_1 = mse_av;
    mse_1 = mse;
    info_1 = info;
    Y_test_1 = Y_test;
    y_1 = y;
    noise_1 = noise;
    save c:\MATLAB704\work\NVD-NHL\MatResults\LMS1.mat mse_av_1 mse_1 info_1
Y_test_1 y_1 noise_1 actual_isi;
elseif delta==delta_vector(2)
    mse_av_2 = mse_av;
    mse_2 = mse;
    info_2 = info;
    Y_test_2 = Y_test;
    y_2 = y;
    noise_2 = noise;
    save c:\MATLAB704\work\NVD-NHL\MatResults\LMS2.mat mse_av_2 mse_2 info_2
Y_test_2 y_2 noise_2 actual_isi;
else
    mse_av_3 = mse_av;
    mse_3 = mse;
    info_3 = info;
```

```
Y_test_3 = Y_test;
y_3 = y;
noise_3 = noise;
save c:\MATLAB704\work\NVD-NHL\MatResults\LMS3.mat mse_av_3 mse_3 info_3
Y_test_3 y_3 noise_3 actual_isi;
end
% Plotting commands follow.
plot(mse_av);
xlabel('Sè b-íc lÆp','FontName','.VnTime','color','b','FontSize',12);
ylabel('MSE_a_v_g','FontName','.VnTime','color','b','FontSize',14);
title('Sù héi tô cña bé c@n b»ng kãnh thÝch øng trªn c- sè gi¶i thuËt LMS',...
'FontName','.VnTime','color','b','FontSize',14);
grid on;

% function y = NGVD_NHL_COMPASION
clear all;
close all;
clc;
load c:\MATLAB6p5\work\NVD-NHL\MatResults\LMS1.mat
load c:\MATLAB6p5\work\NVD-NHL\MatResults\LMS2.mat
load c:\MATLAB6p5\work\NVD-NHL\MatResults\LMS3.mat

figure(1)
xx=1:length(mse_av_1);
plot(xx,mse_av_1,xx,mse_av_2,xx,mse_av_3);
xlabel('Sè b-íc lÆp','FontName','.VnTime','color','b','FontSize',12);
ylabel('MSE_a_v_g','FontName','.VnTime','color','b','FontSize',14);
title('Sù héi tô cña bé c@n b»ng kãnh thÝch øng trªn c- sè gi¶i thuËt
LMS','FontName','.VnTime','color','b','FontSize',14);
legend('Co buoc DK = 0.115','Co buoc DK = 0.09','Co buoc DK = 0.045');
grid on;
```