

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍ MINH
KHOA VIỄN THÔNG 2
Bộ môn Vô Tuyến



BÀI THÍ NGHIỆM THỰC HÀNH

Môn: Đa Truy Nhập Vô Tuyến

Giảng viên biên soạn:

- PGS. TS. Võ Nguyễn Quốc Bảo
- ThS. Phạm Thanh Đàm

TP. Hồ Chí Minh, ngày 21 tháng 9 năm 2021

MỤC LỤC

BÀI THÍ NGHIỆM 1: KỸ THUẬT TRUY NHẬP KÊNH TRUYỀN THEO PHƯƠNG THỨC ALOHA VÀ SLOTTED ALOHA.....	3
BÀI THÍ NGHIỆM 2: KHẢO SÁT VÀ MÔ PHỎNG HỆ THỐNG TDMA.....	8
BÀI THÍ NGHIỆM 3: KHẢO SÁT VÀ MÔ PHỎNG HỆ THỐNG FDMA.....	14
BÀI THÍ NGHIỆM 4: KHẢO SÁT VÀ MÔ PHỎNG HỆ THỐNG CDMA.....	21

BÀI THÍ NGHIỆM 1: KỸ THUẬT TRUY NHẬP KÊNH TRUYỀN THEO PHƯƠNG THỨC ALOHA VÀ SLOTTED ALOHA

<u>Điểm:</u>	<u>Nhận xét:</u>
---------------------	-------------------------

Mục đích của bài thí nghiệm là giúp sinh viên hiểu được nguyên tắc của hệ thống truy nhập theo kỹ thuật Aloha và slotted Aloha.

PHẦN 1: CHUẨN BỊ TRƯỚC Ở NHÀ

Câu 1: Nghiên cứu nguyên tắc của hệ thống Aloha như bên dưới. Chứng minh lại các công thức

Giao thức Aloha là giao thức OSI lớp 2 cho mạng LAN với Broadcast topology. Nguyên tắc cơ bản của hệ thống Aloha là:

- Nếu các đầu cuối có dữ liệu thì tiến hành truyền phát dữ liệu. Sau khi gửi dữ liệu, đầu cuối đợi tín hiệu ACK để xác nhận trong một khoảng thời gian. Nếu sau một khoản thời gian đợi mà không có ACK, đầu cuối tự hiểu rằng gói dữ liệu đã mất.
- Nếu xảy ra xung đột (ít nhất hai đầu cuối cùng truyền tại một điểm thời gian) thì đều dừng truyền và sẽ tiến hành gửi lại sau một khoản thời gian ngẫu nhiên gọi là backoff

Trong bài thí nghiệm này chúng ta có hai phần, khảo sát hiệu năng của kỹ thuật ALOHA và mô phỏng kỹ thuật ALOHA.

Để mô phỏng kỹ thuật ALOHA, chúng ta gọi x là biến ngẫu nhiên thể hiện thời gian bắt đầu của các gói dữ liệu cần gửi của mạng ALOHA. Theo lý thuyết xác suất x sẽ theo phân bố Poisson và xác suất K gói dữ liệu phát trong khoản thời gian T là

$$\Pr(x = K) = \frac{(\lambda T)^K e^{-\lambda T}}{K!} \quad (1)$$

với λ là tốc độ đến trung bình của gói dữ liệu có đơn vị là số lượng gói dữ liệu trên giây. (packet per second) được biểu diễn lại như sau:

$$\lambda = \lambda_s + \lambda_r \quad (2)$$

với λ_s là tốc độ đến trung bình của gói dữ liệu gửi thành công và λ_r là tốc độ đến trung bình của gói dữ liệu gửi không thành công.

Gọi S là thông lượng kênh truyền (hay là lưu lượng chuẩn hóa của dữ liệu gửi thành công), có đơn vị là Erlangs, ta có

$$S = \frac{L\lambda_s}{R} \quad (3)$$

với L là chiều dài gói tin có đơn vị là bit và R là dung lượng kênh truyền có đơn vị là bit/s. Tương tự như vậy, ta có lưu lượng chuẩn hóa của kênh truyền là

$$G = \frac{L\lambda}{R} \quad (3)$$

Thời gian để truyền một gói tin với đơn vị là bit/s là

$$T = \frac{L}{R} \quad (4)$$

khi đó ta dễ dàng có được $S = \lambda_s T$ và $G = \lambda T$.

Trong hệ thống ALOHA, để một gói tin ở một đầu cuối bất kỳ truyền thành công trong khoảng thời gian T , thì phải không có gói tin nào được truyền trong khoảng thời gian T sau khoảng thời gian T trước. Nghĩa là chúng ta cần $2T$ giây cho một gói tin thành công. Do đó, xác suất thành công của một gói tin là

$$P_s = \Pr(K = 0) = \frac{(\lambda T)^0 e^{-2\lambda T}}{0!} = e^{-2\lambda T} \quad (5)$$

Ngoài ra ta sẽ thấy

$$P_s = \frac{\lambda_s}{\lambda_s + \lambda_r} \quad (6)$$

Kết hợp hai biểu thức ở trên, ta thấy rằng

$$S = Ge^{-2G} \quad (7)$$

Một câu hỏi đặt ra là S có giá trị lớn nhất khi nào? Để giải bài toán này chúng ta thực hiện đạo hàm S theo G , và giải phương trình $dS/dG = 0$, dẫn đến S đạt giá trị lớn nhất là $1/(2e) = 0.184$ khi $G = 1/2$.

Câu 2: Nguyên tắc của Slotted ALOHA là gì? Mối liên hệ giữa S và G của Slotted ALOHA là gì?

Câu 3: Khảo sát các hàm Matlab sau

TT	Hàm	Cú pháp và Chức năng?
1	rand	

2	sort	
3	diff	
4	sum	
5	plot	
6	legend	
7	xlabel	
8	ylabel	
9	title	

PHẦN 2: THỰC HIỆN TẠI PHÒNG THÍ NGHIỆM TRÊN PHẦN MỀM MATLAB

Câu 4: Giải thích mã nguồn Matlab như bên dưới.

Mã nguồn Matlab	Giải thích
<pre> % Số lượng gói tin trong đơn vị thời gian T N = 10^5; % Thời gian đến của N gói tin PacketArrivalTimes = rand(1,N); % Sắp xếp thời gian đến x = sort(PacketArrivalTimes); % Thời gian đến khác nhau giữa các gói tin trong (0,T) t1=diff(x); % Thời gian đến khác nhau giữa các gói tin trong (T,@T) x(2:end) = x(1:end-1); t2=diff(x); % Khởi tạo G G = 0.1:0.01:2.2; % Thời gian truyền của một gói tin T = G/N; </pre>	

```

%% Tính toán lý thuyết
S_t = G.*exp(-2*G);

%% Tính toán mô phỏng
% N_s: Số lượng gói tin gửi thành công
for idx = 1:length(G)
    N_s(idx) = sum((t1 > T(idx)) & (t2 > T(idx)));
end

% Thông lượng
S_s = N_s.*T;

plot(G,S_t,G,S_s,'+');
legend('Lý thuyết','Mô phỏng');
set(gcf,'color','w');
xlabel('G');
ylabel('S');
title('Pure ALOHA');

```

Câu 5: Vẽ đồ thị S theo G cho giao thức pure ALOHA

So sánh với lý thuyết và nhận xét:

Câu 6: Kiểm chứng lý thuyết và mô phỏng Slotted ALOHA trên Matlab. Viết code bên dưới

Câu 7: Vẽ đồ thị S theo G cho pure ALOHA và Slotted ALOHA trên cùng đồ thị và nhận xét.

BÀI THÍ NGHIỆM 2: KHẢO SÁT VÀ MÔ PHỎNG HỆ THỐNG TDMA

<u>Điểm:</u>	<u>Nhận xét:</u>
---------------------	-------------------------

Mục đích của bài thí nghiệm là giúp sinh viên hiểu được nguyên tắc của hệ thống truy nhập theo thời gian.

PHẦN 1: CHUẨN BỊ TRƯỚC Ở NHÀ

Câu 1: Tóm tắt nguyên tắc hệ thống TDMA. Thế nào là mux và demux?

Câu 2: Khảo sát các hàm Matlab bên dưới.

TT	Hàm	Cú pháp và Chức năng?
1	randi	
2	subplot	
3	bar	
4	ylabel	
5	upsample	
6	downsample	

PHẦN 2: THỰC HIỆN TẠI PHÒNG THÍ NGHIỆM TRÊN PHẦN MỀM MATLAB

Câu 3: Giải thích mã nguồn Matlab như bên dưới

Mã nguồn Matlab	Giải thích
<pre>clc; close all; clear all; % Số lượng bit cho một user N = 10; xA = randi([0 1],1,N); xB = randi([0 1],1,N); xC = randi([0 1],1,N); subplot(11,1,1) bar(xA, 'c') ylabel('x_A'); subplot(11,1,2) bar(xB, 'g') ylabel('x_B'); subplot(11,1,3) bar(xC, 'r') ylabel('x_C'); %% Muxitplexing xA = upsample(xA,3,0); subplot(11,1,4) bar(xA, 'c') ylabel('x_A'); xB = upsample(xB,3,1); subplot(11,1,5) bar(xB, 'g') ylabel('x_B'); xC = upsample(xC,3,2); subplot(11,1,6) bar(xC, 'r') ylabel('x_C'); %% Transmitting Tx = xA + xB + xC; subplot(11,1,7) bar(Tx) ylabel('Tx'); %% Receiving Rx = Tx; xA_ = downsample(Rx,3,0); xB_ = downsample(Rx,3,1); xC_ = downsample(Rx,3,2); subplot(11,1,8)</pre>	

```

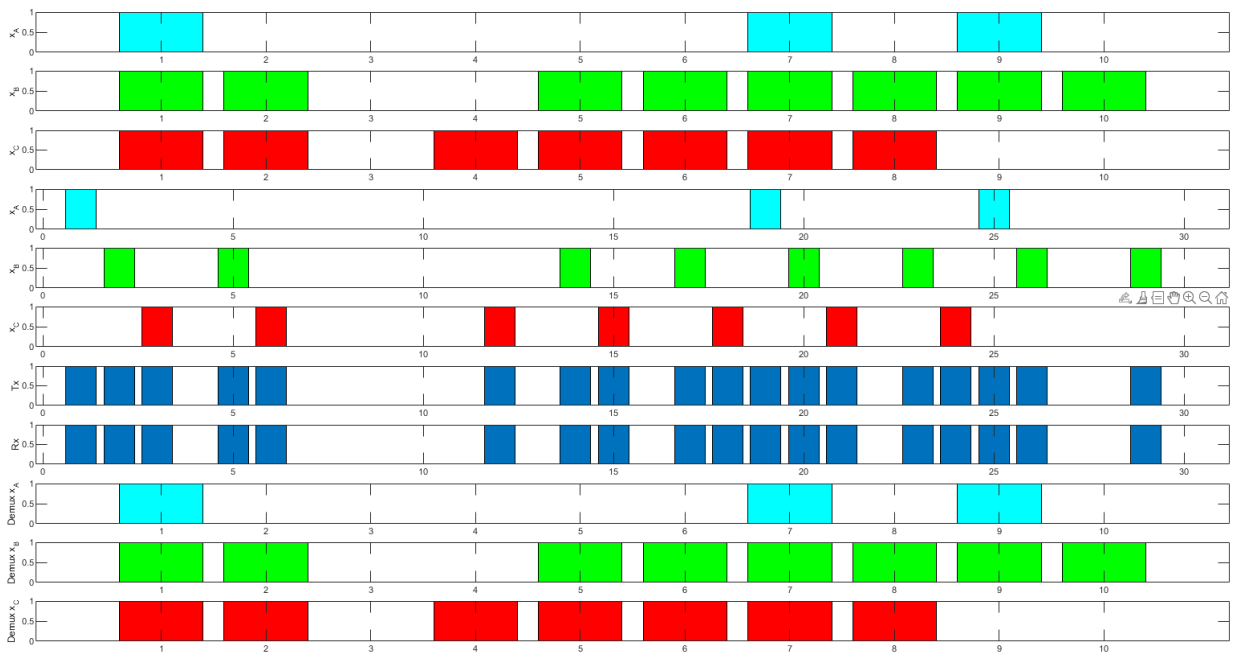
bar(Rx)
ylabel('Rx');

subplot(11,1,9)
bar(xA_, 'c')
ylabel('Demux x_A');

subplot(11,1,10)
bar(xB_, 'g')
ylabel('Demux x_B');

subplot(11,1,11)
bar(xC_, 'r')
ylabel('Demux x_C');
set(gcf, 'color', 'w');

```



Câu 4: Làm lại ví dụ trên với số user là 5 và số bit là 7

Câu 5: Giải thích vai trò của hàm upsample và hàm downsample

BÀI THÍ NGHIỆM 3: KHẢO SÁT VÀ MÔ PHỎNG HỆ THỐNG FDMA

Điểm:

Nhận xét:

Mục đích của bài thí nghiệm là giúp sinh viên hiểu được nguyên tắc của hệ thống truy nhập theo tần số.

PHẦN 1: CHUẨN BỊ TRƯỚC Ở NHÀ

Câu 1: Nguyên tắc hệ thống FDMA

Câu 2: Khảo sát các hàm Matlab như bên dưới

TT	Hàm	Cú pháp và Chức năng?
1	sin	
2	modulate	
3	demod	

4	butter	
5	freqz	
6	pwelch	
7	legend	

PHẦN 2: THỰC HIỆN TẠI PHÒNG THÍ NGHIỆM TRÊN PHẦN MỀM MATLAB

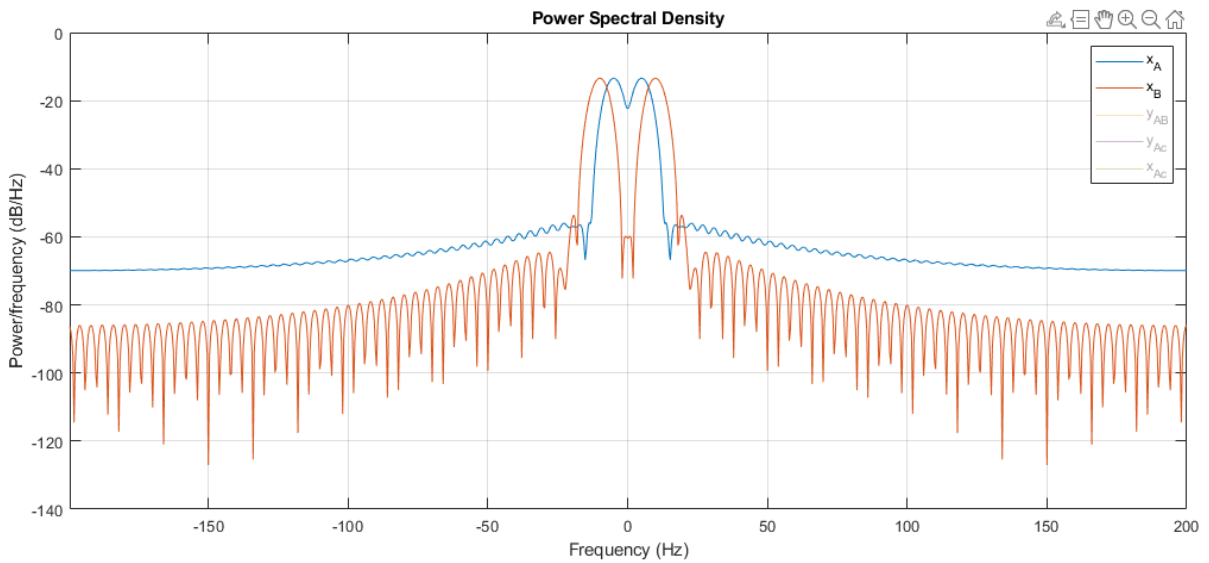
Câu 3: Giải thích mã nguồn Matlab như bên dưới.

Mã nguồn Matlab	Giải thích
<pre> clc clear all fs = 400; t = 0:1/fs:1; % User A xA = sin(2*pi*5*t); % User B xB = sin(2*pi*10*t); % Tần số sóng mang cho user A fcA = 100; yA = modulate(xA,fcA,fs,'amssb'); % Tần số sóng mang cho user B fcB = 120% Carrier freq for user A; yB = modulate(xB,fcB,fs,'amssb'); % Truyền tín hiệu kết hợp lên kênh truyền yAB = yA + yB; %% Tại máy thu % Tần số cắt fcut = (fcA + fcB)/2; [b,a] = butter(40,fcut/(fs/2)); freqz(b,a); yA_ = filter(b,a,yAB); xA_ = demod(yA_,fcA,fs,'amssb'); % Vẽ phổ </pre>	

```
figure(1)
pwelch([xA;xB;yAB;yA_;xA_]',hamming(100),80,1024,fs,'centered');
legend('x_A','x_B','y_{AB}','y_{Ac}','x_{Ac}');
set(gcf,'color','w');
```

```
figure(2)
plot(t,xA,t,xA_)
```

Câu 4: Phổ tín hiệu xA và xB

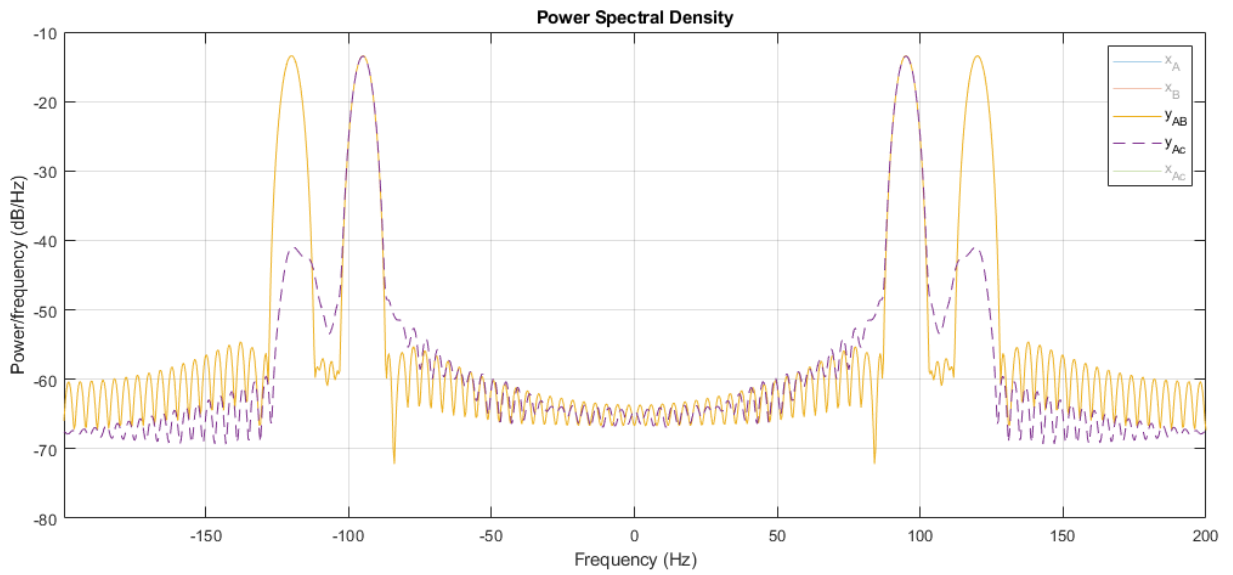


Tần số tại đỉnh của phổ tín hiệu xA:

Tần số tại đỉnh của phổ tín hiệu xB:

Nhận xét và giải thích:

Câu 5: Phổ tín hiệu y_{AB} và y_{Ac}

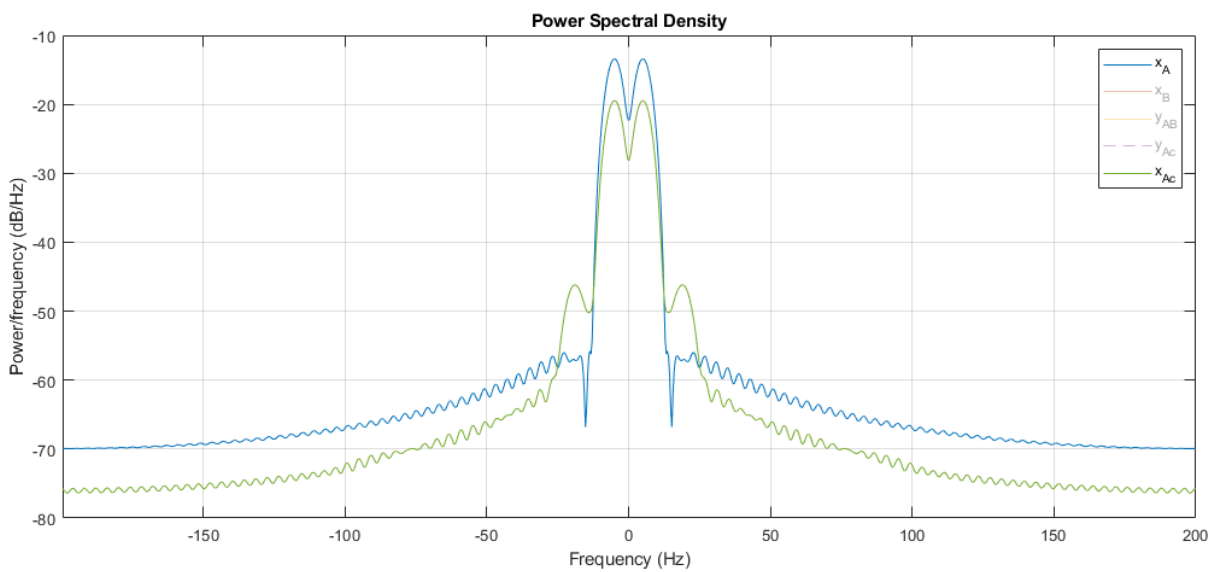


Tần số tại đỉnh của phổ tín hiệu y_{AB} :

Tần số tại đỉnh của phổ tín hiệu y_{Ac} :

Nhận xét và giải thích:

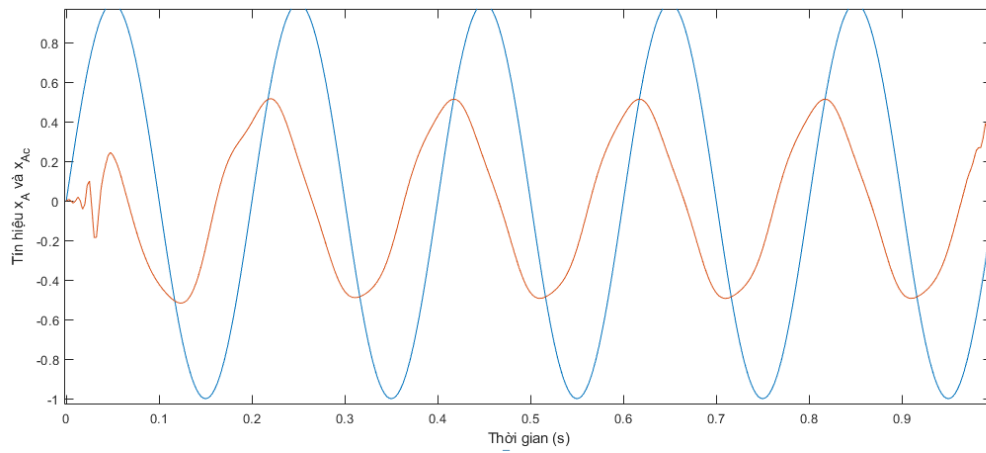
Câu 6: Phổ tín hiệu x_A và x_{Ac}



Tần số tại đỉnh của phổ tín hiệu y_{Ac} :

Nhận xét và giải thích:

Câu 7: So sánh tín hiệu x_A và x_{Ac}



Nhận xét và giải thích:

Câu 8: Thực hiện lại Câu 3 nhưng $f_c B$ bằng 120 Hz, 140 Hz, vẽ lại các hình trên và nhận xét

BÀI THÍ NGHIỆM 4: KHẢO SÁT VÀ MÔ PHỎNG HỆ THỐNG CDMA

Điểm:

Nhận xét:

Mục đích của bài thí nghiệm là giúp sinh viên hiểu được nguyên tắc của hệ thống truy nhập theo tần số.

PHẦN 1: CHUẨN BỊ TRƯỚC Ở NHÀ

Câu 1: Nguyên tắc hệ thống CDMA

Câu 2: Nguyên tắc mã NRZ-L

Câu 3: Khảo sát các hàm Matlab

TT	Hàm	Cú pháp và Chức năng?
1	randi	
2	dot	

PHẦN 2: THỰC HIỆN TẠI PHÒNG THÍ NGHIỆM TRÊN PHẦN MỀM MATLAB

Câu 4: Giải thích mã nguồn Matlab sau.

Mã nguồn Matlab	Giải thích từng dòng lệnh
<pre>clc clear % Số lượng bit N = 5; % Tạo các bit ngẫu nhiên xA = randi([0 1],1,N); xB = randi([0 1],1,N); xC = randi([0 1],1,N); % Sử dụng mã hóa NRZ-L xA = 2*xA - 1; xB = 2*xB - 1; xC = 2*xC - 1; % Chip Pattern for station A, B and C chipA = [1 1 1 1]; chipB = [1 -1 1 -1]; chipC = [1 1 -1 -1]; nc = length(chipA); % Chip Pattern for station A, B and C txA = xA(:).*chipA;</pre>	

<pre> txB = xB(:).*chipB; txC = xC(:).*chipC; % BS phát tín hiệu kết hợp tx = txA + txB + txC; % Giải mã tại BS for i=1:N xA_(i) = dot(chipA,tx(i,:))/nc; xB_(i) = dot(chipB,tx(i,:))/nc; xC_(i) = dot(chipC,tx(i,:))/nc; end % sum(xA~=xA_) sum(xB~=xB_) sum(xC~=xC_) </pre>	
--	--

Câu 5: Làm lại mã nguồn cho trường hợp 5 users

Câu 7: Giải thích mã nguồn Matlab sau.

Mã nguồn Matlab	Giải thích từng dòng lệnh
<pre> clear all; %% Các thông số ban đầu Fs = 1000; fc = 100; fp = 4; bit_t = 0.1; %% Tạo tín hiệu dữ liệu lưỡng cực m = [0 0 1 1 1 1 0 0]; for bit = 1:length(m) if(m(bit)==0) m(bit) = -1; end end message = repmat(m,fp,1); message = reshape(message,1,[]); %% Tạo PN pn_code = randi([0,1],1,length(m)*fp); for bit = 1:length(pn_code) if(pn_code(bit)==0) pn_code(bit) = -1; end end DSSS = message.*pn_code; %% Tạo sóng mang t = 0:1/Fs:(bit_t-1/Fs); s0 = -1*cos(2*pi*fc*t); s1 = cos(2*pi*fc*t); carrier = []; BPSK = []; for i = 1:length(DSSS) if (DSSS(i) == 1) BPSK = [BPSK s1]; elseif (DSSS(i) == -1) BPSK = [BPSK s0]; end carrier = [carrier s1]; end %% Giải điều chế rx = []; for i = 1:length(pn_code) if(pn_code(i)==1) rx = [rx BPSK((((i-1)*length(t))+1):i*length(t))]; else rx = [rx (-1)*BPSK((((i-1)*length(t))+1):i*length(t))]; end end end </pre>	

```

demod = rx.*carrier;
result = [];
for i = 1:length(m)
    x = length(t)*fp;
    cx = sum(carrier(((i-
1)*x)+1:i*x).*demod(((i-1)*x)+1:i*x));
    if(cx>0)
        result = [result 1];
    else
        result = [result -1];
    end
end
pn_codeWrong = randi([0,1],1,length(m)*fp);
resultWrong = [];
rx2 =[];
for i = 1:length(pn_code)
    if(pn_codeWrong(i)==1)
        rx2 = [rx2 BPSK(((i-
1)*length(t))+1:i*length(t))];
    else
        rx2 = [rx2 (-1)*BPSK(((i-
1)*length(t))+1:i*length(t))];
    end
end
demod2 = rx2.*carrier;
for i = 1:length(m)
    x = length(t)*fp;
    cx = sum(carrier(((i-
1)*x)+1:i*x).*demod2(((i-1)*x)+1:i*x));
    if(cx>0)
        resultWrong = [resultWrong 1];
    else
        resultWrong = [resultWrong -1];
    end
end
message1 = repmat(result,fp,1);
message1 = reshape(message1,1,[]);
message2 = repmat(resultWrong,fp,1);
message2 = reshape(message2,1,[]);

%% Vẽ dạng sóng tín hiệu
pn_size = length(pn_code);
tpn = linspace(0,length(m)*bit_t-
bit_t/fp,pn_size);
tm = 0:bit_t/fp:length(m)*bit_t-bit_t/fp;
figure
subplot(311)
stairs(tm,message,'linewidth',2)
title('Message bit sequence')
axis([0 length(m)*bit_t -1 1]);
subplot(312)
stairs(tpn,pn_code,'linewidth',2)
title('Pseudo-random code');
axis([0 length(m)*bit_t -1 1]);
subplot(313)

```

```

stairs(tpn,DSSS,'linewidth',2)

title('signal after spread spectrum');
axis([0 length(m)*bit_t -1 1]);
figure
subplot(311)
stairs(tm,message,'linewidth',2)
title('Message bit sequence')
axis([0 length(m)*bit_t -1 1]);
subplot(312)
stairs(tm,message1,'linewidth',2)
title('Received message using true pseudo-
random code')
axis([0 length(m)*bit_t -1 1]);
subplot(313)
stairs(tm,message2,'linewidth',2)
title('Received message using wrong pseudo-
random code')
axis([0 length(m)*bit_t -1 1]);
%% Vẽ dạng phổ tín hiệu
f = linspace(-Fs/2,Fs/2,1024);
figure
subplot(311)
plot(f,abs(fftshift(fft(message,1024))), 'linewi
dth',2);
title('Message spectrum')
subplot(312)
plot(f,abs(fftshift(fft(pn_code,1024))), 'linewi
dth',2);
title('Pseudo-random code spectrum');
subplot(313)
plot(f,abs(fftshift(fft(DSSS,1024))), 'linewidth
',2);
title('Encoded signal spectrum');
figure;
subplot(311)
plot(f,abs(fftshift(fft(BPSK,1024))), 'linewidth
',2);
title('Transmitted signal spectrum');
subplot(312)
plot(f,abs(fftshift(fft(rx,1024))), 'linewidth',
2);
title('Received signal multiplied by pseudo
code');
subplot(313)
plot(f,abs(fftshift(fft(demod,1024))), 'linewidt
h',2);
title('Demodulated signal spectrum before
decision device ');

```

Giải thích các đồ thị vẽ được