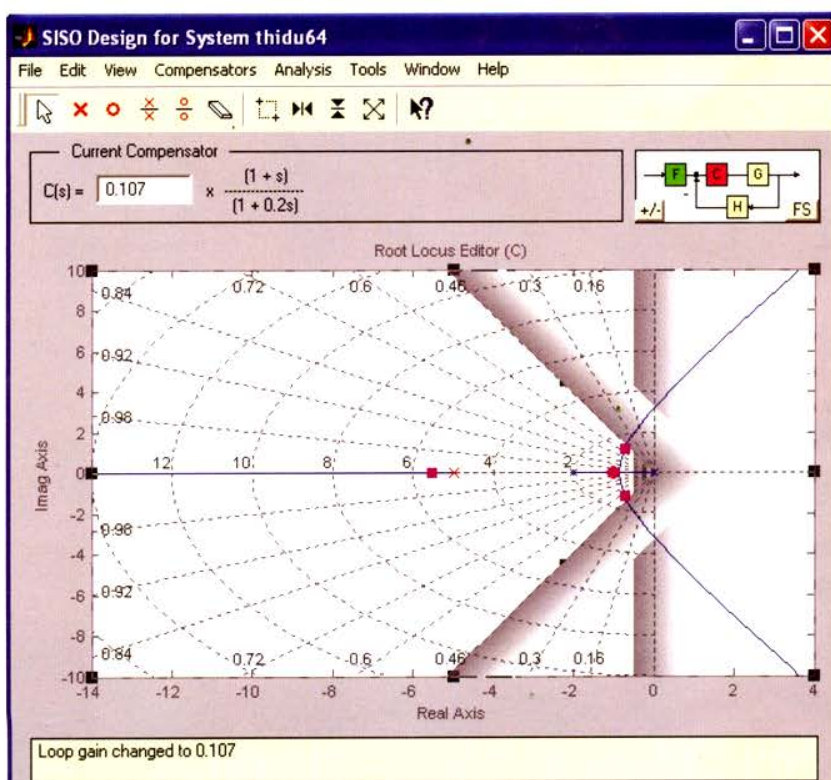


THÍ NGHIỆM ĐIỀU KHIỂN TỰ ĐỘNG



Biên soạn: HUỖNH MINH NGỌC

Lưu hành nội bộ

NĂM 2009

LỜI NÓI ĐẦU

Thí nghiệm điều khiển tự động là môn học minh họa và bổ sung phần thực tế cho giáo trình Lý thuyết điều khiển tự động. Sinh viên tiếp cận công cụ Matlab để khảo sát và thiết kế hệ thống tự động và khảo sát các hệ thống điều khiển tự động thực tế. Điều khiển tự động ngày nay có mặt trong các lĩnh vực điện - điện tử, cơ khí, hóa, giao thông, qui trình sản xuất ở nhà máy.

Giáo trình gồm có bảy bài thí nghiệm:

1. Khảo sát hệ thống tự động dùng Matlab.
2. Thiết kế hệ thống tự động dùng Matlab.
3. Khảo sát và mô phỏng hệ thống tự động dùng Simulink.
4. Hệ thống điều khiển nhiệt độ.
5. Hệ thống điều khiển vị trí và tốc độ động cơ DC.
6. Hệ thống điều khiển số động cơ một chiều.
7. Hệ thống điều khiển mức nước.

Giáo trình dùng để giảng dạy cho sinh viên ngành Điện tử tự động hệ đại học. Chắc chắn giáo trình còn nhiều thiếu sót. Tác giả chân thành cảm ơn các ý kiến đóng góp của các thầy cô giáo trong bộ môn Điều khiển tự động, các đồng nghiệp, và bạn đọc để giáo trình ngày hoàn thiện hơn. Thư góp ý xin gửi về bộ môn Điều khiển tự động, Khoa Công nghệ Điện tử, trường Đại học Công nghiệp Tp. Hồ Chí Minh, số 12 Nguyễn Văn Bảo, P.4, Q. Gò Vấp, Tp. HCM, ĐT: 38940390.

Tác giả

ThS. Huỳnh Minh Ngọc

THIẾT BỊ CHÍNH CHO CÁC BÀI THỰC TẬP

I. Thiết bị chính bao gồm các chức năng sau:

- Máy tính Pentium IV.
- Phần mềm Matlab 6.5, 7.0 hay cao hơn và Simulink.
- Dao động kí và VOM/DMM.
- Dây nguồn, cable và dây có chốt cắm hai đầu.
- Card PCI -1711.
- Mô hình thực tập điều khiển nhiệt độ.
- Mô hình thực tập điều khiển tốc độ và vị trí động cơ một chiều.
- Mô hình điều khiển số động cơ một chiều.
- Mô hình điều khiển mức nước.

II. Đặc trưng và chức năng của thiết bị chính

- Dùng minh họa và bổ sung phần thực tế cho giáo trình Lý thuyết điều khiển tự động. Có bảy bài thí nghiệm chia làm ba phần: Khảo sát và thiết kế hệ thống điều khiển tự động dùng Matlab, khảo sát các hệ thống điều khiển tương tự và hệ thống điều khiển rời rạc.
1. Khảo sát hệ thống điều khiển tự động dùng MATLAB.
 2. Thiết kế hệ thống điều khiển tự động dùng MATLAB.
 3. Khảo sát và mô phỏng hệ thống tự động dùng Simulink.
 4. Hệ thống điều khiển nhiệt độ.
 5. Hệ thống điều khiển tốc độ và vị trí động cơ một chiều.
 6. Hệ thống điều khiển số động cơ một chiều.
 7. Hệ thống điều khiển mức nước.

III. Các bài thực tập về điều khiển tự động sử dụng với thiết bị chính

STT	Danh mục		Vật tư
1	Khảo sát hệ thống điều tự động dùng Matlab	Automatic control system analysis with MATLAB	Máy tính Pentium IV và phần mềm MATLAB
2	Thiết kế hệ thống điều khiển tự động dùng Matlab	Automatic control system design with Matlab	Máy tính Pentium IV và phần mềm Matlab
3	Khảo sát và mô phỏng hệ thống tự động dùng Simulink	Analysis and simulation of control systems with Simulink	Máy tính Pentium IV và phần mềm Matlab
4	Hệ thống điều khiển nhiệt độ	Temperature Control Systems	Mô hình điều khiển nhiệt độ
5	Hệ thống điều khiển tốc độ và vị trí động cơ một chiều	Speed and Position Control systems	Mô hình điều khiển tốc độ và vị trí động cơ một chiều
6	Điều khiển số động cơ một chiều	Digital DC motor control system	Mô hình điều khiển số động cơ một chiều
7	Hệ thống điều khiển mức nước	Water level control systems	Mô hình điều khiển mức nước trong bồn

BÀI 1

KHẢO SÁT HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG DÙNG MATLAB

1.1. Mục tiêu

- Căn bản Matlab: giới thiệu, mô tả toán học hệ thống dùng hàm truyền đạt, phương pháp biến trạng thái, đặc tính động học, khảo sát tính ổn định hệ, sai số xác lập.
- Phân tích ở miền tần số: Biểu đồ Bode, Nyquist.
- Khảo sát hệ dùng phương pháp quỹ đạo nghiệm số.

1.2. Nội dung

1.2.1. Phần mềm MATLAB và cách sử dụng

1.2.1.1. Giới thiệu về Matlab

Matlab được phát triển bởi công ty Math Works Inc, là một chương trình phân tích ma trận, thiết kế điều khiển, nhận dạng hệ thống và đồ thị kỹ thuật.

Gõ đường dẫn: MATLAB khi đó sẽ xuất hiện dấu nhắc ">>" hoặc double-click vào biểu tượng Matlab trên màn hình nền.

Matlab cũng có khả năng thi hành một dãy lệnh chứa trong một tập tin, tập tin file.M. Một chương trình có thể được viết và ghi lại ở dạng ASCII với tên tập tin có phần mở rộng ra trong thư mục mà Matlab đang chạy. Tên tập tin phải ở dạng chữ thường. Dạng tập tin file.M này được coi như là tập tin hay tập lệnh nguyên bản. Để chạy chương trình tại dấu nhắc ">>" đánh tên tập tin không có phần mở rộng .m.

1.2.1.2. Lệnh và Biến số

Lệnh có dạng ở hình sau. Matlab dùng phép gán để dấu « = » chỉ ra rằng gán biểu thức vào một biến.

>>biến= biểu thức

Dạng lệnh Matlab

>>: dấu nhắc lệnh

Biểu thức nhập vào mà không có tên biến thì Matlab tính toán và kết quả được nhớ và hiển thị sau từ ans. Kết quả một biểu thức gán vào một tên biến để tiện cho việc sử dụng khác. Tên biến có thể đến 19 ký tự (bao gồm chữ và số), tuy nhiên ký tự đầu tiên của một tên biến phải bắt đầu bằng một chữ.

Chuỗi ký tự:

Một chuỗi ký tự nối tiếp nhau trong các câu sau được gọi là một chuỗi ký tự hoặc biến câu.

```
>>C='Good'
```

Kết quả sẽ là:

```
C=Good
```

Bảng 1.1. Các phép toán học

+	Cộng
-	Trừ
*	Nhân
/	Chia
^	lũy thừa

1.2.1.3. Các phép tính toán vector

Một vector bậc n là một dãy hàng hay dãy cột của n số hạng. Để biểu diễn vector cột ta nhập: các phần tử nằm trong dấu ngoặc vuông [], cách nhau bởi dấu chấm phẩy (;). Ví dụ:

```
>>X=[2;-4;8]
```

Kết quả là:

```
X=
```

```
2
```

```
-4
```

```
8
```

Nếu các phần tử được phân cách bởi các dấu phẩy hoặc khoảng trắng thì đó là vector hàng. Ví dụ:

```
>>R=[tan(pi/4) sqrt(9) -5]
```

Kết quả sẽ là:

R=

1.0000 3.0000 -5.0000

Chuyển vị một vector. Ví dụ:

```
>>Y=R'
```

sẽ cho ra:

Y=

1.0000

3.0000

-5.0000

Các vector có cùng kích thước có thể cộng hoặc trừ. Ví dụ:

```
>>P=5*R
```

 sẽ cho ra kết quả sau:

P=

5.0000 15.0000 -25.0000

Toán tử * thực hiện phép tính nhân tương ứng các phần tử với nhau. Ví dụ: với X và Y ở trên ta có:

```
>>E=X*Y
```

Kết quả sẽ là:

E=

2

-12

-40

Bài 1: Khảo sát hệ thống điều khiển tự động dùng MATLAB

Tích vô hướng của 2 vector X và Y là số vô hướng được xác định bởi $\sum_{i=1}^n X_i Y_i$. Nếu X và Y là hai vector cột được xác định ở trên thì tích vô hướng của chúng là: $S=X'*Y$.

Kết quả sẽ là:

S=

-50

Có các hàm của Matlab để tính chuẩn của vector. Ví dụ tính Norm Euclid

>>N=norm(X)

Đưa ra kết quả:

N=

9.1652

Góc giữa hai vector X và Y được xác định bằng $X.Y/(\text{Modun}X. \text{Modun}Y)$. Dòng lệnh:

» theta=acos(X'*Y/(norm(X)*norm(Y)))

Kết quả là:

theta =

2.7444

trong đó theta được đo bằng radian.

Biểu diễn một vector zero. Ví dụ:

» Z=zeros(1,4)

Z =

0 0 0 0

Trong MATLAB (:) có thể sử dụng để tính vector hàng. Ví dụ như:

» x=1:8

cho ra một vector hàng của các số nguyên từ 1 đến 8.

x =

1 2 3 4 5 6 7 8

1.2.1.4. Ma trận

Trong MATLAB một ma trận được tạo bởi một dãy số trong ngoặc vuông. Các phần tử trong mỗi hàng được phân biệt bởi các khoảng trống hoặc dấu phẩy. Dấu chấm phẩy được dùng để kết thúc một hàng. Ví dụ:

```
» A=[6 1 2;-1 8 3;2 4 9]
```

Kết quả là:

A =

```
6   1   2
-1   8   3
2   4   9
```

Một cột hoặc hàng của một ma trận có thể được ký hiệu bằng (:). Ví dụ:

```
» r3=A(3,:)
```

Kết quả là:

r3 =

```
2   4   9
```

Tương tự A(:,2) biểu thị tất cả các phần tử của cột thứ 2 trong A.

```
» A(:,2)
```

ans =

```
1
8
4
```

Cộng 2 ma trận A và B. Ví dụ:

```
» A=[6 1 2;-1 8 3;2 4 9]
```

Bài 1: Khảo sát hệ thống điều khiển tự động dùng MATLAB

A =

```
6   1   2
-1   8   3
2   4   9
```

» B=[1 2 3;-2 4 6;1 3 7]

B =

```
1   2   3
-2   4   6
1   3   7
```

» D=A+B

D =

```
7   3   5
-3  12   9
3   7  16
```

Trừ hai ma trận A và B. Ví dụ:

» C=A-B

C =

```
5  -1  -1
1   4  -3
1   1   2
```

Nhân hai ma trận A và B từ phép tính $A*B$ nếu chúng tương thích. Ví dụ:

» $E=A*B$

E =

```
6 22 38
-14 39 66
3 47 93
```

Hai ký hiệu được sử dụng để chia ma trận. $A \backslash B$ tương ứng $A^{-1}*B$ và A/B tương ứng với $A*B^{-1}$. Ví dụ:

$AX=B$

$$\begin{bmatrix} 4 & -2 & -10 \\ 2 & 10 & -12 \\ -4 & -6 & 16 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -10 \\ 32 \\ -16 \end{bmatrix}$$

» $A=[4 -2 -10;2 10 -12;-4 -6 16]$

A =

```
4 -2 -10
2 10 -12
-4 -6 16
```

» $B=[-10;32;-16]$

B =

```
-10
32
-16
```

» $X=A \backslash B$

Kết quả là:

X =

Bài 1: Khảo sát hệ thống điều khiển tự động dùng MATLAB

2.0000

4.0000

1.0000

Ngoài ra còn rất nhiều hàm toán học khác trong file.m.

Nên sử dụng chức năng **inv** để xác định nghịch đảo của ma trận A rồi sau đó xác định ma trận X.

```
» A=[4 -2 -10;2 10 -12;-4 -6 16]
```

A =

4 -2 -10

2 10 -12

-4 -6 16

```
» B=[-10;32;-16]
```

B =

-10

32

-16

```
» C=inv(A)
```

C =

2.2000 2.3000 3.1000

0.4000 0.6000 0.7000

0.7000 0.8000 1.1000

```
» X=C*B
```

X =

2

4

1

1.2.1.5. Giá trị riêng

Nếu ma trận A là một ma trận có $(n \times n)$ phần tử, thì có n số λ thỏa mãn $Ax = \lambda x$ là giá trị riêng của A . Chúng tìm được bằng cách sử dụng lệnh **eig**(A). Giá trị riêng và vector riêng của A cũng có thể tìm được bằng lệnh **[X,D]=eig(A)**. Các phần tử trên đường chéo chính của ma trận chéo D là các λ , còn các cột của ma trận X là các vector riêng thỏa mãn $AX=XD$. Ví dụ: Tìm giá trị riêng và vector riêng của ma trận A cho bởi:

» $A=[0 \ 1 \ -1;-6 \ -11 \ 6;-6 \ -11 \ 5]$

A =

0 1 -1

-6 -11 6

-6 -11 5

» **[X,D]=eig(A)**

X =

0.7071 -0.2182 -0.0921

0.0000 -0.4364 -0.5523

0.7071 -0.8729 -0.8285

D =

```
-1.0000    0    0
    0 -2.0000    0
    0    0 -3.0000
```

1.2.1.6.Số phức

Hầu hết các phép tính số phức đều có thể sử dụng được trong chương trình Matlab. Số ảo $\sqrt{-1}$ được ngầm định trước bởi hai biến số i và j trong chương trình. Nếu i và j được sử dụng cho các giá trị khác thì ta phải định nghĩa phần ảo như sau:

```
» j=sqrt(-1)
```

j =

```
0 + 1.0000i
```

hoặc:

```
» i=sqrt(-1)
```

i =

```
0 + 1.0000i
```

Ví dụ: Tính $Z_c \cosh g + \sinh g/Z_c$, với $Z_c=200+ i300$ và $g=0.02 +j1.5$

```
» i=sqrt(-1)
```

i =

```
0 + 1.0000i
```

```
» Zc=200+300*i;
```

```
» g=0.02+1.5*j;
```

» $V = Z_c * \cosh(g) + \sinh(g) / Z_c$

Kết quả là:

V =

8.1672 + 25.2172i

1.2.1.7. Đồ thị

Chương trình MATLAB có thể tạo ra các loại đồ thị 2-D, 3-D, đường log, semilog, đồ thị cực, đồ thị khối và đường viền trên máy vẽ, máy in kim, máy in laser. Một số đồ thị 2-D có thể vẽ bằng các lệnh **plot**, **loglog**, **semilogx**, **semilogy**, **polar('text')**, **xlabel('text')**, **ylabel('text')**, và **title('text')** có thể được sử dụng để đặt tên và thêm các chú thích trên đồ thị. Cú pháp của các câu lệnh trên bao gồm các ký hiệu (.,+,*,0,x) và màu sắc (r,b,g,w). Ví dụ: dòng lệnh sau

» `plot(t,y1,'r',t,y2,'+b')`

sẽ cho màu đỏ trên đường cong thứ nhất và màu xanh trên đường cong thứ hai.

Lệnh `plot(x,y)`- vẽ đồ thị y theo x.

Ví dụ: Vẽ đồ thị quan hệ x-y với nhiều biến khác nhau:

X 0 1 2 3 4 5 6 7 8 9 10 11 12

Y 0 0.5 1 2 4 7 11 14 15.5 16 16 16 16

» `x=[0 1 2 3 4 5 6 7 8 9 10 11 12];`

» `y=[0 0.5 1 2 4 7 11 14 15.5 16 16 16 16];`

» `plot(x,y)`

» `grid`

» `meta EX18`

Đồ thị x-y của ví dụ.

Lệnh meta cho phép mở file EX18.MET và ghi đồ thị vào đó.

Đồ thị mặt lưới 3 chiều:

Lệnh mesh (Z) tạo một đồ thị 3 chiều của các phần tử trong ma trận Z. Bề mặt lưới được định nghĩa bởi tọa độ Z của các điểm ở trên một ô lưới trong mặt phẳng X-Y. Biểu đồ được hình thành bởi sự liên kết các điểm gần kề với các đường thẳng. Meshdom làm biến đổi phạm vi định rõ bởi vector x và y vào trong các dãy X và Y.

Ví dụ: Để có đồ thị 3D của các của hàm Bessel $j_0 \sqrt{x^2 + y^2}$ trong khoảng $-12 < x < 12$, $-12 < y < 12$.

Ta có các dòng lệnh sau:

```
» clf;% thay cho clg
» [x,y]=meshgrid(-12:.6:12,-12:.6:12);% thay cho meshdom
» r=sqrt(x^2+y^2);
» z=bessel(0,r);
» m=[-45 60];
» mesh(z,m)
```

1.2.1.8. Script - Tập tin .m

Matlab cho phép thực hiện một chuỗi lệnh trong một tập tin. Tập tin này gọi là M-file vì tên có dạng filename.m. Một scripts là một loại M-file. Một script là một chuỗi các lệnh thông thường và hàm được dùng ở dấu nhắc lệnh.

Hộp công cụ hệ thống điều khiển (Control systems Toolbox) là tập hợp các hàm để mô hình hóa, phân tích và thiết kế hệ thống điều khiển. Nó cung cấp các công cụ cổ điển như là vẽ Bode, biểu đồ Nichols, quỹ đạo nghiệm số, cũng như các kỹ thuật hiện đại như không gian trạng thái LQG và thiết kế đặt cực.

Thí dụ:

```
>>alpha=50;
```

```
>>plotdata
```

```
plotdata.m
```

```
%Day la script ve ham y=sin(alpha*t)
```

```
%Giá trị alpha phải tồn tại trong workspace trước khi kích hoạt script.
```

```
t=[0:0.01:1];  
y=sin(alpha*t);  
Plot(t,y)  
Xlabel('Time(sec)')  
Ylabel('y(t)=sin(\alpha t)')  
Grid on
```

Matlab có các lệnh vòng lặp như for, while và logic như là if. Cách sử dụng chúng tương tự trong ngôn ngữ Pascal hay C.

Lệnh điều kiện: if và switch

Cú pháp:

If term command [elseif term command...] [else command] end

Switch term case term command[...] [otherwise command] end

Trong đó term là điều kiện, command là lệnh.

Thí dụ:

```
>> test=5;
```

```
>> if test<=2;a=2,elseif test<=5; a=5,else a=10, end;
```

```
a =
```

```
5
```

```
>> switch test case 2;a=2,case {3,4,5};a=5,otherwise a=10,end;
```

```
a =
```

```
5
```

Trong cả hai trường hợp trên, các lệnh con được ngăn cách bởi dấu (;) và dấu (,). Trong các Scripts, thường ta hay viết mỗi lệnh con trong một dòng riêng. Thí dụ:

```
>> if test<=2,
```

Bài 1: Khảo sát hệ thống điều khiển tự động dùng MATLAB

```
a=2;
elseif test<=5,
    a=5;
else a=10;
end;
>> a
```

a =

5

Vòng lặp và logic: for và while

Bằng vòng lặp ta có thể thực hiện lặp lại nhiều lần một số lệnh nhất định. Cú pháp:

For variable=term command end

While term command end

Trong cả hai trường hợp lệnh break đều có tác dụng kết thúc vòng lặp.

Thí dụ:

```
>> for k=0:1, k^2, end;
```

ans =

0

ans =

1

```
>> n=1;
```

```
>> while 1,n=n+1;m=n^2;if m>10, break;end;end
```

m =

4

m =

9

m =

16

```
>>
```

Bài tập:

1. Cho hai ma trận sau:

$$A = \begin{bmatrix} 4 & 2\pi \\ 6j & 10 + \sqrt{2}j \end{bmatrix}$$

$$B = \begin{bmatrix} 6j & -13\pi \\ \pi & 16 \end{bmatrix}$$

Dùng Matlab tính toán sau:

- | | |
|---------------------|-----------|
| a) $A+B$ | b) AB |
| c) A^2 | d) A' |
| e) B^{-1} | f) $B'A'$ |
| g) $A^2 + B^2 - AB$ | |

2. Cho hệ phương trình đại số tuyến tính:

$$5x+6y+10z=4$$

$$-3x+14z=10$$

$$-7y+21z=0$$

Xác định giá trị x,y, và z để tập phương trình đại số thỏa mãn.

3. Viết một Matlab Script (Tập tin .m) để vẽ hàm sau:

$$y(x) = \frac{4}{\pi} \cos \omega x + \frac{4}{9\pi} \cos 3\omega x$$

trong đó ω là ngõ vào biến ở dòng lệnh. Đặt nhãn trục x là time(sec) và trục y là y(x).

1.2.2. Hàm truyền đạt, phương pháp biến trạng thái

1.2.2.1. Hàm truyền đạt

Khai báo hàm truyền:

- Sử dụng lệnh tf(num, den): Nhập đa thức tử số num và đa thức mẫu số den dưới dạng vector tham số của s theo trình tự số mũ của s bé dần.

```
>> h=tf([2 -3], [1 1])
```

Transfer function:

2 s - 3

s + 1

```
>>
```

- Khai báo dưới dạng hàm hữu tỷ của s. Trước hết ta phải khai báo s là biến mô hình TF, sau đó nhập hàm truyền đạt dưới dạng hàm hữu tỷ của s.

```
>> s=tf('s')
```

Transfer function:

s

```
>> h=(s+2)/(s^2+5*s+4)
```

Transfer function:

$$\frac{s + 2}{s^2 + 5s + 4}$$

```
>>
```

Hàm truyền:

Nếu P là một vector hàng chứa các hệ số của đa thức, thì hàm **roots(P)** sẽ cho ta một vector cột các phần tử của nó là nghiệm của đa thức.

Ví dụ: Tìm nghiệm của đa thức sau:

$$S^6 + 9s^5 + 31.25s^4 + 61.25s^3 + 67.75s^2 + 14.75s + 15$$

Nhập:

```
» P=[1 9 31.25 61.25 67.75 14.75 15]
```

P =

```
1.0000 9.0000 31.2500 61.2500 67.7500 14.7500 15.0000
```

```
» R=roots(P)
```

Được:

R =

```
-4.0000
```

```
-3.0000
```

```
-1.0000 + 2.0000i
```

```
-1.0000 - 2.0000i
```

```
0.0000 + 0.5000i
```

```
0.0000 - 0.5000i
```

Bài 1: Khảo sát hệ thống điều khiển tự động dùng MATLAB

Nếu r là vector cột chứa các nghiệm của đa thức thì hàm **poly(r)** cho vector hàng, các phần tử là hệ số của đa thức.

Ví dụ: Nghiệm của đa thức là: -1,-2,-3+4i,-3-4i. Hãy xác định phương trình đa thức.

```
» i=sqrt(-1);
```

```
» R=[-1 -2 -3+4*i -3-4*i]
```

R =

```
-1.0000    -2.0000   -3.0000 + 4.0000i  -3.0000 - 4.0000i
```

```
» P=poly(R)
```

P =

```
1    9   45   87   50
```

Từ đó phương trình đa thức là;

$$S^4 + 9s^3 + 45s^2 + 87s + 50 = 0$$

Nghiệm và zero của hàm truyền:

Ta sử dụng hàm **tf2zp** để tìm zero, nghiệm và độ lợi của hàm truyền.

Ví dụ: tìm nghiệm và zero của hàm truyền sau:

$$H(s) = \frac{(s^3 + 11s^2 + 30s)}{(s^4 + 9s^3 + 45s^2 + 87s + 50)}$$

Nhập :

```
» num=[1 11 30 0]
```

num =

```
1    11   30    0
```

```
» den=[1 9 45 87 50]
```

den =

```
1    9   45   87   50
```

```
» [z,p,k]=tf2zp(num,den)
```

Ta có:

z =

```
0
```

```
-6
```

```
-5
```

p =

```
-3.0000 + 4.0000i
```

```
-3.0000 - 4.0000i
```

```
-2.0000
```

```
-1.0000
```

k =

```
1
```

Hàm **zp2tf** dùng để xác định hàm truyền từ nghiệm và xem hệ số khuếch đại của hệ thống.

```
» z=[-6; -5; 0];k=1
```

```
» i=sqrt(-1);
```

```
» p=[-3+4*i;-3-4*i;-2;-1];
```

```
» [num,den]=zp2tf(z,p,k)
```

num =

0 1 11 30 0

den =

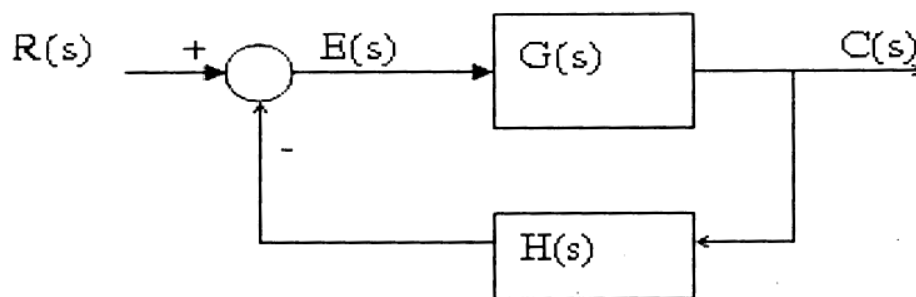
1 9 45 87 50

Lệnh đơn giản hàm truyền: minreal

Tính hàm truyền của hệ thống nối tiếp: lệnh series.

Tính hàm truyền của hệ song song: lệnh parallel.

Cho hệ thống hồi tiếp âm như sau:



Hình 1.1

Hồi tiếp âm: $G_k = G/(1+G.H)$

Hồi tiếp dương: $G_k = G/(1-G.H)$

$G_k = \text{feedback}(G,H)$ tính hàm truyền hệ thống hồi tiếp âm.

$G_k = \text{feedback}(G,H,+1)$ tính hàm truyền hệ thống hồi tiếp dương

Thí dụ:

```
>> G=tf([1 1],[1 3 2])
```

Transfer function:

$s + 1$

$$s^2 + 3s + 2$$

```
>> H=tf(1,[1 5])
```

Transfer function:

$$1$$

$$s + 5$$

```
>> Gk=feedback(G,H) % Hàm truyền kín hệ hồi tiếp âm.
```

Transfer function:

$$s^2 + 6s + 5$$

$$s^3 + 8s^2 + 18s + 11$$

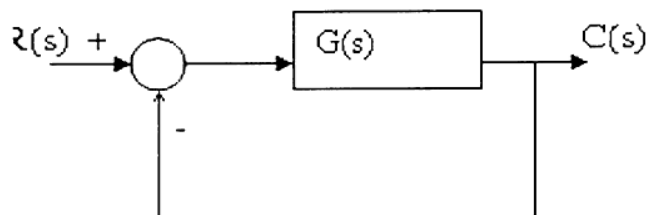
```
>> Gk=feedback(G,H,+1) % hàm truyền kín hệ hồi tiếp dương
```

Transfer function:

$$s^2 + 6s + 5$$

$$s^3 + 8s^2 + 16s + 9$$

```
>> Gk=feedback(G,1) % hàm truyền kín hệ hồi tiếp âm đơn vị.
```



Hình 1.2

Transfer function:

$$\frac{s + 1}{s^2 + 4s + 3}$$

1.2.2.2. Phương pháp biến trạng thái

Phương trình vi phân:

$$\frac{dx}{dt} = Ax(t) + Bu(t)$$

$$y = Cx(t)$$

Là hệ phương trình vi phân bậc nhất của hệ thống và x gọi là biến trạng thái. Ưu điểm của phương pháp biến trạng thái là các phương trình mô tả hệ được giải dễ dàng nhờ máy tính. Ngoài ra phương pháp biến trạng thái có thể mở rộng để khảo sát hệ phi tuyến. Ta lập các phương trình biến trạng thái từ một phương trình vi phân bậc n bằng cách chỉ định các biến trạng thái thích hợp.

Ví dụ: lập phương trình biến trạng thái cho phương trình vi phân dưới đây:

$$2\frac{d^3y}{dt^3} + 4\frac{d^2y}{dt^2} + 6\frac{dy}{dt} + 8y = 10u(t)$$

Ta sử dụng hàm **ode2phv.m** để chuyển phương trình vi phân bậc n thành phương trình biến trạng thái.

Cú pháp: $[A,B,C]=\text{ode2phv}(ai,k)$

» $ai=[2 \ 4 \ 6 \ 8]$

$ai =$

2 4 6 8

» $k=10$

$k =$

10

```
» [A,B,C]=ode2phv(ai,k)
```

Hàm truyền - mô hình biến trạng thái:

Hàm [A,B,C]=tf2ss(num,den) sẽ biến đổi hàm truyền thành mô hình biến trạng thái.

Ví dụ: Cho hàm truyền $\frac{s^2 + 7s + 2}{s^3 + 9s^2 + 26s + 24}$

```
» num=[1 7 2]
```

num =

1 7 2

```
» den=[1 9 26 24]
```

den =

1 9 26 24

```
» [A,B,C,D]=tf2ss(num,den)
```

A =

-9 -26 -24
1 0 0
0 1 0

B =

Bài 1: Khảo sát hệ thống điều khiển tự động dùng MATLAB

1

0

0

C =

1 7 2

D =

0

Mô hình biến trạng thái - Hàm truyền:

Cho hệ $\dot{x}/dt = Ax + Bu$

$$Y = Cx + Du$$

Áp dụng biến đổi Laplace suy ra $G(s) = C(sI - A)^{-1}B + D$

Hàm `[num,den]=ss2tf(A,B,C,D,i)` sẽ biến đổi phương trình biến trạng thái thành hàm truyền.

Ví dụ: Hệ thống được biểu diễn bằng phương trình biến trạng thái sau:

$$\dot{x}/dt = Ax + Bu$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ -1 & -2 & -3 \end{bmatrix}$$

$$B = \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x$$

Tìm hàm truyền $G(s)=Y(s)/X(s)$

```
» A=[0 1 0;0 1 1;-1 -2 -3]
```

A =

```
0 1 0
```

```
0 1 1
```

```
-1 -2 -3
```

```
» B=[10; 0;0]
```

B =

```
10
```

```
0
```

```
0
```

```
» C=[1 0 0]
```

C =

```
1 0 0
```

```
» D=[0]
```

D =

```
0
```

```
» [num,den]=ss2tf(A,B,C,D,1)
```

num =

0 10.0000 20.0000 -10.0000

den =

1.0000 2.0000 -1.0000 1.0000

Do đó hàm truyền là:

$$G(s) = (10s^2 + 20s - 10) / (s^3 + 2s^2 - s + 1)$$

1.2.3. Đáp ứng của hệ thống

Hàm truyền chuẩn của hệ thống bậc 2:

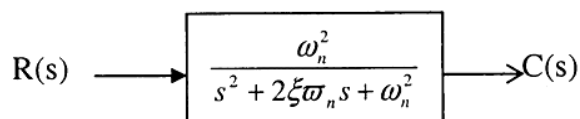
$$G(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

1.2.3.1. Đáp ứng miền thời gian

Ta có thể dùng hàm `c=stepzwn(z,w,R,a,T,t)` để tính đáp ứng của hệ. Ở đây z -thông số suy giảm, ω_n - tần số tự nhiên, R - biên độ hàm bậc thang. Đối với hệ bậc 2 thì $a=0$, $T=0$ còn t - khoảng thời gian cần khảo sát. Đáp ứng của hệ suy giảm nhanh và có vọt lố cũng được khảo sát bằng hàm trên. Các hàm `c=impzwn(num,den)`, `c=step(num,den,t)` và `c=lsim(num,den,u,t)` cũng có thể được sử dụng để khảo sát đáp ứng chuyển tiếp hệ thống.

Ví dụ: Xác định đáp ứng bậc thang của hệ thống với $\xi=0.6$ và $\omega_n=5$.

$$G(s) = (\omega_n^2) / (s^2 + 2\xi\omega_n s + \omega_n^2)$$



$$G(s) = \frac{25}{s^2 + 6s + 25}$$

Cách 1:

Num=25;

Den=[1 6 25];

Gs=tf(num,den)

T=0:0.02:2;

C=step(gs,t);

Plot(t,c)

Xlabel('t-sec'),ylabel('c(t)'),grid,pause

Cách 2:

```
>> den=[1 6 25]
```

den =

1 6 25

```
>> t=0:0.02:2;
```

```
>> c=step(num,den,t);
```

```
>> plot(t,c);
```

```
>> xlabel('t-sec'),ylabel('c(t)');
```

```
>> grid,pause
```

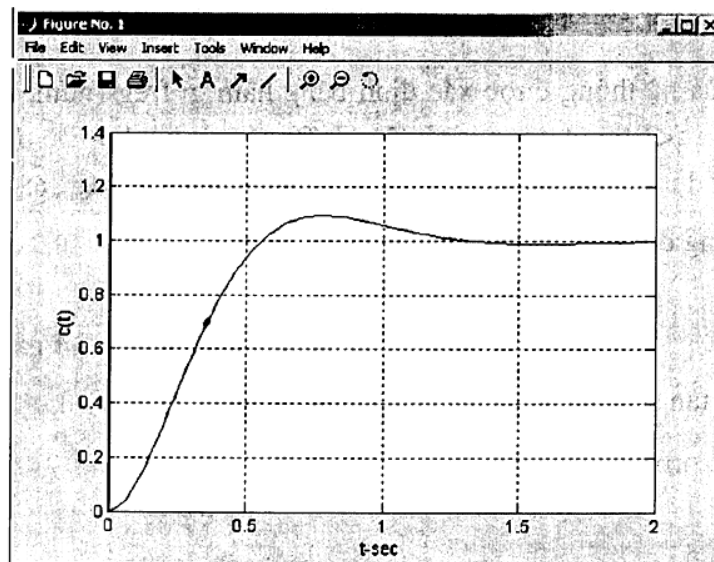
```
>>
```

Kết quả là:

$T_p=0.776667$ Phần trăm vọt lố=9.47783

$T_r=0.373333$

$T_s=1.18667$



Hình 1.3.Đáp ứng thời gian

Lệnh step: Cho hàm truyền đạt của hệ sau: $G(s) = \frac{30}{s^2 + 4s + 30}$. Lệnh Matlab vẽ đáp ứng nấc là:

```
num=30;den=[1 4 30]; G=tf(num,den)
step(G)
```

Lệnh impulse: vẽ đáp ứng xung

Cho hàm truyền đạt của hệ sau: $G(s) = \frac{30}{s^2 + 4s + 30}$. Lệnh Matlab vẽ đáp ứng xung là:

```
num=30;den=[1 4 30]; G=tf(num,den)
impulse(G)
```

1.2.3.2. Đáp ứng tần số của hệ thống

Khi $\xi < 0.707$ thì tần số cộng hưởng (tần số tại đỉnh vọt lố) ω_r được tính bởi:

$$\omega_r = \omega_n \sqrt{1 - 2\xi^2} \quad (1)$$

Biên độ cộng hưởng (Đỉnh vọt lố) được ký hiệu $M_{p\omega}$ và bằng:

$$M_{p\omega} = \frac{1}{2\xi\sqrt{1 - \xi^2}} \quad \text{cho } 0 < \xi < 0.707 \quad (2)$$

Băng thông:

$$BW = \omega_n \sqrt{(1 - 2\xi^2) + \sqrt{4\xi^4 - 4\xi^2 + 2}} \quad (3)$$

Đáp ứng tần số của hệ thống được xác định bằng hàm $g = \text{fregs}(\text{num}, \text{den}, w)$. Đối với hệ thống bậc 2 với $\xi < 1$ thì phương trình (1) và (2) có thể kết hợp để tính ω_r , $M_{p\omega}$. Tuy nhiên, ta có thể sử dụng hàm $\text{frspec}(w, \text{mag})$ để xác định ω_r , $M_{p\omega}$, ω_B .

Ví dụ: Cho hệ thống được mô tả bởi hàm truyền vòng kín

$$G(s) = \frac{4}{s^2 + 2s + 4}$$

Xác định đáp ứng tần số với các thông tin ω_r , $M_{p\omega}$, ω_B .

Cách giải:

```
>> num=4;
```

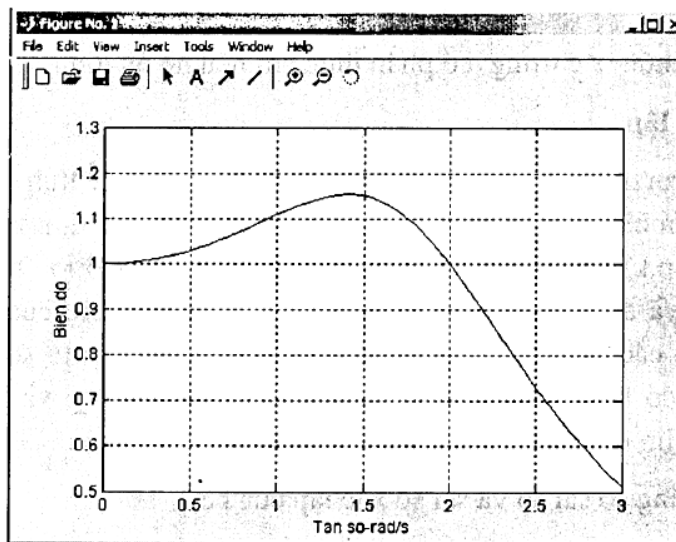
```
>> den=[1 2 4];
```

```
>> w=0:0.1:3;  
>> g=freqs(num,den,w);  
>> mag=abs(g);  
>> frqspec(w,mag)  
>> plot(w,mag);  
>> ylabel('Bien do'),xlabel('Tan so-rad/s')  
>> grid  
>>
```

Kết quả là:

$$\omega_r=1.41, M_{p\omega}=1.15, \omega_B=2.55$$

Kết quả được biểu diễn trên hình:



Hình 1.4. Đáp ứng biên độ tần số

1.2.4. Đặc tính của hệ thống điều khiển

1.2.4.1. Tiêu chuẩn ổn định Routh-Hurwitz

Hàm roots(a) được xây dựng dựa vào tiêu chuẩn Routh-Hurwitz sẽ giúp khảo sát tính ổn định tuyệt đối của hệ.

Ví dụ: Cho hệ với phương trình đặc tính:

$$S^4 + 10s^3 + 35s^2 + 50s + 24 = 0$$

Bài 1: Khảo sát hệ thống điều khiển tự động dùng MATLAB

```
» a=[1 10 35 50 24]
```

```
a =
```

```
1    10    35    50    24
```

```
» roots(a)
```

```
ans =
```

```
-4.0000
```

```
-3.0000
```

```
-2.0000
```

```
-1.0000
```

Vì nghiệm của đa thức đặc trưng có phần thực âm nên hệ ổn định.

1.2.4.2. Sai số xác lập

Hai hàm `errorzp(z,p,k)` và `errorf(num,den)` được viết để tính toán sai số xác lập của hệ thống khi tín hiệu vào là tín hiệu bậc thang đơn vị, dốc đơn vị và parabolic đơn vị. Hàm `errorzp(z,p,k)` tính được sai số xác lập khi hệ thống được biểu diễn bằng các zero, các nghiệm và độ lợi. Ở đây z là vector cột chứa các zero của hàm truyền, p là vector cột bao gồm các nghiệm cực và k là độ lợi. Nếu bậc của tử số nhỏ hơn mẫu số, tức là $n < m$ thì ta có $n-m$ số $\inf(\omega)$ để chiều cao của 2 vector p và z như nhau. Hàm `errorf(num,den)` tìm được sai số xác lập khi hệ thống được mô tả bởi hàm truyền.

Ví dụ: Xác định hằng số sai số và sai số xác lập của hệ:

$$G(s) = \frac{10(s+4)}{s(s+1)(s+2)(s+5)}$$

```
» k=10
```

```
k =
```

```
10
```

```
» z=[-4;inf;inf;inf]
```

z =

-4

Inf

Inf

Inf

» p=[0;-1;-2;-5]

p =

0

-1

-2

-5

» errorzp(z,p,k)

Ví dụ: $G(s)=10/(s^2+14s+50)$

>>num=10

>>den=[1 14 50]

>>errorzf(num,den)

1.2.5.Phân tích ở miền tần số

1.2.5.1. Biểu đồ Bode

Hàm [mag, phase]= bode(num,den,ω) cho ta biết biên độ và góc pha của hàm truyền trong dãy tần số ω đang khảo sát.

Ví dụ 1: Vẽ biểu đồ Bode cho hệ thống điều khiển hồi tiếp đơn vị với hàm truyền hở.

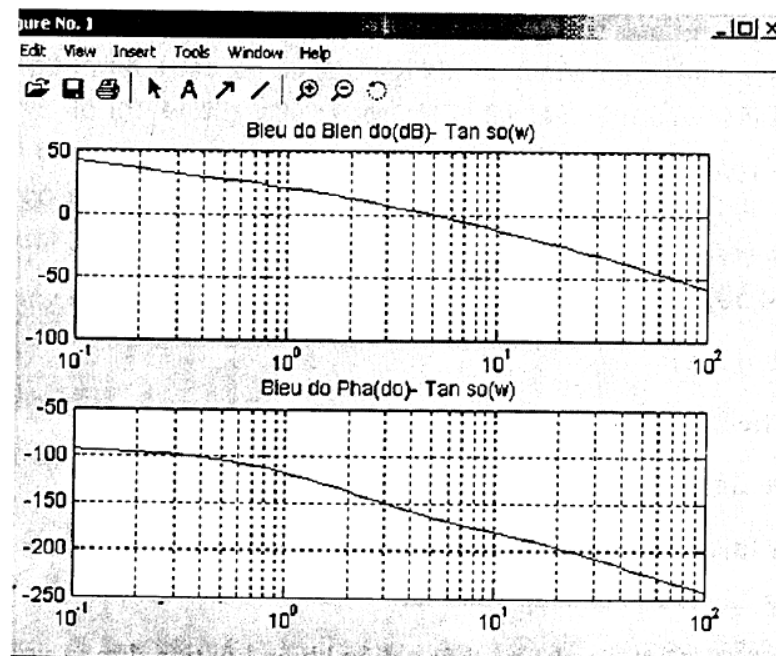
$$GH(s) = \frac{K}{s(s+2)(s+50)} = \frac{K}{s^3 + 52s^2 + 100s}$$

Bài 1: Khảo sát hệ thống điều khiển tự động dùng MATLAB

Cho $K=1300$ chúng ta sử dụng dòng lệnh sau:

```
>> clf
>> k=1300;
>> num=[k];
>> den=[1 52 100 0];
>> w=logspace(-1,2);
>> [mag,phase]=bode(num,den,w);
>> dB=20*log10(mag);
>> subplot(211),semilogx(w,dB)
>> title('Bieu do Bien do(dB)- Tan so(w)'),grid
>> subplot(212),semilogx(w,phase)
>> title('Bieu do Pha(do)- Tan so(w)'),grid
>>
```

Kết quả được thể hiện trên hình sau:



Hình 1.5. Đáp ứng biên độ - tần số và pha - tần số

1.2.5.2. Biểu đồ nghiệm (Biểu đồ Nyquist)

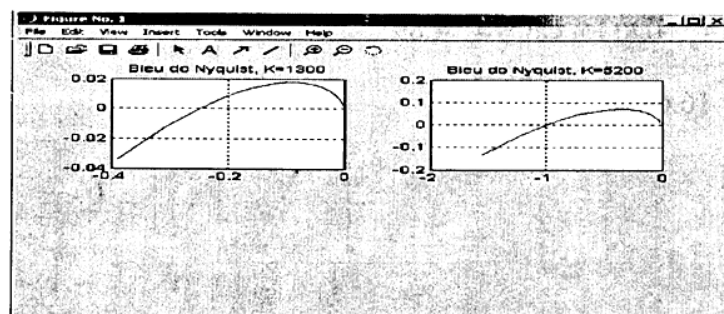
Hàm $[Re, Im]=\text{Nyquist}(\text{num}, \text{den}, \omega)$ của MATLAB cho biết phần thực và phần ảo của hàm truyền khi tần số thay đổi trong khoảng được chỉ định.

Ví dụ 2: Vẽ biểu đồ nghiệm Nyquist cho hệ thống của ví dụ 1 với $K=1300$ và $K=2500$.

Lệnh Matlab:

```
>> clf
>> k1=1300;
>> k2=5200;
>> w=8:1:80;
>> num1=[k1];
>> num2=[k2];
>> den=[1 52 100 0];
>> [Re1, Im1]=Nyquist(num1,den,w);
>> [Re2, Im2]=Nyquist(num2,den,w);
>> subplot(221),plot(Re1,Im1);
>> title('Bieu do Nyquist, K=1300')
>> grid,subplot(222),plot(Re2,Im2)
>> title('Bieu do Nyquist, K=5200')
>> grid
>>
```

Đồ thị được trình bày ở hình vẽ sau:



Hình 1.6. Biểu đồ Nyquist

1.2.5.3. Độ ổn định tương đối

Độ lợi và pha tới hạn

Hàm $[Gm, Pm, \omega_{pc}, \omega_{gc}] = \text{margin}(\text{mag}, \text{phase}, w)$ có thể được sử dụng với hàm Bode để ước lượng độ lợi và pha tới hạn ω_{pc} và ω_{gc} .

Ví dụ: Trong ví dụ 1 cho $K=1300$ hãy tính độ lợi tới hạn ω_{pc} và pha tới hạn ω_{gc} :

Tập tin **doloi_pha.m**

% Tính độ lợi tới hạn, ω_{pc} và pha tới hạn, ω_{gc}

```
k=1300;
```

```
num=[k];
```

```
den=[1 52 100 0];
```

```
w=.1:1:20;
```

```
[mag,phase]=bode(num,den,w);
```

```
[Gm,Pm,wpc,wgc]=margin(mag,phase,w);
```

```
fprintf('Do loi toi han=%7.3g',Gm),fprintf('Do loi cat w=%7.3g\n',wgc)
```

```
fprintf('Pha toi han=%7.3g',Pm),fprintf('Pha cat w=%7.3g\n\n',wpc)
```

Kết quả thực hiện:

```
>> doloi_pha
```

```
Do loi toi han= 4Do loi cat w= 4.89
```

```
Pha toi han= 16.6Pha cat w= 10
```

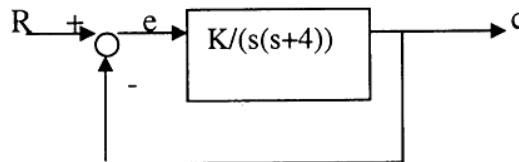
```
>>
```

1.2.6. Phân tích quỹ đạo nghiệm số

Xem tài liệu [1] về 11 quy tắc vẽ quỹ đạo nghiệm số.

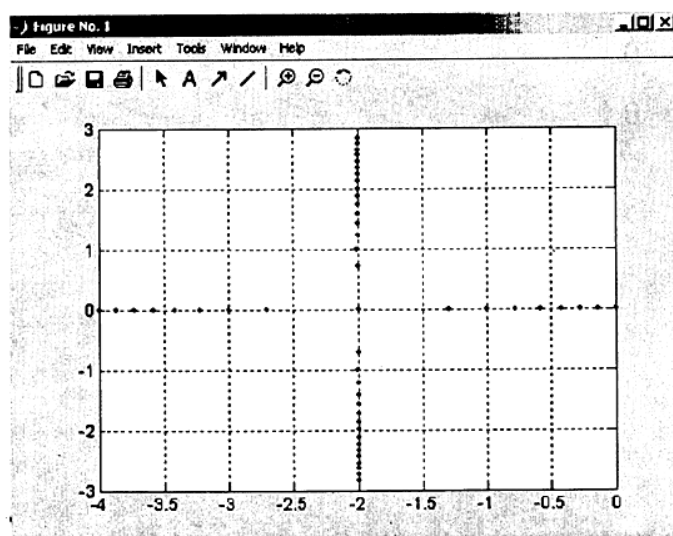
Hàm rlocus: vẽ quỹ đạo nghiệm

Thí dụ: Tìm quỹ đạo nghiệm số ($K > 0$) cho hệ thống đơn giản



Hàm truyền vòng hở: $G(s)H(s) = K/(s(s+4))$

```
» axis([-6,0,-3,3])
» axis('square')
» K=0:0.5:12;
» num=1
» den=[1 4 0]
» r=rlocus(num,den,K);
» plot(r, '.')
» grid
```



Hình 1.7. Quỹ đạo nghiệm số

1.3. Thí nghiệm

1.3.1. Hàm truyền đạt, phương pháp biến trạng thái

Mục đích: biểu diễn hệ bằng hàm truyền và hệ phương trình trạng thái.

Thí nghiệm:

1. Cho hàm truyền $\frac{10}{s^3 + 8.5s^2 + 20.5s + 15}$. Biểu diễn hệ dùng hàm truyền và hệ phương trình biến trạng thái.

Lệnh Matlab:

```
» num=[10]
```

```
» den=[1 8.5 20.5 15]
```

```
» G=tf(num,den)
```

```
» [A,B,C,D]=tf2ss(num,den)
```

A =

```
-8.5000 -20.5000 -15.0000
 1.0000      0      0
      0  1.0000      0
```

B =

```
1
0
0
```

C =

$$\begin{bmatrix} 0 & 0 & 10 \end{bmatrix}$$

$$D =$$

$$0$$

2. Hệ thống được biểu diễn bằng phương trình biến trạng thái sau:

$$dx/dt = Ax + Bu$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 64.4 & 0 & -16 \\ 0 & 0 & -100 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 100 \end{bmatrix}$$

$$y = [1 \ 0 \ 0]x$$

Tìm hàm truyền $G(s) = Y(s)/X(s)$

$$\gg A = [0 \ 1 \ 0; 64.4 \ 0 \ -16; 0 \ 0 \ -100]$$

$$\gg B = [0; 0; 100]$$

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

$$\gg D = [0]$$

$$\gg [\text{num}, \text{den}] = \text{ss2tf}(A, B, C, D, 1)$$

$$\text{num} =$$

Bài 1: Khảo sát hệ thống điều khiển tự động dùng MATLAB

1.0e+003 *

0 0.0000 0.0000 -1.6000

den =

1.0e+003 *

0.0010 0.1000 -0.0644 -6.4400

Do đó hàm truyền là:

$$G(s) = (-1600)/(s^3 + 100s^2 - 64.4s - 6440)$$

3. Lệnh ss:

Lệnh Matlab tạo ra hệ thống mô tả bởi phương trình trạng thái có các ma trận a, b, c, d: a=[0 1;-3 -2]; b=[0;1]; c=[1 0]; d=0

>>PTTT=ss(a,b,c,d)

Biến đổi mô tả toán học từ dạng phương trình trạng thái về dạng hàm truyền: a=[0 1;-3 -2]; b=[0;1]; c=[1 0]; d=0;. Lệnh Matlab là:

>>PTTT=ss(a,b,c,d)

>>g=tf(PTTT)

4. Dùng tiêu chuẩn Routh-Hurwitz xét ổn định các hệ thống cho 3 hàm truyền được khảo sát trong thí nghiệm.

Hệ thống 1: $G(s) = \frac{20}{s(1+0,1s)(1+0,5s)}$

Hệ thống 2: $G(s) = \frac{K(s+5)}{s(s+2)(s+1)}$ khi K=1.

Hệ thống 3: $G(s) = \frac{K(s+1)}{s(s+3)(s^2+8s+20)}$ khi K=1.

1.3.2. Khảo sát hệ thống tự động dùng giản đồ Bode và Nyquist

1.3.2.1. Khảo sát hệ thống tự động dùng giản đồ Bode

Mục đích: Vẽ biểu đồ biên độ và pha theo tần số hàm truyền hệ hở của hệ thống phản hồi âm đơn vị. Trên đồ thị đo độ dự trữ pha, biên độ suy ra sự ổn định và chất lượng hệ thống kín.

Thí nghiệm:

1. Khảo sát hệ thống có hàm truyền hở:

$$G(s) = \frac{20}{s(1+0,1s)(1+0,5s)}$$

$$\text{Ta có: } G(s) = \frac{20}{(0,05s^3 + 0,6s^2 + s)}$$

Hàm Bode (num,den) vẽ bode biên và bode pha.

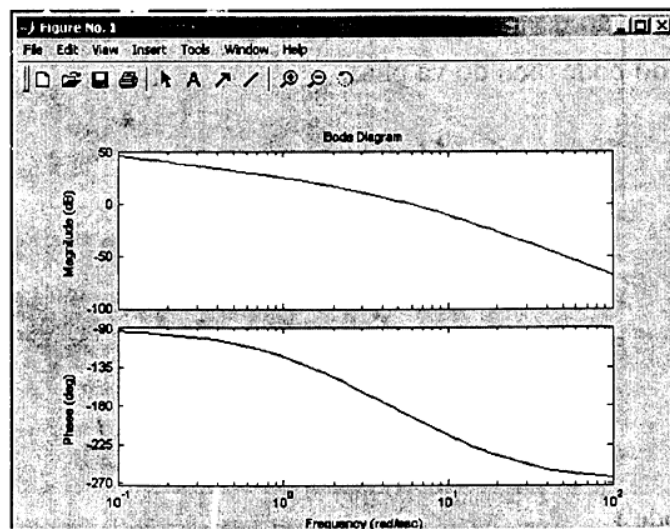
Lệnh Matlab:

```
>> num=[20]
```

```
>> den=[0.05 0.6 1 0]
```

```
>> bode(num,den)
```

```
>>
```



Hình 1.8. Biểu đồ Bode

2. Cho hệ có hàm truyền là:

$$G(s) = (10s^2 + 20s - 10) / (s^3 + 2s^2 - s + 1)$$

Lệnh Matlab:

```
>> num=[10 20 -10]
```

```
>> den=[1 2 -1 1]
```

```
» bode(num,den)
```

3. Cho hàm truyền $G(s) = \frac{s+4}{s^2+3s+2}$, lệnh MATLAB vẽ đồ thị biên độ - tần số và pha - tần số là:

```
num=[1 4]
```

```
den=[1 3 2]
```

```
bode(num,den)
```

4. Xét hệ sau:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -25 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 25 \end{bmatrix} u$$
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Hệ có một ngõ vào u và một ngõ ra y . Bằng cách dùng lệnh bode (A, B, C, D) của Matlab vẽ biểu đồ bode biên độ và pha.

Lệnh Matlab:

```
>> A=[0 1;-25 -4];
```

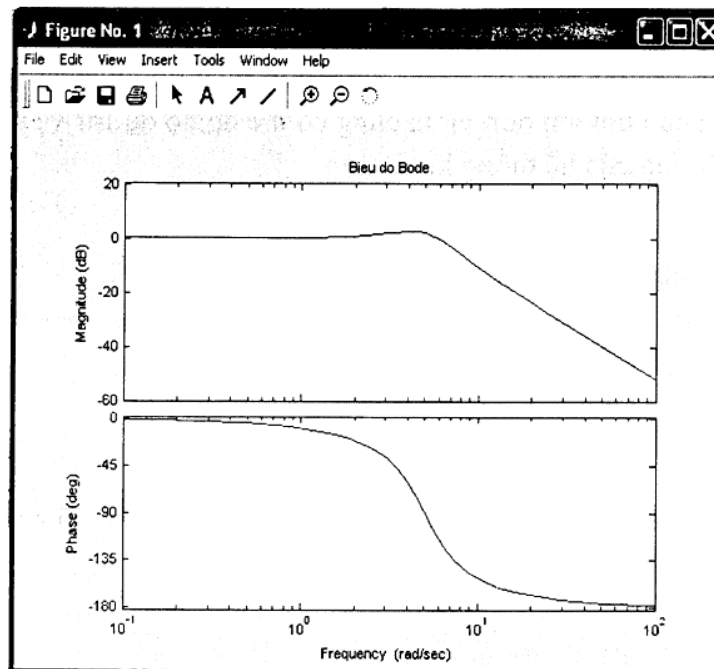
```
>> B=[0;25];
```

```
>> C=[1 0];
```

```
>> D=[0];
```

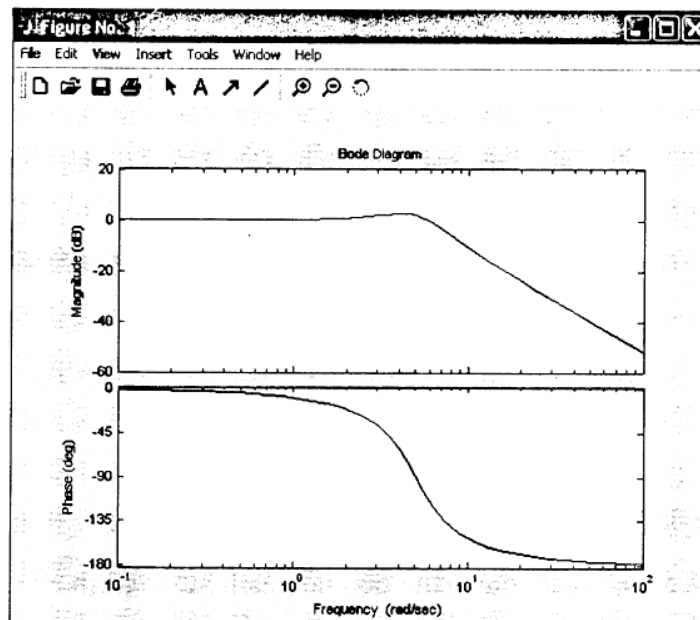
```
>> bode(A,B,C,D)
```

```
>> title('Bieu do Bode')
```



Hình 1.9. Biểu đồ Bode

Nếu ta thay lệnh bode (A, B, C, D) bằng lệnh bode (A, B, C, D, 1) thì MATLAB tạo ra giản đồ bode giống như ví dụ trên.



Hình 1.10

1.3.2.2. Khảo sát hệ thống tự động dùng giản đồ Nyquist

Mục đích: tương tự như biểu đồ Bode, từ bản in biểu đồ Nyquist của hàm truyền hệ hở của hệ thống phản hồi âm đơn vị, ta cũng có thể đo độ dự trữ về pha, biên độ suy ra sự ổn định chất lượng của hệ thống kín.

Thí nghiệm:

1. Xét ổn định hệ thống các hàm truyền hở:

$$G(s) = \frac{K(s+5)}{s(s+2)(s+1)} \quad \text{với } K=1$$

Lệnh Nyquist: vẽ biểu đồ Nyquist

2. Cho hàm truyền đạt của hệ sau: $G(s) = \frac{30}{s^2 + 4s + 30}$. Vẽ biểu đồ Nyquist.

Lệnh Matlab:

```
num=30;den=[1 4 30]; G=tf(num,den)
nyquist(G)
```

1.3.3. Khảo sát hệ thống dùng phương pháp quỹ đạo nghiệm số:

Mục đích:

- Khảo sát đặc tính của hệ thống tuyến tính có hệ số khuếch đại K thay đổi, tìm giá trị K giới hạn K_{gh} để hệ thống ổn định. K_{gh} là hệ số khuếch đại để hệ thống có nghiệm số nằm trên trục ảo.
- Cũng trong phần này, ta tính chọn K để hệ thống có cặp nghiệm số không chế (các cực của hàm truyền hệ kín ảnh hưởng chủ yếu đến đặc tính quá độ) có thông số cho trước.

Với mỗi hệ thống, ta đều vẽ đặc tính quá độ hệ thống kín với đầu vào hàm nấc để kiểm tra.

Thí nghiệm:

1. Vẽ quỹ đạo nghiệm số (QĐNS) của hệ có hàm truyền hở:

$$G(s) = \frac{K(s+1)}{s(s+3)(s^2+8s+20)}, \quad 0 \leq K \leq \infty$$

Tìm K_{gh} và giá trị K để quá trình quá độ của hệ thống kín có tính dao động với tần số dao động riêng $\omega_n = 4$.

Hàm `rlocus(num,den,K)` dùng để vẽ quỹ đạo nghiệm số.

2. Xem xét hệ thống mà hàm truyền vòng hở $G(s)H(s)$ là:

$$G(s)H(s) = \frac{K}{s(s+0.5)(s^2+0.6s+10)}$$

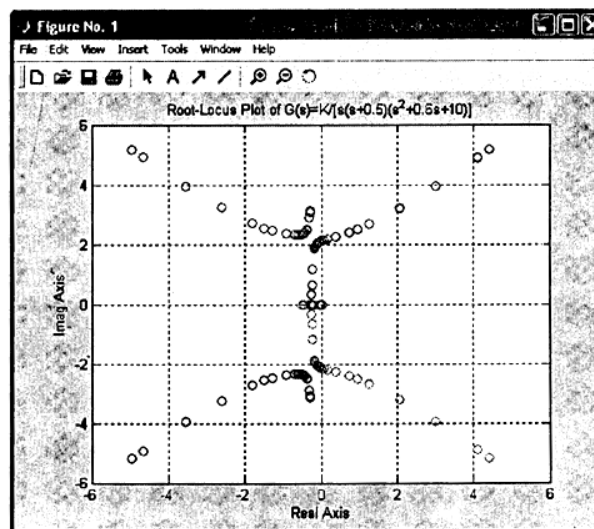
$$= \frac{K}{s^4 + 1.1s^3 + 10.3s^2 + 5s}$$

Viết lệnh Matlab vẽ quỹ đạo nghiệm.

Không có zero vòng hở, cực vòng hở là $s=-0.3+j3.1480$, $s=-0.3-j3.1480$, $s=-0.5$ và $s=0$.

Lệnh Matlab:

```
>> num=[0 0 0 1];
>> den=[1 1.1 10.3 5 0];
>> K=0:0.1:100;
>> r=rlocus(num,den,K);
>> plot(r,'o')
>> v=[-6 6 -6 6];axis(v)
>> grid
>> title('Root-Locus Plot of G(s)=K/[s(s+0.5)(s^2+0.6s+10)]')
>> xlabel('Real Axis')
>> ylabel('Imag Axis')
>>
```



Hình 1.11. Quỹ đạo nghiệm số bài 2.

Bài 1: Khảo sát hệ thống điều khiển tự động dùng MATLAB

3. Xem xét hệ sau:

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

$$u = y - r$$

Trong bài này ta vẽ quỹ đạo nghiệm trong không gian trạng thái. Trong ví dụ trên cho các ma trận A, B, C, và D như sau:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -160 & -56 & -14 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \\ -14 \end{bmatrix}$$
$$C = [1 \ 0 \ 0], D = [0]$$

Dùng Matlab vẽ quỹ đạo nghiệm.

Lệnh Matlab:

Lệnh rlocus (A, B, C, D) vẽ quỹ đạo nghiệm. Lệnh này tương đương với lệnh rlocus(num, den) trong đó num, den đặt được từ lệnh sau:

$$[\text{num}, \text{den}] = \text{ss2tf}(A, B, C, D)$$

như sau:

$$\text{num} = [0 \ 0 \ 1 \ 0]$$

$$\text{den} = [1 \ 14 \ 56 \ 160]$$

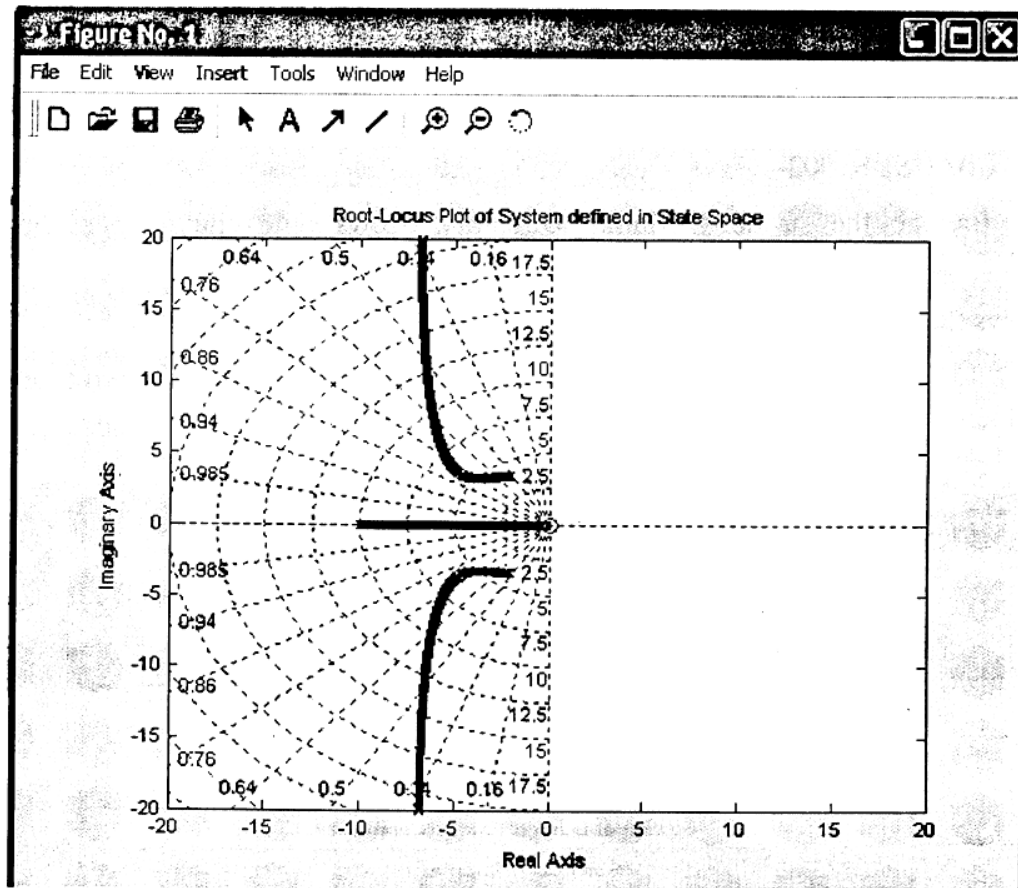
Lệnh Matlab:

$$>> A = [0 \ 1 \ 0; 0 \ 0 \ 1; -160 \ -56 \ -14];$$

$$>> B = [0; 1; -14];$$

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

```
>> D=[0];  
>> K=0:0.1:400;  
>> rlocus(A,B,C,D,K);  
>> v=[-20 20 -20 20];axis(v)  
>> grid  
>> title('Root-Locus Plot of System defined in State Space')  
>>
```



Hình 1.12. Quỹ đạo nghiệm số bài 3.

1.4. Kiểm tra đánh giá

1.5. Phần mềm ACSYS2002

Phần này mô tả và giới thiệu công cụ (toolbox) ACSYS (Automatic Control Systems).

1.5.1. Cài đặt ACSYS

Đầu tiên người sử dụng phải làm theo bước 1 và bước 2 trước khi sử dụng ACSYS:

1. Tạo thư mục C:\ACSYS2002.

2. Sao chép tất cả các tập MATLAB và ảnh từ thư mục ACSYS2002 trong CDROM đi kèm sách vào thư mục C:\ACSYS2002.

Để chạy ACSYS2002, đơn giản khởi động MATLAB R12 hay cao hơn (Matlab version 6.0 hay cao hơn 6.1, 6.5, 7.0).

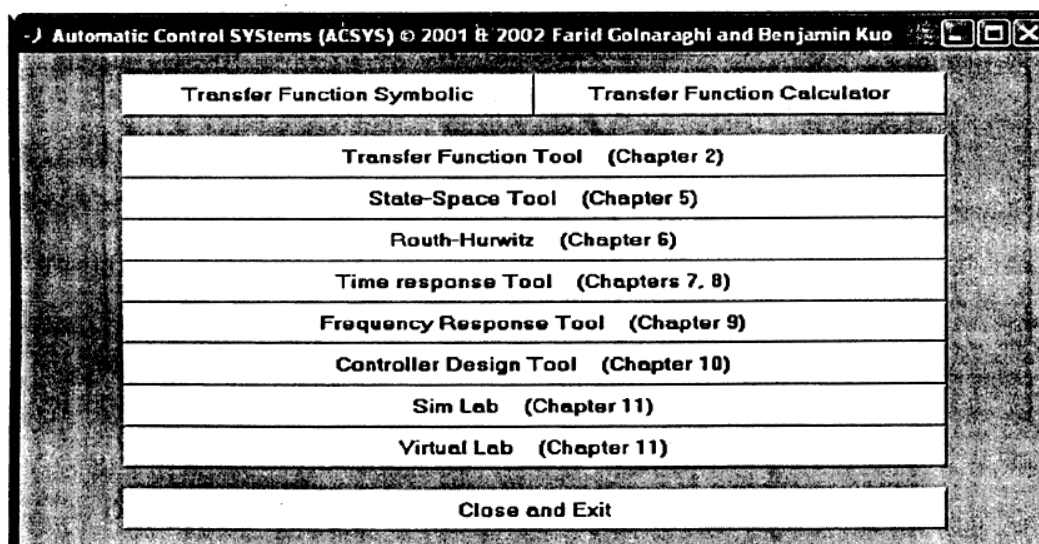
Từ dấu nhắc MATLAB, gõ lệnh:

```
>>cd C:\ACSYS2002
```

Bạn có thể sử dụng Matlab Directory Browser để chuyển đến thư mục ACSYS2002. Từ dấu nhắc “>>” trong cửa sổ lệnh Matlab, ta gõ:

```
>>acsys (hay ACSYS).
```

Hình 1.13 xuất hiện. Dựa trên việc nhấn chuột vào bất cứ nút nào, công cụ tương ứng xuất hiện.



Hình 1.13 : Cửa sổ chính toolbox Automatic Control Systems(ACSYS)

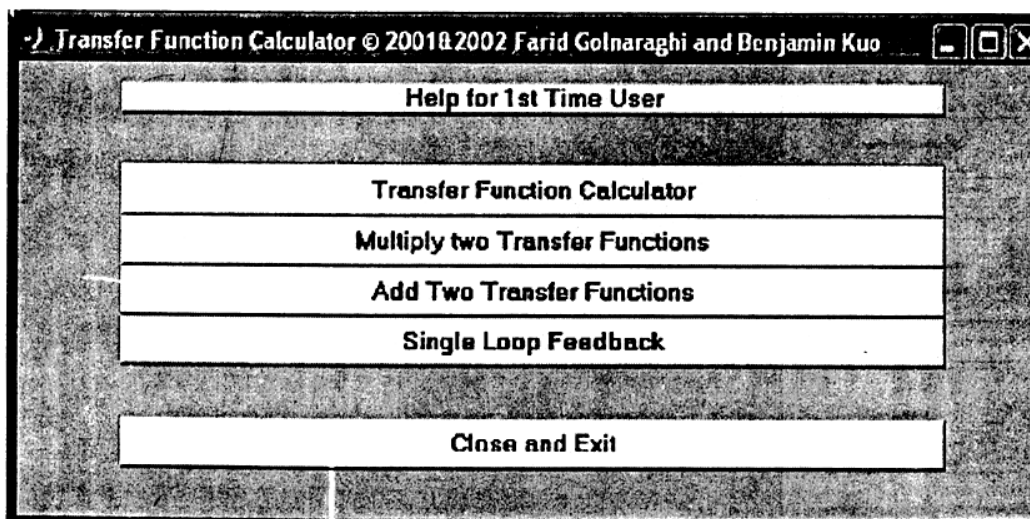
1.5.2. Mô tả phần mềm

Công cụ ACSYS gồm có một số m-files và GUI (graphical user interfaces) để phân tích 10 hàm truyền kỹ thuật điều khiển đơn giản.

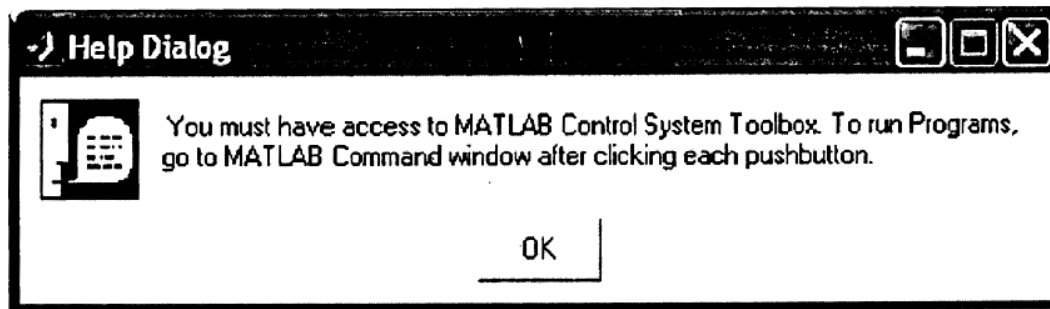
Cụ thể ACSYS gồm có 10 công cụ khác nhau và một công cụ cho chuyển đổi đại lượng kỹ thuật giữa hệ SI và đơn vị khác.

Những thành phần này bao gồm các chủ đề khác nhau trong nghiên cứu hệ thống điều khiển. Chúng bao gồm:

- . Tính toán hàm truyền: tfcal.
- . Mô hình ký hiệu hàm truyền: tfsym.
- . Công cụ hàm truyền: tftool.
- . Công cụ không gian trạng thái: statetool.
- . Công cụ Routh-Hurwitz: stabtool.
- . Công cụ phân tích đáp ứng thời gian: timetool.
- . Công cụ phân tích đáp ứng tần số: freqtool.
- . Công cụ thiết kế bộ điều khiển: controls.
- . Thí nghiệm mô phỏng SIM: simlab.
- . Thí nghiệm ảo Virtual: virtuallab.



Hình 1.14: Cửa sổ tính toán hàm truyền.



Hình 1.15: Hộp thoại trợ giúp MATLAB cho công cụ tfcal.

Các thành phần ACSYS hay toolbox có thể gọi trực tiếp bằng cách gõ tên ở dấu nhắc lệnh MATLAB.

1.5.3. Sử dụng công cụ phân tích không gian trạng thái:

Mục đích: Sử dụng công cụ phân tích không gian trạng thái để khảo sát hệ phương trình trạng thái, biểu diễn hệ phương trình trạng thái, tính trị riêng và vector riêng, xác định hàm truyền đạt, cực và zero và xét tính điều khiển được và quan sát được.

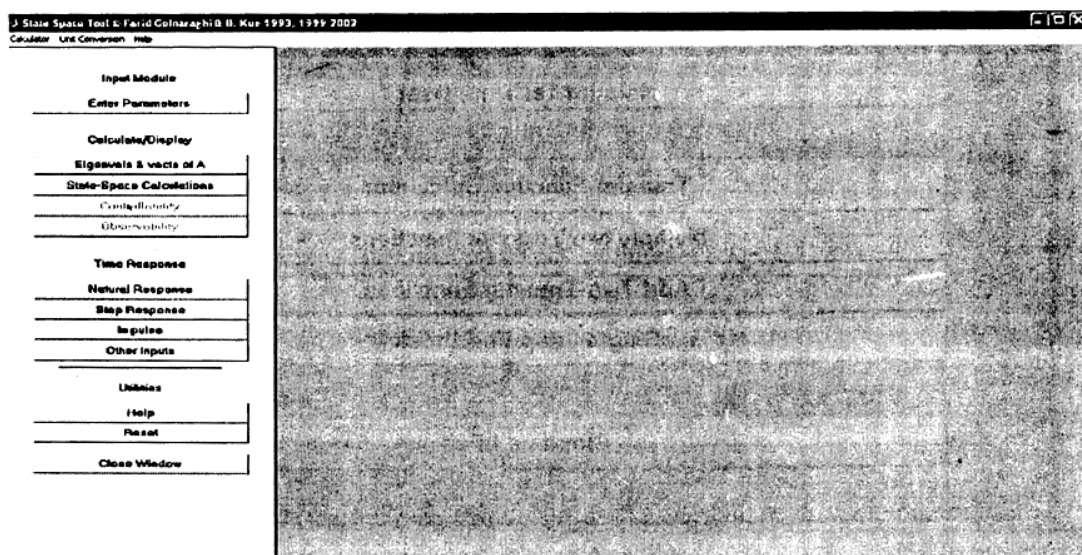
Thí nghiệm:

Bước 1: Khởi động ACSYS

```
>> acsys
```

Bước 2: Chọn State-Space Tool từ màn hình menu ACSYS

Cửa sổ State-Space Tool xuất hiện:



Hình 1.16

Thí nghiệm: Xét hệ sau:

$$\frac{dx}{dt} = Ax + Bu$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 64.4 & 0 & -16 \\ 0 & 0 & -100 \end{bmatrix}$$

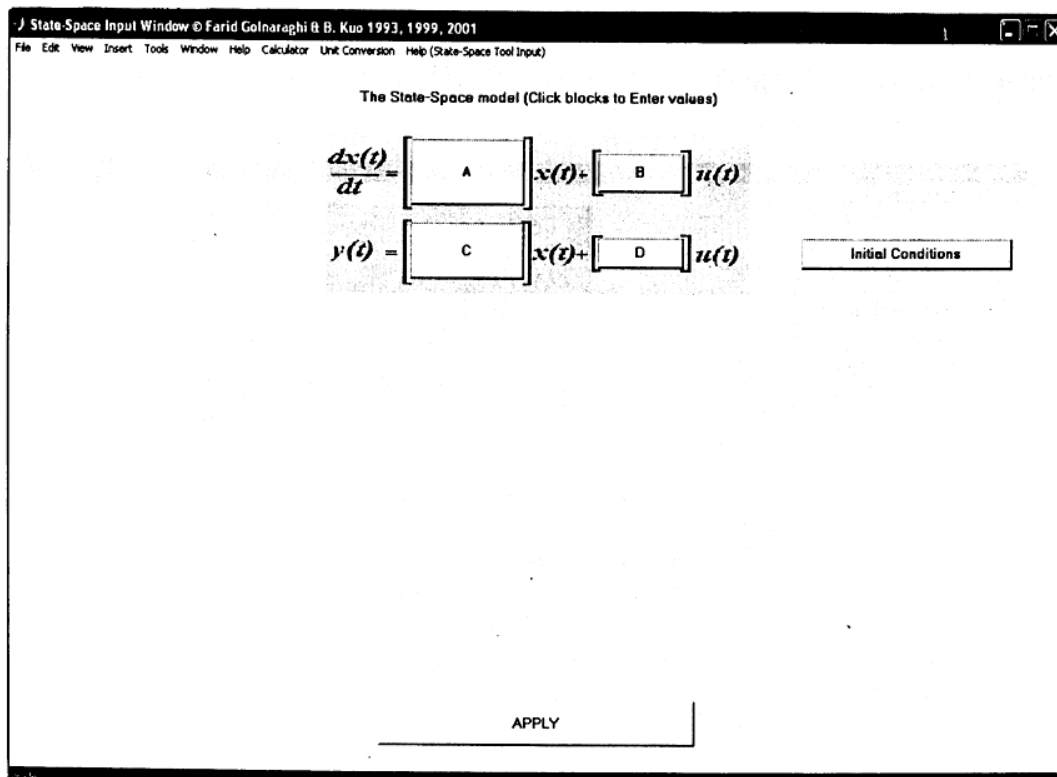
$$B = \begin{bmatrix} 0 \\ 0 \\ 100 \end{bmatrix}$$

$$y = [1 \ 0 \ 0]x$$

Bước 3: Nhấp chuột vào Enter Parameter để nhập thông số ma trận A và B như sau:

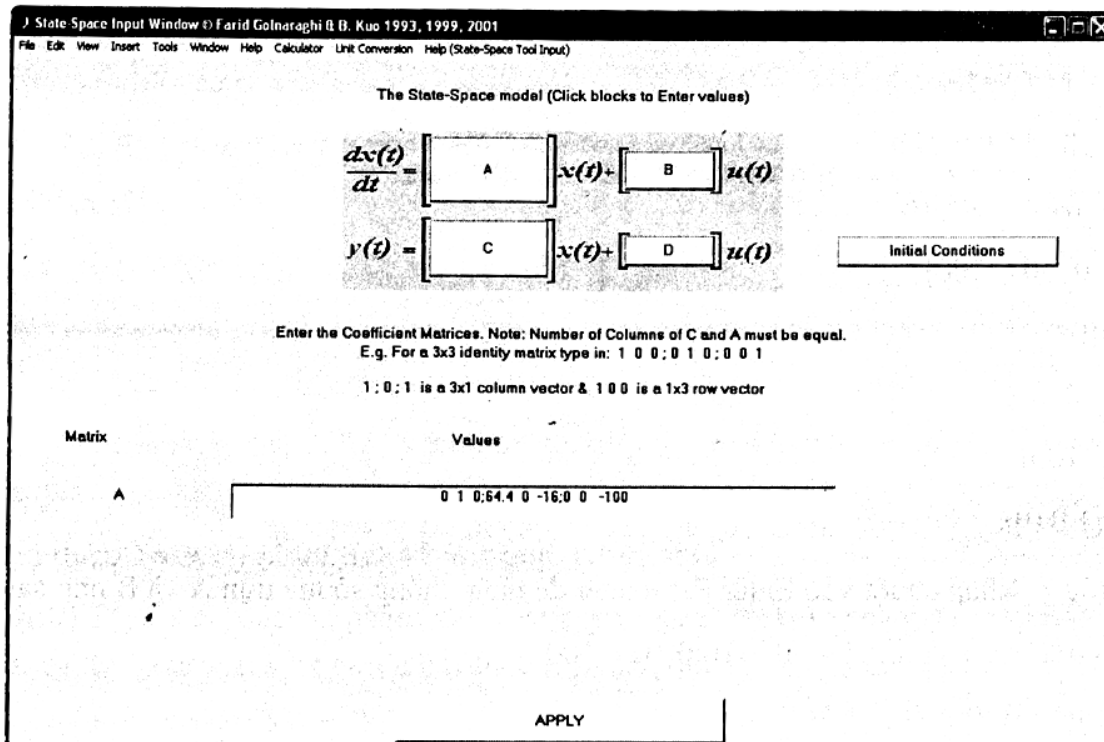
Ma trận A: 0 1 0; 64.4 0 -16; 0 0 -100

Ma trận B: 0 ; 0 ; 100



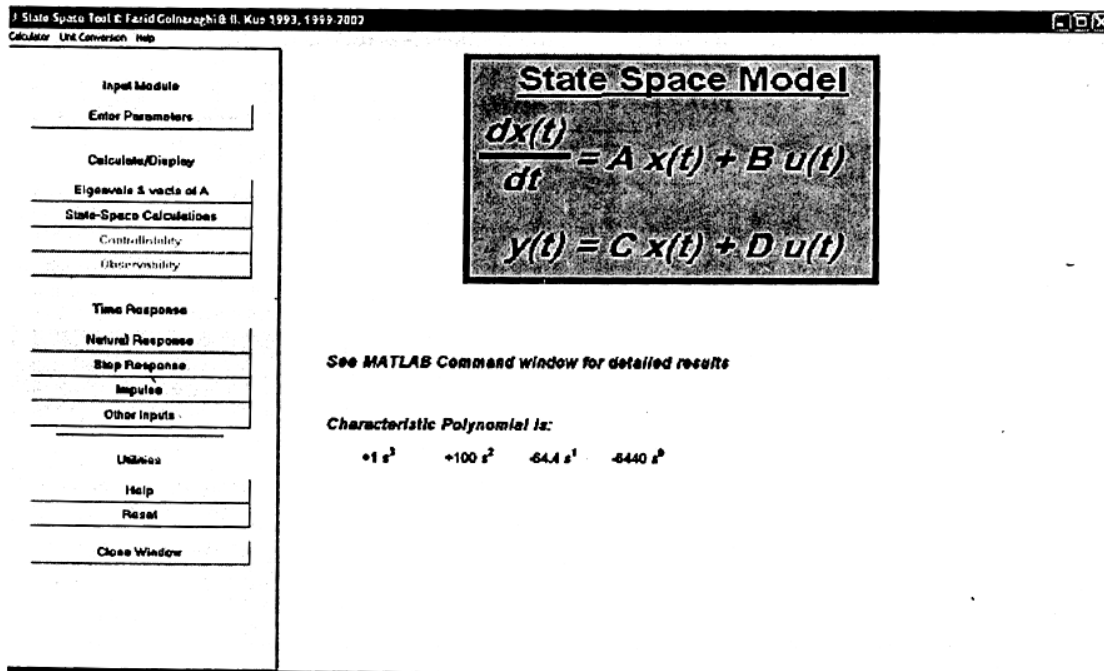
Hình 1.17

Bài 1: Khảo sát hệ thống điều khiển tự động dùng MATLAB



Hình 1.18 : Nhập ma trận A trong cửa sổ Input State-Space

Bước 4: Nhấp vào “Eigenval & vects of A” (Trị riêng và vector riêng của A”):



Hình 1.19

Cửa sổ lệnh Matlab sau khi ta nhập vào “Eigenval & vects of A”:

The A matrix is:

Amat =

```
0    1.0000    0
64.4000    0 -16.0000
0    0 -100.0000
```

Characteristic Polynomial:

ans =

```
s^3+100*s^2-2265873562520787/35184372088832*s-6440
```

Eigenvalues of A = diagonal canonical form of A is:

Aeigs =

```
8.0250    0    0
0 -8.0250    0
0    0 -100.0000
```

Eigenvectors are

T =

```
0.1237 -0.1237 -0.0016
0.9923  0.9923  0.1590
0    0  0.9873
```

Bước 5: Nhấp chuột vào State-Space Calculation (tính toán không gian trạng thái).
Cửa sổ lệnh Matlab:

State-space model is:

a =

	x1	x2	x3
x1	0	1	0
x2	64.4	0	-16
x3	0	0	-100

b =

	u1
x1	0
x2	0
x3	100

c =

	x1	x2	x3
y1	1	0	0

d =

	u1
y1	0

Continuous-time model.

Characteristic polynomial:

ans =

$s^3 + 100s^2 - 2265873562520787/35184372088832s - 6440$

Equivalent transfer function model is:

Bài 1: Khảo sát hệ thống điều khiển tự động dùng MATLAB

Transfer function:

$$4.263\text{e-}014 s^2 + 8.527\text{e-}014 s - 1600$$

$$s^3 + 100 s^2 - 64.4 s - 6440$$

pole, zero form:

Zero/pole/gain:

$$4.2633\text{e-}014 (s+1.937\text{e}008) (s-1.937\text{e}008)$$

$$(s+100) (s+8.025) (s-8.025)$$

Bước 6: Nhấp chuột vào “Controllability”(tính điều khiển được). Cửa sổ lệnh Matlab:

The controllability matrix [B AB A²B ...] is =

Smat =

$$\begin{bmatrix} 0 & 0 & -1600 \\ 0 & -1600 & 160000 \\ 100 & -10000 & 1000000 \end{bmatrix}$$

The system is therefore controllable, rank of S matrix is =

rankS =

$$3$$

Mmat =

$$\begin{bmatrix} -64.4000 & 100.0000 & 1.0000 \end{bmatrix}$$

Bài 1: Khảo sát hệ thống điều khiển tự động dùng MATLAB

```
100.0000  1.0000    0
1.0000    0    0
```

The controllability canonical form (CCF) Transformation matrix is:

Ptran =

```
-1600    0    0
    0 -1600    0
-6440    0   100
```

The transformed matrices using CCF are

Abar =

1.0e+003 *

```
    0  0.0010    0
    0    0  0.0010
6.4400  0.0644 -0.1000
```

Bbar =

```
0
0
1
```

Cbar =

```
-1600    0    0
```

Dbar =

0

Bước 7: Nhấp chuột vào “Observability” (tính quan sát được). Cửa sổ lệnh Matlab:

The observability matrix (transpose:[C CA CA^2 ...]) is =

Vmat =

```
1.0000    0    0
    0 1.0000    0
64.4000    0 -16.0000
```

The System is therefore observable, rank of V matrix is =

rankV =

3

Mmat =

```
-64.4000 100.0000  1.0000
100.0000  1.0000    0
 1.0000    0    0
```

The observability canonical form (OCF) Transformation matrix is:

Qtran =

```
0    0  1.0000
0  1.0000 -100.0000
```

-0.0625 6.2500 -625.0000

The transformed matrices using OCF are

Abar =

1.0e+003 *

0.0000 -0.0000 6.4400

0.0010 -0.0000 0.0644

0 0.0010 -0.1000

Bbar =

-1600

0

0

Cbar =

0 0 1

Dbar =

0

Tham khảo: [1]. B.C. Kuo, M.F. Golnaraghi, Automatic Control Systems, Wiley, 2003

[2].Appendix K: ACSYS2002: Description of the Software, của sách Automatic Control Systems, tác giả: B. C. Kuo và M.F. Golnaraghi, 8th ed. ,NXB: Wiley, 2003.

BÀI 2

THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG DÙNG MATLAB

2.1. Mục tiêu

Trong bài thí nghiệm này sinh viên sẽ tìm hiểu cách thiết kế một bộ hiệu chỉnh sớm trễ pha theo phương pháp dùng QĐNS theo yêu cầu chất lượng cho trước. Matlab hỗ trợ một công cụ rất mạnh dùng để thiết kế hệ thống điều khiển tuyến tính một đầu vào một đầu ra đó là công cụ **sisotool**. Dựa vào công cụ này, sinh viên sẽ thiết kế được bộ hiệu chỉnh sớm trễ pha và nhiều bộ hiệu chỉnh khác một cách dễ dàng nhờ giao diện dễ hiểu, dễ sử dụng và hỗ trợ nhiều chức năng mạnh có trong **sisotool**.

2.2. Nội dung

2.2.1. Công cụ Sisotool

Để thực hiện các yêu cầu trong bài thí nghiệm này, sinh viên cần chuẩn bị kỹ và hiểu rõ trình tự các thao tác tiến hành thiết kế một bộ điều khiển trong Sisotool.

Sinh viên cần tham khảo phần phụ lục của chương 6 trong sách Lý thuyết điều khiển tự động [1] để làm quen và hiểu rõ trình tự các bước thiết kế một bộ điều khiển trong Sisotool. Chú ý, sinh viên phải đọc kỹ phần này để nắm rõ trình tự thiết kế. Nếu sinh viên không hiểu kỹ thì sẽ không thực hiện được các yêu cầu trong bài thí nghiệm này.

Để kích hoạt công cụ Sisotool, từ cửa sổ Command Window gõ lệnh `sisotool`. Tiến hành thao tác từ bước 1 đến bước 3, cửa sổ SISO Design Tool xuất hiện như sau:

Trình tự thiết kế như sau:

Bước 1: Khai báo đối tượng điều khiển

```
G=tf(20,conv([1 1 0],[1 2]))
```

Transfer function:

20

Bài 2: Thiết kế hệ thống điều khiển tự động dùng Matlab

$$s^3 + 3s^2 + 2s$$

$$H=tf(1,1)$$

Bước 2: Kích hoạt SiSotool

Cửa sổ SISO Design Tool xuất hiện

Bước 3: Nhập đối tượng điều khiển vào sisotool.

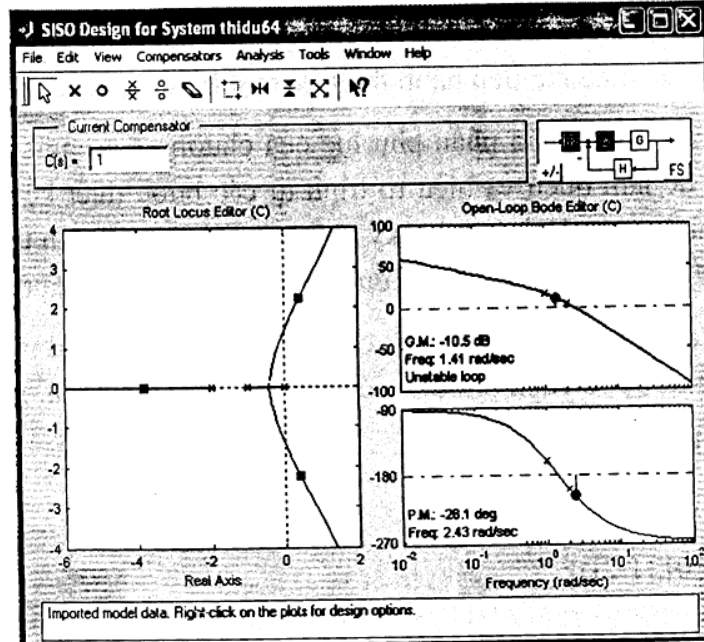
Trong cửa sổ SISO Design Tool chọn [File]→[Import...]. Cửa sổ Import Data xuất hiện.

Đặt tên hệ thống tùy ý, như là thidu64. Ban đầu tất cả các khối trong hệ thống điều khiển đều có hàm truyền bằng 1, ta thay đổi đối tượng điều khiển (plant) là G, cảm biến (sensor) là H, bộ lọc F(prefilter) bằng 1, khâu hiệu chỉnh (compensator) C chưa thiết kế nên cũng bằng 1. Sau khi thực hiện xong nhấp chuột vào nút Ok.

Bước 4: Khảo sát hệ thống trước khi hiệu chỉnh.

Bước 5: Thiết kế khâu hiệu chỉnh.

Bước 6: Kiểm tra lại đáp ứng của hệ thống.



Hình 2.1

Cửa sổ gồm các vùng:

- Vùng hiển thị sơ đồ cấu trúc của hệ thống đang thiết kế. Có thể thay đổi cấu trúc bằng cách kích chuột vào nút [+/-] và [FS] ở góc trái bên dưới. Trong bài thí nghiệm này ta sử dụng cấu trúc như hiển thị.

G: đối tượng điều khiển (plant).

H: cảm biến hồi tiếp (sensor).

F: bộ lọc (prefilter).

C: bộ hiệu chỉnh cần thiết kế.

- Hàm truyền của bộ hiệu chỉnh $C(s)$.
- Cửa sổ hiển thị kết quả trong quá trình thao tác.
- Quỹ đạo nghiệm số của hệ thống vòng kín sau khi hiệu chỉnh.
- Biểu đồ Bode biên độ và pha của hệ thống vòng hở sau khi hiệu chỉnh.

2.2.2. Thiết kế hệ thống điều khiển dùng phương pháp quỹ đạo nghiệm số bằng Matlab

1. Thiết kế quỹ đạo nghiệm: dùng hàm `sisotool.m` hoặc `rltool.m`

Thí dụ: Cho hệ thống hồi tiếp âm đơn vị có hàm truyền vòng hở

$$G_p(s) = \frac{1}{s(s+1)(s+4)}$$

Dùng `sisotool.m`:

```
>> Num=1;  
>> Den=[1 5 4 0];  
>> sysc=tf(num,den);  
>> sisotool('rlocus', sysc)
```

hoặc dùng `rltool.m`:

```
>> Num=1;  
>> Den=[1 5 4 0];  
>> sysc=tf(num,den);  
>> rltool(sysc)
```


Xét hàm truyền hở của hệ thống điều khiển vòng kín đơn:

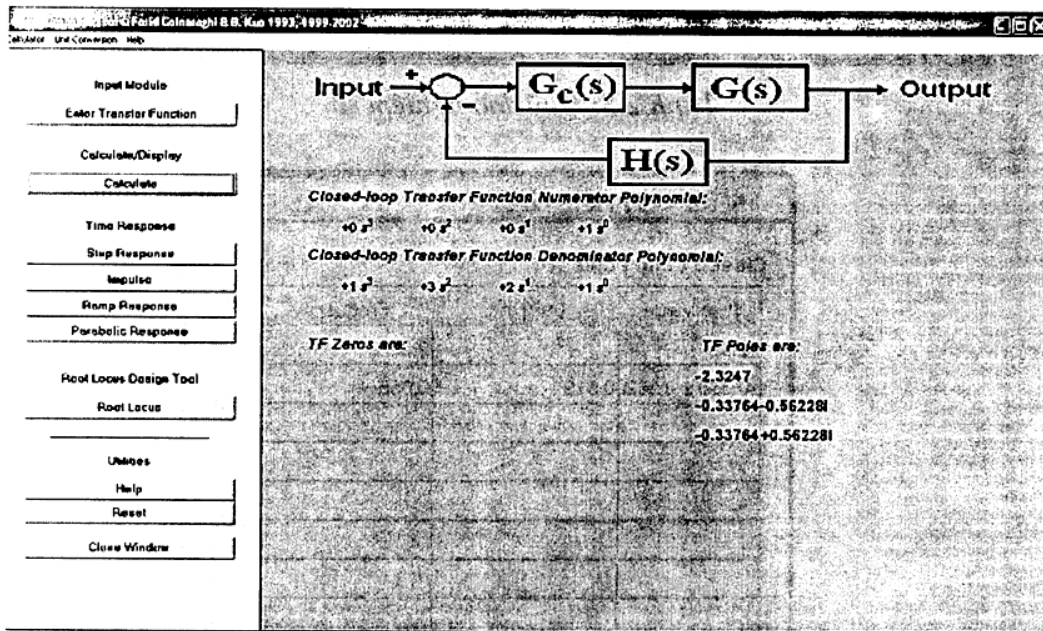
$$G(s)H(s) = \frac{K}{s(s+1)(s+2)} = \frac{K}{s^3 + 3s^2 + 2s}$$

Để tìm quỹ đạo nghiệm, kích hoạt timetool và nhấn vào nút “Enter Transfer Function”. Nhập giá trị của $G(s)$ và $H(s)$. Có thể gán $H(s)=K$, nghĩa là $K=1$, trong trường hợp này $G(s)$ là:

$$G(s) = \frac{1}{s^3 + 3s^2 + 2s}$$

$G_c(s)=1$, $H(s)=1$, $K=1$. Hình vẽ trên minh họa modul nhập hàm truyền sau khi nhập giá trị $G(s)$. Tiếp theo nhấn nút “APPLY” để trở về cửa sổ Time Response Analysis, nhấn nút “Calculate” để tính hàm truyền.

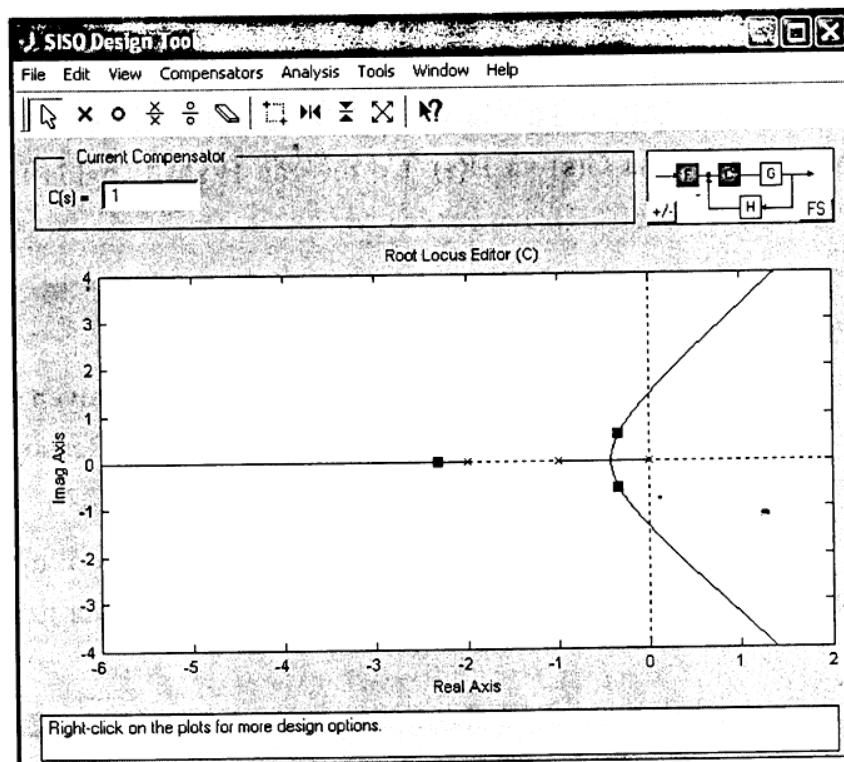
Để vẽ quỹ đạo nghiệm, ta nhấn “Root Locus”. Hình vuông đỏ định nghĩa cực vòng kín, ta có thể xem bằng cách kích hoạt Closed-Loop Pole Viewer từ menu View.



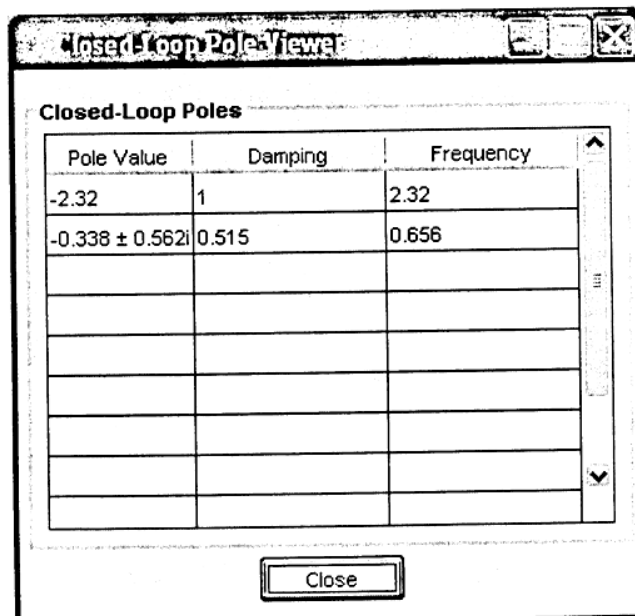
Hình 2.4: Tính toán hàm truyền

Bài 2: Thiết kế hệ thống điều khiển tự động dùng Matlab

Nhấn vào “Root Locus”, kích hoạt SISO design tool của Matlab.



Hình 2.5: Quỹ đạo nghiệm số



Hình 2.6: Cực của hệ thống vòng kín với $K=1$.

Cửa sổ lệnh Matlab thể hiện hàm truyền:

```
>> acsys
```

G=

Transfer function:

$$1$$

$$s^3 + 3 s^2 + 2 s$$

Gc=

Transfer function:

$$1$$

H=

Transfer function:

$$1$$

G*G_c ==>open loop

Transfer function:

$$1$$

$$s^3 + 3 s^2 + 2 s$$

G*G_c*H ==>loop

Bài 2: Thiết kế hệ thống điều khiển tự động dùng Matlab

Transfer function:

$$\frac{1}{s^3 + 3s^2 + 2s}$$

$G*G_c/(1+G*G_c*H) \Rightarrow$ closed loop

Transfer function:

$$\frac{1}{s^3 + 3s^2 + 2s + 1}$$

system Coefficients are

Closed Loop TF in zero/pole format

Zero/pole/gain:

$$\frac{1}{(s+2.325)(s^2 + 0.6753s + 0.4302)}$$

System Zeros are

zeroTF =

Empty matrix: 1-by-0

System Poles are

poleTF =

-2.3247 -0.3376 - 0.5623i -0.3376 + 0.5623i

>>

2.2.3. Thiết kế ở miền tần số

Ví dụ: Vẽ biểu đồ Bode, độ lợi pha tới hạn của hệ thống điều khiển.

Cho hệ thống hồi tiếp âm đơn vị, có hàm truyền vòng hở

$$G_p(s) = \frac{8}{s(s+1)(s+4)} = \frac{8}{s^3 + 5s^2 + 4s}$$

Dùng hàm sisotool.m để thiết kế tần số.

>> num=1;

>> den=[1 5 4 0];

>> sysc=tf(num,den);

>>sisotool('bode', sysc)

Thí dụ:

Cho hàm truyền hở: $G_p(s) = \frac{1}{s(s+1)(s+4)}$

Lệnh Matlab:

>> num=1

num =

1

>> den=[1 5 4 0]

den =

1 5 4 0

>> sysc=tf(num,den)

Transfer function:

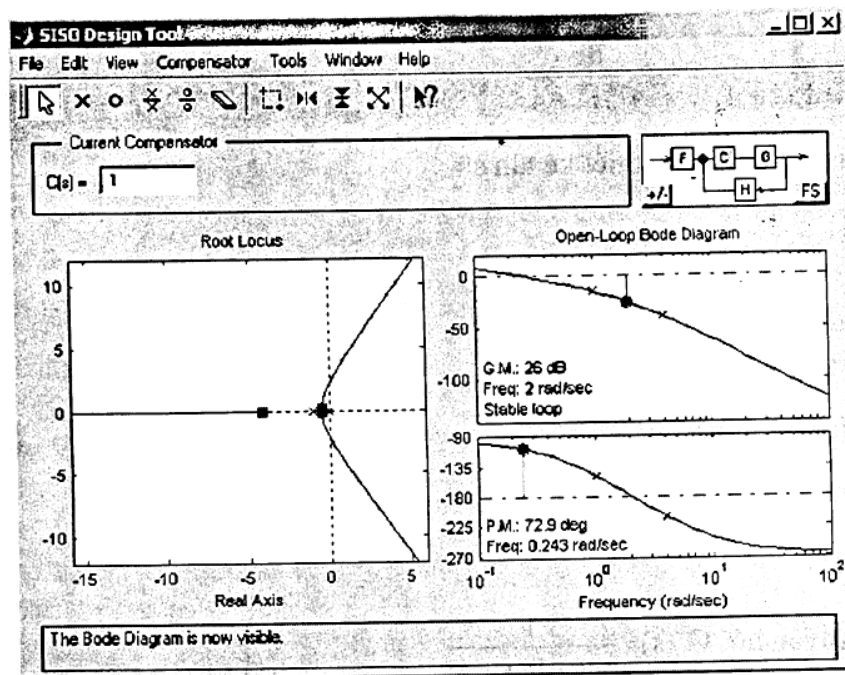
$$1$$

$$s^3 + 5s^2 + 4s$$

```
>> rltool(sysc) % Thiết kế quy dao nghiem
```

```
>> sisotool('bode',sysc) % Thiết kế tan so
```

```
>> sisotool('rlocus',sysc) % Thiết kế quy dao nghiem
```



Hình 2.7: Quỹ đạo nghiệm và biểu đồ Bode

Minh họa kết quả:

Plant Output (step)

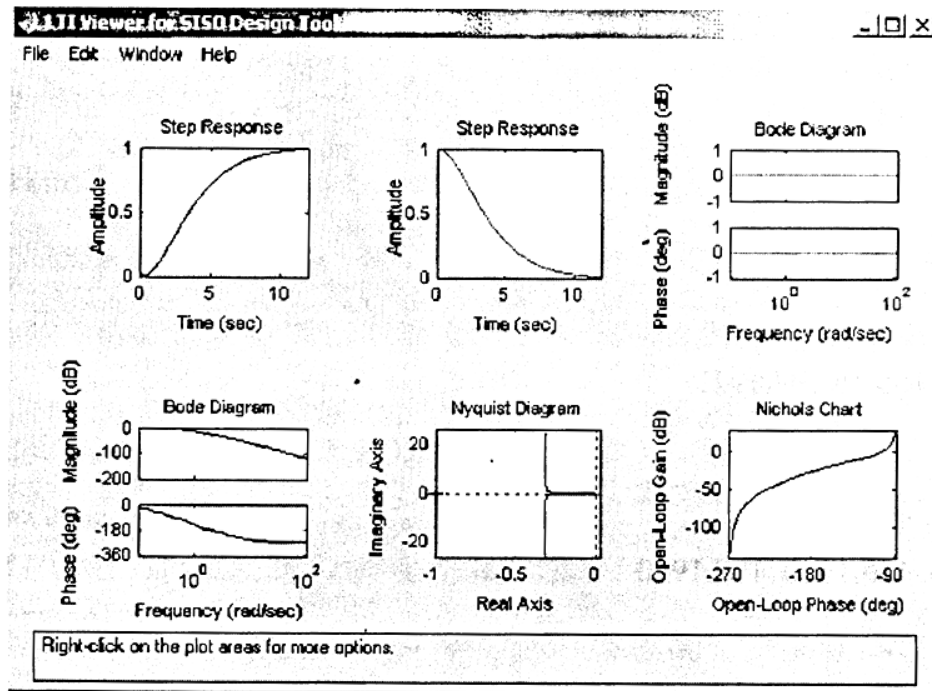
Control signal (step)

Compensator(Bode)

Closed-loop Bode

Open-loop Nyquist

Open-loop Nichols



Hình 2.8

2.2.4. Chuyển từ mô hình liên tục theo thời gian ra rời rạc và ngược lại:

Hàm **c2d.m**

Cú pháp:

$$\text{SYSD} = \text{C2D}(\text{SYSC}, \text{TS}, \text{METHOD})$$

Chuyển mô hình LTI liên tục thời gian SYSC ra mô hình rời rạc thời gian SYSD với thời gian lấy mẫu TS. Chọn lựa phương pháp rời rạc hóa METHOD.

- 'zoh' Zero-order hold on the inputs.
- 'foh' Linear interpolation of inputs (triangle appx.)
- 'tustin' Bilinear (Tustin) approximation.
- 'prewarp' Tustin approximation with frequency prewarping.

The critical frequency W_c is specified as fourth input by $\text{C2D}(\text{SYSC}, \text{TS}, \text{'prewarp'}, W_c)$.

'matched' Matched pole-zero method (for SISO systems only).

The default is 'zoh' when METHOD is omitted.

Bài 2: Thiết kế hệ thống điều khiển tự động dùng Matlab

Thí dụ:

```
» num=[0 314 0]
```

num =

0 314 0

```
» den=[1 155.15 197.192]
```

den =

1.0000 155.1500 197.1920

Chuyển sang hàm truyền $G(s)$

```
» gs=tf(num,den)
```

Transfer function:

314 s

$s^2 + 155.2 s + 197.2$

Chuyển sang hàm rời rạc bằng biến đổi song tuyến tính:

```
» c2d(gs,0.1,'tustin')
```

Transfer function:

1.697 z² - 1.697

$z^2 - 0.1096 z - 0.6772$

Sampling time: 0.1

Hàm **d2cm**: đổi từ rời rạc về liên tục.

Biến đổi từ liên tục sang rời rạc và ngược lại dùng Matlab:

Thí dụ 1: Cho hệ: $G_1(s) = \frac{1}{s(s+1)}$ Tìm $G_1(z)$

Hàm vd2_14a.m:

```
numc1=[1];
```

```
denc1=[1 1 0];
```

```
T=1;
```

```
[numz,demz]=c2dm(numc1,denc1,T,'ZOH') ;
```

```
printsys(numz,demz,'z')
```

```
>> vd2_14a
```

```
num/den =
```

0.36788 z + 0.26424

z² - 1.368 z + 0.3679

$$G(z) = \frac{0,369z + 0,2642}{z^2 - 1,368z + 0,3679}$$

Thí dụ 2: Đổi $G(z)$ sang $G(s)$

Hàm vd2_14.m:

```
numz=[0.369 0.2642];
```

```
denz=[ 1 -1.368 0.3679];
```

```
T=1;
```

```
[numc1,denc1]=d2cm(numz,denz,T,'zoh');
```

```
printsys(numc1,denc1)
```

```
>> vd2_14
```

```
num/den =
```

$$0.00078371 s + 1.0017$$

$$s^2 + 0.99994 s - 0.00015819$$

$$G_1(s) = \frac{1}{s(s+1)}$$

2.3.Thí nghiệm

Dùng công cụ Sisotool trong Matlab

Nguyên tắc thiết kế hệ thống điều khiển dùng phương pháp quỹ đạo nghiệm số là dựa vào phương trình đặc tính của hệ thống sau khi hiệu chỉnh:

$$1 + G_c(s).G(s) = 0$$

$$\Leftrightarrow \begin{cases} |G_c(s).G(s)| = 1 \\ \angle G_c(s).G(s) = -180^\circ \end{cases} (*)$$

Ta cần tính toán các thông số của bộ hiệu chỉnh $G_c(s)$ sao cho đáp ứng ngõ ra của hệ thống đạt được các chất lượng về đáp ứng quá độ và sai số xác lập nhưng phải thỏa mãn điều kiện biên độ và điều kiện pha ở (*). Chú ý, trong công cụ sisotool thì hàm truyền bộ hiệu chỉnh $C(s)$ chính là $G_c(s)$ ở (*).

2.3.1.Thiết kế bộ hiệu chỉnh sớm pha

Mục đích:

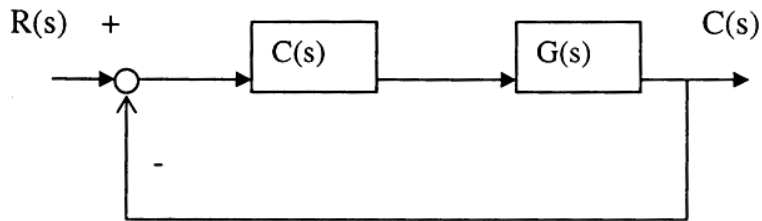
Trong phần này sinh viên sẽ thiết kế bộ hiệu chỉnh sớm pha để hệ thống đạt được các tiêu chuẩn về đáp ứng quá độ như độ vọt lố và thời gian xác lập. Hàm truyền sớm pha có dạng:

$$C(s) = K_c \frac{1 + \alpha Ts}{1 + Ts}, \text{ với } \alpha > 1.$$

Từ các yêu cầu về đáp ứng quá độ ta tìm được vị trí của cặp cực quyết định trên QĐNS. Sau đó ta tính các thông số của bộ hiệu chỉnh $C(s)$ để sao cho QĐNS của hệ thống sau khi hiệu chỉnh đi qua cặp cực quyết định này.

Thí nghiệm:

Cho hệ thống như hình vẽ (H2.9):



Hình 2.9

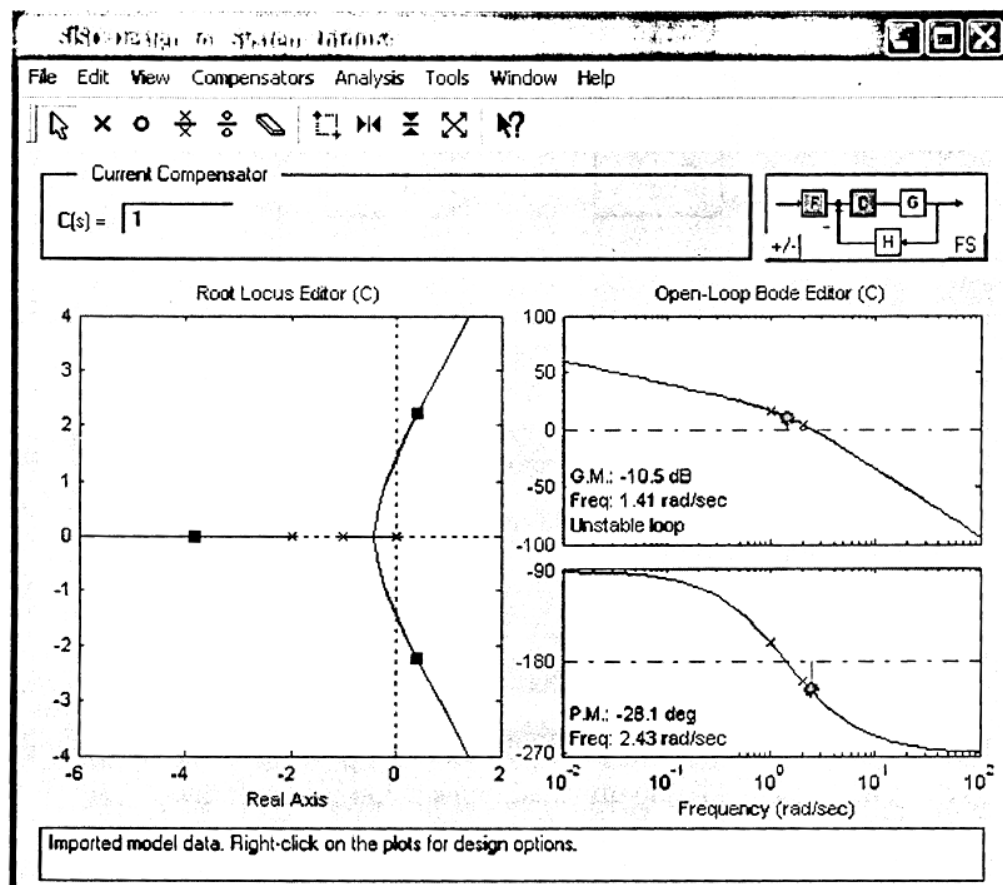
$$G(s) = \frac{20}{s(s+1)(s+2)}.$$

- Dùng công cụ sisotool nhập hàm truyền hệ thống. Dựa vào QĐNS của hệ thống khảo sát hệ thống có ổn định hay không. Giải thích. Vẽ đáp ứng quá độ của hệ thống với đầu vào hàm nấc để minh họa hệ thống có ổn định hay không. Lưu hình vẽ này để so sánh với đáp ứng hệ thống sau khi hiệu chỉnh.
- Thiết kế bộ hiệu chỉnh sớm pha để hệ thống có độ vọt lố POT nhỏ hơn 20% và thời gian xác lập nhỏ hơn 8s. Trình bày rõ quá trình thiết kế này.
- Vẽ đáp ứng quá độ của hệ thống sau khi hiệu chỉnh để chứng minh hệ thống đạt được các yêu cầu ở câu b. Lưu hình vẽ này để viết báo cáo.

Hướng dẫn:

Nhập hàm truyền và khởi động siso để import G và H vào sisotool với chú ý: $G=tf(20,conv([1\ 1\ 0],[1\ 2]))$ và $H=tf(1,1)$.

Cửa sổ sisotool hiện ra như sau:



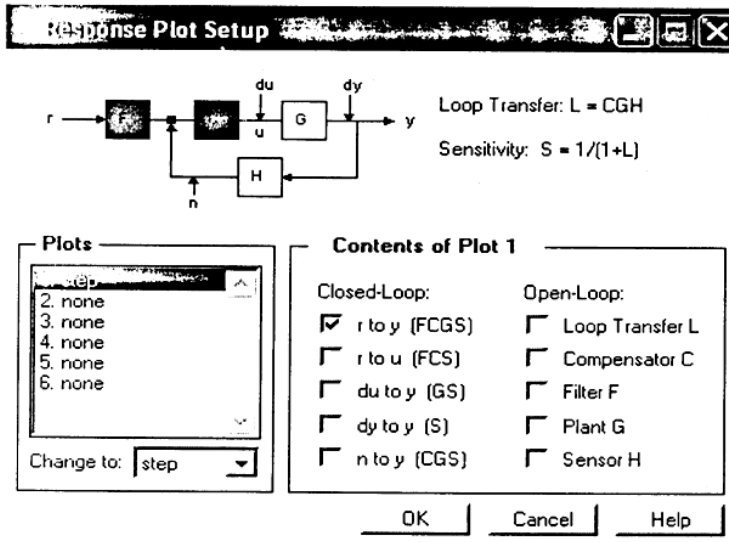
Hình 2.10

Quan sát QĐNS ta thấy phương trình đặc tính vòng kín có 3 nghiệm (dấu vuông màu đỏ):

$$s_1 = -3.84, \quad s_2 = 0.419 + j2.24, \quad s_3 = 0.419 - j2.24$$

Nhận thấy hệ thống có 2 nghiệm s_2 và s_3 nằm bên phải mặt phẳng phức nên hệ thống không ổn định. Ta cũng có thể quan sát trên biểu đồ Bode và nhận thấy $GM = -10.5 \text{ dB} < 0$ và $PM = -28.1^\circ < 0$ nên kết luận hệ thống không ổn định. Tuy nhiên, ở phần thí nghiệm này ta dùng phương pháp quỹ đạo nghiệm số nên sẽ không dựa vào biểu đồ Bode để đánh giá mà chỉ dựa vào QĐNS.

Để xem đáp ứng quá độ của hệ thống với đầu vào hàm nấc, vào menu [Analysis] → [Other Loop Responses]. Cửa sổ Response Plot Setup hiện ra. Tiến hành cài đặt các tín hiệu cần vẽ đáp ứng. Ở đây ta chọn như hình bên dưới:



. r to y: vẽ đáp ứng của ngõ ra $y(t)$ theo tín hiệu đầu vào $r(t)$. Đây chính là đáp ứng quá độ cần vẽ.

. r to u: vẽ đáp ứng của tín hiệu điều khiển $u(t)$.

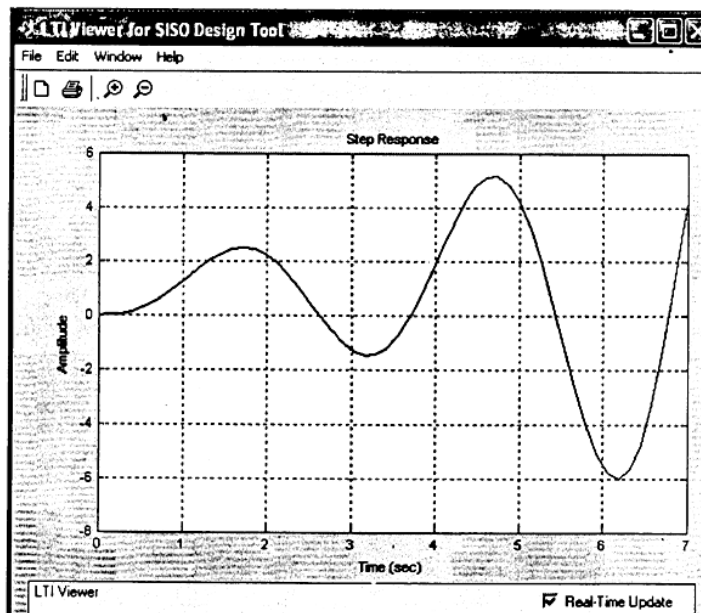
...

$$s(s^2 + 7s + 12)$$

$$s^3 + 7s^2 + 12s$$

Hình 2.11

Vì cửa sổ LTI Viewer này không hỗ trợ việc lưu hình vẽ nên phải chuyển sang cửa sổ Figure bằng cách vào menu [File]→[Print to Figure]. Cửa sổ Figure hiện ra và sau đó tiến hành lưu hình vẽ.



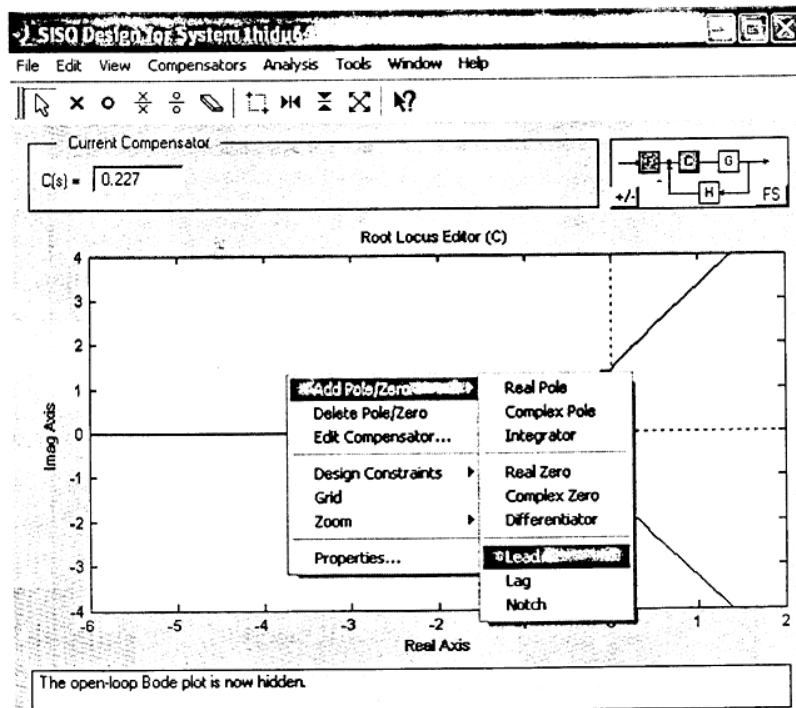
Hình 2.12

Bây giờ thiết kế bộ hiệu chỉnh để hệ thống có $POT < 20\%$ và $t_{x1} < 8s$. Ta quay trở lại với cửa sổ sisotool. Vì trong phần này ta không sử dụng biểu đồ Bode và để mở rộng vùng QĐNS nên ta xóa vùng biểu đồ Bode đi bằng cách vào menu [View] bỏ dấu chọn mục [Open – Loop Bode].

Kích chuột phải vào vùng QĐNS, menu kiểu pop-up xuất hiện:

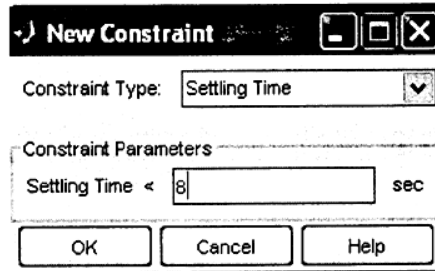
- Lead: bộ hiệu chỉnh sớm pha.
- Lag: bộ hiệu chỉnh trễ pha.
- Notch: bộ hiệu chỉnh sớm trễ pha.
- Delete Pole/Zero: xóa các cực và zero của bộ hiệu chỉnh.
- Edit Compensator...: thay đổi các thông số của bộ hiệu chỉnh.
- Design Constraints: giới hạn vùng thỏa mãn các tiêu chuẩn chất lượng.

Chọn [Add Pole/Zero] → [Lead] để thêm khâu hiệu chỉnh sớm pha vào hệ thống. Nhấp chuột vào một vị trí bất kì trên trục thực của QĐNS để xác định vị trí cực và zero của bộ hiệu chỉnh, sisotool sẽ gán tự động vị trí của zero nằm gần gốc tọa độ hơn cực.

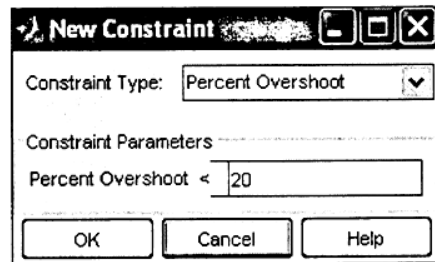


Hình 2.13

Kích chuột phải vào vùng QĐNS, ta chọn [Design Constraints]→New để cài độ vọt lố và thời gian xác lập như sau:

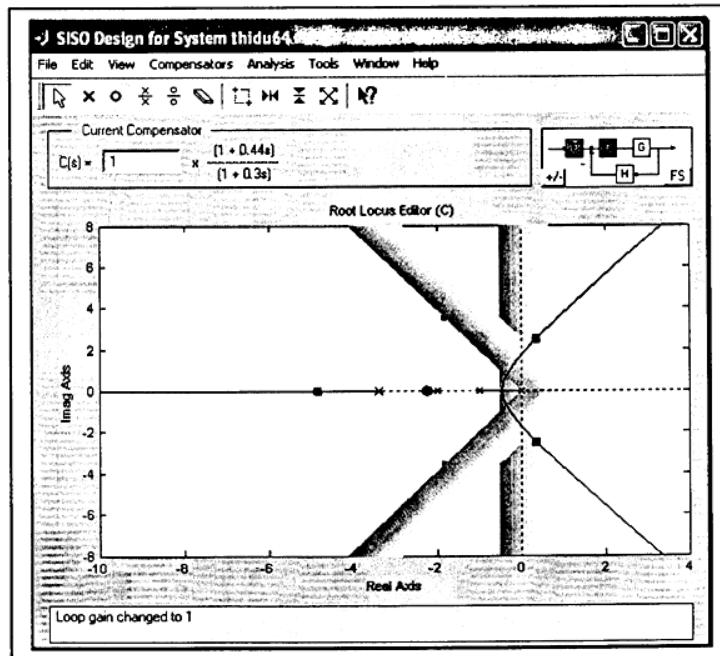


Hình 2.14 a



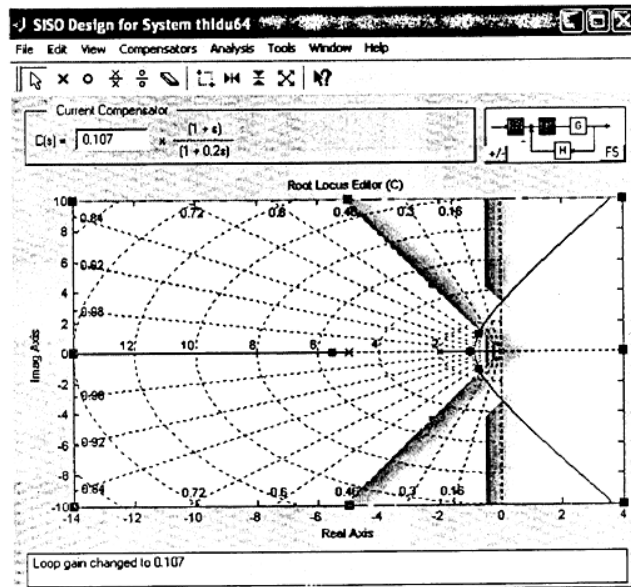
Hình 2.14b

Sau khi tiến hành cài đặt xong, QĐNS lúc này sẽ thay đổi như sau:

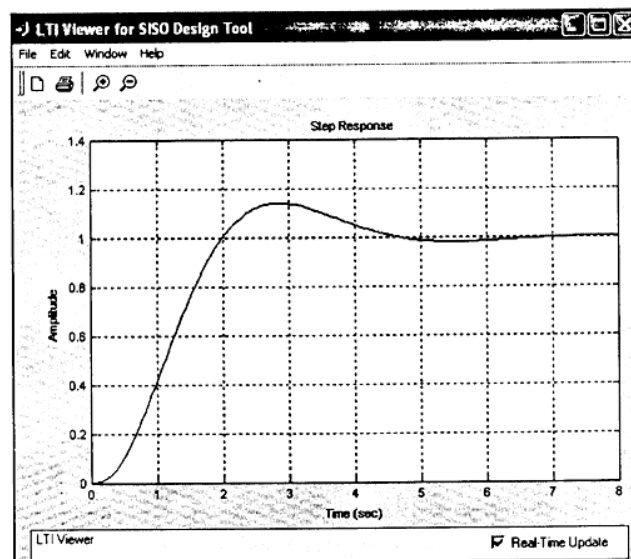


Hình 2.15

Bây giờ ta sẽ di chuyển các cực và zero của bộ hiệu chỉnh $C(s)$ trên trục thực sao cho nhánh QĐNS(A) kéo vào vùng thỏa mãn thiết kế. Có 2 phương pháp tìm cực và zero của bộ hiệu chỉnh là phương pháp đường phân giác và phương pháp khử cực (sinh viên xem lại sách lý thuyết để hiểu 2 phương pháp này). Ở đây ta chọn phương pháp khử cực vì dễ thao tác và trực quan trên cửa sổ QĐNS.



Hình 2.16



Hình 2.17

2.3.2. Thiết kế bộ hiệu chỉnh trễ pha

Mục đích: Sinh viên thiết hàm truyền khâu hiệu chỉnh trễ pha có dạng:

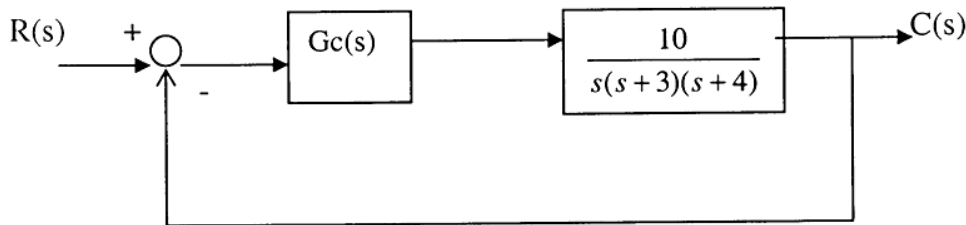
$$G_c(s) = K_c \frac{s + (1/\beta T)}{s + (1/T)} \text{ với } \beta < 1$$

Bài toán đặt ra là chọn giá trị K_c , β và T để đáp ứng hệ thống thỏa mãn yêu cầu về sai số xác lập mà “không” làm ảnh hưởng đến đáp ứng quá độ (ảnh hưởng không đáng kể).

Thí nghiệm: Thiết kế khâu hiệu chỉnh trễ pha dùng phương pháp QĐNS.

Hãy thiết kế khâu hiệu chỉnh $G_c(s)$ sao cho hệ thống có sơ đồ khối dưới đây sau khi hiệu chỉnh có sai số đối với tín hiệu vào là hàm dốc là 0,02 và đáp ứng quá độ thay đổi không đáng kể.

Dùng công cụ Sisotool để thiết kế.



Xem [1] về phương pháp thiết kế khâu hiệu chỉnh trễ pha dùng QĐNS.

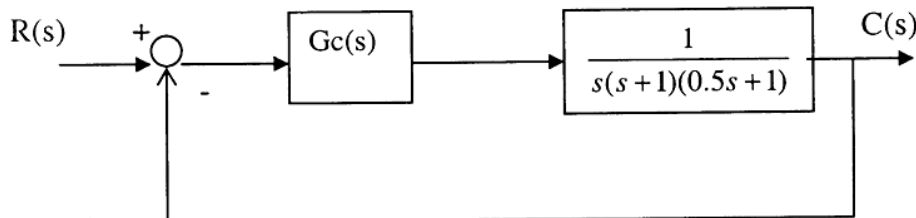
Đáp số là: $G_c(s) = \frac{s + 0,1}{s + 0,0017}$

Thí nghiệm 2:

Thí dụ: Thiết kế khâu hiệu chỉnh trễ pha dùng phương pháp biểu đồ Bode

Hãy thiết kế khâu hiệu chỉnh trễ pha sao cho hệ thống sau khi hiệu chỉnh có:

$$K_v^* = 5; \phi M^* \geq 40; GM^* \geq 10dB.$$



Hình 2.18

Bài 2: Thiết kế hệ thống điều khiển tự động dùng Matlab

Giải:

Hàm truyền khâu hiệu chỉnh trễ pha cần thiết kế là:

$$G_c(s) = K_c \frac{1 + \alpha Ts}{1 + Ts}, (\alpha < 1)$$

Bước 1: Xác định K_c .

Hệ số vận tốc của hệ sau khi hiệu chỉnh là:

$$K_v^* = \lim_{s \rightarrow 0} s G_c(s) G(s) = \lim_{s \rightarrow 0} s K_c \frac{1 + \alpha Ts}{1 + Ts} \cdot \frac{1}{s(s+1)(0.5s+1)} = K_c$$

$$\Rightarrow K_c = K_v^* \Rightarrow K_c = 5.$$

Bước 2:

Hàm truyền hệ $G(s)$:

$$>> G1=tf(1,conv([1 1 0],[0.5 1]))$$

Transfer function:

$$\frac{1}{0.5s^3 + 1.5s^2 + s}$$

$$\text{Đặt } G_1(s) = K_c G(s) = 5 \cdot \frac{1}{s(s+1)(0.5s+1)}$$

$$\Rightarrow G_1(s) = \frac{5}{s(s+1)(0.5s+1)}$$

Bước 3: Xác định tần số cắt mới

Cách 1: Tìm ω_c bằng phương pháp giải tích. Ta có:

$$\begin{aligned}\varphi_1(\omega_c) &= -180^\circ + \Phi M^* + \theta \\ \Rightarrow -90^\circ - \arctan(\omega_c) - \arctan(0.5\omega_c) &= -180^\circ + 40^\circ + 5^\circ \\ \Rightarrow \arctan(\omega_c) + \arctan(0.5\omega_c) &= 45^\circ \\ \Rightarrow \frac{(\omega_c) + (0.5\omega_c)}{1 - 0.5(\omega_c)^2} &= \tan(45^\circ) = 1 \\ \Rightarrow 0.5(\omega_c)^2 + 1.5(\omega_c) - 1 &= 0 \\ \Rightarrow \omega_c &= 0.56(\text{rad/sec})\end{aligned}$$

Cách 2: dựa vào biểu đồ Bode

Ta có:

$$\begin{aligned}\varphi_1(\omega_c) &= -180^\circ + \Phi M^* + \theta \\ \Rightarrow \varphi_1(\omega_c) &= -180^\circ + 40^\circ + 5^\circ \\ \Rightarrow \varphi_1(\omega_c) &= -135^\circ\end{aligned}$$

Vẽ đường thẳng có hoành độ -135° . Hoành độ giao điểm của đường thẳng này với biểu đồ Bode về pha $\varphi_1(\omega)$ chính là giá trị tần số cắt mới. Theo hình vẽ ta có $\omega_c = 0,5(\text{rad/sec})$.

Bước 4:

Cách 1: Tính α từ điều kiện:

$$\begin{aligned}|G_1(j\omega_c)| &= \frac{1}{\alpha} \\ \Rightarrow \left| \frac{5}{s(s+1)(0,5s+1)} \right|_{s=j\omega_c} &= \frac{1}{\alpha} \\ \Rightarrow \left| \frac{5}{j0,56(j0,56+1)(0,5xj0,56+1)} \right| &= \frac{1}{\alpha} \\ \Rightarrow \frac{5}{0,56(\sqrt{0,56^2+1})(\sqrt{0,28^2+1})} &= \frac{1}{\alpha} \\ \Rightarrow \frac{5}{0,56x1,146x1,038} = \frac{1}{\alpha} &\Rightarrow \alpha = 0,133.\end{aligned}$$

Cách 2: Tính α từ điều kiện: $L_1(\omega_c) = -20\lg\alpha$

Dựa vào biểu đồ Bode ta thấy : $L_1(\omega_c) = 18\text{dB}$

Suy ra: $\lg\alpha = -0,9$

$$\alpha = 10^{-0,9}$$

$$\alpha = 0,126$$

Ta thấy giá trị α sai khác không đáng kể ở hai cách. Ở các bước thiết kế tiếp theo ta chọn $\alpha = 0,133$.

Bài 2: Thiết kế hệ thống điều khiển tự động dùng Matlab

Bước 5: Chọn zero của khâu trễ pha

$$\frac{1}{\alpha T} \ll \omega_c = 0,56$$

$$\text{Chọn } \frac{1}{\alpha T} = 0,05 \\ \Rightarrow \alpha T = 20$$

Bước 6: Tính thời hằng T

$$\frac{1}{T} = \alpha \frac{1}{\alpha T} = 0,133 \times 0,05 = 0,067 \\ \Rightarrow T = 150$$

$$\text{Vậy: } G_c(s) = 5 \cdot \frac{(20s + 1)}{(150s + 1)}$$

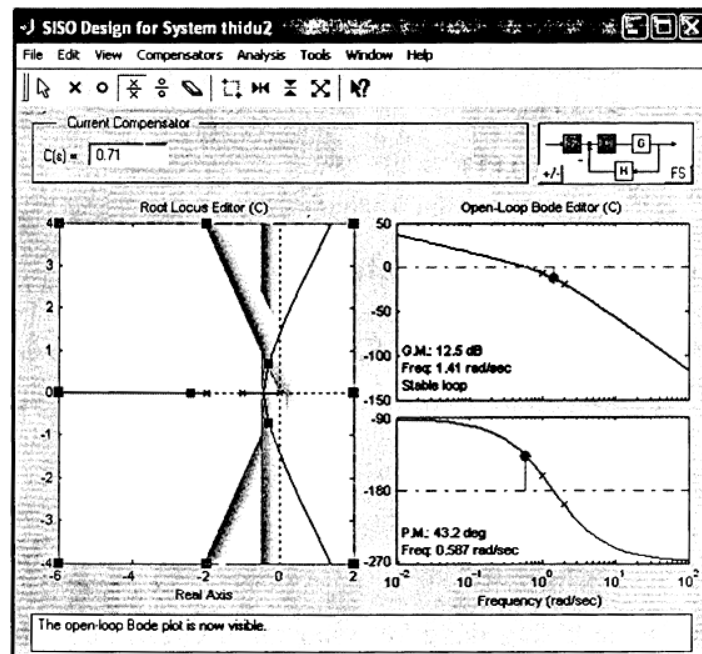
Bước 7: Kiểm tra lại điều kiện biên độ

Dựa vào biểu đồ Bode ta thấy độ dự trữ biên sau khi hiệu chỉnh là: $GM^* = 10 \text{ dB}$.

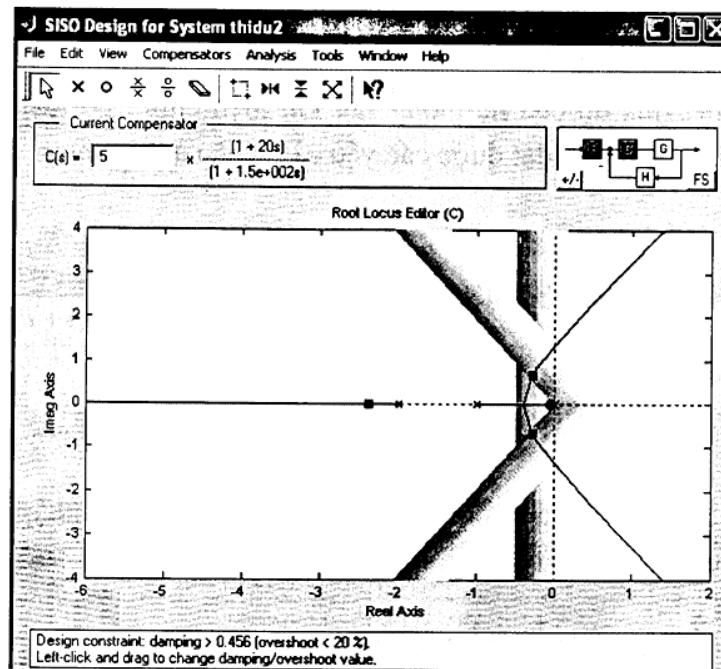
Kết luận: Khâu hiệu chỉnh vừa thiết kế đạt yêu cầu về độ dự trữ biên.

Kết quả:

Dùng Sisotool vẽ biểu đồ Bode



Hình 2.19



Hình 2.20

2.3.3. Thiết kế bộ điều khiển sớm trễ pha

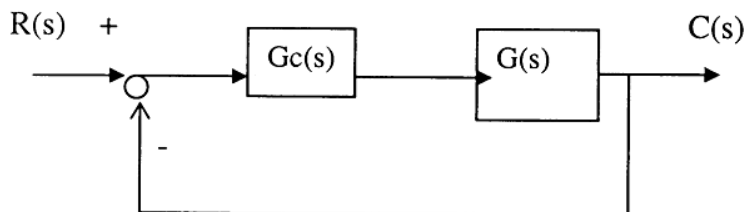
Mục đích:

Trong phần này sinh viên sẽ thiết kế bộ hiệu chỉnh sớm trễ pha để hệ thống đạt được các tiêu chuẩn về đáp ứng quá độ và chất lượng xác lập. Hàm truyền sớm trễ pha có dạng:

$$C(s) = K_{c1} \frac{1 + \alpha T_1 s}{1 + T_1 s} \cdot K_{c2} \frac{1 + \beta T_2 s}{1 + T_2 s} \quad (\alpha > 1, \beta < 1)$$

Thí nghiệm:

Cho hệ thống như hình vẽ:



$$G(s) = \frac{4}{s(s + 0.5)}$$

- Thiết kế bộ hiệu chỉnh sớm trễ pha để hệ thống có $\xi=0.5$, $\omega_n=5$ (rad/sec) và hệ số vận tốc $K_v=80$. Trình bày rõ quá trình thiết kế.
- Vẽ đáp ứng quá độ của hệ thống sau khi hiệu chỉnh với đầu vào hàm dốc để chứng minh hệ thống đạt được các yêu cầu ở câu a. Lưu hình vẽ này để báo cáo.

Hướng dẫn:

Trước tiên ta thiết kế bộ hiệu chỉnh sớm pha $C_1(s) = K_{c1} \frac{1 + \alpha T_1 s}{1 + T_1 s}$ để hệ thống có $\xi=0.5, \omega_n=5$ cho đối tượng $G(s)$. Sau đó, thiết kế bộ hiệu chỉnh trễ pha $C_2(s) = K_{c2} \frac{1 + \beta T_2 s}{1 + T_2 s}$ cho đối tượng mới $G1(s)=G(s).C1(s)$. Chú ý, sau khi thiết kế xong bộ hiệu chỉnh sớm pha ta phải khởi động lại sisotool và nhập lại hàm truyền G và H với hàm truyền G lúc này chính là $G1(s)$ và $H=1$.

với $\xi=0.5, \omega_n=5 \rightarrow s_{1,2}^* = -2.5 \pm j4.33$

Do đó, khi thiết kế bộ hiệu chỉnh sớm pha $C1(s)$, để hệ thống có $\xi=0.5, \omega_n=5$ ta sẽ di chuyển zero của $C1(s)$ tới vị trí -0.5 (vị trí cực của $G(s)$ để khử cực này) và di chuyển cực của $C1(s)$ (phải cách xa gốc tọa độ hơn zero) sao cho QĐNS đi qua hai nghiệm $s_{1,2}^*$. Sau đó dùng chuột di chuyển nghiệm $s2$ (dấu vuông màu đỏ) lại vị trí $s_{1,2}^*$ này.

Kết quả: Hàm truyền khâu sớm pha

$$C1(s) = 0.251x(1+2s)/(1+0.2s)$$

2.3.4. Thiết kế bộ điều khiển PID bằng ACSYS

Bộ điều khiển PID dùng để cải thiện đáp ứng động lực học cũng như giảm hay loại trừ sai số xác lập. Có thể dùng công cụ sisotool hoặc ACSYS để thiết kế bộ điều khiển PID.

Thí dụ 1: Thiết kế PD

Xét hệ có hàm truyền sau:

$$G(s) = \frac{4500}{s(s + 361.2)}$$

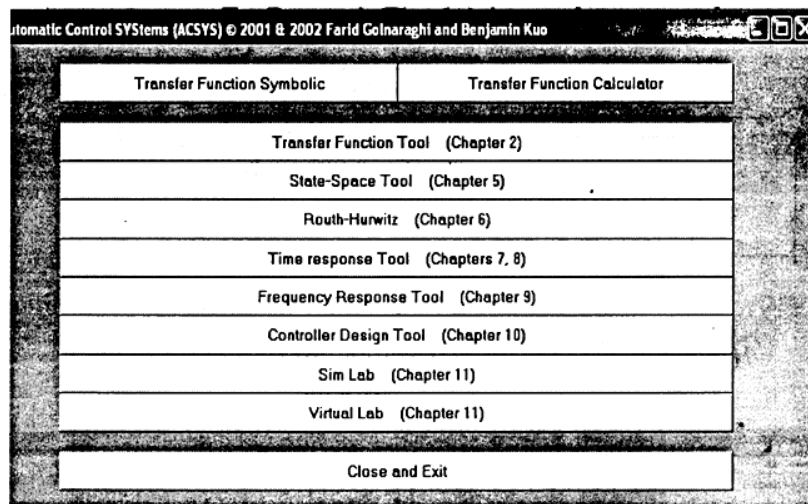
Thiết kế miền thời gian: Thiết kế bộ điều khiển để hệ thống vòng kín thỏa mãn yêu cầu sau:

- Sai số xác lập với ngõ vào hàm unit-ramp ≤ 0.000443 .

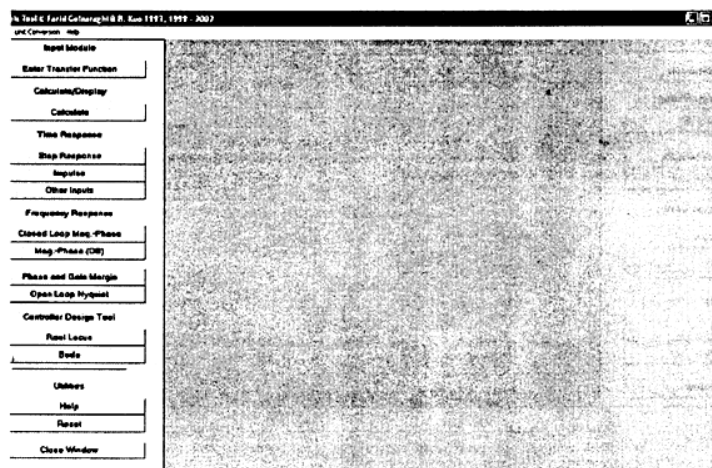
- Vọt lố cực đại $\leq 5\%$.
- Thời gian lên: $t_r \leq 0.005$ sec.
- Thời gian thiết lập $t_s \leq 0.005$ sec.

Dùng ACSYS

Để xem xét chất lượng của hệ ta bắt đầu dùng bộ điều khiển tỉ lệ. Khởi động ACSYS, kích chọn Controller Design tool bằng cách gõ controls ở dòng lệnh Matlab hay nhấp vào nút trong ACSYS.



Hình 2.21: ACSYS



Hình 2.22: Cửa sổ chính thiết kế bộ điều khiển.

Bài 2: Thiết kế hệ thống điều khiển tự động dùng Matlab

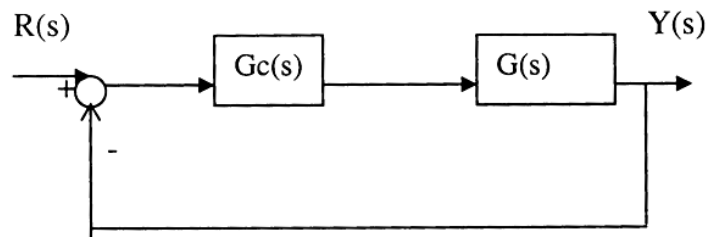
Chọn “Enter Transfer Function” và nhập vào hàm truyền $G(s)$:

Num $G(s)$: 4500

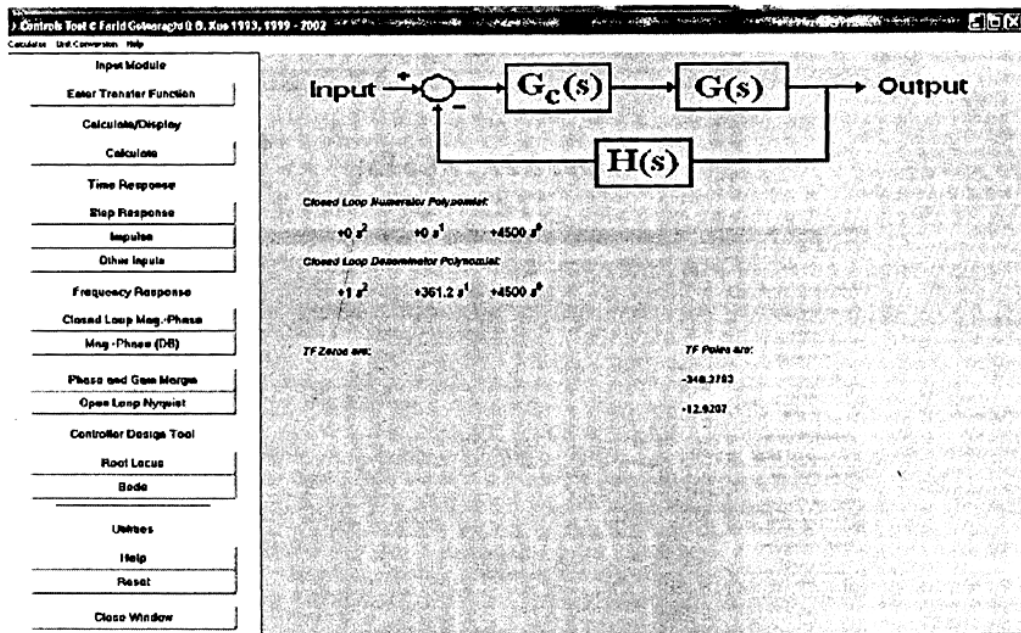
Den $G(s)$: 1 361.2 0

Kế tiếp nhập $G_c(s)=1$ và $H(s)=1$.

Nhấp nút “APPLY” trở về cửa sổ chính và nhấn “Calculate” để đánh giá hàm truyền vòng kín. Chú ý rằng bạn luôn tham khảo cửa sổ lệnh Matlab để xem thể hiện chính xác hàm truyền.



Hình 2.23: Hệ thống điều khiển hồi tiếp âm đơn vị.



Hình 2.24: Cửa sổ chính bộ điều khiển minh họa hàm truyền vòng kín.

Cửa sổ lệnh Matlab:

```
>> acsys
```

G=

Transfer function:

4500

$s^2 + 361.2 s$

Gc=

Transfer function:

1

H=

Transfer function:

1

$G \cdot G_c \Rightarrow \text{open loop } G_1$

Transfer function:

4500

$s^2 + 361.2 s$

$G \cdot G_c \cdot H \Rightarrow \text{loop } G_2$

$G_1 \cdot G_2 = G_2$

Bài 2: Thiết kế hệ thống điều khiển tự động dùng Matlab

Transfer function:

4500

$s^2 + 361.2 s$

$G*G_c/(1+G*G_c*H) ==>$ closed loop $\rightarrow \omega_d$

Transfer function:

4500

$s^2 + 361.2 s + 4500$ $\times$

system Coefficients are

Closed Loop TF in zero/pole format

Zero/pole/gain:

4500

$(s+348.3)(s+12.92)$

System Zeros are

zeroTF =

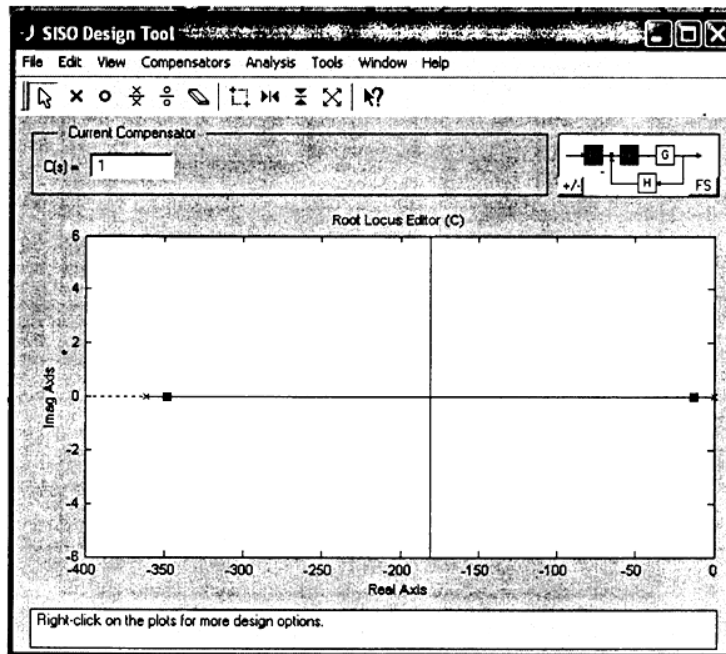
Empty matrix: 1-by-0

System Poles are

poleTF =

-348.2793 -12.9207

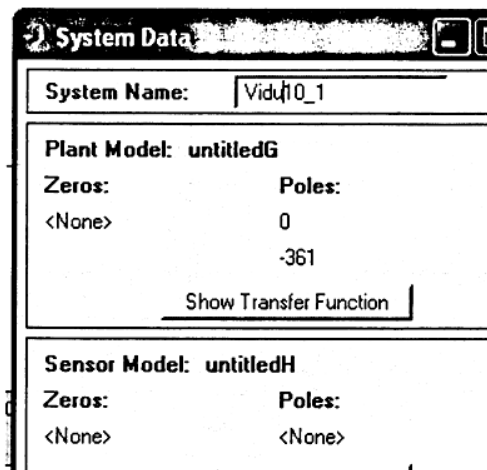
Để xem chất lượng của bộ điều khiển tỉ lệ, ta cần tìm quỹ đạo nghiệm hệ thống. Nhấn vào nút “Root Locus” trong cửa sổ chính điều khiển. Điều này kích hoạt công cụ thiết kế SISO của Matlab.



Hình 2.25: Quỹ đạo nghiệm số.

Hình vuông màu đỏ trong hình thể hiện cực của hệ vòng kín với $K=1$.

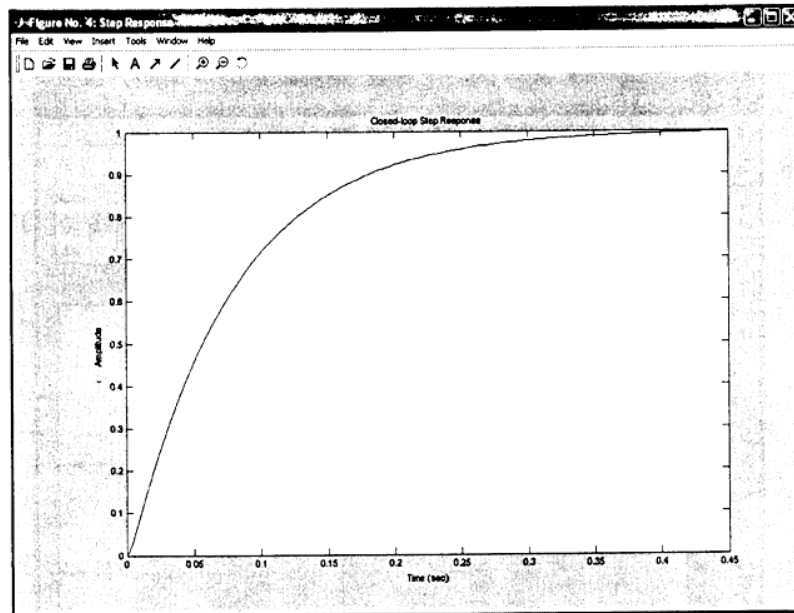
Vào menu View, chọn System Data



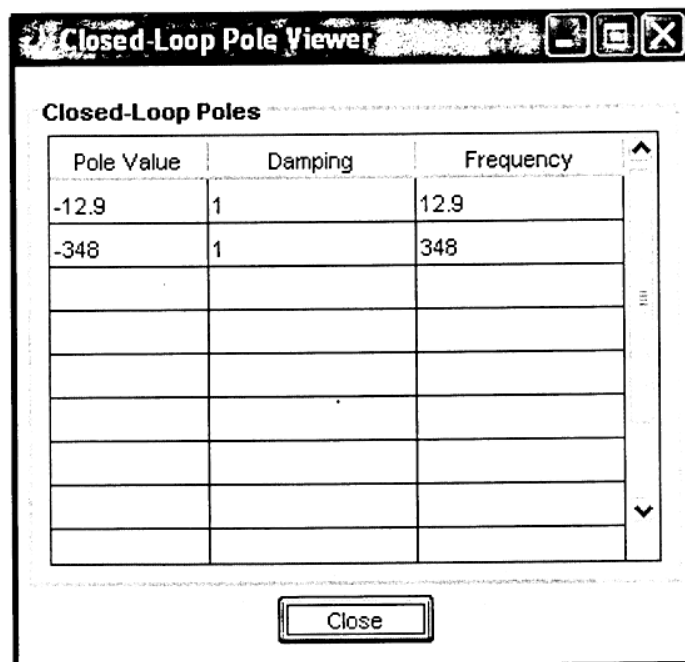
Hình 2.26: Cực và zero của G và H .

Bài 2: Thiết kế hệ thống điều khiển tự động dùng Matlab

Chọn Step Response: xem đáp ứng thời gian hệ kín với ngõ vào hàm bậc đơn vị.

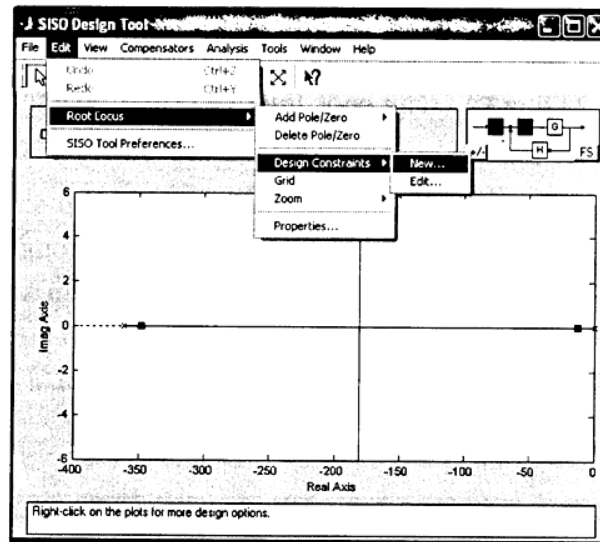


Hình 2.27: Đáp ứng unit-step của hệ kín với $K=1$.

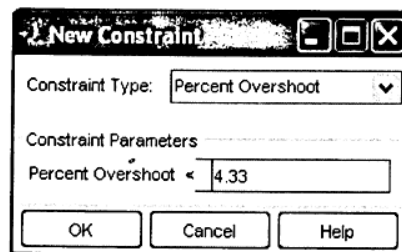
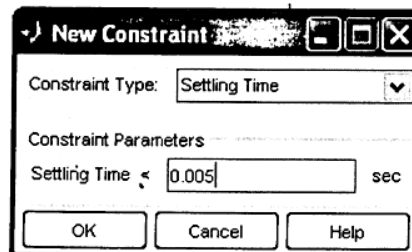


Hình 2.28: Cực và zero của hệ với $K=1$.

Tích hợp tiêu chuẩn thiết kế: Khi bắt đầu thiết kế bộ điều khiển, ta xây dựng tùy chọn tiêu chuẩn thiết kế có sẵn bên trong công cụ SISO Design để thiết lập vùng cực mong muốn trên quỹ đạo nghiệm. Để thêm ràng buộc thiết kế, sử dụng menu Edit và chọn Root Locus, mục New và sau đó chọn tùy chọn Design Constrains như hình sau:



Hình 2.29a



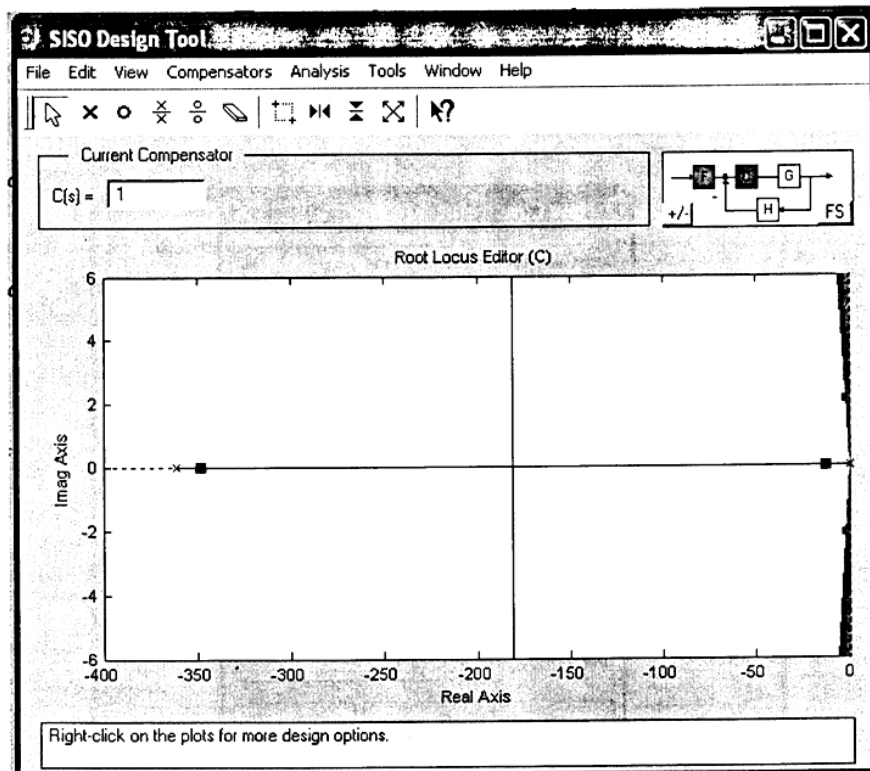
Hình 2.29b: Sử dụng Design Constrains tùy chọn cho ví dụ thiết kế (H2.29a) và Thời gian lên <0.005 sec và phần trăm vọt lố <4.33(H2.29b).

Tùy chọn ràng buộc gồm:

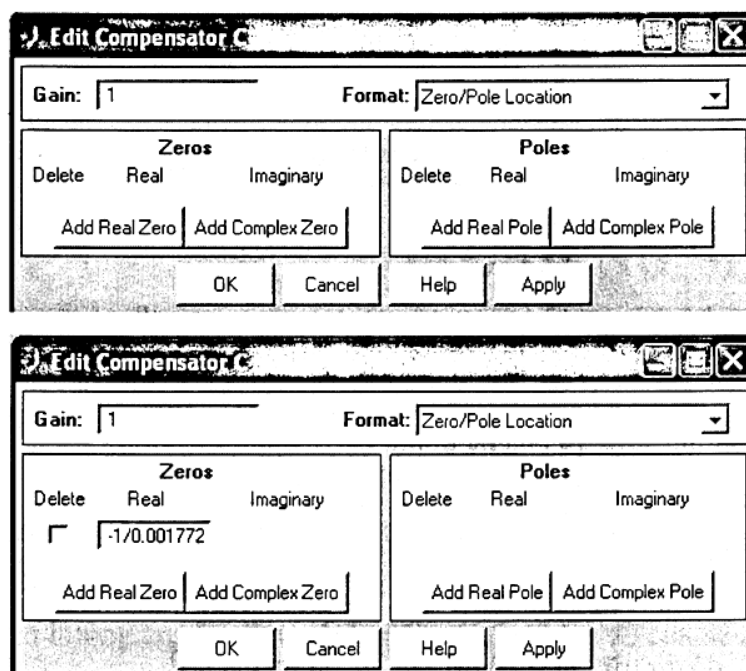
- . Settling time (thời gian thiết lập).
- . Percent Overshoot (phần trăm vọt lố).
- . Damping ratio (tỉ số suy giảm).
- . Natural frequency (tần số tự nhiên).

Trong ví dụ này, ta đưa vào ràng buộc thiết kế là thời gian thiết lập và phần trăm vọt lố.

Hình sau minh họa vị trí cực của hệ kín trên quỹ đạo nghiệm sau khi đưa vào ràng buộc thiết kế. Rõ ràng là cực của hệ khi $K=1$ không nằm trong vùng mong muốn. Không thể dùng bộ điều khiển tỉ lệ để di chuyển cực của hệ kín xa hơn về mặt phẳng trái. Do đó ta dùng bộ điều khiển PD để đạt được điều này, bộ điều khiển PD có zero tại $z=-1/0.001772$. Thành phần tỉ lệ $K_p=1$, sai số xác lập với ngõ vào unit-ramp là $ess=0.000443$ với $K_d=0.001772$.

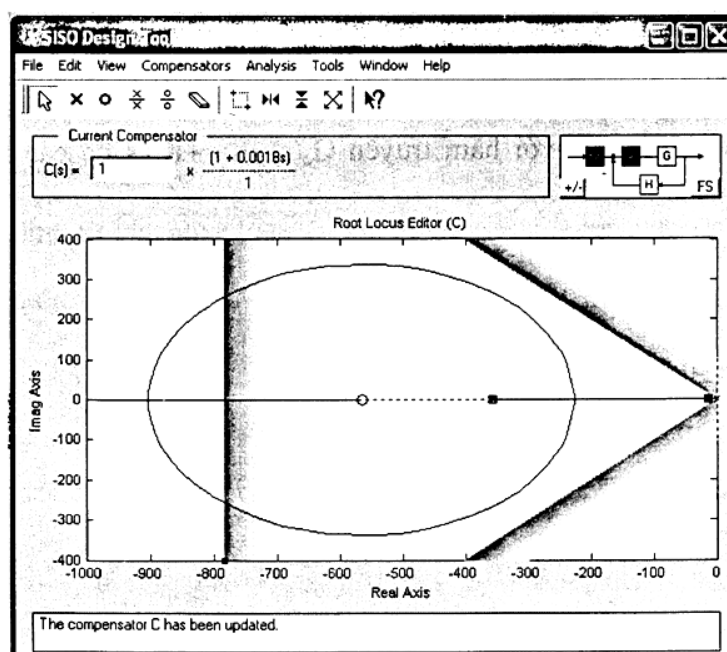


Hình 2.30: Quỹ đạo nghiệm sau khi tích hợp ràng buộc thiết kế là phần trăm vọt lố và thời gian thiết lập.



Hình 2.31: Sự thêm vào zero của bộ điều khiển $C(s)$ để tạo ra bộ điều khiển PD

Quỹ đạo nghiệm mới xuất hiện như hình sau:



Hình 2.32: Quỹ đạo nghiệm sau khi tích hợp zero trong bộ điều khiển PD ở $-1/0.001772$.

Các bài thực tập:

1. Thiết kế mẫu:

Mục đích: sinh viên hãy thiết kế bộ bù sớm trễ pha hệ thống sau:

$$G(s) = \frac{A}{s(s+1)(s+2)} \text{ với } A=20$$

Sao cho: $K_V=10$, độ dự trữ biên=12 dB, độ dự trữ pha là 50° .

Thí nghiệm:

2. Thiết kế hiệu chỉnh sớm trễ pha hệ thống 2:

Có hàm truyền sau:

$$G(s) = \frac{80}{s(0.02s+1)(0.05s+1)}$$

Sao cho: $K_V=80$, độ dự trữ biên=10 dB, độ dự trữ pha là 50° .

Thí nghiệm:

3. Xem xét mô hình bậc 2 của hệ thống điều khiển hành vi máy bay. Hàm truyền của đối tượng là:

$$G_p(s) = \frac{4500K}{s(s+361,2)}$$

Thiết kế bộ điều khiển PD với hàm truyền $G_c(s)=K_P + K_D.s$ để các tiêu chuẩn chất lượng sau thỏa mãn:

- Sai số xác lập với ngõ vào unit-ramp $\leq 0,001$.
- Vọt lò cực đại $\leq 5\%$.
- Thời gian lên $t_r \leq 0,005$ sec.
- Thời gian thiết lập $t_s \leq 0,005$ sec.

2.4. Kiểm tra đánh giá

BÀI 3

KHẢO SÁT VÀ MÔ PHỎNG HỆ THỐNG TỰ ĐỘNG DÙNG SIMULINK

3.1. Mục tiêu

Simulink là một công cụ rất mạnh của Matlab để xây dựng các mô hình một cách trực quan và dễ hiểu. Để mô tả hay xây dựng hệ thống ta chỉ cần liên kết các khối có sẵn trong thư viện của simulink lại với nhau. Sau đó tiến hành mô phỏng hệ thống để xem xét ảnh hưởng của bộ điều khiển đến đáp ứng quá độ của hệ thống và đánh giá chất lượng hệ thống. Sau đó sinh viên thực hiện mô phỏng, khảo sát hệ thống điều khiển nhiệt độ và hệ thống điều khiển tốc độ và vị trí động cơ một chiều.

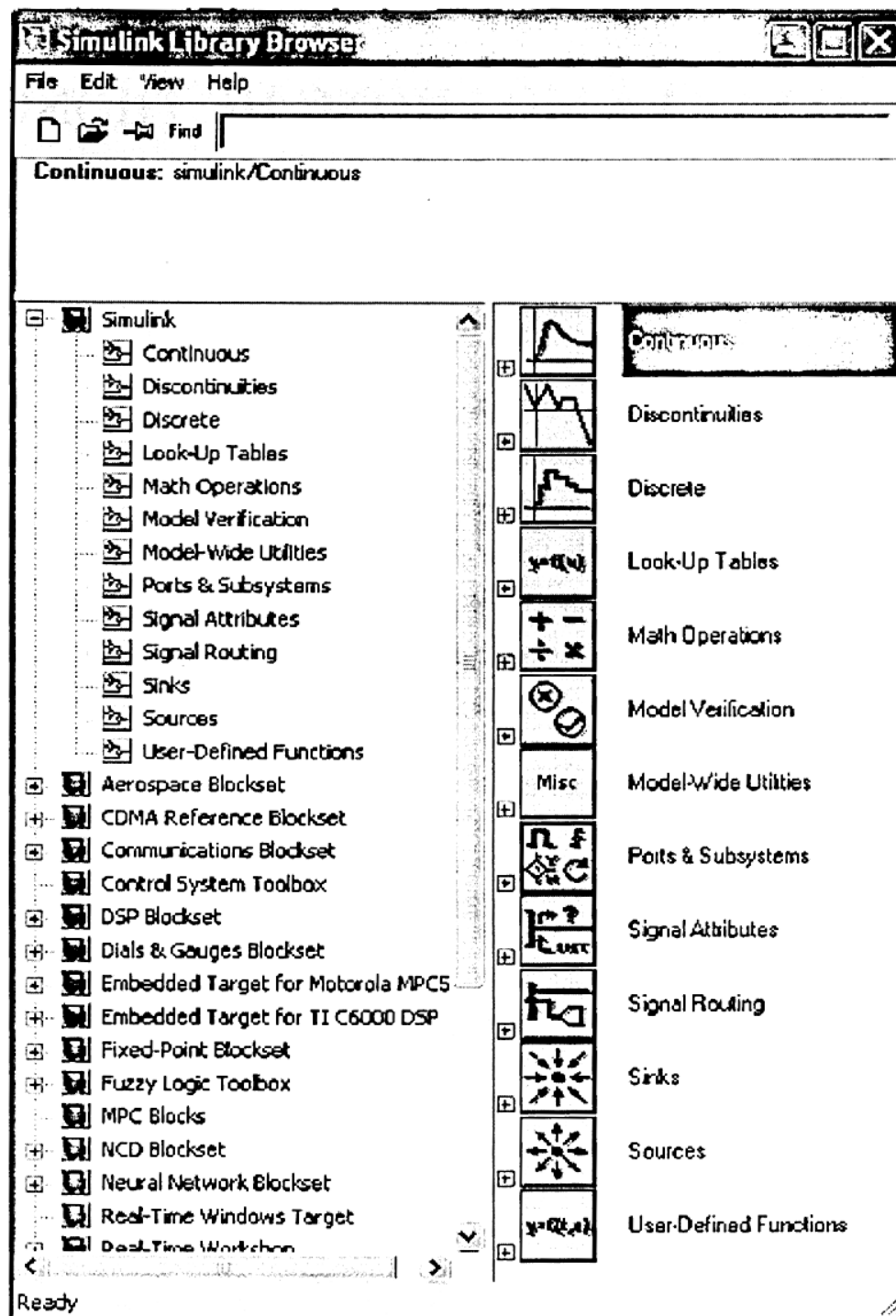
3.2. Nội dung

3.2.1. Giới thiệu về Simulink

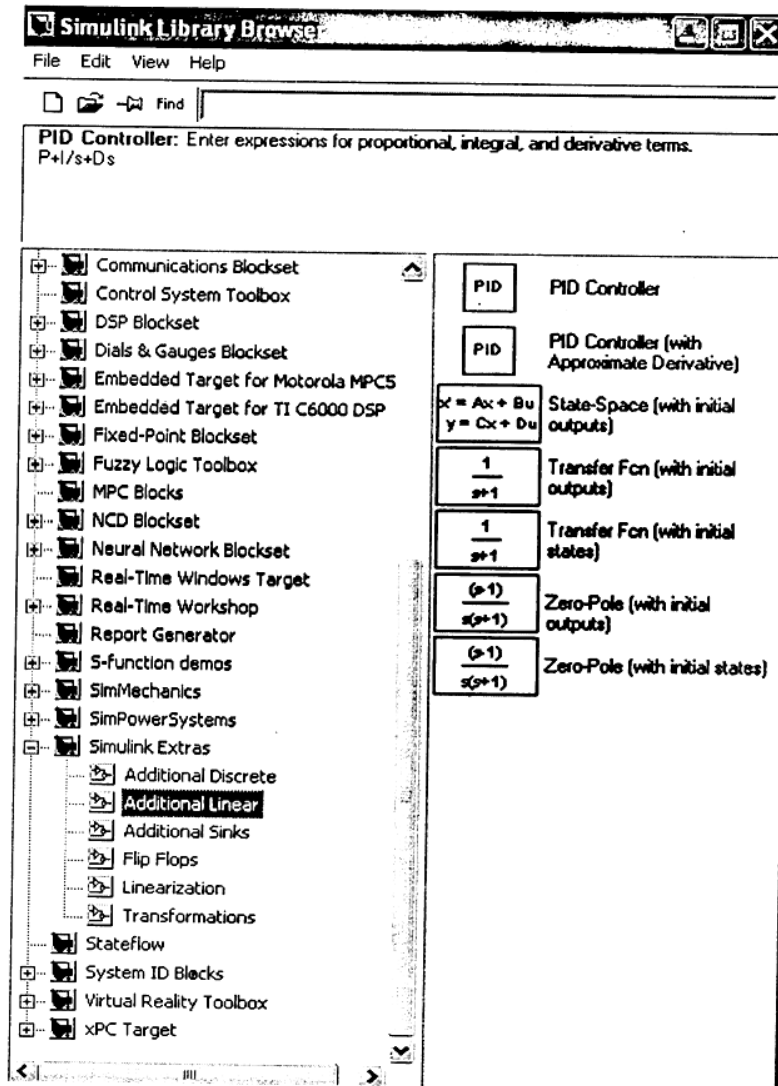
Khởi động Simulink:

- Khởi động Matlab.
- Gõ lệnh simulink từ dấu nhắc lệnh.

Cửa sổ đầu tiên được trình bày bởi Simulink.



Hình 3.1: Cửa sổ Simulink.

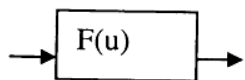


Hình 3.2: Simulink Extras

Khối chức năng

Khối thư viện Continuous gồm có vi phân, tích phân, hàm truyền, và không gian trạng thái,...

Thí dụ: khối Transfer Fcn (hàm truyền)



Fcn

Tạo mới và soạn thảo lưu đồ tín hiệu:

- Sao chép, di chuyển, đánh dấu, xoá, hệ thống con, nối hai khối, di chuyển đường nối,...

Simulink có các thư viện khối chuẩn, được tổ chức thành các khối con theo chức năng. Các khối thông dụng là:

- Sources.

- Sinks.

- Discrete.

Continuous.

- Math Operations.

Tín hiệu và các loại dữ liệu:

Đối với Simulink, khái niệm tín hiệu chỉ nhằm vào dữ liệu xuất hiện ở đầu ra của các khối chức năng trong quá trình mô phỏng. Ta tạm hình dung rằng các tín hiệu (các dữ liệu) đó chạy dọc theo đường nối từ đầu ra của khối chức năng này tới đầu vào của các khối chức năng khác mà không tốn thời gian.

Mỗi tín hiệu thuộc sơ đồ cấu trúc Simulink đều được gán một loại số liệu nhất định, và do đó quyết định đến dung lượng bộ nhớ dành cho một tín hiệu. Simulink hỗ trợ tất cả các loại số liệu của Matlab: double, single, int8, uint8, int16, uint16, int32, uint32, và Boolean.

Thư viện sources và Sinks:

- Sources như là Constant, Step và Ramp, Signal Generator và Pulse Generator,...

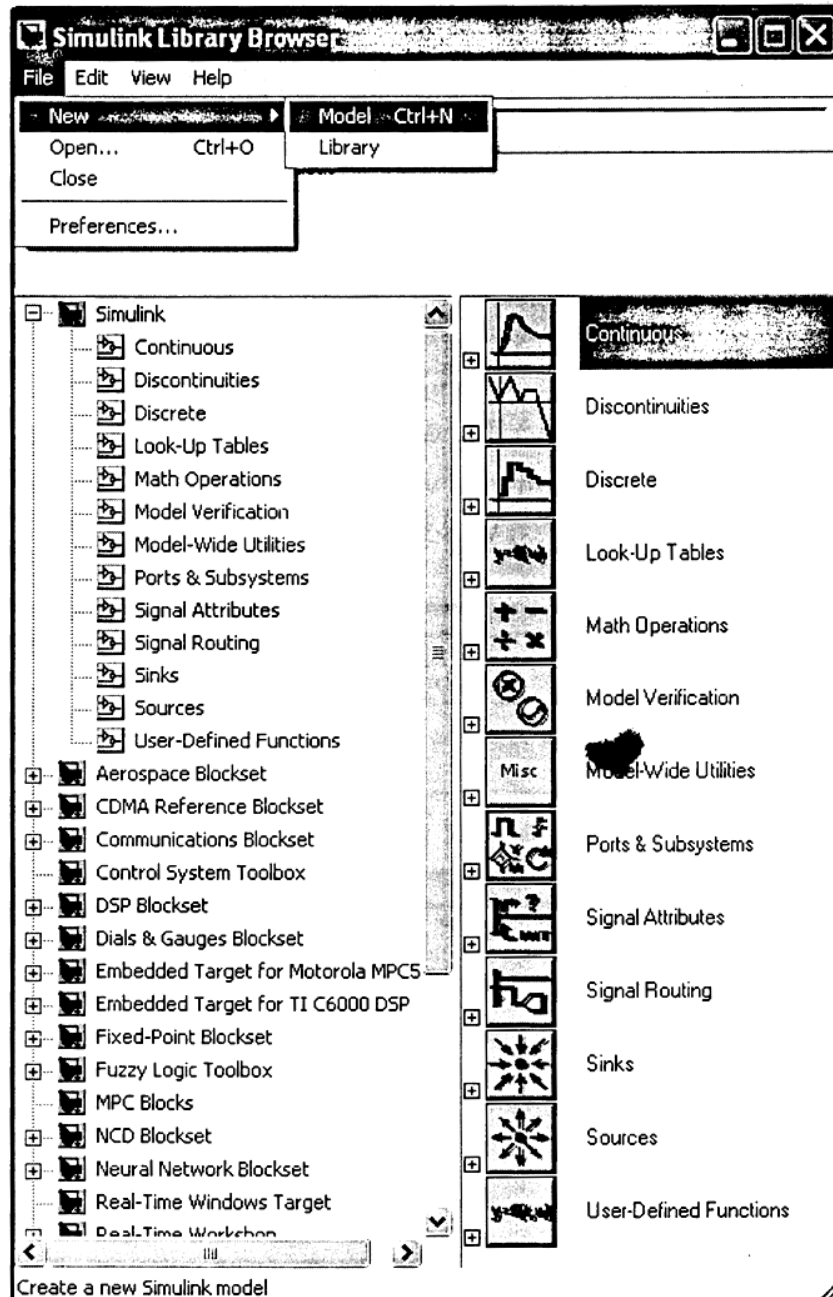
- Sinks: Scope, XYGraph, ToWorkspace,...

Thư viện Math:

- Sum, Product và Dot Product, Math Function, Gain,...

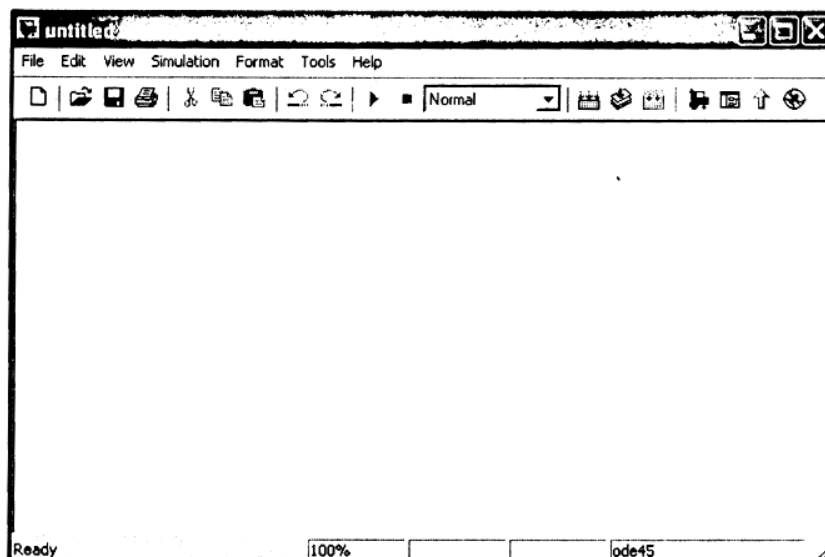
3.2.2. Các bước tiến hành mô phỏng

Từ cửa sổ thư viện khối, ta tạo mới bằng File, New hay mở file có sẵn File, Open. Untitled được tạo ra đầu tiên và có thể đặt tên lại bằng chọn Save As từ menu File.



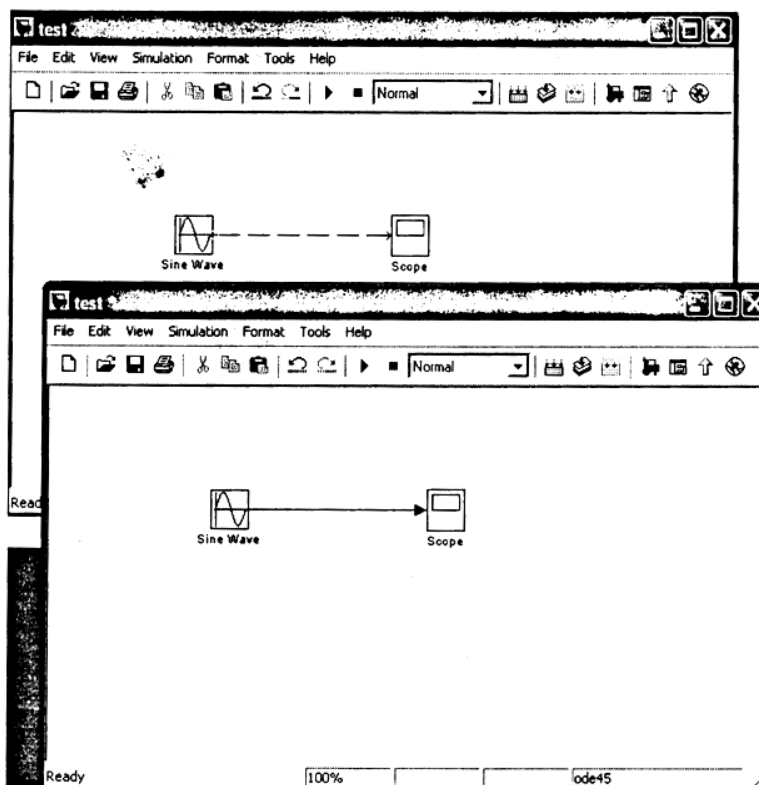
Hình 3.3

Bài 3: Khảo sát và mô phỏng hệ thống tự động dùng Simulink



Hình 3.4

Để tạo mô hình, ta bấm và thả (drag and drop) các khối từ thư viện khối chuẩn vào mô hình. Trong mô hình đơn giản vừa tạo ra ở trên, ta sẽ định vị và sao chép khối Sine Wave (sóng sin) (từ Sources) và Scope (xem dạng sóng) từ Sinks.

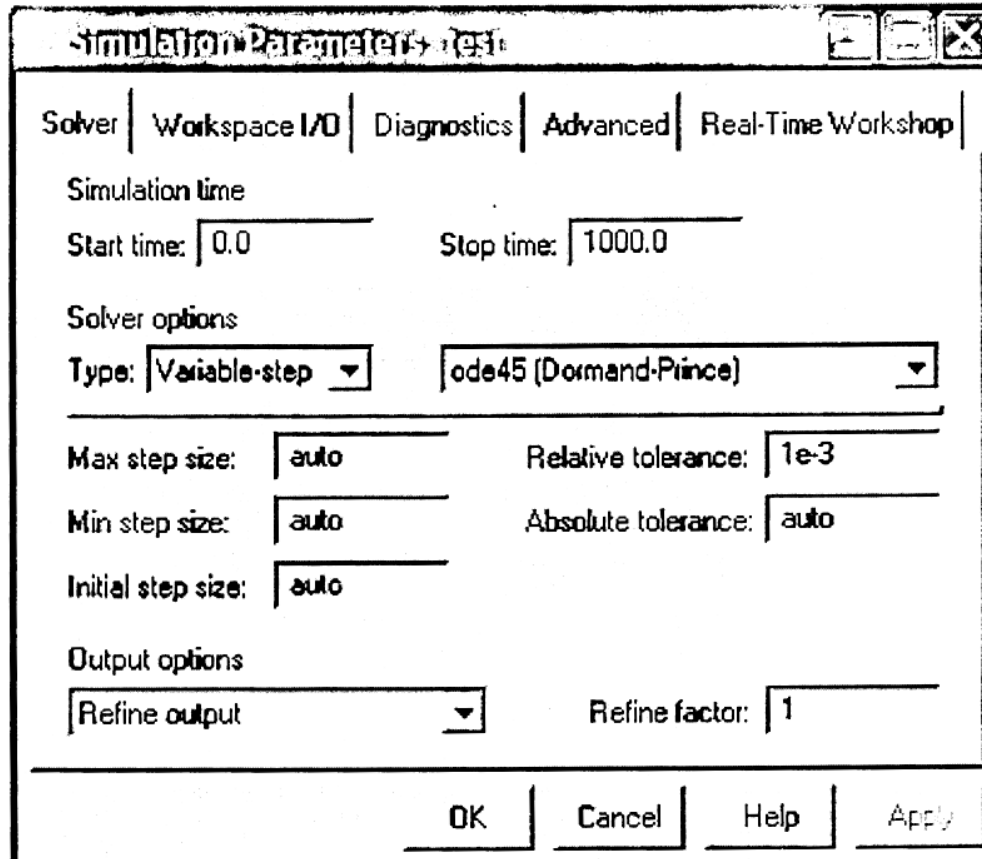


Hình 3.5 và 3.6

Sine Wave 1 là output port, và Scope là Input port.

Chuẩn bị mô phỏng: khai báo tham số và phương pháp tích phân.

Chọn thông số: chọn Simulation Parameters từ menu Simulation.

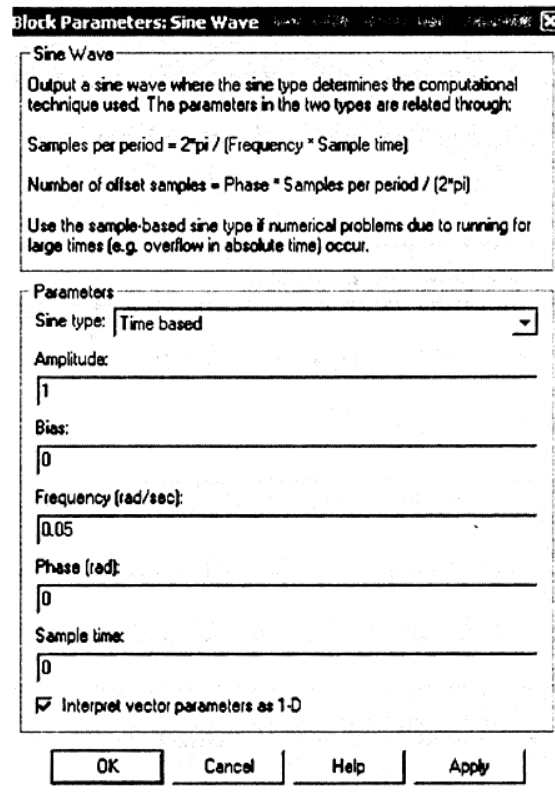


Hình 3.7.

Thời gian mô phỏng bắt đầu từ 0 và kết thúc là 1000. Nhấn vào nút Ok khi làm xong.

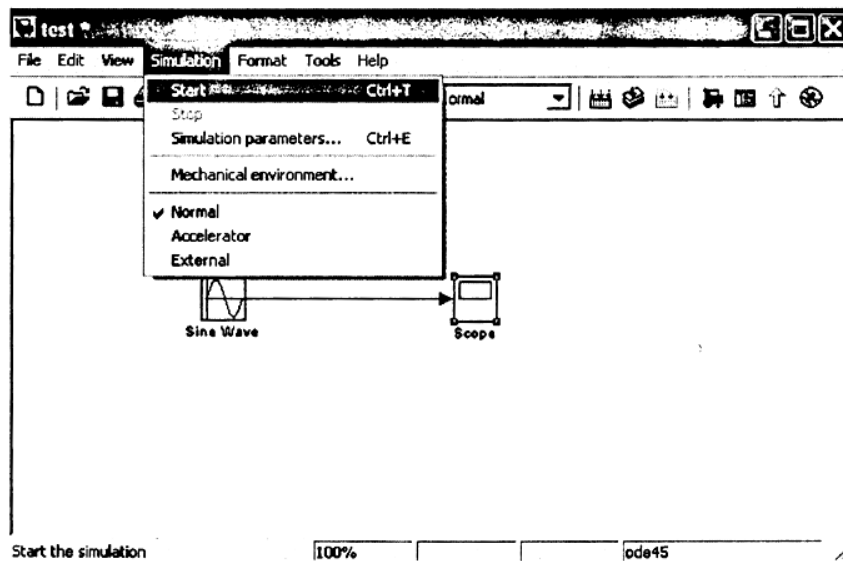
Nhấp đúp chuột vào khối Sine Wave để hiệu chỉnh thông số, chọn tần số là 0.05 rad/sec.

Bài 3: Khảo sát và mô phỏng hệ thống tự động dùng Simulink



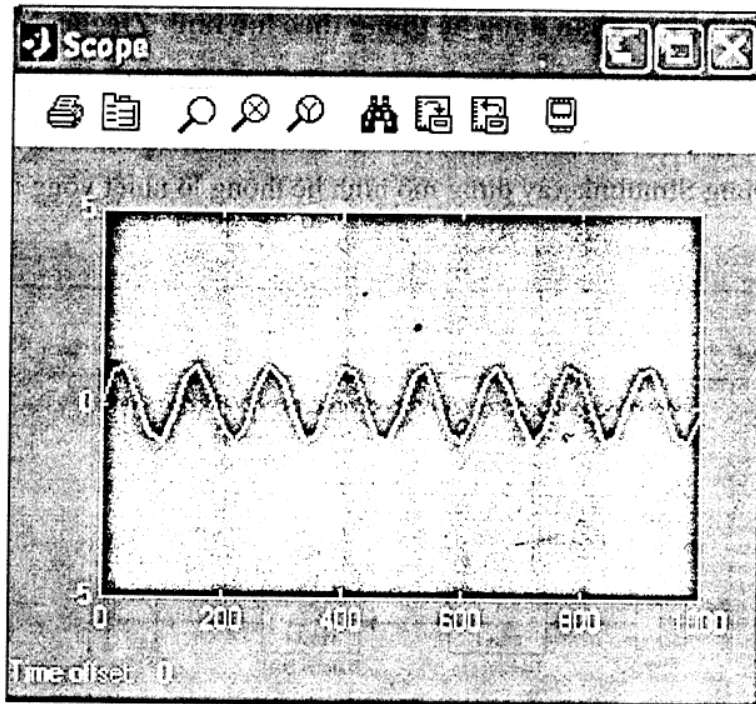
Hình 3.8

Để bắt đầu mô phỏng, chọn Simulation, start.



Hình 3.9.

Kết quả mô phỏng: nhấp chuột vào khối Scope



Hình 3.10

Để kết thúc Simulink chọn Close từ menu File và trở về dấu nhắc lệnh Matlab để tiếp tục với Matlab. Để kết thúc phiên Simulink mà không rời Matlab, ta chỉ cần đóng tất cả cửa sổ Simulink. Để thoát cả Matlab và Simulink, chọn Exit Matlab.

Trong phân tích và thiết kế hệ thống điều khiển hồi tiếp, Simulink cung cấp tập các khối mô hình hàm truyền và không gian trạng thái. Hình vẽ sau minh họa các thư viện khối Simulink của các mô hình liên quan đến điều khiển như là Continuous và Simulink Extras. Thí dụ bộ điều khiển PID nằm trong Simulink Extras.

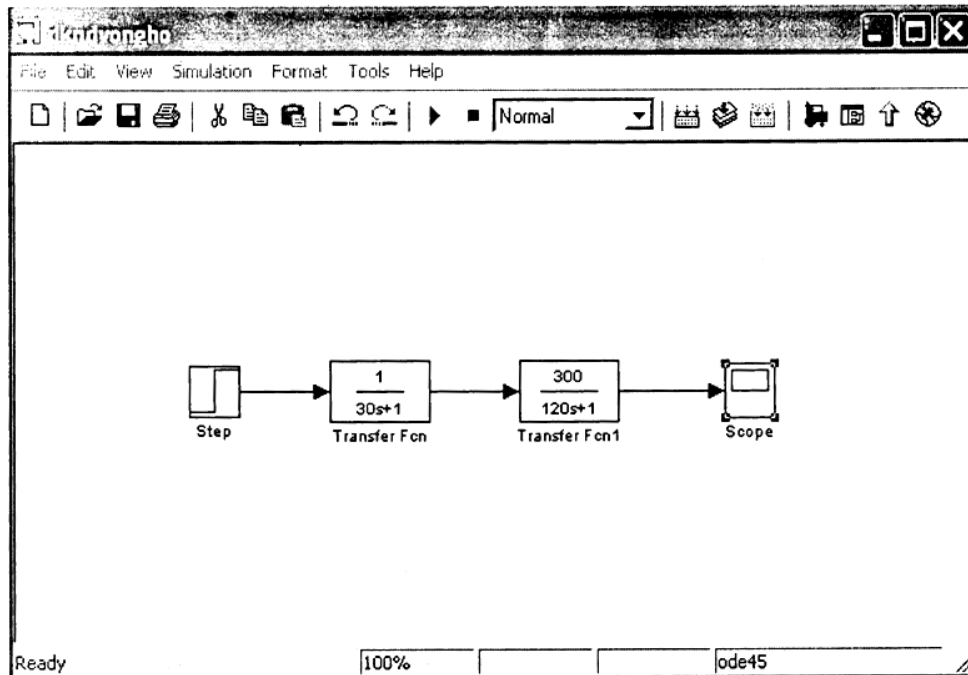
3.3. Thí nghiệm

3.3.1. Khảo sát hệ thống ổn định nhiệt độ

3.3.1.1. Khảo sát hệ hờ, nhận dạng hệ thống theo mô hình Ziegler-Nichols:

Mục đích: Khảo sát khâu quán tính bậc hai, dùng để so sánh với mô hình Ziegler-Nichols (xem bài 4).

Thí nghiệm: Dùng Simulink xây dựng mô hình hệ thống lò nhiệt vòng hờ như sau:



Hình 3.11. Mô hình khảo sát vòng hờ.

Step: là tín hiệu hàm nấc thể hiện phần trăm công suất cung cấp cho lò nhiệt.

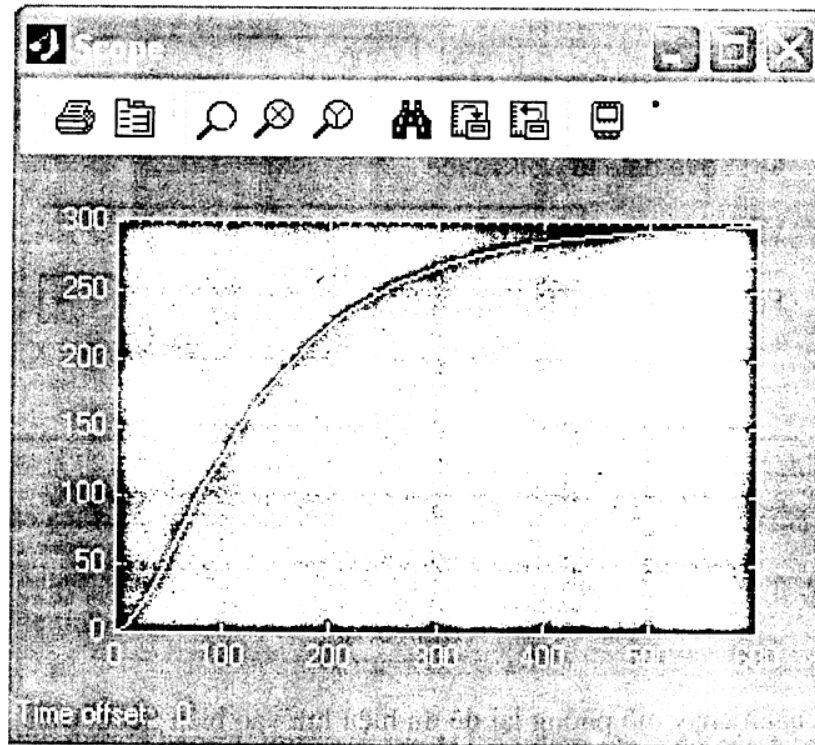
Giá trị của hàm nấc từ 0→1 tương ứng công suất cung cấp 0%→100%.

Transfer Fcn-Transfer Fcn1: mô hình lò nhiệt tuyến tính hóa.

- Chỉnh giá trị của hàm nấc bằng 1 để công suất cung cấp cho lò là 100% (Step time=0, Initial time=0, Final time=1). Chỉnh thời gian mô phỏng Stop time=600 s. Mô phỏng và vẽ quá trình quá độ của hệ thống trên.
- Trên hình vẽ ở câu trên, vẽ tiếp tuyến tại điểm uốn để tính thông số L và T theo như hướng dẫn trong bài thí nghiệm 4. Chỉ rõ các giá trị này trên hình vẽ. So sánh giá trị L, T vừa tìm được với giá trị của mô hình lò nhiệt tuyến tính hóa.

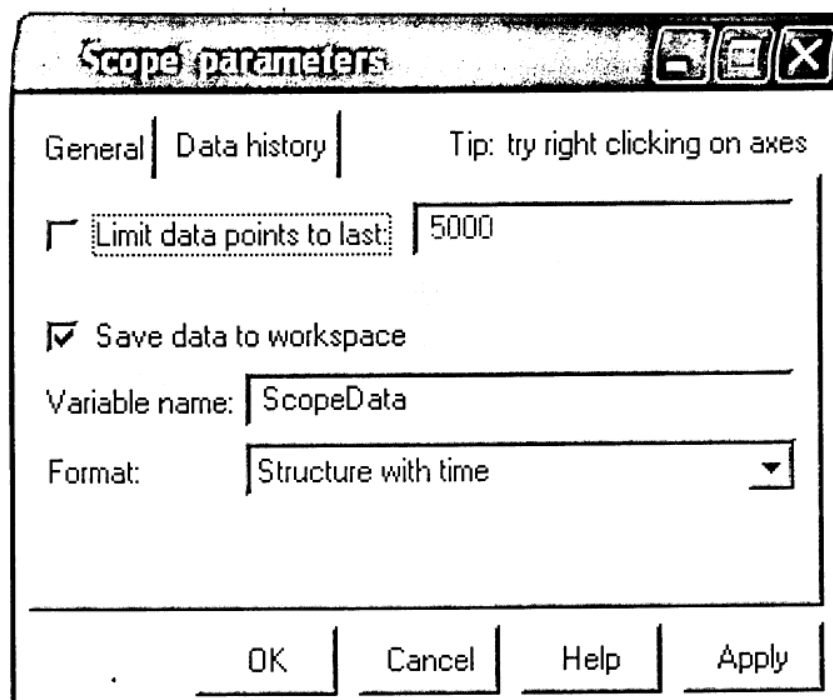
Hướng dẫn:

Sau khi chạy xong mô phỏng, để xem quá trình quá độ của tín hiệu ta nhấp đúp chuột vào khối Scope. Cửa sổ Scope hiện ra như sau:



Hình 3.12

Vì cửa sổ Scope chỉ có thể xem đáp ứng hoặc in trực tiếp ra máy in nhưng không lưu hình vẽ thành file *.bmp được nên ta phải chuyển Scope này sang cửa sổ Figure để lưu. Thực hiện điều này bằng cách nhấp chuột vào ô Parameters. Cửa sổ Parameters hiện ra, nhấp chuột vào trang Data history và tiến hành cài đặt các thông số như hình bên dưới:



Hình 3.13

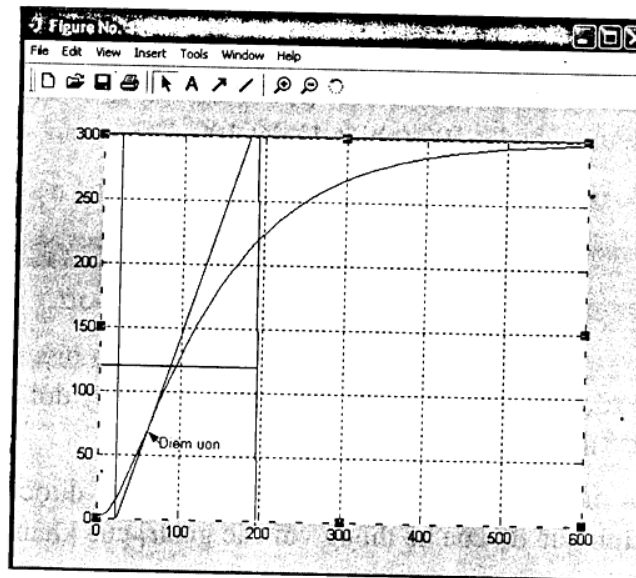
Tiến hành chạy mô phỏng lại để tín hiệu lưu vào biến ScopeData. Chú ý là nếu sau khi khai báo mà không tiến hành mô phỏng lại thì tín hiệu sẽ không lưu vào biến ScopeData mặc dù trên cửa sổ Scope vẫn có hình vẽ.

Sau đó vào cửa sổ Command Window nhập lệnh sau:

```
>> plot(ScopeData.time,ScopeData.signals.values) %vẽ đáp ứng
```

```
>> grid on % kẻ lưới
```

Lúc này cửa sổ Figure hiện ra với hình vẽ giống như hình vẽ ở cửa sổ Scope. Vào menu Insert/Line, Insert/Text để tiến hành kẻ tiếp tuyến và chú thích cho hình vẽ. Kết quả cuối cùng như hình bên dưới.



Hình 3.14

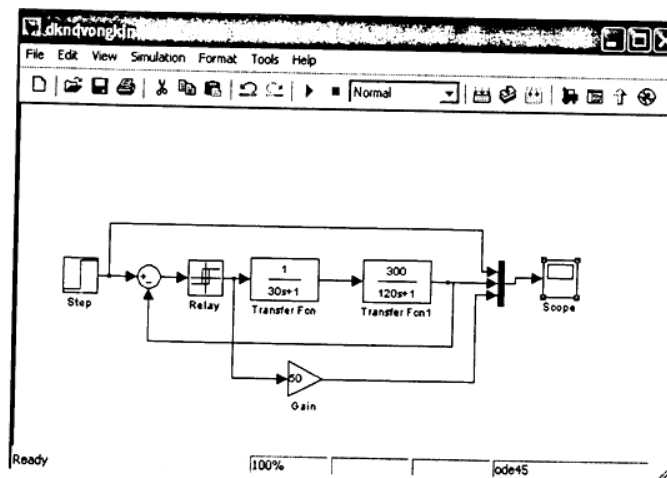
Vào menu [File] → [Export] để lưu thành file *.bmp như ở bài thí nghiệm 1.

3.3.1.2. Khảo sát mô hình hệ thống ổn định nhiệt, điều khiển ON-OFF

Mục đích: Khảo sát hệ thống (HT) ổn định nhiệt độ điều khiển ON-OFF, xét ảnh hưởng của đặc tính rơle có trễ.

Thí nghiệm:

Mô hình được khảo sát tương tự với HT ổn định nhiệt độ của bài 4 (với thông số khác). Sơ đồ khối như sau:



Hình 3.15

Trong đó:

- Lò nhiệt được biểu diễn bằng hai khâu quán tính nối tiếp.
- Đầu vào: u =hàm nấc, đặt 100 (nhiệt độ đặt 100°C).
- Tín hiệu ra đo ở Scope.
- Sum lấy hiệu số tín hiệu đặt và phản hồi.
- Relay là khâu rơle có trễ (Smith trigơ), điều khiển On-off..
- Giá trị độ lợi ở khối Gain =50 dùng để khuếch đại tín hiệu ngõ ra khỏi Relay để quan sát cho rõ. Lưu ý rằng giá trị này không làm thay đổi cấu trúc của hệ thống mà chỉ hỗ trợ việc quan sát tín hiệu.

- a. Chính thời gian mô phỏng Stop time=600s để quan sát được 5 chu kì điều khiển.
Khảo sát quá trình quá độ của hệ thống với các giá trị của khâu role theo bảng sau:

Vùng trễ (Switch on/off point)	Ngõ ra cao (Output when on)	Ngõ ra thấp (Output when off)
+1/-1	1(công suất 100%)	0 (công suất 0%)
+5/-5	1(công suất 100%)	0 (công suất 0%)
+10/-10	1(công suất 100%)	0 (công suất 0%)
+20/-20	1(công suất 100%)	0 (công suất 0%)

- b. Tính sai số ngõ ra so với tín hiệu đặt và thời gian đóng ngắt ứng với các trường hợp của khâu Role ở câu a theo bảng sau:

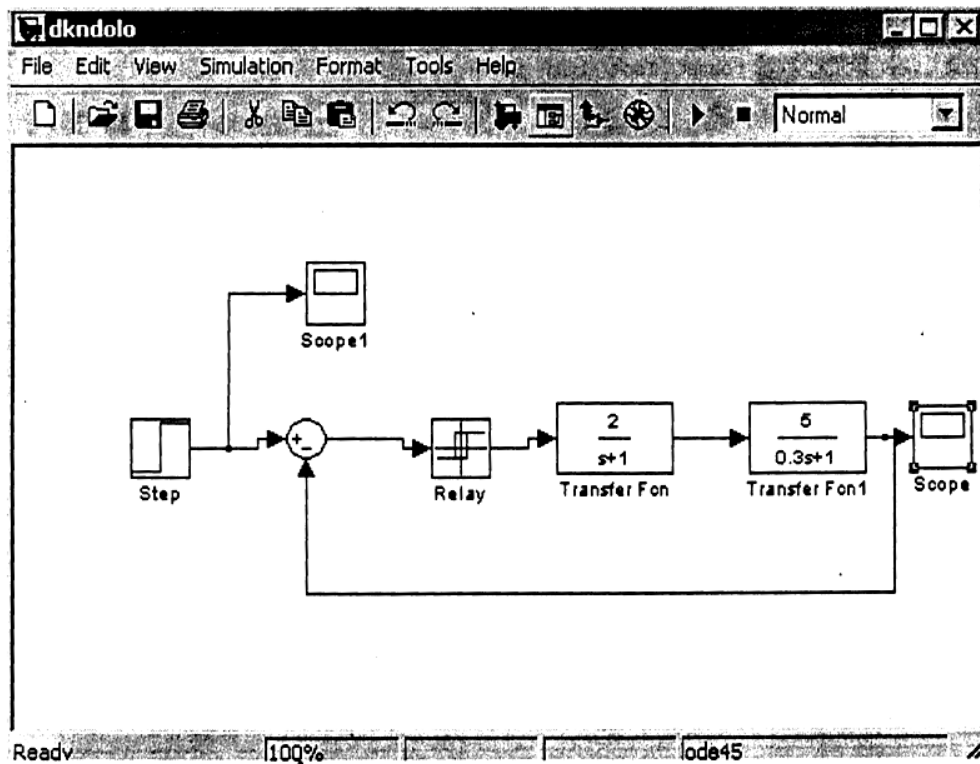
Vùng trễ	$\Delta e1$	$\Delta e2$	Chu kì đóng ngắt(s)
+1/-1			
+5/-5			
+10/-10			
+20/-20			

Nhận xét ảnh hưởng của vùng trễ đến sai số ngõ ra và chu kì đóng ngắt của khâu role (khoảng thời gian ngõ ra khâu rơle thay đổi 1 chu kì).

Bài tập:

1. Khảo sát mô hình hệ thống ổn định nhiệt, điều khiển on-off, xét ảnh hưởng của đặc tính rơle có trễ. Đầu vào $u=1(t)$.

Khảo sát thực hiện lại bài thí nghiệm với sơ đồ mô phỏng như sau:



Hình 3.16

3.3.1.3. Hệ thống điều khiển nhiệt độ hiệu chỉnh Ziegler-Nichols (điều khiển PID):

Mục đích: Sử dụng hiệu chỉnh Ziegler-Nichols (PID) cho HT điều khiển nhiệt độ đang khảo sát. Từ đó so sánh chất lượng của hệ thống ở hai bộ điều khiển PID với bộ điều khiển On/off.

Thí nghiệm:

Bộ hiệu chỉnh có hàm truyền: $PID = K_p + K_i/s + K_d \cdot s = K_p (1 + \frac{1}{T_i s} + T_d s)$

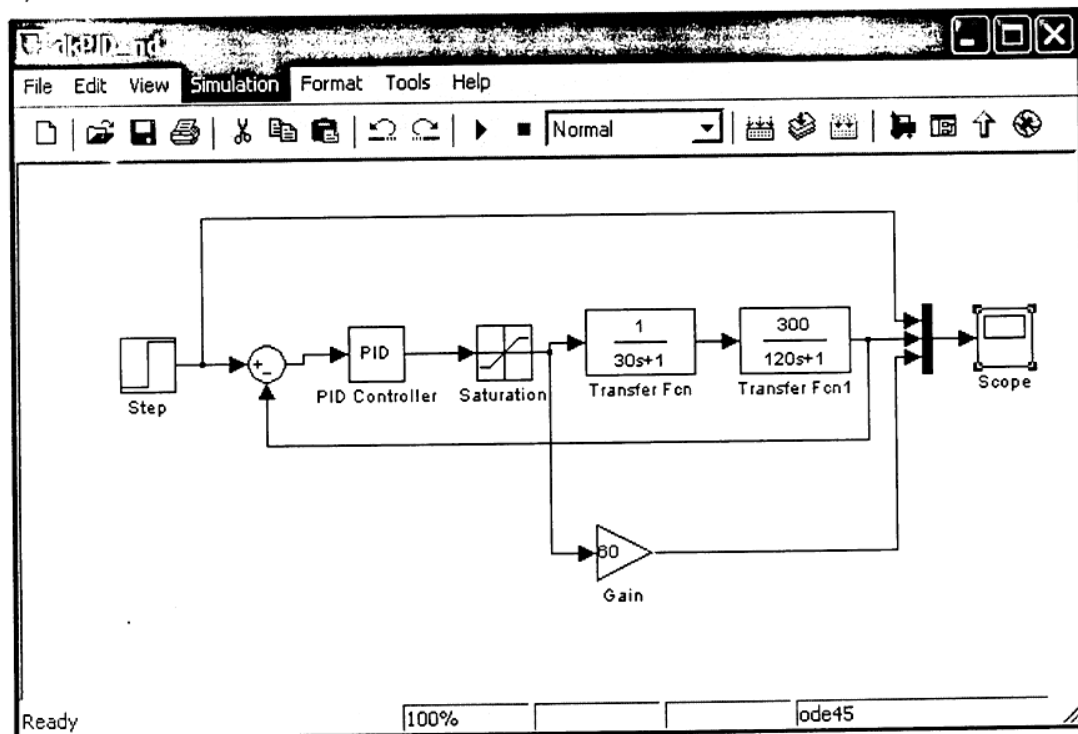
Trong đó: $K_p = (1.2 \cdot T)/L$; $K_i = K_p/(2 \cdot L)$; $K_d = 0.5 \cdot K_p \cdot L$

Hay $K_p = 1.2 \frac{T}{L}$; $T_i = 2L$; $T_d = L/2$

Bài 3: Khảo sát và mô phỏng hệ thống tự động dùng Simulink

Trong đó L, T, K là các giá trị đã tìm được ở phần 3.3.1.1. Chú ý giá trị K đã cho trước ở mô hình hàm truyền lò nhiệt $K=300$.

Xây dựng mô hình hệ thống điều khiển nhiệt độ PID như sau:



1

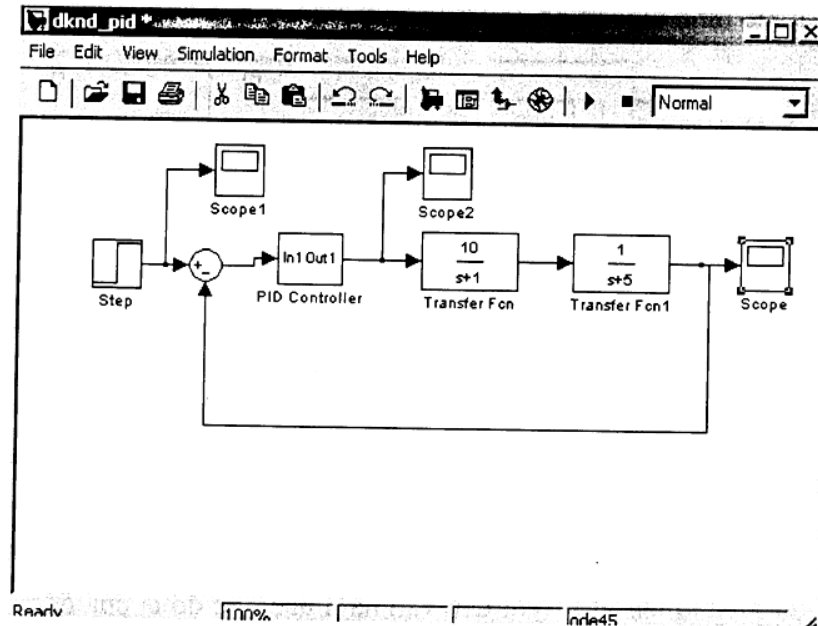
Hình 3.17

Trong đó:

- Tín hiệu đầu vào hàm nấc $u(t)=100$ (tương trưng nhiệt độ đặt 100°C).
- Khâu bảo hoà Saturation có giới hạn upper limit=1, lower limit=0 (tương trưng ngõ ra bộ điều khiển có công suất cung cấp từ 0% đến 100%).
- Bộ điều khiển PID có thông số cần tính toán.
- Transfer Fcn-Transfer Fcn1: mô hình lò nhiệt tuyến tính hóa.

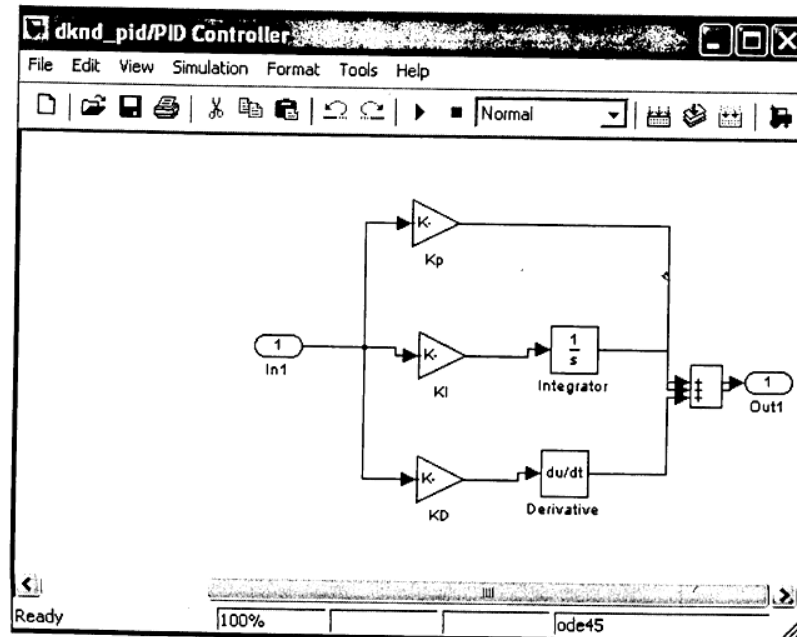
Bài tập: 2. Hệ thống điều khiển nhiệt độ hiệu chỉnh Ziegler-Nichols:

Thực hiện lại bài thí nghiệm theo sơ đồ mô phỏng như sau (H3.18):



Hình 3.18

Bên trong bộ điều khiển PID (H3.19)

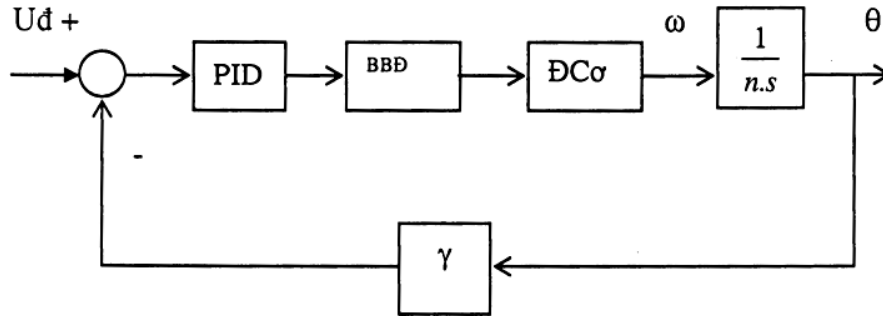


Hình 3.19

3.3.2. Khảo sát mô hình hệ thống điều khiển vị trí và tốc độ động cơ DC

3.3.2.1. HT điều khiển dùng một khâu hiệu chỉnh PID vị trí

Khảo sát mô hình tuyến tính có sơ đồ khối sau:

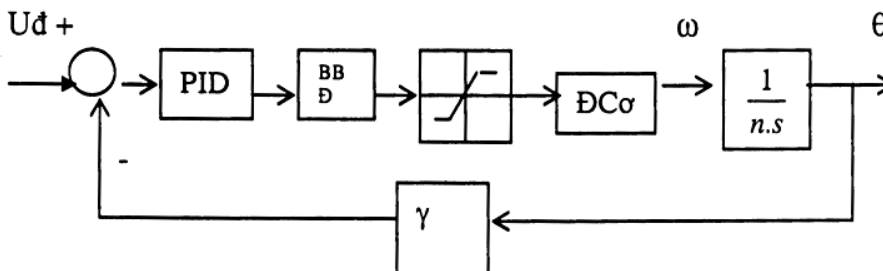


Trong đó: U_d là tín hiệu đặt; PID là bộ hiệu chỉnh; BBĐ là bộ biến đổi cung cấp điện áp thay đổi được cho động cơ một chiều kích từ độc lập ĐCσ; ω, θ là tốc độ trực động cơ và vị trí của tải, quan hệ qua hệ số giảm tốc n ; γ là hệ số phản hồi vị trí.

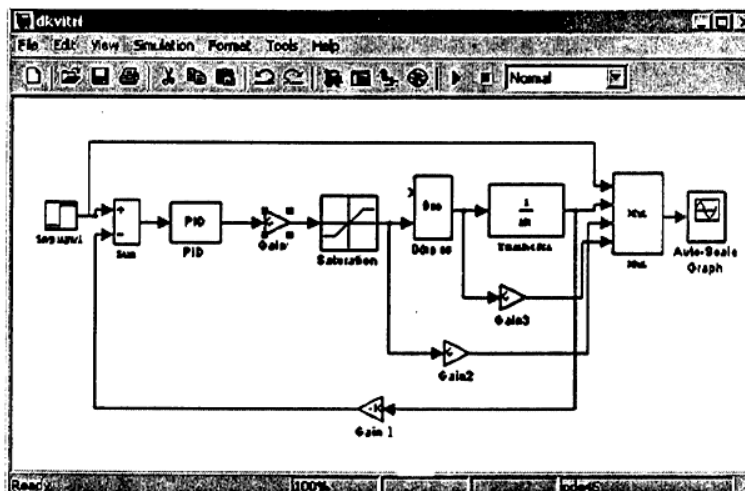
Đề ý, khi điều khiển vị trí với đầu vào hàm nấc, tốc độ ω của động cơ sẽ giảm về zero trong trạng thái xác lập, khi mà trục động cơ đã di chuyển đến vị trí mới. Với đầu vào hàm dốc, tốc độ động cơ sẽ là không đổi khi xác lập để có ngõ ra vị trí tăng (giảm) tuyến tính.

- a. Khảo sát HT điều khiển vị trí hiệu chỉnh PD có tính đến sự bão hòa của bộ biến đổi BBĐ với các ngõ vào khác nhau.

Khi xét đến sự bão hòa của bộ điều khiển (khi điện đặt vào động cơ bị giới hạn) hệ thống có sơ đồ khối như sau:



Xây dựng sơ đồ mô phỏng Simulink dùng một bộ PID.



Hình 3.20

Trong đó:

- gain1: khối phản hồi vị trí, gain1=1.
- gain2, gain3: tỉ lệ các tín hiệu để tiện quan sát trên cùng đồ thị; gain2=0,5 và gain3=0,1.
- Bộ biến đổi (Gain) cung cấp công suất cho động cơ có mô hình là bộ khuếch đại (ở đây $K_{BD}=5$).

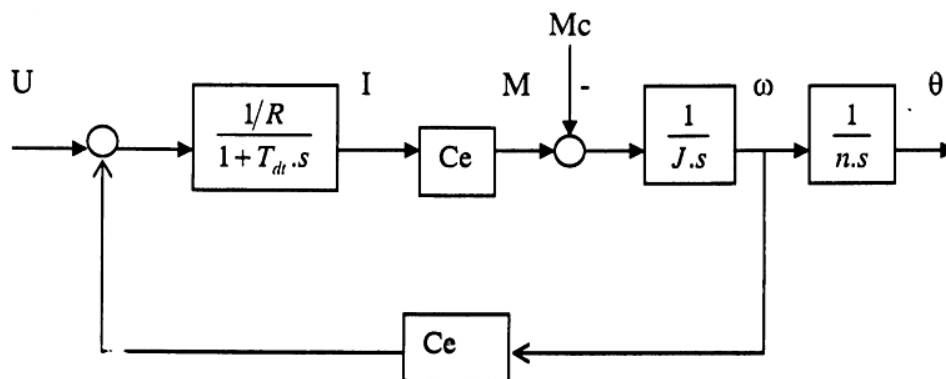
b. Khảo sát HT khi hiệu chỉnh PD và PID với ngõ vào hàm dốc.

Chỉnh thông số PID: $K_i=48,83$; $K_p=29$; $K_d=2,913$

Dùng menu Simulation/start để chạy mô phỏng hệ thống khi hiệu chỉnh PID.

Sinh viên tham khảo bài thí nghiệm 5 để biết rõ phương trình mô tả động cơ DC.

Từ phương trình mô tả động cơ, ta có sơ đồ khối biểu diễn mô hình động cơ như sau:

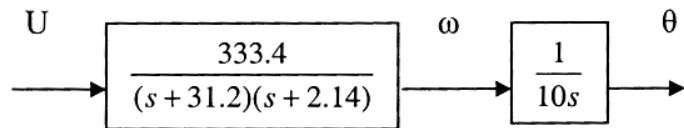


$$T_{dt} = \frac{L}{R}$$

Trong đó:

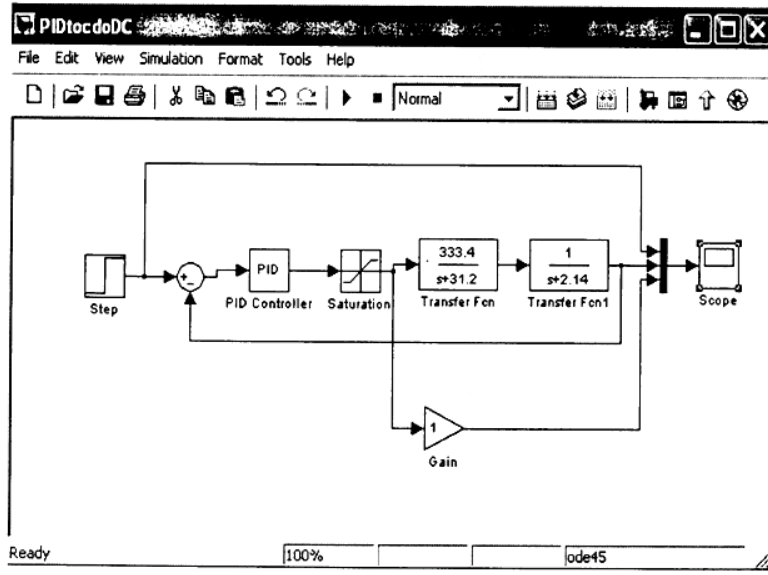
- . Phản ứng: $R=1 \text{ Ohm}$, $L=0.03\text{H} \rightarrow T_{dt}=0.03\text{s}$.
- . Tốc độ xác lập $\omega=100 \text{ rad/sec}$ ở $I=4 \text{ A}$, $U=24\text{V}$
- . C_e : hằng số điện từ, $C_e=(24-1*4)/100=0.2\text{V.s/rad}$.
- . M : moment động cơ, M_c : moment cản.
- U : giá trị điện áp đặt vào động cơ.
- J : moment quán tính của các phần chuyển động, $J=0.02 \text{ kgm/s}^2$.
- ω : tốc độ quay của động cơ (rad/s).
- θ : vị trí góc quay của động cơ (rad).
- N : tỉ số truyền, trong khảo sát: $n=10$.

Với điều kiện không tải $M_c=0$, thu gọn sơ đồ khối thành:



Khảo sát mô hình điều khiển tốc độ động cơ DC:

Tương ứng sơ đồ Simulink:



Hình 3.21

Trong đó:

- Tín hiệu đặt đầu vào hàm nấc $u(t)=100$ (tương trưng tốc độ đặt 100)
 - Khâu bão hòa Saturation có giới hạn là $+30/-30$ (tương trưng điện áp cung cấp cho phản ứng động cơ từ $-30V$ đến $+30V$).
 - Transfer Fcn-Transfer Fcn1 thể hiện mô hình tốc độ động cơ DC.
- a. Chính thời gian mô phỏng Stop time =10s. Thực hiện khảo sát hệ thống với bộ điều khiển P($K_i=0$, $K_d=0$) và tính độ vọt lố, sai số xác lập, thời gian xác lập của ngõ ra theo bảng sau:

K_p	1	10	20	50	100
POT					
E_{xl}					
t_{xl}					

Nhận xét chất lượng của hệ thống thay đổi như thế nào khi K_p thay đổi. Giải thích.

Bài 3: Khảo sát và mô phỏng hệ thống tự động dùng Simulink

b. Thực hiện khảo sát với bộ điều khiển PI ($K_p=2$, $K_d=0$) và tính độ vọt lố, sai số xác lập, thời gian xác lập của ngõ ra theo bảng sau:

K_I	0.1	0.5	0.8	1	2
POT					
E_{xl}					
t_{xl}					

Nhận xét chất lượng của hệ thống thay đổi thế nào khi K_i thay đổi. Giải thích.
So sánh chất lượng bộ điều khiển PI với bộ điều khiển PI.

c. Thực hiện khảo sát hệ thống với bộ điều khiển PID ($K_p=2$, $K_i=2$) và tính độ vọt lố, sai số xác lập, thời gian xác lập của ngõ ra theo bảng sau:

K_D	0.1	0.2	0.5	1	2
POT					
E_{xl}					
t_{xl}					

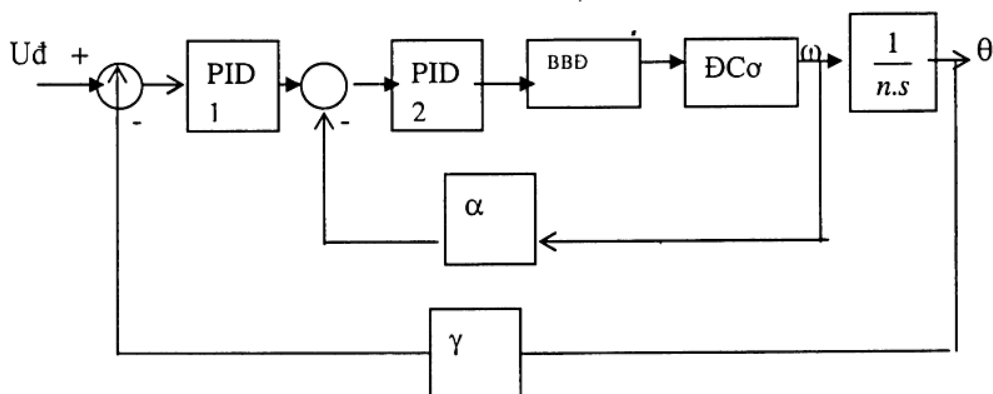
Nhận xét chất lượng của hệ thống thay đổi thế nào khi K_d thay đổi. Giải thích.
So sánh chất lượng bộ điều khiển PID với bộ điều khiển P và PI.

d. Nhận xét ảnh hưởng của các khâu P, I, D lên chất lượng hệ thống.

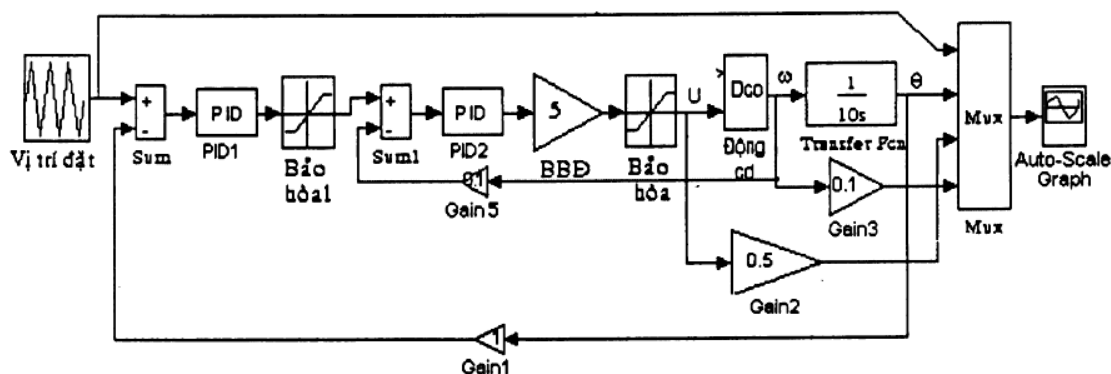
3.3.2.2. Hệ thống điều khiển dùng hai hiệu chỉnh PID vị trí và tốc độ

Mục đích: Khảo sát HT điều khiển vị trí hiệu chỉnh PID với ngõ vào hàm dốc có phản hồi tốc độ, so sánh với hệ thống không có phản hồi tốc độ.

Sơ đồ khối như hình sau: trong đó BBĐ và bộ hiệu chỉnh vòng vị trí PID1 bị bão hoà (giới hạn biên độ); α là hệ số phản hồi tốc độ.



Tương ứng sơ đồ mô phỏng Simulink hệ thống điều khiển dùng hai hiệu chỉnh PID vị trí và tốc độ:



3.4. Kiểm tra đánh giá