# BÀI THỰC HÀNH MATLAB 4

Họ và tên: Ngô Mẫn Đạt  
MSSV: 19521333  
Lớp: CE212.M21.MTCL

# GIỚI THIỆU

MATLAB, tên viết tắt của từ tiếng Anh **MAT**rix **LAB**oratory, là một môi tr ư ờ n g mạ n h dành cho các tính toán khoa hoc. Nó tích hợp các phép tính ma trận và phân tích số dựa trên các hàm cơ bán. Hơn nữa, cấu trúc đồ họa hướng đối tượng cúa Matlab cho phép tạo ra các hình vẽ 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)
* Finacial 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 bieu thúc)
* System Iđếntification (Nhận dạng)

- …

Một tính chất rất mạnh cúa Matlab là nở 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 goi từ các ngôn ngữ này…

Các ban có thể xem phần Help trong Matlab để tham kháo cách sử dụng và ví dụ cúa từng lệnh, hoặc download (mien phí) các file help dang \*.pdf tai trang Web cúa Matlab cở địa chí [http://www.mathworks.com](http://www.mathworks.com/)

# Control System Toolbox

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

# Định nghĩa một hệ thống tuyến tính

* + 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, chí dùng cho hệ gián đoạn (tính bằng *s*)

Ví dụ:

Định nghĩa hàm truyền trong Matlab

𝑝+2

*F(p)* = 3 𝑃2+2𝑝+4 *num=3\*[1 2];den=[1 2 4];sys1=tf(num,den);*

𝑧−0,6

F(z) = 2,1\* 𝑧2− 0,56𝑧 + 0,4 *num=2.1\*[1 -0.6];den=[1 -0.56];*

*T=0.5;sys2=tf(num,den,T)*

# Hệ thống nhiều tín hiệu vào/ra

U1 Y1

G11(r) G12(r) ……. G1n(r)

G21(r) G22(r) ……. G2n(r)

…….

Gp1(r) Gp2(r) ……. Gpn(r)

**G(r)**

G(r) =

Un Yn

Câu lệnh :

*G11=tf(num11,den11,T); G12=tf(num12,den12,T);...; G1n=tf(num1n,den1n,T); G21=tf(num21,den21,T); G22=tf(num22,den22,T);...; G2n=tf(num2n,den2n,T);*

*Gp1=tf(nump1,denp1,T); G12=tf(nump2,denp2,T);...; Gpn=tf(numpn,denpn,T); sys=[G11,G12,...,G1n;G21;G22;...;G2n;...;Gp1,Gp2,...,Gpn];*

# Ðịnh nghĩa bằng zero và cực

**Hệ thống nhiều tín hiệu vào/ra**

Câu lệnh: *sys=zpk(Z,P,K,T)*

* + - * *Z,P* là các Vectơ hàng chứa danh sách các điểm zero và cực cúa hệ thống.
      * *K* là hệ số khuếch đại

Chú ý: nếu hệ thống không cở diểm zero (cực) thì ta đặt là [] Ví dụ:

*F(p)* = 𝑝+2

𝑝(𝑝+5)

Z = -2; P = [0 -5]; K=1; sys = zpk(Z,P,K);

# Hệ thống nhiều tín hiệu vào/ra

Câu lệnh :

*G11=zpk(Z11,P11,T); G12=zpk(Z12,P12,T);...; G1n=zpk(Z1n,P1n,T);*

*G21=zpk(Z21,P21,T); G22=zpk(Z22,P22,T);...; G2n=zpk(Z2n,P2n,T);*

*Gp1=zpk(Zp1,Pp1,T); G12=zpk(Zp2,Pp2,T);...; Gpn=zpk(Zpn,Ppn,T);*

*sys=[G11,G12,...,G1n;G21;G22;...;G2n;...;Gp1,Gp2,...,Gpn];*

# Phương trình trang thái

Câu lệnh: *sys=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.

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 trang thái

*[A,B,C,D]=tf2ss(num,den)*

# Chuyển đổi giữa hệ liên tục và gián đoạn

**Số hóa một hệ thông 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 cở hàm truyền: G(p) = 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. ***Biến đổi sơ đồ tương đương***
     1. **Mắc nối tiếp**

**U**

**Y**

**sys2**

**sys1**

Câu lệnh: *sys=series(sys1,sys2)*

# Mắc song song

Câu lệnh: *sys=parallel(sys1,sys2)*

# Mắc phản hồi

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

# U Y

**sys2**

**sys1**

*sign* = +1 nếu phản hồi dương và *sign*= -1 (hoặc không cở *sign*) nếu phản hồi âm

* 1. ***Phân tích hệ thống***

# 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 thòi điểm *t\_end*.
* *step(sys,T)*: vẽ hàm quá độ trong khoáng thời gian *T*. *T* được định nghĩa như sau T=Ti:dt:Tf. Ðối với hệ liên tục, dt là bước vẽ, Ðối 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 úng

# Hàm trọng lượng 𝝎(t)

Câu lệnh: *impulse(sys)*

# Trong miền tần số Đặc tính bode

Câu lệnh: *bode (sys)*

Vẽ đặc tính tần số Bode cúa hệ thống tuyến tính *sys*. Dải tần số vẽ do Matlab tự chọn . Một so 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

nghia bằng hàm *logspace.*

* + - * Ví dụ: w=logspace(-2,2,100) định nghia vectơ w gồm 100 điểm, từ tần số 10-2 đến

102.

* + - * *bode (sys1,sys2,sys3,…)* Vẽ đặc tính bode cúa nhieu hệ thống đồng thời.
      * *[mag,phi,w]=bode (sys,…)* lưu tất cả các điểm tính toán cúa đặc tính bode vào

vectơ *mag*, *phi* úng với tần số *w* tương ứng.

Chú ý: Ðối với hệ thống gián đoạn, dải tần số để vẽ phái 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(w)|, arg[G(w)] và vẽ trong mặt phẳng Black. Ví dụ: Vẽ các đặc tính tần số cúa hệ thống sau

𝜔2

G(p) = 0

𝑝2− 2𝜀𝜔0𝑝 +𝜔2

0

với 𝜔0

= 1rad/s và 𝜀 = 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 cúa Matlab nyquist(G); % vẽ đặc tính nyquist

# Một số hàm để phân tích Hàm *margin*

* + - * *margin(sys)* Vẽ đặc tính *Bode* cúa hệ thống SISO và chí ra độ dự trữ biên độ, độ dự trữ

pha tai 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*

tai tần số *w\_L*, lưu độ dự trữ về pha vào biến *delta\_phi* tai tần số *w\_phi*.

**Hàm *pole***

*vec\_pol=pole(sys)* tính các điểm cực cúa hệ thống và lưu vào biến *vec\_pol*.

**Hàm *tzero***

*vec\_zer=tzero(sys)* tính các điểm zero cúa hệ thống và lưu vào biến *vec\_zer*.

**Hàm *pzmap***

* + - * *[vec\_pol,vec\_zer]=pzmap(sys)* tính các điểm cực và zero cúa hệ thống và lưu vào các biến tương ứng.
      * *pzmap(sys)* tính các điểm cực, zero và biểu diễn trên mặt phẳng phúc.

**Hàm *dcgain***

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

# Một số hàm đặc biệt trong không gian trang thái Hàm *ctrl*

Câu lệnh: *C\_com=ctrl(A,B)*

*C\_com=ctrl(sys)*

Tính ma trận “điều khiển được” **C** cúa một hệ thống. Ma trận **C** được định nghia như sau:

**C**=[B AB A2B … An-1B] với A = Rnxn

**Hàm *obsv***

Câu lệnh: *O\_obs=obsv(A,C)*

*O\_obs=obsv(sys)*

Tính ma trận “*quan sát* được” **O** cúa một hệ thống. Ma trận **O** được định nghia như sau:

**O**=[C CA CA2 … CAn-1]

**Hàm *ctrbf***

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

Chuyển vẽ dạng chuẩn (canonique) “điều khiển được” cúa một hệ thống biểu diễn dưới dạng phương trình trang thái.

Trong đở: Ab=TAT-1, Bb=TB, Cb=CT-1, T là ma trận chuyển đổi.

**Hàm *o*b*svf***

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

Chuyển vẽ dạng chuẩn “quan sát được“ cúa một hệ thống biểu diễn dưới dạng phương trình

trang thái.

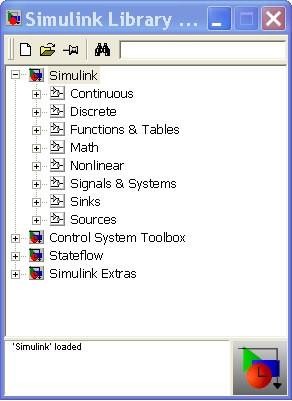
-1 -1

Trong đó: Ab=TAT

# SIMULINK

, Bb=TB, Cb=CT

, T là ma trận chuyển đổi.

Simulink được tích hợp vào Matlab (vào khoáng đầu những năm1990) 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 cúa hệ thống với nhiều tín hiệu vào khác nhau như: tín hiệu bậc thang, tín hiệu sin, 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 giao diện đồ họa. Vì vậy tất cả các hàm trong Matlab đều có thể truy cập được từ Simulink, ngay cá các hàm do người sử dụng tạo ra. Ngược lai, 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 khá 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.