

MATLAB**Control System Toolbox & SIMULINK***Ứng dụng để phân tích, thiết kế và mô phỏng các hệ thống tuyến tính*

Trần Đình Khôi Quốc, BM. Tự động hóa. Email : tdkquoc@dng.vnn.vn

GIỚI THIỆU

MATLAB, tên viết tắt của từ tiếng Anh **MA**TriX **LAB**oratory, là một môi trường mạnh dành cho các tính toán khoa học. Nó tích hợp các phép tính ma trận và phân tích số dựa trên các hàm cơ bản. Hơn nữa, cấu trúc đồ họa hướng đối tượng của Matlab cho phép tạo ra các hình vẽ chất lượng cao. Ngày nay, Matlab trở thành một ngôn ngữ « chuẩn » được sử dụng rộng rãi trong nhiều ngành và nhiều quốc gia trên thế giới.

Về mặt cấu trúc, Matlab gồm một cửa sổ chính và rất nhiều hàm viết sẵn khác nhau. Các hàm trên cùng lĩnh vực ứng dụng được xếp chung vào một thư viện, điều này giúp người sử dụng dễ dàng tìm được hàm cần quan tâm. Có thể kể ra một số thư viện trong Matlab như sau :

- Control System (dành cho điều khiển tự động)
- Financial Toolbox (lĩnh vực kinh tế)
- Fuzzy Logic (điều khiển mờ)
- Signal Processing (xử lý tín hiệu)
- Statistics (toán học và thống kê)
- Symbolic (tính toán theo biểu thức)
- System Identification (nhận dạng)
- ...

Một tính chất rất mạnh của Matlab là nó có thể liên kết với các ngôn ngữ khác. Matlab có thể gọi các hàm viết bằng ngôn ngữ Fortran, C hay C++, và ngược lại các hàm viết trong Matlab có thể được gọi từ các ngôn ngữ này...

Các bạn có thể xem phần Help trong Matlab để tham khảo cách sử dụng và ví dụ của từng lệnh, hoặc download (miễn phí) các file help dạng *.pdf tại trang Web của Matlab ở địa chỉ <http://www.mathworks.com>

1 Control System Toolbox

Control System Toolbox là một thư viện của Matlab dùng trong lĩnh vực điều khiển tự động. Cùng với các lệnh của Matlab, tập lệnh của Control System Toolbox sẽ giúp ta thiết kế, phân tích và đánh giá các chỉ tiêu chất lượng của một hệ thống tuyến tính.

1.1 Định nghĩa một hệ thống tuyến tính**1.1.1 Định nghĩa bằng hàm truyền**

Hệ thống một tín hiệu vào/ra

Câu lệnh: $sys=tf(num,den,T)$

- *num*: vectơ chứa các hệ số của đa thức ở tử số, bậc từ cao đến thấp theo toán tử Laplace (hệ liên tục) hoặc theo toán tử z (hệ gián đoạn)
- *den*: vectơ chứa các hệ số của đa thức ở mẫu số, bậc từ cao đến thấp
- *T*: chu kỳ lấy mẫu, chỉ dùng cho hệ gián đoạn (tính bằng s)

Ví dụ:

Định nghĩa một hàm truyền trong Matlab

$$F(p) = 3 \frac{p+2}{p^2+2p+4}$$

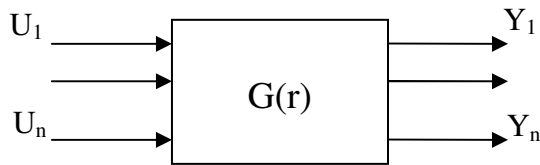
$$\text{num}=3*[1 \ 2]; \text{den}=[1 \ 2 \ 4]; \text{sys1}=\text{tf}(\text{num}, \text{den});$$

$$F(z) = 2,1 * \frac{z-0,6}{z^2-0,56z+0,4}$$

$$\text{num}=2.1*[1 \ -0.6]; \text{den}=[1 \ -0.56];$$

$$T=0.5; \text{sys2}=\text{tf}(\text{num}, \text{den}, T)$$

Hệ thống nhiều tín hiệu vào/ra



$$G(r) = \begin{bmatrix} G_{11}(r) & G_{12}(r) & \dots & G_{1n}(r) \\ G_{21}(r) & G_{22}(r) & & G_{2n}(r) \\ \dots & & & \\ G_{p1}(r) & G_{p2}(r) & & G_{pn}(r) \end{bmatrix}$$

Câu lệnh :

$$G11=\text{tf}(\text{num11}, \text{den11}, T); G12=\text{tf}(\text{num12}, \text{den12}, T); \dots; G1n=\text{tf}(\text{num1n}, \text{den1n}, T); \\ G21=\text{tf}(\text{num21}, \text{den21}, T); G22=\text{tf}(\text{num22}, \text{den22}, T); \dots; G2n=\text{tf}(\text{num2n}, \text{den2n}, T);$$

$$Gp1=\text{tf}(\text{nump1}, \text{denp1}, T); G12=\text{tf}(\text{nump2}, \text{denp2}, T); \dots; Gpn=\text{tf}(\text{numpn}, \text{denpn}, T); \\ \text{sys}=[G11, G12, \dots, G1n; G21; G22; \dots; G2n; \dots; Gp1, Gp2, \dots, Gpn];$$

1.1.2 Định nghĩa bằng zero và cực

Hệ thống một tín hiệu vào/ra

Câu lệnh: $\text{sys}=\text{zpk}(Z, P, K, T)$

- Z, P là các vectơ hàng chứa danh sách các điểm zerô và cực của hệ thống.
- K là hệ số khuếch đại

Chú ý: nếu hệ thống không có điểm zerô (cực) thì ta đặt là []

Ví dụ:

$$F(p) = \frac{p+2}{p(p+5)}$$

$$Z=-2; P=[0 \ -5]; K=1; \text{sys}=\text{zpk}(Z, P, K);$$

Hệ thống nhiều tín hiệu vào/ra

Câu lệnh :

$$G11=\text{zpk}(Z11, P11, T); G12=\text{zpk}(Z12, P12, T); \dots; G1n=\text{zpk}(Z1n, P1n, T); \\ G21=\text{zpk}(Z21, P21, T); G22=\text{zpk}(Z22, P22, T); \dots; G2n=\text{zpk}(Z2n, P2n, T);$$

$$Gp1=\text{zpk}(Zp1, Pp1, T); G12=\text{zpk}(Zp2, Pp2, T); \dots; Gpn=\text{zpk}(Zpn, Ppn, T); \\ \text{sys}=[G11, G12, \dots, G1n; G21; G22; \dots; G2n; \dots; Gp1, Gp2, \dots, Gpn];$$

1.1.3 Phương trình trạng thái

Câu lệnh: $\text{sys}=\text{ss}(A, B, C, D, T)$

- A, B, C, D là các ma trận trạng thái định nghĩa hệ thống
- T là chu kỳ lấy mẫu.

1.1.4 Chuyển đổi giữa các dạng biểu diễn

- Chuyển từ phương trình trạng thái sang hàm truyền
 $[num,den] = ss2tf(A,B,C,D)$
- Chuyển từ dạng zero/cực sang hàm truyền
 $[num,den] = zp2tf(Z,P,K)$
- Chuyển từ hàm truyền sang phương trình trạng thái
 $[A,B,C,D]=tf2ss(num,den)$

1.1.5 Chuyển đổi giữa hệ liên tục và gián đoạn

Số hóa một hệ thống liên tục

Câu lệnh: `sys_dis=c2d(sys,T,method)`

- `sys, sys_dis` hệ thống liên tục và hệ thống gián đoạn tương ứng
- `Ts` thời gian lấy mẫu
- `method` phương pháp lấy mẫu: 'zoh' lấy mẫu bậc 0, 'foh' lấy mẫu bậc 1, 'tustin' phương pháp Tustin...

Ví dụ: chuyển một khâu liên tục có hàm truyền $G(p) = \frac{2}{0.5p+1}$ sang khâu gián đoạn bằng

phương pháp giữ mẫu bậc 0, chu kỳ lấy mẫu $T=0.01s$

`num=2`

`den=[0.5 1]`

`sysc=tf(num,den)`

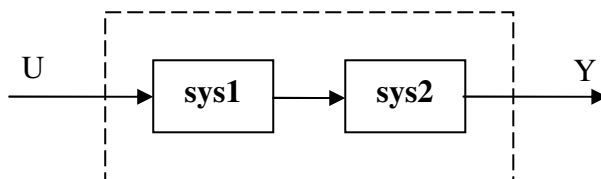
`sysd=c2d(sysc,0.01,'zoh')`

Hệ liên tục tương đương của một hệ thống gián đoạn

Câu lệnh: `sys=d2c(sys_dis,method)`

1.2 Biến đổi sơ đồ tương đương

1.2.1 Mắc nối tiếp



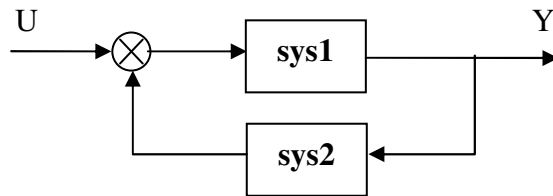
Câu lệnh: `sys=series(sys1,sys2)`

1.2.2 Mắc song song

Câu lệnh: `sys=parallel(sys1,sys2)`

1.2.3 Mắc phản hồi

Câu lệnh: `sys=feedback(sys1,sys2,sign)`



$sign = +1$ nếu phản hồi dương và $sign = -1$ (hoặc không có $sign$) nếu phản hồi âm.

1.3 Phân tích hệ thống

1.3.1 Trong miền thời gian

Hàm quá độ $h(t)$

Câu lệnh: `step(sys)`

Vẽ hàm quá độ của hệ thống tuyến tính sys . Khoảng thời gian vẽ và bước thời gian do Matlab tự chọn.

Một số trường hợp khác

- `step(sys, t_end)`: vẽ hàm quá độ từ thời điểm $t=0$ đến thời điểm t_{end} .
- `step(sys, T)`: vẽ hàm quá độ trong khoảng thời gian T . T được định nghĩa như sau $T=T_i:dt:T_f$. Đối với hệ liên tục, dt là bước vẽ, đối với hệ gián đoạn, $dt=Ts$ là chu kỳ lấy mẫu.
- `step(sys1, sys2, sys3, ...)`: vẽ hàm $h(t)$ cho nhiều hệ thống đồng thời.
- `[y, t]=step(sys)`: tính đáp ứng $h(t)$ và lưu vào các biến y và t tương ứng

Hàm trọng lượng $\omega(t)$

Câu lệnh: `impulse(sys)`

1.3.2 Trong miền tần số

Đặc tính bode

Câu lệnh: `bode(sys)`

Vẽ đặc tính tần số Bode của hệ thống tuyến tính sys . Dải tần số vẽ do Matlab tự chọn.

Một số trường hợp khác

- `bode(sys, {w_start, w_end})`: vẽ đặc tính bode từ tần số w_{start} đến tần số w_{end} .
- `bode(sys, w)` vẽ đặc tính bode theo vectơ tần số w . Vectơ tần số w được định nghĩa bằng hàm `logspace`. Ví dụ: `w=logspace(-2, 2, 100)` định nghĩa vectơ w gồm 100 điểm, từ tần số 10^{-2} đến 10^2 .
- `bode(sys1, sys2, sys3, ...)` vẽ đặc tính bode của nhiều hệ thống đồng thời.
- `[mag, phi, w]=bode(sys, ...)` lưu tất cả các điểm tính toán của đặc tính bode vào vectơ mag , phi ứng với tần số w tương ứng.

Chú ý: Đối với hệ thống gián đoạn, dải tần số để vẽ phải thỏa mãn định lý Shannon.

Đặc tính Nyquist

Câu lệnh: `nyquist(sys)`
`nyquist(sys, {w_start, w_end})`

```
nyquist(sys,w)
nyquist(sys1,sys2,sys3,...,w)
[real,ima,w]=nyquist(sys,...)
```

Đặc tính Nichols

```
Câu lệnh: nichols(sys)
nichols(sys,{w_start,w_end})
nichols(sys,w)
nichols(sys1, sys2, sys3,...,w)
[mag,phi,w]=nichols(sys,...)
```

Tính toán $|G(\omega)|$, $\arg[G(\omega)]$ và vẽ trong mặt phẳng Black.

Ví dụ: Vẽ các đặc tính tần số của hệ thống sau

$$G(p) = \frac{\omega_0^2}{p^2 + 2\xi\omega_0 p + \omega_0^2} \quad \text{với } \omega_0=1\text{rad/s và } \xi=0,5$$

```
w0=1 ;xi=0.5 ;num=w0^2 ;den=[1 2*xi*w0^2 w0^2] ;G=tf(num,den);
```

```
w=logspace(-2,2,100) ;
```

```
bode(G,w) ; % vẽ đặc tính bode trong dải tần số w
```

```
nichols(G) ; % vẽ đặc tính nichols trong dải tần số tự chọn của Matlab
```

```
nyquist(G) ; % vẽ đặc tính nyquist
```

1.3.3 Một số hàm để phân tích

Hàm *margin*

- *margin(sys)* vẽ đặc tính Bode của hệ thống SISO và chỉ ra độ dự trữ biên độ, độ dự trữ pha tại các tần số tương ứng.
- *[delta_L,delta_phi,w_L,w_phi]=margin(sys)* tính và lưu độ dự trữ biên độ vào biến *delta_L* tại tần số *w_L*, lưu độ dự trữ về pha vào biến *delta_phi* tại tần số *w_phi*.

Hàm *pole*

vec_pol=pole(sys) tính các điểm cực của hệ thống và lưu vào biến *vec_pol*.

Hàm *tzero*

vec_zer=tzero(sys) tính các điểm zero của hệ thống và lưu vào biến *vec_zer*.

Hàm *pzmap*

- *[vec_pol,vec_zer]=pzmap(sys)* tính các điểm cực và zero của hệ thống và lưu vào các biến tương ứng.
- *pzmap(sys)* tính các điểm cực, zero và biểu diễn trên mặt phẳng phức.

Hàm *dcgain*

G0=dcgain(sys) tính hệ số khuếch đại tĩnh của hệ thống và lưu vào biến *G0*.

1.3.4 Một số hàm đặc biệt trong không gian trạng thái

Hàm *ctrl*

```
Câu lệnh: C_com=ctrl(A,B)
C_com=ctrl(sys)
```

Tính ma trận “điều khiển được” C của một hệ thống. Ma trận C được định nghĩa như sau:

$$C=[B \ AB \ A^2B \ \dots \ A^{n-1}B] \text{ với } A \in \mathbb{R}^{n \times n}$$

Hàm *obsv*

Câu lệnh: $O_obs=obsv(A,C)$
 $O_obs=obsv(sys)$

Tính ma trận “quan sát được” O của một hệ thống. Ma trận O được định nghĩa như sau:

$$O=[C \ CA \ CA^2 \ \dots \ CA^{n-1}]$$

Hàm *ctrbf*

Câu lệnh: $[Ab,Bb,Cb,T,k]=ctrbf(A,B,C)$

Chuyển về dạng chuẩn (canonique) “điều khiển được” của một hệ thống biểu diễn dưới dạng phương trình trạng thái.

Trong đó: $A_b=TAT^{-1}$, $B_b=TB$, $C_b=CT^{-1}$, T là ma trận chuyển đổi.

Hàm *obsvf*

Câu lệnh: $[Ab,Bb,Cb,T,k]=obsvf(A,B,C)$

Chuyển về dạng chuẩn “quan sát được” của một hệ thống biểu diễn dưới dạng phương trình trạng thái.

Trong đó: $A_b=TAT^{-1}$, $B_b=TB$, $C_b=CT^{-1}$, T là ma trận chuyển đổi.

1.4 Ví dụ tổng hợp

Cho một hệ thống kín phản hồi -1, trong đó hàm truyền của hệ hở là

$$G(p) = \frac{K}{p(1+\tau p)} * \frac{\omega_0^2}{p^2 + 2\xi\omega_0 p + \omega_0^2} \quad \text{với } K=1, \tau=10s, \omega_0=1\text{rad/s và } \xi=0.5$$

1. Vẽ đặc tính tần số Nyquist. Chứng tỏ rằng hệ kín không ổn định.
2. Vẽ đáp ứng quá độ của hệ kín.
3. Để hệ thống ổn định, người ta hiệu chỉnh hệ số khuếch đại $K=0.111$. Xác định tần số cắt, độ dự trữ biên độ và độ dự trữ về pha của hệ thống trong trường hợp này.
4. Xác định các thông số quá độ (thời gian quá độ lớn nhất T_{\max} , độ quá điều chỉnh lớn nhất σ_{\max}) của hệ thống đã hiệu chỉnh.

THỰC HIỆN

Câu 1

```
>>K=1;to=10;w0=1;xi=0.5;
>>num1=K;den1=[to 1 0];
>>num2=w0^2;den2=[1 2*xi*w0 w0^2];
>>G=tf(num1,den1)*tf(num2,den2)
```

Transfer function:

1

 $10 s^4 + 11 s^3 + 11 s^2 + s$

```
>>w=logspace(-3,2,100); % tạo vector tần số để vẽ các đặc tính tần số
```

```
>>nyquist(G,w);
```

Đặc tính được biểu diễn trên hình 1.1

Để xét tính ổn định của hệ kín dùng tiêu chuẩn Nyquist, trước tiên ta xét tính ổn định của hệ hở. Nghiệm của phương trình đặc tính của hệ hở được xác định :

```
>>pole(G)
```

```
ans =
```

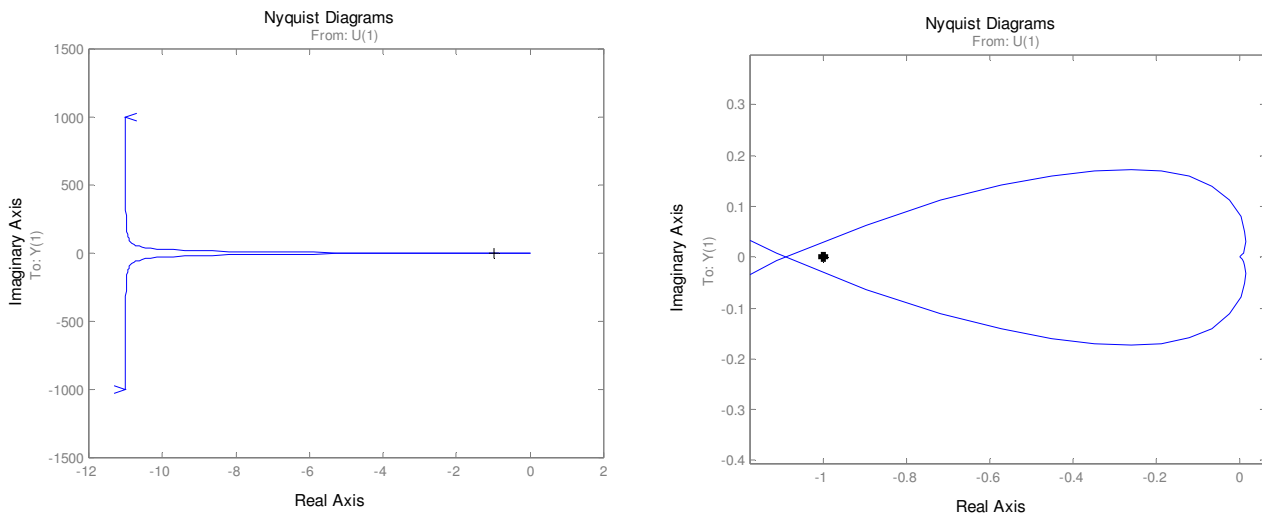
```
0
```

```
-0.5000 + 0.8660i
```

```
-0.5000 - 0.8660i
```

```
-0.1000
```

Hệ hở có 1 nghiệm bằng 0 nên ở biên giới ổn định.



Hình 1.1 : Đặc tính tần số Nyquist của hệ hở

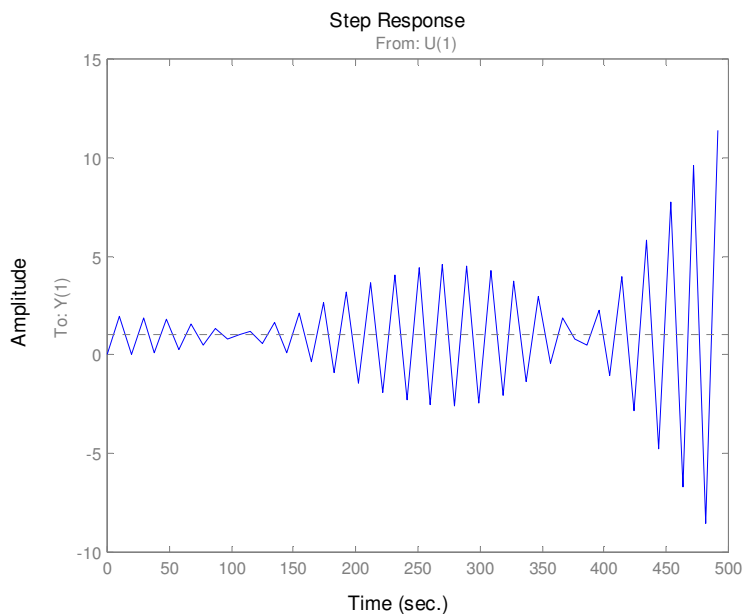
Quan sát đặc tính tần số Nyquist của hệ hở trên hình 1.1 (phần zoom bên phải), ta thấy đặc tính Nyquist bao điểm $(-1, j0)$, và do hệ hở ở biên giới ổn định nên theo tiêu chuẩn Nyquist, **hệ thống kín sẽ không ổn định.**

Câu 2

```
>>G_loop=feedback(G,1,-1);
```

```
% hàm truyền hệ kín
```

```
>>step(G_loop);
```



Hình 1.2 :
Đáp ứng quá độ hệ kín

Câu 3

```
>>K=0.111 ;num1=K ; % thay đổi hệ số khuếch đại K
```

```
>>GK=tf(num1,den1)*tf(num2,den2)
```

```
Transfer function:
```

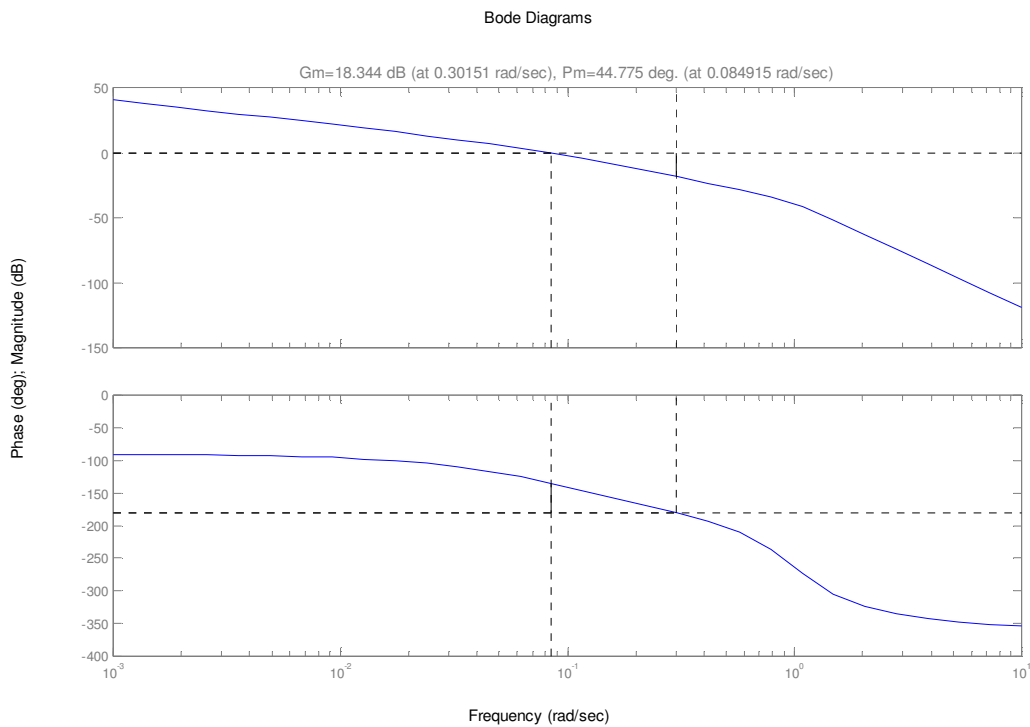
```
0.111
```

```
-----
10 s^4 + 11 s^3 + 11 s^2 + s
```

```
>>margin(GK)
```

Đặc tính tần số Bode của hệ hở đã hiệu chỉnh được biểu diễn trên hình 1.3. Từ đặc tính này, ta có thể xác định được

$$\Delta L=18.34\text{dB} ; \Delta\varphi = 44.78^\circ ; \omega_c=0.085\text{rad/s}$$

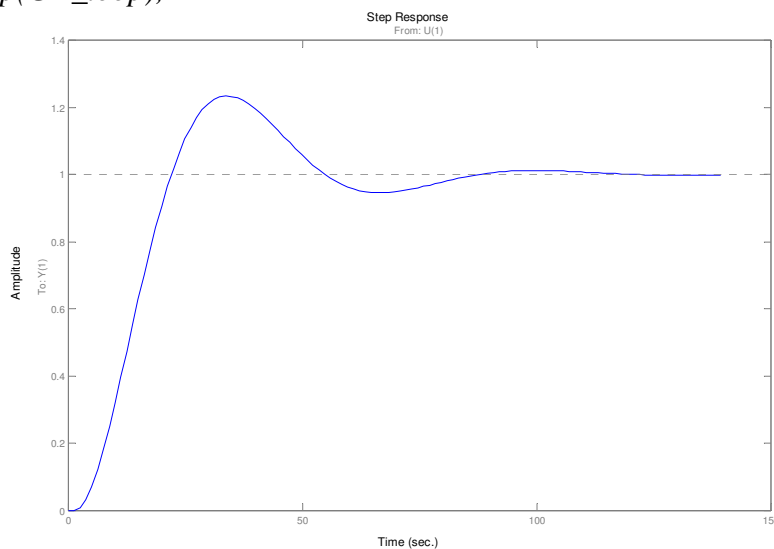


Hình 1.3 : Đặc tính tần số Bode của hệ hở đã hiệu chỉnh

Câu 4

```
>>GK_loop=feedback(GK,1,-1) ;
```

```
>>step(GK_loop);
```



Hình 1.4
Đáp ứng quá độ hệ kín đã hiệu chỉnh

Sử dụng con trỏ chuột và kích vào các điểm cần tìm trên đặc tính, ta xác định được

$$\sigma_{\max}=23\%; T_{\max}= 70.7s$$

2 SIMULINK

Simulink được tích hợp vào Matlab (vào khoảng đầu những năm 1990) như một công cụ để mô phỏng hệ thống, giúp người sử dụng phân tích và tổng hợp hệ thống một cách trực quan. Trong Simulink, hệ thống không được mô tả dưới dạng dòng lệnh theo kiểu truyền thống mà ở dưới dạng sơ đồ khối. Với dạng sơ đồ khối này, ta có thể quan sát các đáp ứng thời gian của hệ thống với nhiều tín hiệu vào khác nhau như : tín hiệu bậc thang, tín hiệu sinus, xung chữ nhật, tín hiệu ngẫu nhiên... bằng cách thực hiện mô phỏng. Kết quả mô phỏng có thể được xem theo thời gian thực trên các Oscilloscope trong môi trường Simulink, hay trong môi trường Matlab.

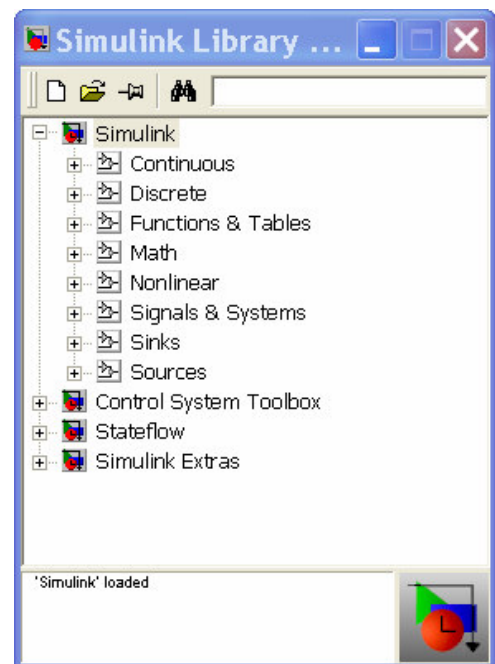
Simulink hoàn toàn tương thích với Matlab, nhưng nó là một dao diện đồ họa. Vì vậy tất cả các hàm trong Matlab đều có thể truy cập được từ Simulink, ngay cả các hàm do người sử dụng tạo ra. Ngược lại, các kết quả tìm được trong Simulink đều có thể được sử dụng và khai thác trong môi trường Matlab.

Cuối cùng, Simulink cho phép người sử dụng khả năng tạo ra một thư viện khối riêng. Ví dụ, nếu bạn muốn làm việc trong lĩnh vực điều khiển các máy điện, bạn có thể tạo ra một thư viện riêng chứa các mô hình máy điện... Như vậy, với công cụ Simulink, ta có thể tự tiến hành mô phỏng thí nghiệm, quan sát kết quả, kiểm chứng với lý thuyết trước khi tiến hành thí nghiệm trên mô hình thật.

2.1 Khởi động Simulink

Để khởi động Simulink từ môi trường Matlab, ta gõ dòng lệnh *simulink*. Lúc này một cửa sổ như trên hình 2.1 sẽ xuất hiện, trên đó có các thư mục chính và các thư viện con của Simulink. Để bắt đầu làm việc, ta tạo cửa sổ mới bằng cách kích vào biểu tượng « New ». Có 8 thư viện chính của Simulink được phân loại như sau :

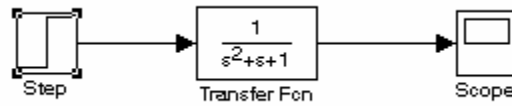
- *Continuous* : hệ thống tuyến tính và liên tục
- *Discrete* : hệ thống tuyến tính gián đoạn
- *Nonlinear* : mô hình hóa những phần tử phi tuyến như role, phần tử bão hòa...
- *Source* : các khối nguồn tín hiệu
- *Sinks* : các khối thu nhận tín hiệu
- *Function & Table* : các hàm bậc cao của Matlab
- *Math* : các khối của simulink với các hàm toán học tương ứng của Matlab
- *Signals & System* : các khối liên hệ tín hiệu, hệ thống con...



Hình 2.1 Cửa sổ chính của Simulink

2.2 Tạo một sơ đồ đơn giản

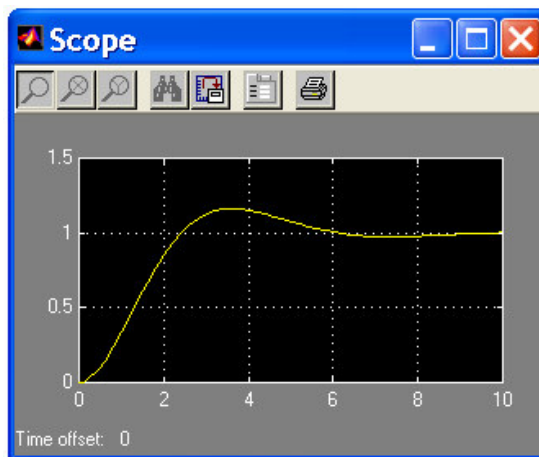
Để làm quen với Simulink, ta bắt đầu bằng một ví dụ đơn giản : phân tích hàm quá độ của một khâu bậc hai có hàm truyền $G(p) = \frac{\omega_0^2}{p^2 + 2\xi\omega_0 p + \omega_0^2}$ với $\omega_0=1\text{rad/s}$ và $\xi=0,5$. Các bước thực hiện để được sơ đồ mô phỏng như hình 2.2 như sau :



Hình 2.2 : Một sơ đồ Simulink đơn giản

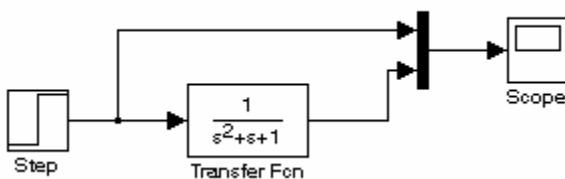
- Khởi động Simulink từ Matlab bằng dòng lệnh *simulink*
- Trong cửa sổ chính của Simulink, chọn biểu tượng « New » để tạo cửa sổ ứng dụng.
- Muốn tạo một khối trong cửa sổ ứng dụng, ta tìm khối đó trong các thư viện của Simulink, kích chọn và kéo nó vào cửa sổ ứng dụng. Ví dụ, để tạo khối *Step*, ta vào thư viện *Simulink* -> *Continuous* -> *Sources* -> *Step*, khối *Transfer Fcn* trong *Simulink* -> *Continuous* -> *Transfer Fcn*...
- Để đặt thông số cho từng khối, ta mở khối đó ra bằng cách double-click chuột vào nó. Lúc này đặt các thông số theo hướng dẫn trên màn hình.
- Đường nối giữa các khối được thực hiện bằng cách dùng chuột kéo các mũi tên ở đầu (cuối) mỗi khối đến vị trí cần nối.

Sau khi tạo được sơ đồ khối như hình 2.2, ta có thể bắt đầu tiến hành mô phỏng (với các tham số mặc định) bằng cách chọn **Simulation** -> **Start**. Xem kết quả mô phỏng bằng cách mở khối *Scope* như hình 2.3.

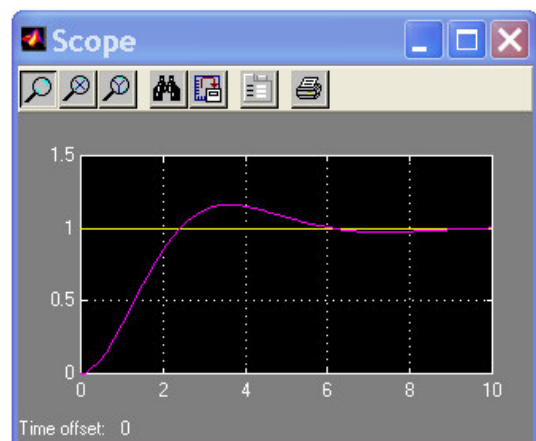


Hình 2.3 : Kết quả mô phỏng

Để xem đồng thời tín hiệu vào và ra trên cùng một *Scope*, ta tạo sơ đồ mô phỏng như hình 2.4. Kết quả mô phỏng biểu diễn trên hình 2.5.



Hình 2.4



Hình 2.5

2.3 Một số khối thường dùng

2.3.1 Thư viện « Sources »

<i>Step</i>	Tạo ra tín hiệu bậc thang liên tục hay gián đoạn.
<i>Ramp</i>	Tạo tín hiệu dốc tuyến tính (rampe) liên tục.
<i>Sine Wave</i>	Tạo tín hiệu sinus liên tục hay gián đoạn.
<i>Constant</i>	Tạo tín hiệu không đổi theo thời gian.
<i>Clock</i>	Cung cấp đồng hồ chỉ thời gian mô phỏng. Có thể xem được « đồng hồ » này khi đang thực hiện mô phỏng.

Chú ý : Muốn khối clock chỉ đúng thời điểm đang mô phỏng, tham số **Sample time** được đặt như sau

→ 0 : hệ liên tục

→ >0 : hệ gián đoạn, *clock* lúc này sẽ chỉ số chu kỳ lấy mẫu đặt trong **Sample time**.

2.3.2 Thư viện « Sinks »

<i>Scope</i>	Hiển thị các tín hiệu được tạo ra trong mô phỏng.
<i>XY Graph</i>	Vẽ quan hệ giữa 2 tín hiệu theo dạng XY. Khối này cần phải có 2 tín hiệu vào, tín hiệu thứ nhất tương ứng với trục X, tín hiệu vào thứ hai tương ứng với trục Y.
<i>To Workspace</i>	Tất cả các tín hiệu nối vào khối này sẽ được chuyển sang không gian tham số của Matlab khi thực hiện mô phỏng. Tên của biến chuyển vào Matlab do người sử dụng chọn.

2.3.3 Thư viện « Continuous »

<i>Transfer Fcn</i>	Mô tả hàm truyền của một hệ thống liên tục dưới dạng <i>Đa thức tử số/Đa thức mẫu số</i> . Các hệ số của đa thức tử số và mẫu số do người sử dụng nhập vào, theo bậc giảm dần của toán tử <i>Laplace</i> . Ví dụ để nhập vào hàm truyền có dạng $\frac{2s+1}{s^2+s+1}$, ta nhập vào như sau : <i>Numerator</i> [2 1], <i>Denominator</i> [1 1 1].
<i>State Space</i>	Mô tả hàm truyền của một hệ thống liên tục dưới dạng <i>phương trình trạng thái</i> . Các ma trận trạng thái A, B, C, D được nhập vào theo qui ước ma trận của Matlab.
<i>Integrator</i>	Khâu tích phân.
<i>sDerivative</i>	Khâu đạo hàm
<i>Transport Delay</i>	Khâu tạo trễ

2.3.4 Thư viện « Discrete »

<i>Discrete Transfer Fcn</i>	Mô tả hàm truyền của một hệ thống gián đoạn dưới dạng <i>Đa thức tử số/Đa thức mẫu số</i> . Các hệ số của đa thức tử số và mẫu số do người sử dụng nhập vào, theo bậc giảm dần của toán tử <i>z</i> .
<i>Discrete State Space</i>	Mô tả hàm truyền của một hệ thống gián đoạn dưới dạng <i>phương trình trạng thái</i> . Người sử dụng phải nhập vào các ma trận trạng thái A,B,C,D và chu kỳ lấy mẫu.
<i>Discrete-Time Integrator</i>	Khâu tích phân của hệ thống gián đoạn.
<i>First-Order Hold</i>	Khâu giữ mẫu bậc 1. Người sử dụng phải nhập vào chu kỳ lấy mẫu.
<i>Zero-Order Hold</i>	Khâu giữ mẫu bậc 0. Người sử dụng phải nhập vào chu kỳ lấy mẫu.

2.3.5 Thư viện « Signal&Systems »

<i>Mux</i>	Chuyển nhiều tín hiệu vào (vô hướng hay vectơ) thành một tín hiệu ra duy nhất dạng vectơ. Vectơ ngõ ra có kích thước bằng tổng kích
------------	---

2.5 LTI Viewer

Như ta đã biết, khi thực hiện mô phỏng trên Simulink, ta chỉ có thể quan sát được các đặc tính thời gian của hệ thống. Để có thể phân tích toàn diện một hệ thống, ta cần các đặc tính tần số như đặc tính Bode, đặc tính Nyquist, quỹ đạo nghiệm số v.v...

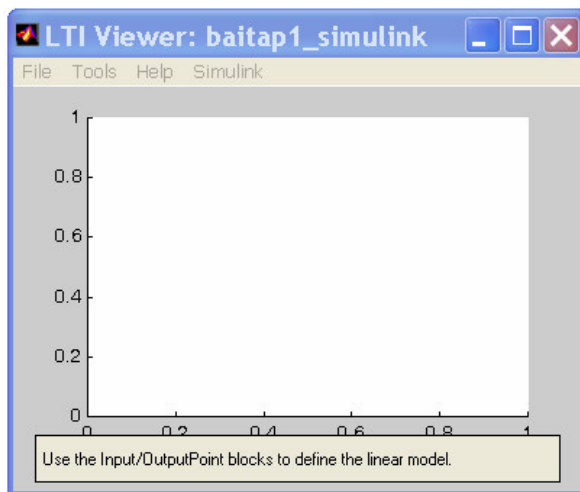
« **LTI Viewer** » là một giao diện đồ họa cho phép quan sát đáp ứng của một hệ thống tuyến tính, trong lĩnh vực tần số cũng như thời gian, mà không cần gõ lại lệnh hay lập trình theo từng dòng lệnh như trong Control System Toolbox. Nó sử dụng trực tiếp sơ đồ khối trong Simulink.

2.5.1 Khởi động LTI Viewer

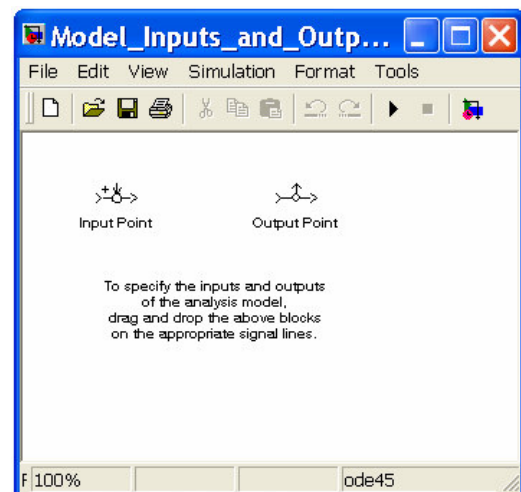
Để khởi động LTI Viewer từ Simulink, ta chọn menu **Tool -> Linear Analysis**.

Lúc này, Matlab sẽ mở 2 cửa sổ mới:

- Cửa sổ LTI Viewer (hình 2.9) có 2 phần chính:
 - o Phần cửa sổ đồ họa dùng để biểu diễn các đường đặc tính.
 - o Thanh công cụ phía dưới chỉ dẫn cách sử dụng LTI Viewer
- Cửa sổ chứa các điểm input và output (hình 2.10). Các điểm này được dùng để xác định điểm vào/ra trên sơ đồ Simulink cần phân tích.



Hình 2.9



Hình 2.10

2.5.2 Thiết lập các điểm vào/ra cho LTI Viewer

Dùng chuột kéo rê các điểm “input point”, “output point” trên cửa sổ hình 2.10 và đặt lên các vị trí tương ứng trên sơ đồ Simulink.

Chú ý: Việc chọn các điểm đặt “input”, “output” phải phù hợp yêu cầu phân tích. LTI Viewer tính hàm truyền bằng cách tuyến tính hóa hệ thống với 2 điểm input/output đã được định nghĩa. Khi vẽ các đặc tính tần số cũng như thời gian, LTI sử dụng các hệ thống đã được tuyến tính hóa này.

2.5.3 Tuyến tính hóa một mô hình

Để tìm mô hình giữa 2 điểm input/output đã định nghĩa, ta thực hiện như sau:

Chọn cửa sổ LTI Viewer (hình 2.9) → Chọn menu **Simulink** → **Get linearized model**

Lúc này, trong phần đồ họa của cửa sổ LTI Viewer sẽ xuất hiện đặc tính quá độ của mô hình tuyến tính hóa tìm được.

Để xem các đặc tính khác trên LTI Viewer, ta chỉ việc kích chuột phải vào phần đồ họa, chọn menu **Plot Type** → chọn loại đặc tính cần quan sát.

Ghi chú:

- Cứ mỗi lần thực hiện tuyến tính hóa một mô hình (**Simulink** → **Get linearized model**) thì LTI Viewer sẽ nạp mô hình hiện hành tại cửa sổ Simulink vào không gian của nó. Nếu giữa 2 lần thực hiện tuyến tính hóa, mô hình không có sự thay đổi (cấu trúc hay thông số) thì 2 mô hình tìm được tương ứng sẽ giống nhau.
- Có thể bật/tắt đặc tính của một hay nhiều mô hình đã tìm được trong LTI Viewer bằng cách: kích chuột phải vào cửa sổ đồ họa → chọn *Systems* → chọn mô hình cần bật/tắt. Tiện ích này rất cần thiết khi ta muốn so sánh tác động do sự biến đổi một thông số nào đó đến hệ thống.

2.5.4 Lưu và sử dụng các thông số của mô hình tuyến tính hóa

- Để lưu mô hình tuyến tính hóa vừa tìm được, chọn menu **File** → **Export...** ???
- Để sử dụng các thông số của mô hình :
 - o Dạng hàm truyền $[num,den]=tfdata(\text{« biến file »}, 'v')$
 - o Dạng phương trình trạng thái $[A,B,C,D]=ssdata(\text{« biến file »})$

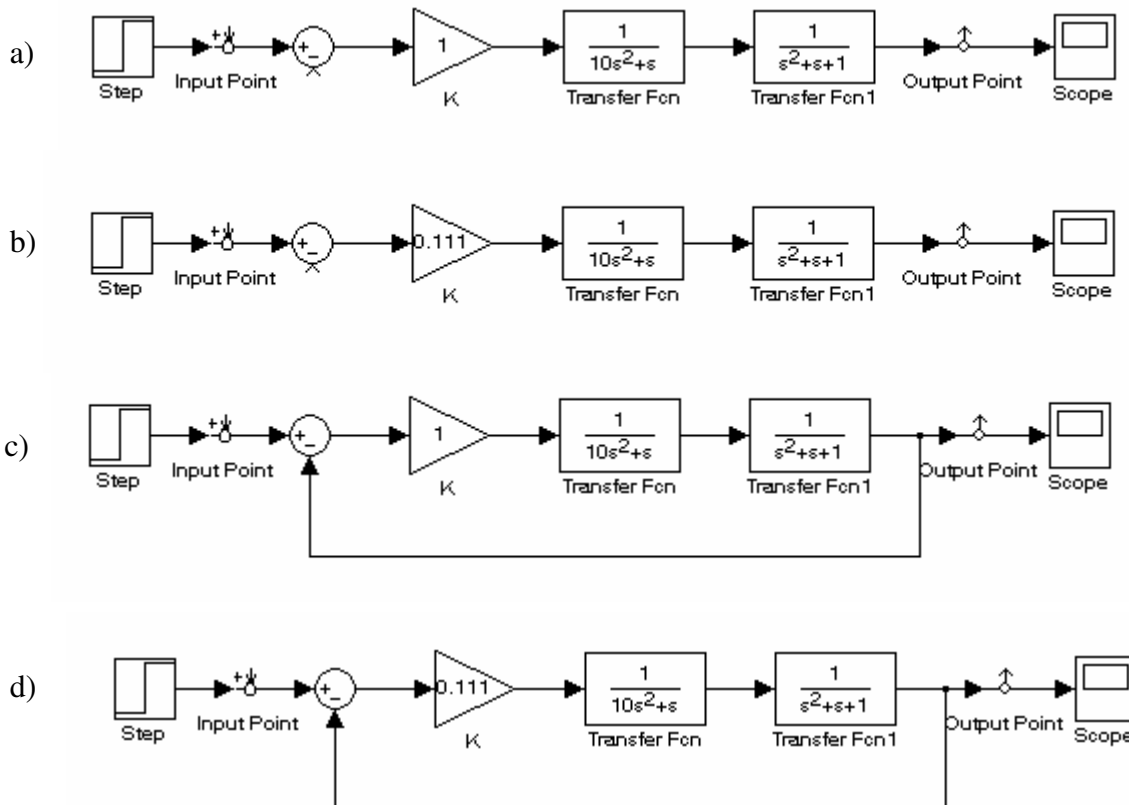
2.5.5 Ví dụ sử dụng LTI Viewer

Giả sử đã có hàm mô hình mô phỏng trên của sổ Simulink như hình 2.6. Sử dụng LTI Viewer để quan sát các đặc tính sau:

- Đặc tính tần số Nyquist của hệ hở khi chưa hiệu chỉnh ($K=1$) và đã hiệu chỉnh ($K=0.111$).
- Đặc tính tần số Bode của hệ hở đã hiệu chỉnh.
- Đặc tính quá độ của hệ kín chưa hiệu chỉnh và đã hiệu chỉnh.

THỰC HIỆN

Theo yêu cầu đặt ra, ta cần phải có 4 hệ thống có thông số và cấu trúc khác nhau: hệ hở với $K=1$, hệ hở với $K=0.111$, hệ kín $K=1$ và hệ kín $K=0.111$. Do vậy, ta cần thực hiện 4 lần tuyến tính hóa để có được 4 mô hình khác nhau trong LTI Viewer. Các bước thực hiện tuần tự như trong hình 2.11.



Hình 2.11 : Sơ đồ và cấu trúc để tuyến tính hóa

Sau 4 lần tuyến tính hóa, trong LTI Viewer, ta được 4 hệ thống lần lượt là *baitap1_simulink_1* đến *baitap1_simulink_4* (sơ đồ trong Simulink có tên là *baitap1_simulink*).

Trên cửa sổ đồ họa lúc này sẽ hiển thị đồng thời đặc tính quá độ của cả 4 mô hình ở trên.

- Để xem đặc tính Nyquist của hệ hở trước và sau hiệu chỉnh:
 - o Kích chuột phải vào phần đồ họa, chọn *Systems*, chọn 2 mô hình 1 và 2.
 - o Tiếp tục kích chuột phải vào phần đồ họa, chọn *Plot Type* → *Nyquist*.Trên cửa sổ đồ họa sẽ xuất hiện 2 đặc tính Nyquist với 2 màu phân biệt.
- Để xem đặc tính quá độ của hệ kín trước và sau hiệu chỉnh:
 - o Kích chuột phải vào phần đồ họa, chọn *Systems*, chọn 2 mô hình 3 và 4.
 - o Tiếp tục kích chuột phải vào phần đồ họa, chọn *Plot Type* → *Step*.

Các đặc tính khác được tiến hành một cách tương tự.