

Subtask 2.1:

a)

Tín hiệu $s(n-D)$ với $D \in \mathbb{Z}^+$ được gọi là tín hiệu trễ đi D mẫu của $s(n)$ với D là số nguyên dương.

b)

Muốn tín hiệu $s(n-D)$ trễ với $s(n)$ một khoảng thời gian t_0 (giây) thì ta có công thức sau:

$$T = nT_s = \frac{n}{F_s} \Rightarrow n = F_s \cdot t \quad \Leftrightarrow D = F_s \cdot t_0$$

Trong đó:

+) t là khoảng thời gian $s(n-D)$ trễ đi D mẫu so với $s(n)$

+) n số mẫu trễ đi từ tín hiệu ban đầu hay n chính là D trong tín hiệu $s(n-D)$

+) T_s là chu kỳ lấy mẫu, F_s là tần số lấy mẫu

c)

Phần mềm sử dụng : Matlab.

Code:

```
addpath('C:\Users\Admin\Desktop\CN3_32\task2\data');
[sn,Fs] = audioread('task0201.wav');
% vẽ đồ thị sn
plot(sn);
% khai báo t1,t2,t3
t1=0.5; t2=1; t3=1.5;
% khai báo a1,a2,a3
a1=0.5; a2=0.2; a3=0.1;
% tính D1,D2,D3
D1=Fs*t1; D2=Fs*t2; D3=Fs*t3;
```

```

% lấy ra số hàng và cột của sn
[h , c]= size(sn);
% lấy ra 1 kênh trong 2 kênh của sn
sn = sn(:,1);
% tính sn1,sn2,sn3
sn1=[zeros(D1,1) ; sn];
sn2=[zeros(D2,1) ; sn];
sn3=[zeros(D3,1) ; sn];
% độ dài của từng ma trận theo hàng
l=length(sn);
l1=length(sn1);
l2=length(sn2);
l3=length(sn3);
% độ dài max là n3
l_max=l3;
% độ dài cần bổ sung cho sn là:
l_sn=l_max-l;
% độ dài cần bổ sung cho sn1 là:
l_sn1=l_max-l1;
% độ dài cần bổ sung cho sn là:
l_sn2=l_max-l2;
% bổ sung độ dài cho các ma trận
sn=[sn ; zeros(l_sn,1)];
sn1=[sn1 ; zeros(l_sn1,1)];
sn2=[sn2 ; zeros(l_sn2,1)];
% thiết lập biểu thức yn
yn = sn + a1*sn1 + a2*sn2 + a3*sn3;
% nghe âm thanh
soundsc(yn,Fs);

```

- Nhận xét: âm thanh nghe bị vọng lại, nhỏ dần so với ban đầu.

d)

Code:

```
addpath('C:\Users\Admin\Desktop\CN3_32\task2\data')
[sn,Fs] = audioread('task0201.wav');
% vẽ đồ thị sn
plot(sn);
% khai báo t1,t2,t3
t1=0.5; t2=1; t3=1.5;
% khai báo a1,a2,a3
a1=0.2; a2=0.5; a3=1;
% tính D1,D2,D3
D1=Fs*t1; D2=Fs*t2; D3=Fs*t3;
% lấy ra số hàng và cột của sn
[h , c]= size(sn);
% lấy ra 1 kênh trong 2 kênh của sn
sn = sn(:,1);
% tính sn1,sn2,sn3
sn1=zeros(D1,1) ; sn];
sn2=zeros(D2,1) ; sn];
sn3=zeros(D3,1) ; sn];
% độ dài của từng ma trận theo hàng
l=length(sn);
l1=length(sn1);
l2=length(sn2);
l3=length(sn3);
% độ dài max là n3
l_max=l3;
% độ dài cần bổ sung cho sn là:
l_sn=l_max-l;
% độ dài cần bổ sung cho sn1 là:
l_sn1=l_max-l1;
% độ dài cần bổ sung cho sn là:
```

```

l_sn2=l_max-l2;
% bổ sung độ dài cho các ma trận
sn=[sn ; zeros(l_sn,1)];
sn1=[sn1 ; zeros(l_sn1,1)];
sn2=[sn2 ; zeros(l_sn2,1)];
% thiết lập biểu thức yn
yn = 0.1*sn + a1*sn1 + a2*sn2 + a3*sn3;
% nghe âm thanh
soundsc(yn,Fs);

```

- Nhận xét: Khi ta thay đổi hệ số của sn từ 1 thành 0.1, hệ số của sn1, sn2, sn3 lần lượt là a1=0.2, a2=0.5, a3=1 thì ta sẽ được hiệu ứng âm thanh vọng to dần, ngược lại hoàn toàn so với hiệu ứng âm thanh vọng nhỏ dần của phần c.

Subtask 2.2:

a)

a) $y(n) = a_1s(n) + a_2n(n)$.

$y(n)$ được tạo ra từ phép toán tích của 1 dãy với hằng số và tổng của 2 dãy.

$a_1s(n)$ được tạo ra từ phép toán tích của 1 dãy với hằng số, trong đó a_1 là hằng số, $s(n)$ là tín hiệu sạch.

Tương tự ta có: $a_2n(n)$ cũng được tạo ra từ phép toán tích của 1 dãy với hằng số, trong đó a_2 là hằng số, $n(n)$ là tín hiệu nhiễu.

Nguyên tắc thực hiện phép toán:

- Tích của 1 dãy với hằng số: Tích của 1 dãy với các hằng số nhận được bằng cách nhân tất cả các giá trị mẫu của dãy với hằng số đó.

- Tổng của 2 dãy: Tổng của 2 dãy nhận được bằng cách cộng từng đôi một các giá trị mẫu đối với cùng một trị số của biến độc lập.

b)

Sử dụng dữ liệu được giao trong Task 2.1, chọn ra hai phân đoạn $s(n)$ và $n(n)$, trong đó:

- Phân đoạn $s(n)$ tương ứng với phần dữ liệu “sạch” (chỉ gồm giọng nói không có nhiễu nền - không có tiếng nhạc) độ dài khoảng 12-30 giây;
- Phân đoạn $n(n)$ ứng với phần chỉ có nhạc (nhạc hiệu, hoặc nhạc nền). Với $n(n)$ có thể cần Audacity và sử dụng lại dữ liệu được giao ở Task 2.1 để trích ra một đoạn theo yêu cầu. Chiều dài của $n(n)$ cần nhỏ hơn chiều dài của $s(n)$.

Lưu dữ liệu của hai phân đoạn chọn được với các file lần lượt tên là **cleandata0202.wav** và **noisedata0202.wav** vào thư mục **data**.

Phần mềm sử dụng : Matlab.

Code:

```
addpath('C:\Users\Admin\Desktop\CN3_32\task2\data')
[sn,Fs]=audioread('cleandata0202.wav');
[nn,Fs]=audioread('noisedata0202.wav');
% khai báo a1,a2
a1=1.0; a2=0.25;
% độ dài của sn và nn
l_sn=length(sn);
l_nn=length(nn);
l_max=l_sn;
% tìm độ dài bổ sung cho độ dài của nn
l_bs=l_max-l_nn;
% bổ sung chiều dài cho nn
```

```

nn=[nn ; zeros(1_bs,2)];
% thiết lập biểu thức yn
yn=a1*sn+a2*nn;
% nghe âm thanh và nhận xét
soundsc(yn,Fs);

```

- Nhận xét: Khi thay đổi hệ số a_1 , a_2 thì hiệu ứng âm thanh cũng thay đổi theo. Nếu tăng hoặc giảm a_1 thì âm thanh sạch của file **cleandata0202.wav** sẽ tăng hoặc giảm theo. Tương tự với a_2 và âm thanh của file **noisedata0202.wav**.

Subtask 2.3:

a)

$y(n)$ được tạo ra từ tích 2 tín hiệu $a(n)$ và $s(n)$.

Nguyên tắc thực hiện tích của 2 dãy:

- Tích của 2 dãy nhận được bằng cách nhân từng đôi một các giá trị mẫu đối với cùng một trị số của biến độc lập.

b)

Chọn phân đoạn tiếng nói sạch trong thư mục Clean ở phần Task 1 tiền xử lý dữ liệu, gọi là $s(n)$. Lưu tín hiệu phân đoạn được chọn thành file có tên là **cleandata0203.wav** vào thư mục data.

c)

Code:

```

addpath('C:\Users\Admin\Desktop\CN3_32\task2\data')

[sn,Fs] = audioread('cleandata0203.wav');
% tần số đã cho F0

```

```

F0 = 0.5;
% từ F0 => omega
omega = 2 * pi * F0;
% lay ra 1 kênh trong 2 kênh của sn
sn = sn(:,1);
% chiều dài của tín hiệu hình sin bằng độ
dài sn
l_sn = length(sn);
% giả sử biên độ A=2
A = 2;
% giả sử pha ban đầu là 0
pha = 0;
% tạo dãy tín hiệu hình sin an
n = 0 : l_sn - 1;
pha = phase(pha); % hàm chuyển thành radian
arg = omega * n + pha;
an = A * cos(arg);
% định dạng lại kích thước của an cho giống
với sn
an = reshape(an,[l_sn,1]); % an có hàng là l_sn
cột là 1
% thiết lập biểu thức yn
yn = (1.1 + an) .* sn;
soundsc(yn,Fs);

```

- Nhận xét: Âm thanh bị nhiễu so với lúc đầu, kèm thêm những tiếng nghe rất ỉn tai.

d)

Code:

```
addpath('C:\Users\Admin\Desktop\CN3_32\task2\data');

[sn, Fs] = audioread('cleandata0203.wav');

% yn = an*sn;

n = 0:length(sn);

% an tăng tuyến tính.

an = n / length(sn);

for i=1:length(sn) yn(i) = an(i)*sn(i);

end;

plot(yn);

soundsc(yn, Fs);

% an giảm tuyến tính.

an = 1 - (n/length(sn));

for i=1:length(sn) yn(i) = an(i)*sn(i);

end;

plot(yn);

soundsc(yn, Fs);
```


- Nhận Xét: +) Với tín hiệu $a(n)$ có dạng tín hiệu tăng tuyến tính, độ dài bằng $s(n)$ và có giá trị lớn nhất là 1 thì âm thanh của tín hiệu $y(n)$ to dần.
+) Với tín hiệu $a(n)$ có dạng tín hiệu giảm tuyến tính, độ dài bằng $s(n)$ và có giá trị lớn nhất là 1 thì âm thanh của tín hiệu $y(n)$ nhỏ dần.

Subtask 2.4:

a)

Để có được tín hiệu $s_1(n)$ có độ dài hữu hạn từ tín hiệu $s(n)$ có độ dài vô hạn, ta thực hiện lấy mẫu tín hiệu $s(n)$.

b)

Năng lượng của tín hiệu $s(n)$ được tính theo công thức:

$$E_s = \sum_{n=-\infty}^{\infty} |s(n)|^2$$

c)

Công thức:

- Công thức tính năng lượng của tín hiệu $s_1(n)$ với $s_1(n)$ có chiều dài hữu hạn N (từ mẫu 0 đến mẫu $N-1$):

Ta có:

$$E_{s_1} = \sum_{n=-\infty}^{\infty} |s_1(n)|^2 = \sum_{n=-\infty}^{-1} |s_1(n)|^2 + \sum_{n=0}^{N-1} |s_1(n)|^2 + \sum_{n=N}^{\infty} |s_1(n)|^2$$

$$\Leftrightarrow E_{s_1} = \sum_{n=0}^{N-1} |s_1(n)|^2$$

Code:

```
addpath('C:\Users\Admin\Desktop\CN3_32\task2\data')  
[s1n,Fs] = audioread('cleandata0203.wav');  
% lay ra 1 kênh trong 2 kênh của sn  
s1n = s1n(:,1);  
% thiết lập công thức tính năng lượng  
E = sum(abs(s1n.^2));
```

d)

Code:

```
>> help reshape
```