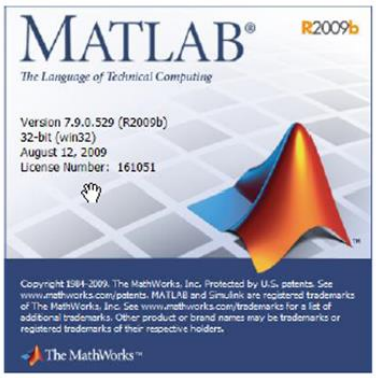
BÀI 1: TÍN HIỆU VÀ HỆ THỐNG RỜI RẠC

1. PHẦN MỀM MATLAB

1.1 KHỞI ĐỘNG MATLAB

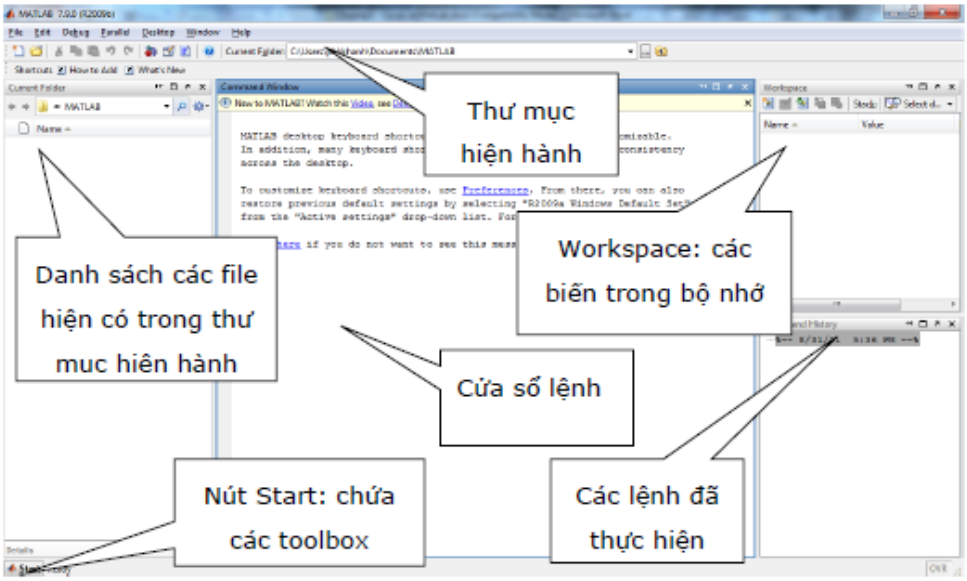
Matlab(Matrix laboratory) là phần mềm dùng để giải các bài toán kỹ thuật, đặc biệt là các bài toán liên quan đến ma trận. Matlab cung cấp các Toolbox, tức các hàm mở rộng môi trường Matlab để giải quyết các vấn đề cơ bản như xử lý tín hiệu số, hệ thống điều khiển, mạng neuron, fuzzy logic, mô phỏng v. v.

Cửa sổ biểu tượng của chương trình Matlab:



Hình 1.1 *– Cửa sổ khởi động Matlab*

Cửa sổ làm việc của Matlab:



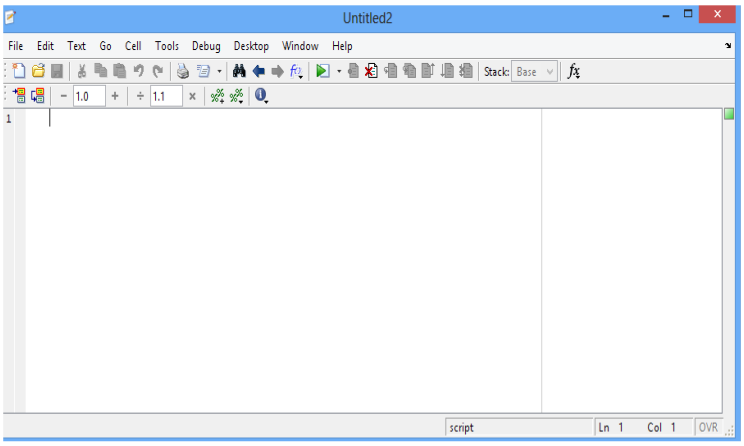
Hình 1.2 *– Cửa sổ làm việc của Matlab*

+ Cửa sổ lệnh(Command window): Là cửa sổ giao tiếp chính của Matlab bởi đây là nơi nhập giá trị các biến, hiển thị giá trị, tính toán giá trị của biểu thức, thực thi các hàm có sẵn trong thư viện hoặc các hàm do người dung lập trình ra trong M-files. Các lệnh được nhập sau dấu nhăc '>>' và thực thi lệnh bằng phím enter. Để mở chương trình soạn thảo trong Matlab, gõ lệnh:

>> edit

+ Cửa sổ Edit: Là cửa sổ dùng để soạn thảo chương trình ứng dụng, được khởi động bằng lệnh edit trong Command window. Chương trình soạn thảo trong cửa sổ edit có hai dạng:

- Dạng Script file: tập hợp các câu lệnh viết dưới dạng liệt kê, không có biến dữ liệu vào và biến ra.
- Dạng function: Có biến dữ liệu vào và biến ra.



Hình 1.3 *– Cửa sổ edit để soạn script hay function*

+ Cửa sổ Command History: Các dòng đã nhập trong Command window được giữ lại trong cửa sổ Command History và cửa sổ này cho phép ta sử dụng lại những lệnh đó bằng cách nhấp đúp chuột lên các lệnh hay biến đó.

+ Cửa sổ Workspace: Là cửa sổ thể hiện tên biến các biến bạn sử dụng cùng với kích thước vùng nhớ(số byte), kiểu dữ liệu(lớp), các biến được giải phóng sau mỗi lần tắt chương trình. Cửa sổ Workspace cho phép thay đổi giá trị của biến bằng cách nhấn phím chuột phải lên các biến và chọn ***Edit.***

1.2 CÁC VẤN ĐỀ CƠ BẢN

1.2.1 Các phép toán và toán tử

Các phép toán : +, -, *,/,\(chia trái), ^(mũ),’(chuyển vị hay số phức liên hợp).

Các phép toán quan hệ: <, <=, >,>=, ==, `~=

Các toán tử logic: &, |(or), ~(not)

Chú ý:

+ Các lệnh kết thúc bằng dấu chấm phẩy, Matlab sẽ không thể hiện kết quả trên màn hình.

+ Các chú thích được đặt phía sau dấu %. Trong quá trình nhập nếu các phần tử trên một hàng dài quá ta có thể xuống dòng bằng toán tử ba chấm(…)

1.2.2 Khai báo biến

- Phân biệt chữ hoa và chữ thường, Không cần phải khai báo kiểu biến
- Tên miền bắt đầu bằng các ký tự và không có khoảng trắng. Không đặt tên trùng với các tên đặc biệt của Matlab.
- Để khai báo biến toàn cục(sử dụng được trong tất cả các chương trình con), phải dùng khóa ***global*** phía trước.

1.2.2.1 Toán tử ‘’ : ’’

Toán tử ‘’ : ’’ là một toán tử quan trọng của Matlab. Nó xuất hiện ở nhiều dạng khác nhau. Biểu thức 1 :10 là một vector hàng chứa 10 số nguyên từ 1 đến 10.

>>1:10;

>>100:-7:50 % tạo dãy số từ 100 đến 51, cách đều nhau 7

>>0: pi/4: pi % tạo một dãy số từ 0 đến π, cách đều nhau π/4

1.2.2.2 Các lệnh xử lý ma trận

- Cộng : X = A + B
- Trừ : X = A – B
- Nhân : X = A* B
- X = A.* B nhân các phần tử tương ứng với nhau, yêu cầu 2 ma trận A và B phải có cùng kích thước.
- Chia : X = A/B lúc đó X = A*inv(B)
- X = A\B lúc đó X = inv(A)*B
- X = A./B chia các phần tử tương ứng với nhau, 2 ma trận có cùng kích thước.
- Lũy thừa: X = A^2; X = A.^2
- Nghịch đảo: X = inv(A)
- Định thức: d = det(A)

1.2.3 Vector

Vector thực chất là ma trận có kích thước n x 1 hay 1 x n, nên ta có thể tạo ra vector như cách tạo ra ma trận. Ngoài ra, có thể dùng một số cách sau:

>>x=0:0.1:1

>>y=linspace(1, 10, 20) % vector 20 phan tu cach deu nhau tu 1den 10

>>z=rand(10,1) ; tạo 10 số ngẫu nhiên phân bố đều

1.3 ĐỒ HỌA

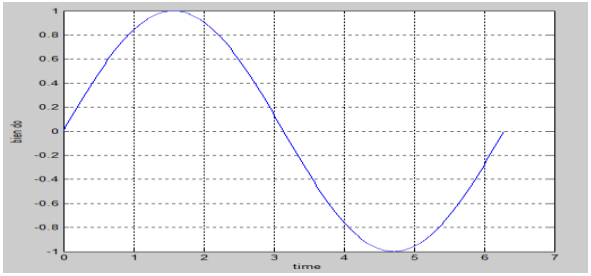
1.3.1 Các lệnh vẽ

Matlab cung cấp một loạt hàm để vẽ biểu diễn các vector cũng như giải thích và in các đường cong này.

plot: đồ họa 2-D với số liệu 2 trục vô hướng và tuyến tính
plot3: đồ họa 3-D với số liệu 2 trục vô hướng và tuyến tính
loglog: đồ hoạ với các trục x, y ở dạng logarit
semilogx: đồ hoạ với trục x logarit và trục y tuyến tính
semilogy: đồ hoạ với trục y logarit và trục x tuyến tính

1.3.2 Tạo hình vẽ

Hàm plot có các dạng khác nhau phụ thuộc vào các đối số đưa vào. Ví dụ nếu y là một vector thì plot(y) tạo ra một đường quan hệ giữa các giá trị của y và chỉ số của nó. Nếu ta có 2 vector x và y thì plot(x,y) tạo ra đồ thị quan hệ giữa x và y.




```
t = 0:pi/100:2*pi;
y = sin(t);
plot(t,y); % Vẽ hình sin từ 0->2π
grid on
```

Hình 1.4 *– Đồ thị hình sin*

1.3.3 Kiểu đường vẽ

Ta có thể dùng các kiểu đường vẽ khác nhau khi vẽ hình.

```
t = [0:pi/100:2*pi];
y = sin(t)
plot(t,y,'.')   % vẽ bằng đường chấm chấm
```

Để xác định màu và kích thước đường vẽ, ta dùng tham số sau :

LineWidth	: độ rộng đường thẳng, tính bằng số điểm
MarkerEdgeColor	: màu của các cạnh của khối đánh dấu
MarkerFaceColor	: màu của khối đánh dấu
MarkerSize	: kích thước của khối đánh dấu

Màu được xác định bằng các tham số :

r: red	**m** magenta	**g**: green	**y**: yellow
b: blue	**k**: black	**c**: cyan	**w**: white

Các dạng đường thẳng xác định bằng :

- đường liền	-- đường đứt nét
: đường chấm chấm	-. đường chấm gạch

Các dạng điểm đánh dấu xác định bằng :

+ dấu cộng	. điểm	**o** vòng tròn	**x** chữ thập	* dấu sao
s hình vuông	**d** hạt kim cương	**v** tam giác hướng xuống		
^ tam giác hướng lên	< tam giác sang trái			
> tam giác sang phải	**h** lục giác	**p** ngũ giác		

```
x = -pi : pi/10 : pi;
y = tan(sin(x)) - sin(tan(x));
plot(x,y,'--rs','LineWidth',2,'MarkerEdgeColor','k',...
'MarkerFaceColor','g','MarkerSize',10)
```

Để vẽ hai hàm trên cùng một đồ thị, ta dùng lệnh :

```
>>hold on
```

1.3.4 Vẽ với hai trục y

Hàm plotyy cho phép tạo một đồ thị có hai trục y. Ta cũng có thể dùng plotyy để cho giá trị trên hai trục y có kiểu khác nhau nhằm tiện so sánh.

```
t = 0 :900 ; A = 1000 ; b =0.005 ; a = 0.005 ;
z2 = sin(b*t) ; z1 = A*exp(-a*t) ;
[haxes, hline1, hline2] = plotyy(t,z1,t,z2, 'semilogy', 'plot');
```

1.3.5 Vẽ đường cong 3-D

Nếu x,y,z là ba vector có cùng độ dài thì plot3 sẽ vẽ đường cong 3D.

```
t = 0:pi/50:10*pi;
plot3(sin(t),cos(t),t)
axis square;
grid on
```

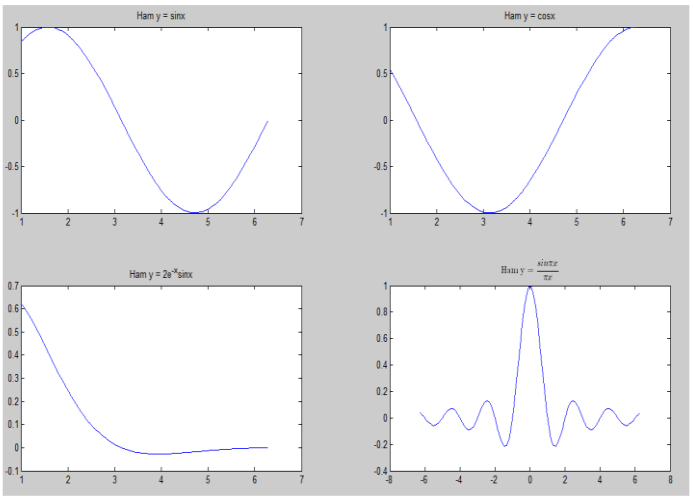
Vẽ nhiều trục tọa độ

Dùng hàm subplot để vẽ nhiều trục tọa độ.

```
subplot(2,3,5)          %2,3: xác định có 2 hàng, 3 cột
% 5: chọn trục thứ 5 (đếm từ trái sang phải, trên xuống dưới)
```

Ví dụ 1 :

```
x = linspace(1,2*pi); y1 = sin(x);y2 = cos(x);
y3 = 2.*exp(-x).*sin(x);
x1 = linspace(-2*pi,2*pi);
y4 = sinc(x1);
subplot(221);plot(x,y1);
title('Ham y = sinx');
subplot(222);plot(x,y2);
title('Ham y = cosx');
subplot(223);plot(x,y3);
title('Ham y = 2e^{-x}sinx');
subplot(224);plot(x1,y4);
title('Ham y = $${sin\pi x \over\pi x}$$','interpreter','latex');
```



1.3.5.1 Giới hạn của trục và chia vạch trên trục

Matlab chọn các giới hạn trên trục tọa độ và khoảng cách đánh dấu dựa trên số liệu dùng để vẽ. Dùng lệnh axis có thể đặt lại giới hạn này.

Cú pháp của lệnh: axis[xmin , xmax , ymin , ymax]

```
x = 0:0.025:pi/2;
plot(x,tan(x),'-ro');
axis([0 pi/2 0 5]);
```

1.3.5.2 Ghi nhãn lên trục tọa độ

Matlab cung cấp các lệnh ghi nhãn lên đồ họa gồm:

title thêm nhãn vào đồ hoạ.

xlabel thêm nhãn vào trục x.

ylabel thêm nhãn vào trục y.

zlabel thêm nhãn vào trục z.

legend thêm chú giải vào đồ thị.

text hiển thị chuỗi văn bản ở vị trí nhất định.

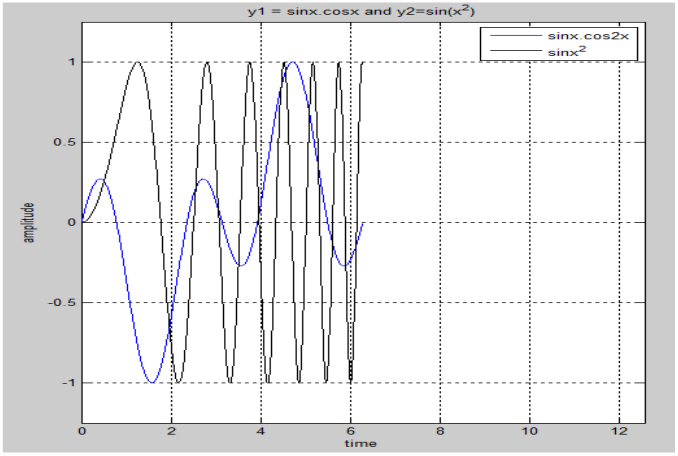
gtext đặt văn bản lên đồ hoạ nhờ chuột.

1.3.6 Một số ví dụ vẽ hàm

Ví dụ 2: Vẽ đồ thị hàm số y1 = sinx.cosx và hàm y2 = $\sin x^2$ trong [0;2π], trên cùng hệ trục tọa độ.

Ta thực hiện mã lệnh như sau:

```
x = 0:0.01:2*pi;
y1 = sin(x).*cos(2*x);
plot(x,y1); grid on;hold on;
y2 = sin(x.^2);
plot(x,y2,'k');
axis([0 4*pi -1.25 1.25]);
xlabel('time');ylabel('amplitude');
title('y1 = sinx.cos2x and y2=sin(x^2)');
legend('sinx.cos2x','sinx^2');
```



***Hình 1.6** – Đồ thị mô phỏng ví dụ 1.1*

1.4 CÁC FILE VÀ HÀM

1.4.1 Script file(file kịch bản)

Kịch bản là M-file đơn giản nhất, không có đối số. Nó dùng khi thi hành một loạt lệnh matlab theo một trình tự nhất định. Ta xét ví dụ vẽ đồ thị hình sin:

```
x = -2*pi:pi/6:2*pi;
y = sin(x) ;plot(x,y) ;
```

Lưu chương trình với tên hinhsin.m bằng cách nhấn Ctrl+S hay nhấn vào biểu tượng Save .

Thực thi chương trình trên bằng cách nhấn phím F5, hoặc nhấn vào biểu tượng Run, hoặc trong Command window bằng dòng lệnh sau:

>> hinhsin

1.4.2 File hàm

Hàm là M-file có chứa các đối số.

Ví dụ 3: Ta có một ví dụ hàm xây dựng hàm ***gptb2*** để giải phương trình bậc 2: $ax^2 + bx + c = 0$. Nội dung hàm như sau:

```
function [x1,x2]=gptb2(a,b,c)
if nargin <3
        error('Error! Nhap 3 he so cua phuong trinh')
 elseif a==0
         x1=-c/b; x2=[];
else
         delta = b^2 - 4*a*c;
 x1 = (-b+sqrt(delta))/(2*a); x2 = (-b-sqrt(delta))/(2*a);
end
```

Từ ví dụ trên ta thấy một hàm M-file gồm các phần tử cơ bản sau :

+ Một dòng định nghĩa hàm : fuction y = gptb2(x) gồm các từ khóa function, đối số trả về y, tên hàm gptb2 và đối số vào x.

+ Thân hàm chứa mã Matlab.

+ Các lời giải thích dùng để cho chương trình rõ ràng. Nó được đặt sau dấu %. Cần lưu ý là tên hàm phải bắt đầu bằng ký tự và cùng tên file chứa hàm. Tên hàm là ***gptb2*** thì tên file cũng là gptb2.m.

Lưu ý rằng trong một tên file.m có thể có nhiều hàm nhưng hàm đầu tiên phải có tên trùng với tên file.

Kiểm tra kết quả :

```
[x1,x2]=gptb2(1,6,-7)
[x1,x2]=gptb2(0,4,3)
[x1,x2]=gptb2(1,6)
gptb2(2,7,14)
```

Các hàm toán học cơ bản

exp(x) hàm mũ cơ số e	*angle(x)* argument của số phức x
sqrt(x) căn bậc hai của x	*imag(x)* phần ảo của x
log(x) logarit cơ số e	*real(x)* phần thực của x
log10(x) logarit cơ số 10	*sign(x)* dấu của x
abs(x) module của số phức x	

II. TÍN HIỆU RỜI RẠC THỜI GIAN

2.1 CÁC DÃY SỐ CƠ BẢN

a) Dãy xung đơn vị.

$$\delta(n)=\begin{cases}1 & n=0\\ 0 & n\neq 0\end{cases} \qquad \delta(n-n_0)=\begin{cases}1 & n=n_0\\ 0 & n\neq n_0\end{cases}$$

Trong Matlab để tạo dãy xung đơn vị bằng cách gõ các dòng lệnh vào cửa sổ soạn thảo (Editor) và lưu lại thành file "**impseq.m**" như sau:

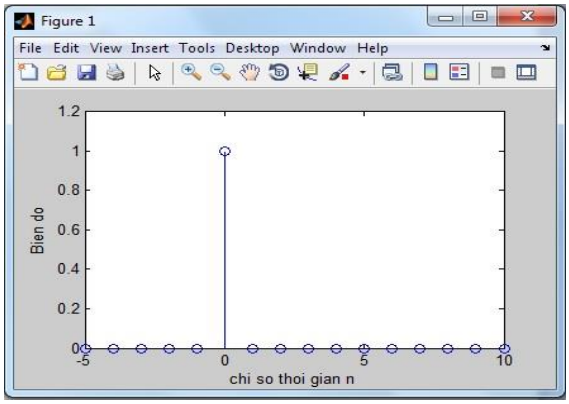
Ví dụ 4:

```
function [x,n] = impseq(n0,n1,n2)
% tao ra day x(n) = delta(n-n0); n1 <= n0,n0 <= n2
if ((n0 < n1) | (n0 > n2) | (n1 > n2))
 error('arguments must satisfy n1 <= n0 <= n2')
end
n = [n1:n2];
x = [(n-n0) == 0];
```

Trên cơ sở xây dựng được hàm **impseq** chúng ta mô phỏng tín hiệu dãy xung đơn vị như sau:

```
% Phát một dãy xung đơn vị
% tạo vector từ -5 đến 10
n= [-5 :10];
% phát ra dãy xung đơn vị
x = impseq(0,-5,10);
% vẽ ra dãy xung
stem(n,x);xlabel('chi so thoi gian n');
ylabel('Bien do');
```

Đồ thị mô phỏng được thể hiện như hình 2.1.



Hình 2.1 *- Dãy xung đơn vị*

b) Dãy nhảy đơn vị (Dãy bậc đơn vị).

$$u(n)=\begin{cases}1 & n\geq 0\\ 0 & n\prec 0\end{cases} \qquad u(n-n_0)=\begin{cases}1 & n\geq n_0\\ 0 & n<n_0\end{cases}$$

Tạo dãy nhảy xung đơn vị theo chương trình mẫu bằng cách gõ các dòng lệnh cho ở bảng dưới đây. Vào của sổ soạn thảo (Editor) và lưu lại thành file "**stepseq.m**".

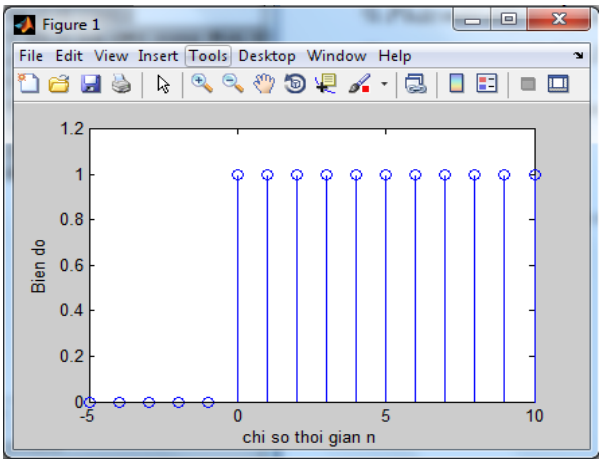
Ví dụ 5:

```
function [x,n] = stepseq(n0,n1,n2)
% tao ra day x(n) = u(n-n0); n1 <= n0,n0 <= n2
if ((n0 < n1) | (n0 > n2) | (n1 > n2))
 error('arguments must satisfy n1 <= n0 <= n2')
end
n = [n1:n2];
x = [(n-n0) >=0];
```

Sau đây là chương trình mô phỏng dãy nhảy bậc:

```
% Phát một dãy nhảy bậc
% tạo vector từ -5 đến 10
n= [-5 :10];
% phát ra dãy nhảy bậc
x = stepseq(0,-5,10);
% vẽ ra dãy xung
stem(n,x);
xlabel('chi so thoi gian n');
ylabel('Bien do');
```

Đồ thị được vẽ như hình 2.3.



Hình 2.3 *- Đồ thị dãy nhảy bậc*

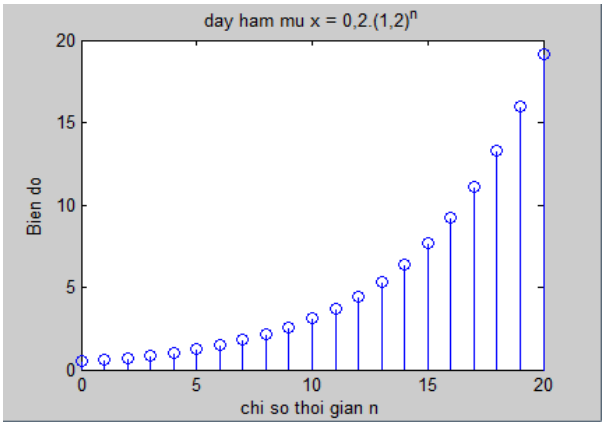
c. Dãy hàm mũ thực.

$$e(n) = \begin{cases} a^n & n \geq 0 \\ 0 & n \prec 0 \end{cases}$$

Ví dụ 6:

```
% Phát ra một dãy hãm mũ thực
% tạo vector từ 0 đến 20
n= [0 :20];   a =1.2; K = 0.5;
x = K*a.^n;
% vẽ ra dãy xung
stem(n,x);
xlabel('chi so thoi gian n');
ylabel('Bien do');
title('day ham mu x = 0,2.(1,2)^n');
```

Đồ thị được vẽ như hình 2.5.



Hình 2.5*- Dãy hàm mũ* thực

a. Dãy sin.

$$s(n) = \sin(\omega 0 n)$$

Dãy sin thực: $x(n) = A\sin(\omega 0 n + \phi)$; với A, ω0 và ϕ là các số thực, A là biên độ, ω0 là tần số góc, ϕ là pha ban đầu. Chương trình Matlab mô phỏng tín hiệu sin như sau:

Ví dụ 7:

% Phát ra dãy sin thực
% tạo vector từ 0 đến 40
n= [0:40];A =2;f = 0.1;phase = 0;
*x = A*sin(2*pi*f*n – phase);*
% vẽ ra dãy xung sin
stem(n,x);
xlabel('chi so thoi gian n');ylabel('Bien do');
title('day ham sin x ');
Đồ thị được biểu diễn như hình vẽ 2.7.

2.2. MỘT SỐ PHÉP TOÁN DÃY SỐ

2.2.1. Trễ(dịch)

$$y(n) = x(n - n0)$$

Trong Matlab phép toán này được thực hiện bởi hàm **sigshift** được định nghĩa như sau:

function [y,n] = sigshift(x,m,n0)
% ham dich chuyen y[n] = x(n - n0);
n = m + n0; y = x;

2.2.2 Phép đảo(Phép chuyển vị)

Phép toán này được thực hiện bằng cách thay đổi dấu của đối số của tín hiệu đó:

y(n) = x(- n) với mọi n.

Trong Matlab phép toán này được thực hiện bởi hàm **sigfold** được định nghĩa như sau:

function [y,n] = sigfold(x,n)
% thuc hien y(n) = x(-n)
y = fliplr(x); n = -fliplr(n);

2.2.3 Tổng của hai dãy.

Để thực hiện phép cộng trong Matlab thì các dãy phải có cùng chiều dài, trong trường hợp các dãy có chiều dài khác nhau thì trước hết phải tăng một số mẫu có giá trị bằng không để các dãy có cùng độ dài. Việc này thực hiện được nhừ toán tử giao "&" và các toán tử "<=" ; "=="; hàm ***find*** khi đó hàm số ***sigadd*** minh họa toán tử đó như sau:

function [y,n] = sigadd(x1,n1,x2,n2) %thuc hien y(n) = x1(n)+x2(n)
n=min(min(n1),min(n2)):max(max(n1),max(n2));
y1=zeros(1,length(n)); y2 = y1; % khoi tao
y1(find((n>=min(n1))&(n<=max(n1))==1))=x1; % x1 voi chi so cua y(n)
y2(find((n>=min(n2))&(n<=max(n2))==1))=x2; % x2 voi chi so cua y(n)
y=y1+y2;

Ví dụ 8

Giả sử x(n) = {1;2;3;4;5;6;7;6;5;4;3;2;1}

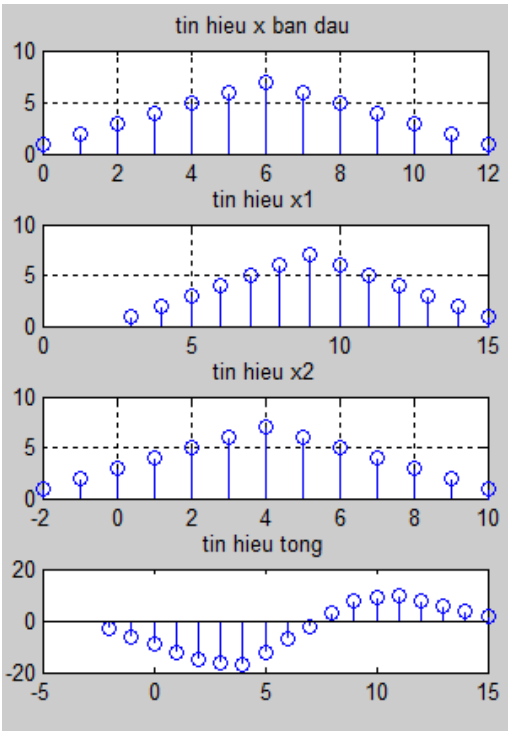
Hãy xác định và vẽ các dãy sau đây:

a. *x1(n) = x(n-3); x2(n) = x(n+2);*

b. *x3(n) = 2.x(n-3)-3 x(n+2) ;*

```
n = [0:12]
x = [1:1:7,6:-1:1]
[x1,n1] = sigshift(x,n,3);
[x2,n2] = sigshift(x,n,-2);
[x3,n3] = sigadd(2.*x1,n1,-3.*x2,n2);
subplot(4,1,1);stem(n,x);grid;
subplot(4,1,2);stem(n1,x1);grid;
subplot(4,1,3);stem(n2,x2);grid;
subplot(4,1,4);stem(n3,x3);
```

Đồ thị được biểu diễn như hình 2.9.



***Hình 2.9** - Tổng của hai dãy*

2.2.4 Tích của hai dãy.

Trong Matlab phép toán này được thực hiện nhờ toán tử mảng(array). Ngoài ra nó còn được thực hiện nhờ hàm Matlab **sigmult** được định nghĩa như sau:

```
function [y,n] = sigmult(x1,n1,x2,n2)   % thuc hien y(n) = x1(n) * x2(n)
n = min(min(n1),min(n2)):max(max(n1),max(n2));
y1 = zeros(1,length(n)); y2 = y1;
y1 (find((n>=min(n1))&(n<=max(n1))==1)) =x1;
y2 (find((n>=min(n2))&(n<=max(n2))==1)) =x2;
y = y1.*y2;
```

Ví dụ 9

Giả sử x(n) = {1;2;3;4;5;6;7;6;5;4;3;2;1}

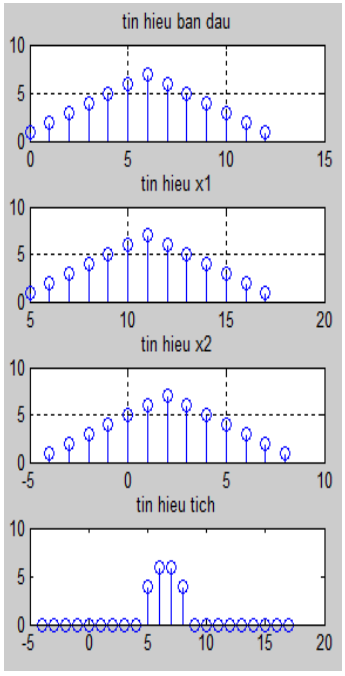
Hãy xác định và vẽ các dãy sau đây:

a. x1(n) = x(n - 5); x2(n) = x(n + 4);

b. x3(n) = x1(n) . x2(n))

```
n = [0:12] ; x = [1,2,3,4,5,6,7,6,5,4,3,2,1]
[x1,n1] = sigshift(x,n,5); [x2,n2] = sigshift(x,n,-4);
[x3,n3] = sigmult(x1,n1,x2,n2);
subplot(4,1,1);stem(n,x);title('tin hieu ban dau');grid;
subplot(4,1,2);stem(n1,x1);title('tin hieu x1');grid;
subplot(4,1,3);stem(n2,x2);title('tin hieu x2');grid;
subplot(4,1,4);stem(n3,x3); title('tin hieu x3');grid;
```

Đồ thị được biểu diễn như hình 2.10.



Hình 2.10.

2.2.5 Tích chập

Trong Matlab có hàm conv thực hiện trả về một dãy là kết quả của phép tính tích chập giữa hai dãy được cho theo tham số đầu vào của hàm conv. Tuy nhiên, các dãy đầu vào và đầu ra cũng như kết quả đều không nói lên chỉ số bắt đầu và chỉ số kết thúc của dãy mà chỉ được ngầm hiểu là các dãy bắt đầu từ chỉ số 0. Tạo hàm tính tích chập có tên conv_m thực hiện việc tính tích chập của hai dãy, mà mỗi dãy được thể hiện bởi 2 vector , một vector thể hiện chỉ số, một vector thể hiện giá trị của dãy, giống như các dãy được biểu diễn ở các bước tiến hành trước bằng cách gõ các dòng lệnh theo bảng dưới đây vào cửa sổ soạn thảo và ghi lại theo tên tệp **conv_m.m**.

```
function [y,ny] = conv_m(x,nx,h,nh)
% Ham tinh tich chap da duoc sua doi danh cho xu ly so tin hiệu
nyb = nx(1)+nh(1); nye = nx(length(x))+nh(length(h));
ny = [nyb:nye]; y = conv(x,h);
```

Ví dụ 10

Cho dãy x(n) = [u(n) – u(n-10)] và h(n) = 0.9^n.u(n). Xác định tích chập y(n) = x(n)*h(n). Chương trình mô phỏng như sau:

```
n = -5:50; u1 = stepseq(0,-5,50); u2=stepseq(10,-5,50);
x = u1-u2;  h = ((0.9).^n).*u1;
figure(1);subplot(2,1,1); stem(n,x); axis([-5,50,0,2]);
title('Day dau vao'); xlabel('n'); ylabel('x(n)');
subplot(2,1,2); stem(n,h); axis([-5,50,0,2]);
title('Dap ung xung'); xlabel('n'); ylabel('h(n)');
 [y,ny] = conv_m(x,n,h,n);
figure(2);stem(ny,y); axis([-5,50,0,8]);title('Output Sequence'); xlabel('n');
ylabel('y(n)');
```

Đồ thị mô phỏng kết quả phép tính tích chập được biểu diễn như hình 2.11.

2.4. Hệ thống tuyến tính bất biến và nhân quả

Trong Matlab, người ta thường sử dụng lệnh h = *impz*(b,a,n+1) để tính đáp ứng xung của hệ thống rời rạc LTI. Ta xét hệ thống có phương trình sai phân:

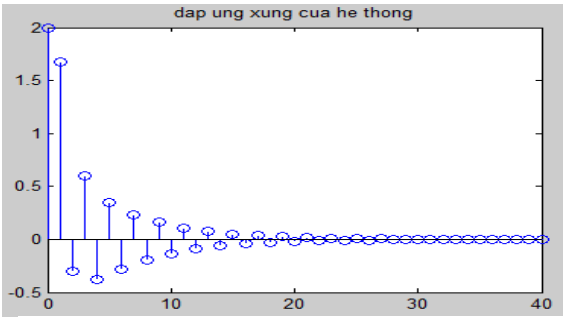
y(n) + 0,5y(n -1) - 0,27y(n -3) = 1,2x(n) + 2,6x(n -1)+ 2,35x(n - 2)

Tìm đáp ứng xung của hệ thống ?.

Chương trình được thực hiện như sau:

```
clear all;
n = 0:40;
a = [1 0.5 0 -0.27];
b = [1.2 2.6 2.35];
h = impz(b,a,n+1);
stem(n,h);title('dap ung xung cua he thong');
```

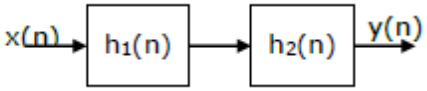
Đồ thị được biểu diễn như hình 2.14.



Hình 2.14 *Đáp ứng xung của hệ thống*

a. Ghép nối tiếp các hệ thống LTI

Trong thực tế, người ta thường ghép nối tiếp các hệ thống LTI nhân quả bậc thấp với nhau để được các hệ thống bậc cao. Chẳng hạn, người ta ghép hai hệ thống bậc 2 có phương trình sai phân:



Hình 2.15 *- Sơ đồ ghép nối hai hệ thống*

$$y_1(n)+0,9y_1(n-1)+0,8y_1(n-2)=0,3x(n)-0,3x(n-1)+0,4x(n-2)$$

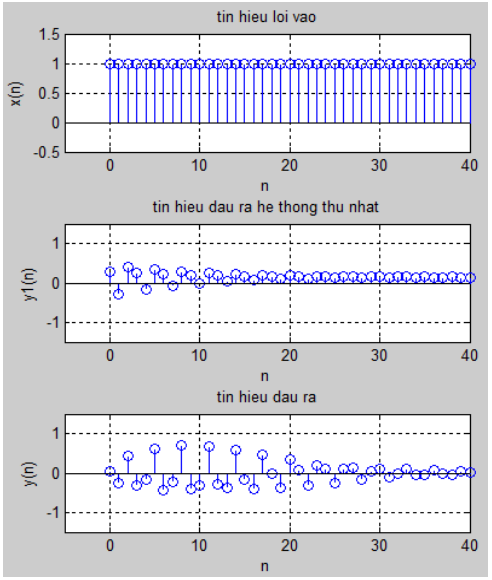
$$y_2(n)+0,7y_2(n-1)+0,85y_2(n-2)=0,2y_1(n)-0,5y_1(n-1)+0,3y_1(n-2)$$

Xét ví dụ mô phỏng hệ thống nối tiếp hai hệ thống bậc 2 giữa hai hệ thống trên. Trong chương trình này, tín hiệu x(n) là tín hiệu kích thích của hệ thống ghép nối.

Ví dụ 11

```
x = stepseq(0,0,40);n=[0:40];
 a1 = [1 0.9 0.8]; b1 = [0.3 -0.3 0.4];
a2 = [1 0.7 0.85]; b2 = [0.2 -0.5 0.3];
y1 = filter(b1,a1,x); % Tinh tin hieu ra y1(n)
y2 = filter(b2,a2,y1);
 subplot(3,1,1)stem(n,x);title('tin hieu loi vao');
axis([-5 40 -0.5 1.5 ]);grid;xlabel('n');ylabel('x(n)');
subplot(3,1,2);stem(n,y1);
title('tin hieu dau ra he thong thu nhat');
axis ([-5 40 -1.5 1.5]);grid;xlabel('n');ylabel('y1(n)');
subplot(3,1,3); stem(n,y2);title('tin hieu dau ra');
axis([-5 40 -1.5 1.5]);grid;xlabel('n');ylabel('y(n)');
```

Đồ thị mô phỏng ở hình vẽ 2. 16.



Hình 2.16. *Mô phỏng ví dụ 2.8*

2.5 BÀI TẬP THỰC HÀNH

Bài 1: Viết chương trình và vẽ dạng tín hiệu hàm u(n-3), u(n+2), $rect_3$(n-2); δ(n+1) với n ∈[-10;10]

Bài 2: Viết chương trình thể hiện trên đồ thị kết quả phép tính tích chập giữa hai dãy:

$$x(n) = rect_6(n)$$

$$h(n) = \begin{cases} 1 - \dfrac{n}{4} & 0 \le n < 4 \\ 0 & n \text{ còn lại} \end{cases}$$

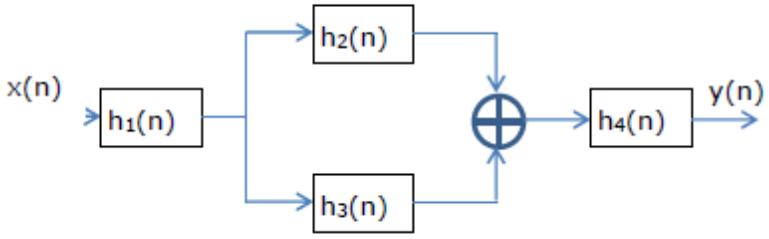
với $-4 \le n \le 10$

Bài 3: Xét hệ thống có phương trình sai phân: y(n) = 0.3x(n) + 0.2x(n - 1) - 0.3x(n -2) - 0.9y(n - 1) + 0.9y(n - 2).

a. Xác định đáp ứng xung đơn vị của hệ thống.

b. Tìm đầu ra của hệ thống khi biết đầu vào $x(n) = \begin{cases} 2 & với\ n = -2 \\ n+1 & với\ -1 \le n \le 1 \\ 0 & n\ còn\ lại \end{cases}$

Bài 4: Xác định đáp ứng xung tương đương của hệ thống sau:



Hình 2.17 *Sơ đồ ghép nối của bài 6*

$h_1 = \{1,0,-1,3\}$; $h_2 = \{2,-2,1\}$; $h_3 = \{3,4,-1,1\}$; $h_4 = \{-3,5,6,-1,1\}$

Viết chương trình xác định ngõ ra của hệ thống khi ngõ vào là $x(n) = (-2)^n u(n-2)$.(Tính toán cho giá trị n từ -20 đến 20).