

Lecture on Matlab

Bài mở đầu

Nguyen Q.Hoang
Department of Applied Mechanics
Hanoi University of Science and Technology

Chương 1. Môi trường Matlab

1. Matlab là gì ?
- MATLAB được viết tắt từ "**MAT**rix **LAB**oratory", là một công cụ tính toán số và mô phỏng số, công cụ này được phát triển dựa trên các thư viện hàm tính toán số viết bằng ngôn ngữ lập trình FORTRAN. Matlab có giao diện thân thiện với người sử dụng.
- Một định nghĩa khác: Matlab là một phần mềm đa năng tính toán số, thể hiện các số liệu bằng hình ảnh trực quan và là một ngôn ngữ đa năng cung cấp một môi trường linh hoạt cho việc tính toán kỹ thuật.
- Matlab bao gồm nhiều công cụ để:
 - thu thập dữ liệu, phân tích và xử lý dữ liệu,
 - hiển thị hình ảnh trực quan và xử lý hình ảnh,
 - tạo mẫu và phát triển thuật toán
 - mô hình hóa và mô phỏng, lập trình và phát triển ứng dụng

Chương 1. Môi trường Matlab

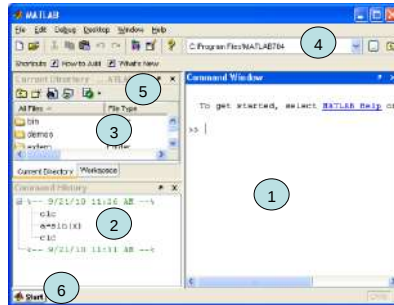
1. Matlab là gì ?
2. Giao diện người sử dụng
3. Các phép tính số học cơ bản
4. Phép gán
5. Các định nghĩa toán học cơ bản
6. Số phức
7. Xử lý lỗi khi gõ lệnh
8. Kết thúc phiên làm việc với Matlab

Chương 1. Môi trường Matlab

- Matlab có thể chạy trên hầu hết các hệ máy tính: máy tính xách tay - Laptop, máy tính cá nhân PC, đến các hệ thống siêu máy tính (super computer). Matlab được điều khiển bởi các tập lệnh, tác động qua bàn phím trên cửa sổ điều khiển. Nó cũng cho phép một khả năng lập trình với cú pháp thông dịch lệnh – còn gọi m-file. Các lệnh hay bộ lệnh của Matlab lên đến con số hàng trăm và ngày càng được mở rộng nhờ các thư viện trợ giúp hay do người sử dụng tạo ra.

Giao diện người sử dụng

1. Cửa sổ lệnh (Command Window).
2. Cửa sổ ghi lại các lệnh đã thực hiện (Command History).
3. Cửa sổ cho biết thư mục hiện thời (Current Directory) và danh sách các biến đang sử dụng (Workspace).
4. Thanh chỉ đường dẫn của thư mục hiện thời, cho phép chọn thư mục hiện thời.
5. Thanh Shortcut.
6. Nút Start.



Phép gán

- Trong Matlab dấu bằng "=" được sử dụng cho phép gán. Mặc dù trong cách viết thông thường ta hiểu dấu bằng thể hiện một phương trình, nhưng trong Matlab dấu bằng được định nghĩa là phép gán. Để phân biệt giữa hai cách thể hiện ta xét ví dụ sau. Nếu trong cửa sổ lệnh ta viết
`>> x + 18 = 120`
- Matlab sẽ báo lỗi với dòng hiển thị màu đỏ:
Error: The expression to the left of the equals sign is not a valid target for an assignment.
- Matlab không hiểu cách bạn viết một phương trình như trên giấy, mà nó chỉ có thể thực hiện được các phép tính và gán giá trị tính được cho một biến nào đó. Chẳng hạn ta có thể gán giá trị của phép tính $120 - 18$ cho biến x bằng cách viết
`>> x = 120 - 18;`
- Các dấu ";" ở cuối dòng lệnh báo cho Matlab biết là ta không muốn các giá trị của x hiện ra ở dòng dưới.
`>> x = 120 - 18`

Các phép tính số học cơ bản

- Đối với người mới bắt đầu sử dụng Matlab, trước hết ta cần làm quen với các phép tính số học (cộng +, trừ -, nhân * và chia /). Với hai số a và b trong Matlab ta có thể thực hiện được các phép tính số học như liệt kê trong bảng dưới đây.

Phép tính	Cách viết toán học	Cách viết trong Matlab
phép cộng	$c = a + b$	$c = a + b$
phép trừ	$c = a - b$	$c = a - b$
phép nhân	$c = ab$	$c = a * b$
phép chia	$c = a : b$	$c = a / b$
phép chia phải	$c = a : b$	$c = a / b$
phép chia trái	$c = b : a$	$c = a \backslash b$
lũy thừa	$c = a^b$	$c = a ^ b$

Các phép tính số học cơ bản

```
>> a=3.5; b=7.5; % gán giá trị số cho hai biến a và b
>> a+b % phép cộng hai số
ans =
    11
>> a-b % phép trừ hai số
ans =
    -4
>> a*b % phép nhân hai số
ans =
   26.2500
>> a/b
% phép chia hai số, a:b (= phép chia phải)
ans =
    0.4667
```

```
>> a/b % phép chia hai số, a:b (= phép chia phải)
ans =
    0.4667
>> a\b % phép chia trái (hiệu là b chia cho a, b/a)
ans =
   2.1429
>> a^b % phép lũy thừa, a mũ b
ans =
  1.2037e+004
```

Một số dạng format

- **format** long, short, rat, short e, bank

- Ví dụ

<pre>>> format long >> co = cos(0.2), si = sin(0.2), tn = tan(0.2) co = 0.98006657784124 si = 0.19866933079506 tn = 0.20271003550867 >> format short >> co = cos(0.2), si = sin(0.2), tn = tan(0.2) co = 0.9801 si = 0.1987 tn = 0.2027 >> format short e >> x=5.125*3.16 x = 1.6195e+001</pre>	<pre>>> format long e >> x=5.125*3.16 x = 1.619500000000000e+001 >> format rat >> x=5.125*3.16 x = 3239/200 >> format bank >> luonggio=35.55 luonggio = 35.55 >> luongtuan=luonggio*40 luongtuan = 1422.00</pre>
---	--

Các định nghĩa và hàm toán học cơ bản

- Để thuận tiện cho việc tính toán, trong Matlab người ta đã định nghĩa sẵn rất nhiều đại lượng toán học và các hàm cơ bản. Ví dụ như số pi đã được định nghĩa sẵn


```
>> R = 2;
>> V = 4/3*pi*R^3
V = 33.5103
```

<pre>>> exp(1) ans = 2.7183 >> exp(2) ans = 7.3891 >> x = sqrt(9) x = 3 >> y = sqrt(11) y = 3.3166</pre>	<pre>>> log(3.2) % ln(3.2) ans = 1.1632 >> x = 5; log(x) ans = 1.6094 >> x = 3; log10(x) ans = 0.4771</pre>
--	--

Các định nghĩa và hàm toán học cơ bản

Các hàm toán học	
sqrt(x)	Tính căn của biến x
exp(x)	Hàm e mũ, Exponential
log(x)	Logarithm cơ số tự nhiên, cơ số e
log1p(x)	Tính giá trị log(1+x)
log10(x)	Logarithm cơ số 10
log2(x)	Logarithm cơ số 2
pow2(x)	Hàm mũ cơ số 2, pow2(x) = 2^x
realpow(x,y)	Tính x^y, với x, y là thực
reallog(x)	Logarithm cơ số tự nhiên của số thực
nthroot(x, n)	Căn bậc n của số thực
nextpow2(n)	Trả lại số p đầu tiên sao cho 2^p >= abs(n)

Các định nghĩa và hàm toán học cơ bản

Các hàm lượng giác - Trigonometric functions		
sin	Hàm sin	>> sin(pi/3) cho 0.8660
sind	Hàm sin với đối số là độ	>> sind(60) cho 0.8660
sinh	Hàm sin Hyperbolic	
asin	Hàm sin ngược, tức arcsin	>> asin(0.866) cho 1.0471
asind	Hàm sin ngược cho kết quả là độ	>> asind(0.866) cho 59.9971
asinh	Hàm sin Hyperbolic ngược	
cos	Hàm cos	
cosd	Hàm cos với đối số là độ	
cosh	Hàm cos Hyperbolic	
acos	Hàm cos ngược, tức arccos	
acosd	Hàm cos ngược cho kết quả là độ	
acosh	Hàm cos Hyperbolic ngược	

$$\sinh(x) = \frac{e^x - e^{-x}}{2}$$

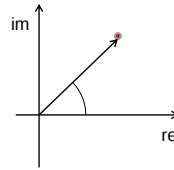
$$\cosh(x) = \frac{e^x + e^{-x}}{2}$$

Số phức

Chúng ta cũng có thể đưa số phức vào trong Matlab. Nhớ lại rằng đơn vị phức được định nghĩa là căn của (-1), Trong Matlab số phức ký hiệu là i hoặc j

- Một số phức có thể được viết dạng $z = x + i y$ trong đó x là phần thực và y là phần ảo của z
- Ví dụ

$$a = 3 + 3i, \quad b = 1 - i \rightarrow a + b = 4 + 2i$$



```
>> a = 3 + 3i, b = 1 - i;
>> c = a + b
c = 4.0000 + 2.0000i
```

```
>> abs(a)      → 4.2426 = 2*sqrt(3)
>> angle(a)    → 0.7854 = pi/4
```

$$|a| = \text{abs}(a) = \sqrt{\text{re}(a)^2 + \text{im}(a)^2}$$

$$\varphi = \text{arg}(a) = \arctan(\text{im}(a) / \text{re}(a))$$

$$a = |a| e^{i\varphi}, \quad e^{i\varphi} = \cos \varphi + i \sin \varphi$$

Xử lý lỗi khi gõ lệnh, Kết thúc phiên làm việc với Matlab

- Như đã thấy ở trên, khi thực hiện các dòng lệnh bạn có thể làm không đúng và Matlab đã báo lỗi cho bạn. Error !. Nếu khi bạn kết thúc dòng lệnh bằng gõ phím ENTER và nhận ra rằng dòng lệnh trên có lỗi, bạn không nhất thiết phải gõ lại dòng lệnh đó, mà đơn giản bạn chỉ cần sử dụng các **phím mũi tên lên hoặc xuống, ↓ ↑**, để hiển thị lại dòng lệnh cần sửa. Sau khi sửa lỗi, và đánh phím ENTER Matlab sẽ đưa ra kết quả.
- Chúng ta vừa bắt đầu với một số lệnh cơ bản của Matlab, và bây giờ có thể bạn muốn ghi lại các công việc vừa thực hiện và thoát khỏi Matlab. Làm thế nào để kết thúc Matlab trên màn hình của bạn? Thật đơn giản bạn vào **exit** trong menu File, như khi sử dụng các chương trình khác trong Windows. Một cách khác là bạn gõ lệnh **quit** vào dấu nhắc trong cửa sổ lệnh và Matlab sẽ đóng lại. Kết thúc phiên làm việc.

Số phức

- Một số hàm liên quan đến số phức được liệt kê trong bảng dưới đây

Các hàm với số phức	
abs	Cho độ lớn hay modul vectơ phức
angle	Cho góc pha hay argument của số phức, radian
complex	Tạo lập dữ liệu phức từ các phần thực vào ảo
conj	Cho số phức liên hợp
imag	Cho phần ảo
real	Cho phần thực
isreal	Trả lại giá trị đúng, 1, nếu đối số là thực
cplxpair	Sắp xếp các số phức thành cặp liên hợp

Lecture 2

Vectơ, ma trận và Matlab

Nguyen Q. Hoang
Department of Applied Mechanics
Hanoi University of Science and Technology
hoang.nguyenquang@hust.edu.vn



Vectơ và các phép tính trên vectơ

- **Nhập vectơ**

```
>> a = [2; 1; 4] % vt cot
>> v = [2 0 4] % vt hang
>> w = [1,1,9] % sd dau phay
```
- **Chuyển vị vectơ (cột thành hàng, hàng thành cột)**

```
>> a = [2; 1; 4];
>> y = a'
```
- **Các phép tính cộng và trừ hai vectơ cùng cỡ**

```
>> a = [1; 4; 5];
>> b = [2; 3; 3];
>> c1 = a + b
>> c2 = a - b
```

$$a = \begin{bmatrix} 2 \\ 1 \\ 4 \end{bmatrix} \Rightarrow y = a' = \begin{bmatrix} 2 & 1 & 4 \end{bmatrix}$$

$$a = \begin{bmatrix} 1 \\ 4 \\ 5 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ 3 \\ 3 \end{bmatrix}$$

$$c_1 = a + b = \begin{bmatrix} 3 \\ 7 \\ 8 \end{bmatrix}, \quad c_2 = a - b = \begin{bmatrix} -1 \\ 1 \\ 2 \end{bmatrix}$$



Hanoi University of Science and Technology

Chương 2. Vectơ, ma trận và Matlab

2.1 Vectơ và các phép tính trên vectơ

2.2 Biểu diễn đa thức và các phép tính đa thức

2.3 Ma trận và các phép tính cơ bản trên ma trận

2.1 Vectơ và các phép tính trên vectơ

Nhập vectơ

Các phép tính cộng và trừ hai vectơ cùng cỡ

Tạo một vectơ từ các vectơ khác

Tạo vectơ hàng có các phần tử cách đều

Các đặc trưng của vectơ

Tích vô hướng và tích có hướng hai vectơ

Tham chiếu đến các phần tử của vectơ

Một số hàm định sẵn cho vectơ



Hanoi University of Science and Technology

Vectơ và các phép tính trên vectơ

Tạo một vectơ từ các vectơ khác

Ghép hai vectơ cột

```
>> a = [1; 4; 5];
>> b = [2; 3; 3];
>> d = [a; b]
```

```
d =
     1
     4
     5
     2
     3
     3
```

Ghép hai vectơ hàng

```
>> a = [1 4 5];
>> b = [2 3 3];
>> d = [a b]
```

```
d = 1 4 5 2 3 3
```

Tạo vectơ hàng có các phần tử cách đều (toán tử hai chấm :)

```
x = [a1 : h : a2]
x(i) = a1 + (i - 1)h, i = 1, 2, 3, ...
```

```
>> x = [0:2:10]
x = 0 2 4 6 8 10
```

```
>> x = [0:0.1:1]
x = 0 0.10 0.20 0.30 0.40 0.50 0.60
    0.70 0.80 0.90 1.00
```

```
>> help linspace
```



Hanoi University of Science and Technology

Véc tơ và các phép tính trên véc tơ

- Cho các vector cùng cỡ v và u ta có các hàm tác động lên véc tơ

Tên hàm	Ý nghĩa
length(v)	Cho biết chiều dài, số phần tử của véc tơ v
sum(v)	Trả lại tổng các phần tử của véc tơ v
prod(v)	Trả lại tích các phần tử của véc tơ v
min(v)	Trả lại phần tử nhỏ nhất của véc tơ v
max(v)	Trả lại phần tử lớn nhất của véc tơ v
sort(v)	Xếp các phần tử tăng dần
find(v)	Tìm các phần tử khác không trong v
dot(u,v)	Trả lại tích vô hướng hai véc tơ
cross(u,v)	Trả lại tích có hướng hai véc tơ cỡ 3×1

Véc tơ và các phép tính trên véc tơ

Ví dụ

```
>> a = [2;3;4;5];
>> length(a)
ans = 5
>> b = [1, 1, 2];
>> length(b)
ans = 3
>> a = [8 4 4 1 7 11 2 0];
>> max(a)
ans = 11
>> min(a)
ans = 0
```

```
>> a = [8 4 4 1 7 11 2 0];
>> sum(a)
ans = 37
>> prod(a)
ans = 0
>> u = [1; 4; 5]; v = [2; 3; 3];
>> x = dot(u,v)
x = 29
>> w = cross(u,v)
w =
    -3
     7
    -5
```

Véc tơ và các phép tính trên véc tơ

Tham chiếu đến phần tử vector, Toán tử :

```
>> a = [12; 17; -2; 0; 4; 27];
>> a(2)
ans = 17
>> a(6)
ans = 27
>> a(:)
ans =
    12
    17
    -2
     0
     4
    27
```

```
>> v = a(1:3)
v =
    12
    17
   -2
```

Biểu diễn đa thức và các phép tính đa thức

2.2 Biểu diễn đa thức và các phép tính đa thức

Nhập đa thức

Các phép tính trên đa thức

Phép nhân đa thức / Phép chia đa thức

Phép cộng và trừ đa thức

Không điểm hay nghiệm của phương trình đa thức

Xây dựng đa thức từ các không điểm cho trước

Giá trị của đa thức tại một điểm

Đạo hàm đa thức

Biểu diễn đa thức và các phép tính đa thức

• Nhập đa thức

Trong Matlab một đa thức được đưa vào dưới dạng một vectơ hàng có các phần tử là các hệ số của đa thức. Khi đưa vào hay khai báo một đa thức ta cần biểu diễn dạng đầy đủ của nó, ví dụ xét đa thức

$$p_5(s) = a_5 s^5 + a_4 s^4 + a_2 s^2 + a_1 s + a_0$$

với dạng đầy đủ như sau

$$p_5(s) = a_5 s^5 + a_4 s^4 + 0s^3 + a_2 s^2 + a_1 s + a_0 s^0$$

Vectơ ứng với đa thức bậc 5 trên là một vectơ hàng gồm sáu phần tử

$$p_5 = [a_5 \quad a_4 \quad 0 \quad a_2 \quad a_1 \quad a_0]$$

Ví dụ

$$\begin{aligned} &>> p = [8 \ 3 \ 0 \ -4 \ 6 \ 0] \\ &p = 8 \ 3 \ 0 \ -4 \ 6 \ 0 \end{aligned} \quad \leftarrow \quad p_5(s) = 8s^5 + 3s^4 - 4s^2 + 6s$$

HUST

Hanoi University of Science and Technology

Biểu diễn đa thức và các phép tính đa thức

```
>> nu = [-2 -1 0 1 2]
```

```
>> poly(nu)
ans =
```

```
1 0 -5 0 4 0
```

$$\begin{aligned} p_5(s) &= s^5 - 5s^3 + 4s \\ &= s(s^4 - 5s^2 + 4) \end{aligned}$$

HUST

Hanoi University of Science and Technology

Biểu diễn đa thức và các phép tính đa thức

- Matlab sẽ xác định bậc của đa thức từ số phần tử của vectơ, trong ví dụ trên

```
>> n = length(p)-1
```

```
n = 5
```

- Các hàm liên quan đến đa thức

p = conv(p1,p2)	Nhân hai đa thức
[q,r]=deconv(p1,p2)	Phép chia đa thức p1 cho p2, bậc của p1 lớn hơn của p2
roots(p)	Tìm các nghiệm của phương trình đa thức, p(x) = 0
p = poly(nu)	Xây dựng đa thức từ các không điểm cho trước, đưa ra cho ta một đa thức nhận các giá trị trong vectơ nu làm nghiệm.
polyval(p,s0)	Giá trị của đa thức tại một điểm
polyder(p)	Đạo hàm đa thức
	Phép cộng và trừ đa thức ??

HUST

Hanoi University of Science and Technology

Biểu diễn đa thức và các phép tính đa thức

- Chia đa thức - Phép chia đa thức p1 cho p2, bậc của p1 lớn hơn của p2

```
[q, r] = deconv(p1, p2)
```

$$\frac{p_1(s)}{p_2(s)} = q(s) + \frac{r(s)}{p_2(s)}$$

- polyder(p) Đạo hàm đa thức

$$p(s) = \sum_{k=0}^n a_k s^k \Rightarrow p'(s) = \sum_{k=1}^n k a_k s^{k-1}$$

HUST

Hanoi University of Science and Technology

Ma trận và các phép tính cơ bản trên ma trận

Ma trận và các phép tính cơ bản trên ma trận

Nhập ma trận
 Nhân ma trận với một số, phép cộng và trừ hai ma trận
 Chuyển vị ma trận
 Phép nhân ma trận
 Một số phép tính cơ bản khác
 Các ma trận đặc biệt
 Tham chiếu đến các phần tử của ma trận
 Ma trận khối
 Các phép nhân và chia phần tử hai ma trận cùng cỡ
 Tính định thức và giải hệ phương trình đại số tuyến tính
 Tìm hạng của ma trận
 Tìm ma trận nghịch đảo và ma trận tựa nghịch đảo (**pseudo-inverse**)
 Trị riêng và vectơ riêng của ma trận vuông
 Phân tích ma trận vuông A thành tích các ma trận

HUST

Hanoi University of Science and Technology

Ma trận và các phép tính cơ bản trên ma trận

• Nhập ma trận

Ma trận là một mảng số hai chiều, được sắp xếp theo hàng và cột. Để tạo một ma trận trong Matlab, chúng ta đặt từng hàng vào trong dấu ngoặc vuông [], các hàng được phân biệt với nhau bằng dấu chấm phẩy (;), và trong mỗi hàng các phần tử được phân biệt với nhau bằng dấu cách hoặc bằng dấu phẩy.

- Ví dụ với hai ma trận

$$A = \begin{bmatrix} -1 & 6 \\ 7 & 11 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 0 & 1 \\ -1 & 7 & 4 \\ 3 & 0 & 1 \end{bmatrix}$$

- Ta đưa vào Matlab như sau:

```
>> A = [-1, 6; 7, 11]
```

A =

```
-1 6
 7 11
```

```
>> B = [2,0,1; -1,7,4; 3,0,1]
```

B =

```
2 0 1
-1 7 4
3 0 1
```

HUST

Hanoi University of Science and Technology

Ma trận và các phép tính cơ bản trên ma trận

Nhập ma trận với một số

```
>> A = [-3, 6; 7, 10];
>> B = [5, 1; -1, 3];
>> C = 2*A
      -6  12
      14  20
```

Cộng và trừ hai ma trận cùng cỡ

```
>> A+B
ans =
     2     7
     6    13
>> A-B
ans =
    -8     5
     8     7
```

Chuyển vị ma trận

$$A = \begin{bmatrix} 1 & -2 \\ 3 & 5 \\ 7 & 4 \end{bmatrix}, \quad A^T = \begin{bmatrix} 1 & 3 & 7 \\ -2 & 5 & 4 \end{bmatrix}$$

Sử dụng dấu nháy đơn (')

```
>> A = [-1 2 0; 6 4 1]
A =
    -1     2     0
     6     4     1
>> B = A'
B =
    -1     6
     2     4
     0     1
```

HUST

Hanoi University of Science and Technology

Ma trận và các phép tính cơ bản trên ma trận

Phép nhân hai ma trận có cỡ thích hợp, $C = A*B$

$$A = \{a_{ij}\}, \quad i = 1..m, j = 1..p,$$

$$B = \{b_{ij}\}, \quad i = 1..p, j = 1..n$$

$$C = AB = \{c_{ij}\}, \quad c_{ij} = \sum_{k=1}^p a_{ik}b_{kj}, \quad i = 1..m, j = 1..n,$$

```
>> A = [2 1; 1 2]; B = [3 4; 5 6];
```

```
>> C = A*B
```

C =

```
11 14
13 16
```

```
>> A = [1 4; 8 0; -1 3];
```

```
>> B = [-1 7 4; 2 1 -2];
```

```
>> C = A*B
```

C =

```
7 11 -4
-8 56 32
7 -4 -10
```

HUST

Hanoi University of Science and Technology

Ma trận và các phép tính cơ bản trên ma trận

Phép nhân phần tử hai ma trận

$$C = A * B = \{c_{ij}\}, \quad c_{ij} = a_{ij} * b_{ij}, \quad i = 1..m, j = 1..n$$

Cần phân biệt phép nhân phần tử (.*) hai ma trận cũng cỡ $C = A .* B$

```
>> A = [2 1; 1 2];
>> B = [3 4; 5 6];
>> C = A .* B
C =
     6     4
     5     2
```

```
>> A = [1 4; 8 0; -1 3];
    B = [-1 7 4; 2 1 -2];
>> C = A * B
C =
     7    11    -4
    -8    56    32
     7     -4   -10
>> C = A .* B
??? Error using ==> times
Matrix dimensions must agree.
```

HUST

Hanoi University of Science and Technology

Ma trận và các phép tính cơ bản trên ma trận

Các ma trận đặc biệt

Khởi tạo các ma trận đặc biệt

eye(m,m)	tạo ma trận đơn vị cỡ mxm
eye(m)	tạo ma trận đơn vị cỡ mxm
eye(m,n)	tạo ma trận đơn vị mở rộng cỡ mxn
zeros(m,n)	tạo ma trận không cỡ mxn
zeros(m,m)	tạo ma trận không cỡ mxm, hay ma trận vuông
zeros(m)	
ones(m,n)	cho ta một ma trận 1 cỡ mxn
ones(m,m)	cho ta một ma trận 1 cỡ mxm, hay ma trận vuông
ones(m)	
magic(n)	tạo ma trận magic vuông cỡ nxn, n > 2 [ma phương]

HUST

Hanoi University of Science and Technology

Ma trận và các phép tính cơ bản trên ma trận

Các ma trận đặc biệt

<pre>>> eye(3) ans = 1 0 0 0 1 0 0 0 1 >> eye(2,3) ans = 1 0 0 0 1 0 >> zeros(3,3) ans = 0 0 0 0 0 0 0 0 0</pre>	<pre>>> zeros(3,2) ans = 0 0 0 0 0 0 >> zeros(2) ans = 0 0 0 0 >> ones(2,2) or ones(2) ans = 1 1 1 1</pre>	<pre>>> magic(3) ans = 8 1 6 3 5 7 4 9 2 >> magic(5) ans = 17 24 1 8 15 23 5 7 14 16 4 6 13 20 22 10 12 19 21 3 11 18 25 2 9</pre>
--	---	--

HUST

Hanoi University of Science and Technology

Ma trận và các phép tính cơ bản trên ma trận

Hàm diag

- Lấy ra đường chéo của ma trận bằng lệnh diag, và tạo ra ma trận đường chéo từ một vector bằng lệnh diag
- $c = \text{diag}(A)$ cho ta đường chéo của ma trận A
- $A = \text{diag}(c)$ cho ta ma trận đường chéo A, với $A(i,i) = c(i)$

<pre>>> A1=[1 2 3; 3 4 5; 3 5 7] A1 = 1 2 3 3 4 5 3 5 7 >> Ad=diag(A1) Ad = 1 4 7</pre>	<pre>>> Add=diag(Ad) Add = 1 0 0 0 4 0 0 0 7</pre>
---	---

HUST

Hanoi University of Science and Technology

Ma trận và các phép tính cơ bản trên ma trận

Tham chiếu đến các phần tử của ma trận

Các phần tử, các cột hay các hàng của ma trận đều có thể được tác động đến nhờ cách đánh chỉ số của chúng. Chỉ số của các phần tử của ma trận là cặp số nguyên (i,j), i là chỉ số hàng và j là chỉ số cột. Các số này bắt đầu từ 1 (1,2,...), Matlab không sử dụng chỉ số 0 như một số ngôn ngữ lập trình khác. Ví dụ đối với ma trận

```
>> A = [1 2 3;
        4 5 6;
        7 8 9];
```

```
>> A(2,3)      >> c2 = A(:,2)
ans = 6         ans =
         2
         5
         8
```

```
>> r3=A(3,:)
r3 =
     7     8     9
>> A(:,2:3)
ans =
     2     3
     5     6
     8     9
```

Ma trận và các phép tính cơ bản trên ma trận

Ma trận khối

- Cho các ma trận có kích cỡ thích hợp: A, B, C, M
- Tạo ma trận cỡ lớn hơn từ các ma trận trên
- $G = \begin{bmatrix} A & B \\ C & M \end{bmatrix}$;

- Ví dụ, với A và B

$$A = \begin{bmatrix} 2 & 4 \\ 7 & 5 \end{bmatrix}, \quad B = \begin{bmatrix} 4 & 2 \\ 5 & 7 \end{bmatrix}$$

- Hãy xây dựng ma trận Q sau

$$Q = \begin{bmatrix} \mathbf{0}_{2 \times 2} & \mathbf{E}_{2 \times 2} \\ -\mathbf{A}^{-1}\mathbf{B} & \mathbf{B}\mathbf{A}^T \end{bmatrix}$$

```
>> A = [2 4; 7 5]
>> B = [4 2; 5 7]
>> Q = [zeros(2) eye(2); -A\B B*A']
Q =
     0     0     1     0
     0     0     0     1
     0    -1    16    38
    -1     0    38    70
```

Ma trận và các phép tính cơ bản trên ma trận

Định thức và giải hệ phương trình đại số tuyến tính

- Định thức của ma trận vuông là một số. Ví dụ đối với ma trận vuông cỡ 2×2

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad \det(A) = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

```
>> A = [1 3; 4 5];      >> B = [3 -1 2 4; 0 2 1 8; -9 17 11 3; 1 2 3 -3];
>> det(A)              >> det(B)
ans = -7               ans = -533
```

- Xét hệ phương trình đại số tuyến tính sau

$$\begin{aligned} 5x + 2y - 9z &= 44 \\ -9x - 2y + 2z &= 11 \\ 6x + 7y + 3z &= 44 \end{aligned}$$

- Viết lại dạng ma trận $Ax = b$

$$A = \begin{bmatrix} 5 & 2 & -9 \\ -9 & -2 & 2 \\ 6 & 7 & 3 \end{bmatrix}, \quad x = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad b = \begin{bmatrix} 44 \\ 11 \\ 44 \end{bmatrix}$$

- Nếu $\det(A) \neq 0$ pt $Ax = b$ có nghiệm duy nhất $x = A \setminus b$

Ma trận và các phép tính cơ bản trên ma trận

Định thức và giải hệ phương trình đại số tuyến tính

```
>> A = [5 2 -9; -9 -3 2; 6 7 3];
>> b = [44;11;5];
```

```
>> det(A)
ans = 368
```

```
>> x = A \ b
x =
    -5.1250
     7.6902
    -6.0272
```

Ma trận và các phép tính cơ bản trên ma trận

Tìm ma trận nghịch đảo

- Nghịch đảo của ma trận vuông A là một ma trận, được ký hiệu là A^{-1} thỏa mãn $A A^{-1} = A^{-1} A = E$, với E là ma trận đơn vị cùng cỡ.
- Điều kiện để tồn tại ma trận nghịch đảo là $\det(A) \neq 0$.
- Nếu tồn tại ma trận nghịch đảo của A , thì phương trình đại số tuyến tính $Ax = b$ có nghiệm duy nhất là $x = A^{-1}b$
- Tính nghịch đảo ma trận vuông bằng lệnh `inv(A)`

<code>>> A = [2 3; 4 5]</code>	<code>>> S = [1 0 -1 2; 4 -2 -3 1; 0 2 -1 1; 0 0 9 8];</code>
<code>>> det(A)</code>	<code>>> det(S)</code>
<code>ans = -2</code>	<code>ans = -108</code>
<code>>> iA = inv(A)</code>	<code>>> iS = inv(S)</code>
<code>iA =</code>	<code>iS =</code>
<code>-2.5000 1.5000</code>	<code>-0.9259 0.4815 0.4815 0.1111</code>
<code>2.0000 -1.0000</code>	<code>-0.6296 0.1574 0.6574 0.0556</code>
	<code>-0.5926 0.1481 0.1481 0.1111</code>
	<code>0.6667 -0.1667 -0.1667 0</code>

HUST

Hanoi University of Science and Technology

Ma trận và các phép tính cơ bản trên ma trận

- Để kiểm tra lại ta tính tích hai ma trận:

<code>>> iS*S</code>	<code>>> S*iS</code>
<code>ans =</code>	<code>ans =</code>
<code>1.0000 0 0 0</code>	<code>1.0000 0 0 0</code>
<code>0 1.0000 0 0.0000</code>	<code>0.0000 1.0000 -0.0000 0</code>
<code>0 0 1.0000 0</code>	<code>0.0000 -0.0000 1.0000 0</code>
<code>0 0 0.0000 1.0000</code>	<code>0 0 0 1.0000</code>

- Ví dụ giải hệ $Ax = b$

<code>>> A = [3 -2; 6 -2]; b = [5; 2];</code>	<code>>> e = A*x - b % thu lại</code>
<code>>> det(A)</code>	<code>e =</code>
<code>ans = 6</code>	<code>1.0e-015 *</code>
<code>>> x = inv(A)*b</code>	<code>0.0000</code>
<code>x =</code>	<code>0.8882</code>
<code>-1.0000</code>	
<code>-4.0000</code>	

HUST

Hanoi University of Science and Technology

Ma trận và các phép tính cơ bản trên ma trận

Trị riêng và vectơ riêng của ma trận vuông

- Cho A là một ma trận vuông cấp n , số λ được gọi là trị riêng và vectơ khác không x là vectơ riêng của A nếu chúng thỏa mãn điều kiện $Ax = \lambda x$ hay $(A - \lambda E)x = 0$ với E là ma trận đơn vị.
- Cho A và B là hai ma trận vuông cấp n , nếu tồn tại số λ và vectơ x khác không thỏa mãn điều kiện $Ax = \lambda Bx$ hay $(A - \lambda B)x = 0$

thì số λ gọi là trị riêng suy rộng của hai ma trận A và B , vectơ x là vectơ riêng tương ứng.

HUST

Hanoi University of Science and Technology

Ma trận và các phép tính cơ bản trên ma trận

Trị riêng và vectơ riêng của ma trận vuông

Lệnh <code>eig</code> tính toán trị riêng và vectơ riêng của ma trận vuông	
<code>l = eig(A);</code>	cho một vectơ <code>l</code> chứa các trị riêng của ma trận vuông A .
<code>[V, D] = eig(A);</code>	cho ta ma trận chéo D chứa các trị riêng của A , ma trận V có các cột là các vectơ riêng của A thỏa mãn $AV = VD$.
<code>d = eig(A,B);</code>	cho ta vectơ chứa các trị riêng suy rộng của các ma trận vuông A và B .
<code>[V, D] = eig(A,B);</code>	cho ta ma trận chéo D chứa các trị riêng suy rộng của hai ma trận A và B , ma trận V có các cột là các vectơ riêng thỏa mãn $AV = BVD$.
<code>eig(A,B,'chol')</code>	tương tự như <code>eig(A,B)</code> đối với ma trận đối xứng A và ma trận đối xứng xác định dương B . Phương pháp tính ở đây dựa trên khai triển Cholesky ma trận B .

HUST

Hanoi University of Science and Technology

Ma trận và các phép tính cơ bản trên ma trận

Trị riêng và vectơ riêng của ma trận vuông

- Ví dụ

<pre>>> C = [2 1 0; 1 4 -1; 0 -1 7]; >> eig(C) ans = 1.5573 4.1233 7.3194</pre>	<pre>>> [V, D]=eig(C) V = 0.9119 -0.4064 -0.0571 -0.4037 -0.8630 -0.3037 -0.0742 -0.3000 0.9510 D = 1.5573 0 0 0 4.1233 0 0 0 7.3194</pre>
---	---

Ma trận và các phép tính cơ bản trên ma trận

Phân tích LU và phân tích Cholesky

- Phân tích LU. Với mỗi ma trận vuông A không suy biến, luôn phân tích được thành tích của hai ma trận tam giác

$A = LU$ với L là ma trận tam giác dưới, U là ma trận tam giác trên

Ý nghĩa của phân tích LU có thể thấy trong việc giải hệ $Ax = b$. Sau khi phân tích LU ta nhận được

$$Ax = LUx = b$$

Đặt $y = Ux \Rightarrow Ly = b$ ta giải hai hệ $Ly = b$ và $Ux = y$

Do L và U là các ma trận tam giác nên việc giải hai hệ này khá đơn giản.

- Phân tích Cholesky. Ma trận vuông A đối xứng xác định dương luôn phân tích được thành tích của hai ma trận

$$A = LL^T \text{ với } L \text{ là ma trận tam giác dưới}$$

Ma trận và các phép tính cơ bản trên ma trận

Ví dụ phân tích LU

```
>> A = [3 2 -9; -9 -5 2; 6 7 3];
b = [-65; 16; 5];
>> [L, U, P] = lu(A)
L =
   -0.3333    0.0909    1.0000
    1.0000     0         0
   -0.6667    1.0000     0
U =
   -9.0000   -5.0000    2.0000
     0       3.6667    4.3333
     0     0      -8.7273
>> x = U\U\b
x =
    2.0000
   -4.0000
    7.0000
```

Ví dụ phân tích cholesky

```
>> M = [2 3 4; 3 5 6; 4 6 9];
>> L = chol(M)
L =
    1.4142    2.1213    2.8284
         0    0.7071    0.0000
         0     0    1.0000
>> L' * L - M
ans =
    1.0e-015 *
    0.4441     0     0
         0     0     0
         0     0     0
```

các phép tính cơ bản trên ma trận

Các phép tính ma trận, vector

Các hàm toán học tác động trên ma trận, vector

Các hàm toán học tác động trên ma trận, vector	
.* ./ .^	Các phép tính trên từng phần tử
\ /	Chia trái và chia phải
transpose(A), A', A.'	Chuyển vị (ma trận, vector)
ctranspose(A), A', A.'	Chuyển vị, ma trận, vector liên hợp
inv(A)	Nghịch đảo ma trận
det(A)	Tính định thức ma trận vuông
linsolve(A, B)	Giải hệ $AX=B$, phân tích LU
eig(A)	Trị riêng, vector riêng ma trận
rank(A)	Hạng ma trận
<[m, n]= size(A <,i>)	Cho biết cỡ ma trận, vector
length(A)	(i=1 số hàng, i=2 số cột)
	Giá trị lớn nhất của số hàng và số cột (=max(sizeA))

các phép tính cơ bản trên ma trận

Các phép tính ma trận, vector	
Các hàm toán học tác động trên ma trận, vector	
<code>sum(v)</code>	Tổng các phần tử của vector <code>v</code>
<code>prod(v)</code>	Tích các phần tử của vector <code>v</code>
<code>min(v)</code>	Phần tử nhỏ nhất của vector <code>v</code>
<code>max(v)</code>	Phần tử lớn nhất của vector <code>v</code>
<code>max(A(:))</code>	Phần tử lớn nhất của ma trận <code>A</code>
<code>sort(v)</code>	Xếp các phần tử tăng dần
<code>find(v)</code>	Tìm các phần tử khác không

HUST

Hanoi University of Science and Technology

HUST

Hanoi University of Science and Technology

HUST

Hanoi University of Science and Technology

HUST

Hanoi University of Science and Technology

Ma trận tựa nghịch đảo

Ma trận tựa nghịch đảo [pseudo-inverse], help pinv

Giải hệ $Ax = b$; A là ma trận chữ nhật $Ax = b$

-Số ẩn > số phương trình

-Số ẩn < số phương trình

-Số ẩn > số phương trình: vô số nghiệm, do đó có thể đưa thêm điều kiện vào bài toán. Ví dụ cần tìm x sao cho độ lớn (chuẩn) của x nhỏ nhất

$$Ax = b, \quad A \in \mathbb{R}^{m \times n}, \quad n > m, \quad \text{rank}(A) = m$$

$$J = \frac{1}{2} x^T x \rightarrow \min$$

Bài toán xuất hiện :

-Robot dư dẫn động (số tọa độ khớp > 6, robot kg; số tọa độ khớp > 3, robot phẳng; [kinematics of redundant manipulator])

-Tàu lặn mini (số cánh quạt > 6).

HUST

Hanoi University of Science and Technology

Ref. Một số bài toán liên quan ma trận

Ma trận tựa nghịch đảo

Sử dụng PP nhân tử Lagrange để giải quyết bài toán tối ưu trên:

ta sẽ tìm cực trị của hàm Lagrange như sau

$$\mathcal{L}(x, \lambda) = \frac{1}{2} (x - x_o)^T W (x - x_o) + \lambda^T [b - Ax] \quad \lambda \in \mathbb{R}^{m \times 1}$$

Đạo hàm theo biến x ta nhận được

$$\frac{\partial}{\partial x} \mathcal{L}(x, \lambda) = W(x - x_o) - A^T \lambda = 0$$

Giải tìm được

$$\Rightarrow x = W^{-1} A^T \lambda + x_o$$

Thay trở lại phương trình ban đầu

$$b = Ax = AW^{-1} A^T \lambda + Ax_o$$

Do A có hạng đầy đủ nên

$$\text{rank}(A) = m \Rightarrow \det[AW^{-1} A^T] \neq 0$$

$$\lambda = [AW^{-1} A^T]^{-1} (b - Ax_o)$$

HUST

Hanoi University of Science and Technology

Ma trận tựa nghịch đảo

Nếu ma trận A suy biến, $\det(A) = 0$, tức là trong ma trận A có những hàng là tổ hợp tuyến tính của các hàng khác. Do đó, ta có thể biến đổi hệ phương trình ban đầu về một hệ gồm m phương trình độc lập ($m < n$). Trường hợp này sẽ được xem xét trong phần sau.

Ma trận A chữ nhật cỡ $m \times n$, ($m < n$), hạng bằng m , [full rank]

Hệ phương trình đại số tuyến tính $Ax = b$ sẽ có vô số nghiệm. Điều này cho phép ta tìm được một nghiệm tối ưu theo một nghĩa nào đó từ tập vô số các nghiệm.

Đưa vào tiêu chuẩn tối ưu cho nghiệm cần tìm

Tìm nghiệm của phương trình

$$Ax = b \quad \text{sao cho} \quad J = \frac{1}{2} (x - x_o)^T W (x - x_o) \rightarrow \min$$

W -Ma trận trọng số (xác định dương)

x_o -Véc tơ tham chiếu (được chọn trước)

$$x_o = 0 \Rightarrow J = \frac{1}{2} x^T W x \rightarrow \min \quad W = 2E \Rightarrow J = x^T x = (\|x\|_2)^2 \rightarrow \min$$

HUST

Hanoi University of Science and Technology

Ma trận tựa nghịch đảo [pseudo-inverse]

$$x = W^{-1} A^T \lambda + x_o \quad \lambda = [AW^{-1} A^T]^{-1} (b - Ax_o)$$

$$= W^{-1} A^T [AW^{-1} A^T]^{-1} (b - Ax_o) + x_o$$

$$= W^{-1} A^T [AW^{-1} A^T]^{-1} b + (E - W^{-1} A^T [AW^{-1} A^T]^{-1} A) x_o$$

Ma trận

$$A_W^\dagger = W^{-1} A^T [AW^{-1} A^T]^{-1}$$

được gọi là ma trận tựa nghịch đảo có trọng số của ma trận A .

Nếu W là ma trận đơn vị ta có là ma trận tựa nghịch đảo (phải) của A .

$$A^\dagger = A^T [AA^T]^{-1}$$

$$x = A^\dagger b + (E - A^\dagger A) x_o;$$

$$\text{Khi } x_o = 0 \Rightarrow x = A^\dagger b$$

Nếu A vuông, ko suy biến

$$A^\dagger = A^T [AA^T]^{-1} = A^T [A^{-T} A^{-1}]$$

$$= A^T A^{-T} A^{-1} = A^{-1}$$

HUST

Hanoi University of Science and Technology

Ma trận tựa nghịch đảo [pseudo-inverse]

Hệ PT đại số tuyến tính với số PT nhiều hơn số ẩn. Thông thường PT không có nghiệm thỏa mãn.

$$\mathbf{Ax} = \mathbf{b}, \quad \mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{x} \in \mathbb{R}^{n \times 1}, \mathbf{b} \in \mathbb{R}^{m \times 1}, m > n$$

$$\text{rank}(\mathbf{A}) = n$$

PT không có nghiệm thỏa mãn, Tuy nhiên, ta có thể tìm được vector $\mathbf{x}$ sao cho sai số

$$\mathbf{e} = \mathbf{Ax} - \mathbf{b} \rightarrow \min$$

Một cách tổng quát ta có thể đặt lại bài toán như sau: tìm vector $\mathbf{x}$ sao cho hàm sau đây đạt cực tiểu

$$J = \frac{1}{2} \mathbf{e}^T \mathbf{W} \mathbf{e} = \frac{1}{2} (\mathbf{Ax} - \mathbf{b})^T \mathbf{W} (\mathbf{Ax} - \mathbf{b})$$

$\mathbf{W}$ là ma trận trọng số đối xứng xác định dương



Ma trận tựa nghịch đảo [pseudo-inverse]

Trường hợp ma trận $\mathbf{A}$ ở điều kiện yếu [ill-condition]

Trong các trường hợp $\det(\mathbf{A})$ khác không nhưng nhỏ hơn chuẩn của nó nhiều lần, thì nghiệm tìm được rất nhạy đối với các phương pháp tính, đặc biệt một sự thay đổi rất nhỏ của $\mathbf{A}$ có thể làm cho nghiệm thay đổi rất nhiều, ($\mathbf{A}$ là ma trận ở điều kiện yếu)

$$\det(\mathbf{A}) \ll \|\mathbf{A}\| \quad \mathbf{Ax} = \mathbf{b}, \quad \mathbf{A} \in \mathbb{R}^{n \times n}, \mathbf{x}, \mathbf{b} \in \mathbb{R}^{n \times 1}$$

Khi giải các hệ ở điều kiện yếu, người ta hay sử dụng phương pháp bình phương tối thiểu có trọng số (*damped least square inverse*). Theo phương pháp này, nghiệm của phương trình được tìm với điều kiện bổ sung như sau:

$$\min_{\mathbf{x}} (\|\mathbf{Ax} - \mathbf{b}\|^2 + \alpha \|\mathbf{x}\|^2)$$

Chọn trọng số $\alpha > 0$

$$f(\mathbf{x}) = (\mathbf{Ax} - \mathbf{b})^T (\mathbf{Ax} - \mathbf{b}) + \mathbf{x}^T (\alpha \mathbf{E}) \mathbf{x}$$

$$= \mathbf{x}^T (\mathbf{A}^T \mathbf{A} + \alpha \mathbf{E}) \mathbf{x} - \mathbf{b}^T \mathbf{Ax} - \mathbf{x}^T \mathbf{A}^T \mathbf{b} + \mathbf{b}^T \mathbf{b} \rightarrow \min$$



Ma trận tựa nghịch đảo [pseudo-inverse]

Đạo hàm J theo biến $\mathbf{x}$ và cho bằng 0, ta nhận được

$$\frac{\partial J}{\partial \mathbf{x}} = \mathbf{A}^T \mathbf{W} (\mathbf{Ax} - \mathbf{b}) = 0 \Rightarrow \mathbf{x} = [\mathbf{A}^T \mathbf{W} \mathbf{A}]^{-1} \mathbf{A}^T \mathbf{W} \mathbf{b}$$

Khi $\mathbf{W}$ là một ma trận đơn vị

$$J = \frac{1}{2} \mathbf{e}^T \mathbf{e} = \frac{1}{2} \sum_{i=1}^n e_i^2 \quad \text{and } \mathbf{x} = [\mathbf{A}^T \mathbf{A}]^{-1} \mathbf{A}^T \mathbf{b} \doteq \mathbf{A}^{\dagger} \mathbf{b}$$

$\mathbf{A}^{\dagger} = [\mathbf{A}^T \mathbf{A}]^{-1} \mathbf{A}^T$ được gọi là ma trận tựa nghịch đảo **trái** của $\mathbf{A}$
(xác định nghiệm hệ có số ẩn < số pt)

$\mathbf{A}^{\dagger} = \mathbf{A}^T [\mathbf{A} \mathbf{A}^T]^{-1}$ được gọi là ma trận tựa nghịch đảo **phải** của $\mathbf{A}$
(xác định nghiệm hệ có số pt < số ẩn)



Ma trận tựa nghịch đảo [pseudo-inverse]

Trường hợp ma trận $\mathbf{A}$ ở điều kiện yếu [ill-condition]

$$\begin{aligned} f(\mathbf{x}) &= (\mathbf{Ax} - \mathbf{b})^T (\mathbf{Ax} - \mathbf{b}) + \mathbf{x}^T (\alpha \mathbf{E}) \mathbf{x} \\ &= \mathbf{x}^T (\mathbf{A}^T \mathbf{A} + \alpha \mathbf{E}) \mathbf{x} - \mathbf{b}^T \mathbf{Ax} - \mathbf{x}^T \mathbf{A}^T \mathbf{b} + \mathbf{b}^T \mathbf{b} \rightarrow \min \end{aligned}$$

Đạo hàm theo biến vector $\mathbf{x}$ và cho kết quả bằng không, ta nhận được

$$\left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right)^T = 2(\mathbf{A}^T \mathbf{A} + \alpha \mathbf{E}) \mathbf{x} - 2\mathbf{A}^T \mathbf{b} = 0$$

Giải được

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A} + \alpha \mathbf{E})^{-1} \mathbf{A}^T \mathbf{b} \quad \alpha = 0 : \mathbf{x} = \mathbf{A}^{\dagger} \mathbf{b} = [\mathbf{A}^T \mathbf{A}]^{-1} \mathbf{A}^T \mathbf{b}$$

Bài toán dạng này thường xuất hiện trong các bài toán động học ngược robot công nghiệp khi mà robot chuyển động gần hoặc qua các cấu hình kỳ dị, tại đó ma trận Jacobi suy biến hay bị giảm hạng.



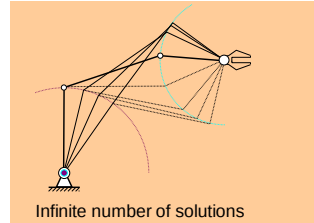
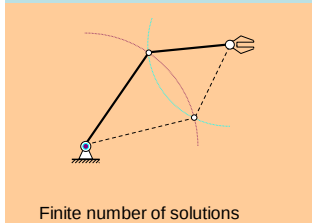
Application of pseudo-inverse: Inverse kinematics

Problems Given $\mathbf{x}(t) = [\mathbf{r}_E, \mathbf{O}_E] \in \mathbb{R}^m$ determined $\mathbf{q}(t) \in \mathfrak{R}^n$

$\mathbf{f}(\mathbf{x}, \mathbf{q}) = 0$
 $\left\{ \begin{array}{l} \text{nonlinear equations} \\ \text{more unknowns than equations} \end{array} \right.$

Methods: analytical and numerical

Number of solutions



HUST

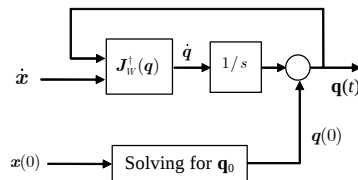
Hanoi University of Science and Technology

Nguyen Q. Hoang, Nguyen V. Khang

45

Application of pseudo-inverse: Inverse kinematics

Blockdiagram of Inverse kinematics



Drawback of this diagram :
 accumulated errors due to rounding and integral method makes $\mathbf{x}(t), \mathbf{q}(t)$ don't satisfy the constraint equations, so the end-effector moves out the given trajectory.

$$\mathbf{f}(\mathbf{x}(t), \mathbf{q}(t)) \neq 0$$

HUST

Hanoi University of Science and Technology

Nguyen Q. Hoang, Nguyen V. Khang

47

Application of pseudo-inverse: Inverse kinematics

Method based on the pseudo-inverse of jacobian matrix

$$\frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}} \frac{d\mathbf{q}}{dt} = \dot{\mathbf{x}} \quad \text{or} \quad \boxed{\mathbf{J}(\mathbf{q})\dot{\mathbf{q}} = \dot{\mathbf{x}}} \quad \mathbf{J}(\mathbf{q}) = \frac{\partial \mathbf{f}}{\partial \mathbf{q}} = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \dots & \frac{\partial f_1}{\partial q_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_m}{\partial q_1} & \frac{\partial f_m}{\partial q_2} & \dots & \frac{\partial f_m}{\partial q_n} \end{bmatrix}$$

system of m linear equations, with n unknown

introduction the functional

$$\mathbf{J} = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{W} \dot{\mathbf{q}} \rightarrow \min \quad \mathbf{W} : \text{positive weighting matrix}$$

Solution

$$\boxed{\dot{\mathbf{q}} = \mathbf{W}^{-1} \mathbf{J}^T(\mathbf{q}) [\mathbf{J}(\mathbf{q}) \mathbf{W}^{-1} \mathbf{J}^T(\mathbf{q})]^{-1} \dot{\mathbf{x}}} \Rightarrow \boxed{\mathbf{q}(t) = \mathbf{q}(0) + \int_0^t \dot{\mathbf{q}} dt}$$

if $\mathbf{W}$ is identical matrix

$$\dot{\mathbf{q}} = \mathbf{J}^T(\mathbf{q}) [\mathbf{J}(\mathbf{q}) \mathbf{J}^T(\mathbf{q})]^{-1} \dot{\mathbf{x}} = \mathbf{J}^+ \dot{\mathbf{x}} \quad \mathbf{J}^+ = \mathbf{J}^T(\mathbf{q}) [\mathbf{J}(\mathbf{q}) \mathbf{J}^T(\mathbf{q})]^{-1}$$

HUST

Hanoi University of Science and Technology

Nguyen Q. Hoang, Nguyen V. Khang

46

Application of pseudo-inverse: Inverse kinematics

Method of Error feedback

$$\dot{\mathbf{e}} = -\mathbf{K}\mathbf{e}, \quad \mathbf{e} = \mathbf{x} - \mathbf{f}(\mathbf{q}) \quad \mathbf{e}(t) \rightarrow 0$$

$$\dot{\mathbf{e}} = \dot{\mathbf{x}} - \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} = -\mathbf{K}\mathbf{e}$$

$$\mathbf{J}(\mathbf{q})\dot{\mathbf{q}} = \dot{\mathbf{x}} + \mathbf{K}\mathbf{e} \Rightarrow \boxed{\dot{\mathbf{q}} = \mathbf{J}^+(\dot{\mathbf{x}} + \mathbf{K}\mathbf{e})}$$

Null space of jacobian matrix $\mathbf{J}(\mathbf{q})$

$$\dot{\mathbf{q}} = \mathbf{J}^+(\dot{\mathbf{x}} + \mathbf{K}\mathbf{e}) + (\mathbf{E} - \mathbf{J}^+ \mathbf{J}) \mathbf{z}_o$$

By putting $\mathbf{z}_o$ the redundancy of manipulator can be exploited:

- avoiding obstacles
- avoiding impact with the joint limitations
- avoiding singular configurations

HUST

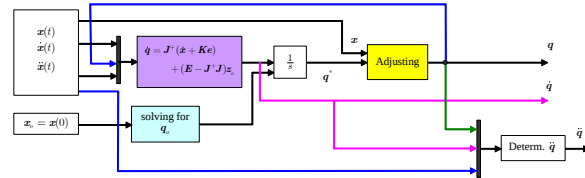
Hanoi University of Science and Technology

Nguyen Q. Hoang, Nguyen V. Khang

48

Application of pseudo-inverse: Inverse kinematics

Method of adjustment in error of joint variables

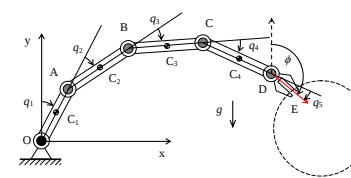


Main idea of this method:

find $\mathbf{q}$ closed to $\mathbf{q}^*$, such that $\mathbf{q}$ satisfy constraint equation $\mathbf{f}(\mathbf{x}, \mathbf{q})=0$

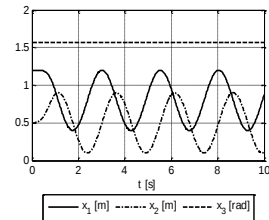
Application of pseudo-inverse: Numerical experiments

A 5-DOF planar manipulator



Manipulator parameter

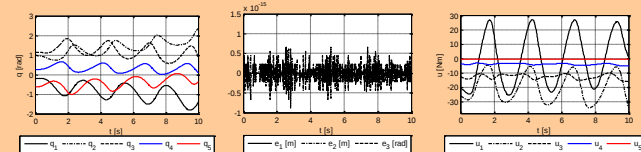
Link	1	2	3	4	5
m [kg]	10	7.5	5	4	1
I_z [kgm ²]	0.11	0.10	0.03	0.02	0.02
l [m]	0.55	0.50	0.45	0.40	0.20
a [m]	0.15	0.16	0.19	0.20	0.10



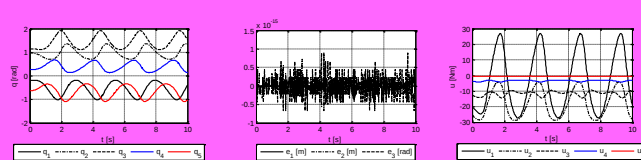
In this simulation, the end-effector (link 5) will be forced to move at the velocity of 2 m/s along a circular trajectory, while its orientation is constant, 1.0 rad. The trajectory has a center at (0.8, 0.5) m and radius of $R = 0.5$ m.

Application of pseudo-inverse: Numerical experiments

A. Without consideration of joint limitations (planar manipulator)



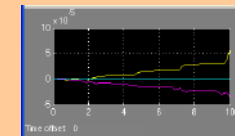
B. Consideration of joint limitations (planar manipulator)



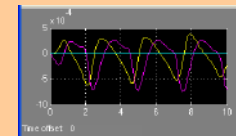
Application of pseudo-inverse:

Numerical experiments / without post-adjusting

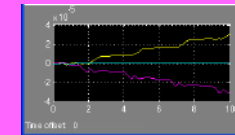
$$\dot{\mathbf{q}} = \mathbf{J}^+ \dot{\mathbf{x}}$$



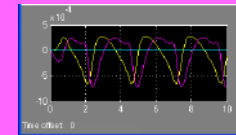
$$\dot{\mathbf{q}} = \mathbf{J}^+ (\dot{\mathbf{x}} + \mathbf{K}_e \mathbf{z}_e)$$



$$\dot{\mathbf{q}} = \mathbf{J}^+ \dot{\mathbf{x}} + [\mathbf{E} - \mathbf{J}^+ (\mathbf{q}) \mathbf{J} (\mathbf{q})] \mathbf{z}_0$$

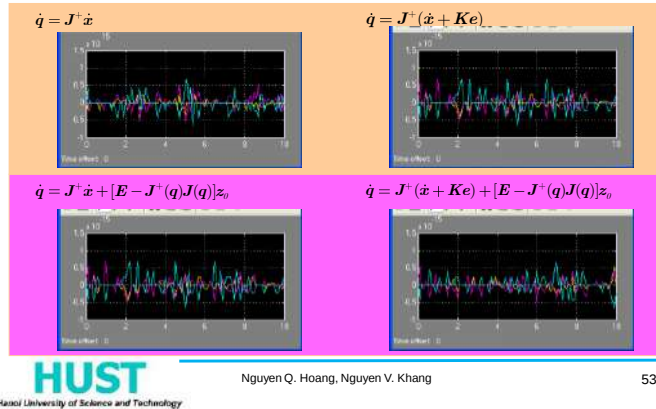


$$\dot{\mathbf{q}} = \mathbf{J}^+ (\dot{\mathbf{x}} + \mathbf{K}_e \mathbf{z}_e) + [\mathbf{E} - \mathbf{J}^+ (\mathbf{q}) \mathbf{J} (\mathbf{q})] \mathbf{z}_0$$



Application of pseudo-inverse:

Numerical experiments / with post-adjusting



53

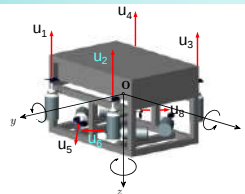
Application of pseudo-inverse: Hệ thống chân vịt đẩy

Quan hệ lực/mômen điều khiển và lực đẩy chân vịt

$$\tau = Bu$$

$$u = [u_1, u_2, \dots, u_{n_p}]^T$$

$$\tau = [F_x, F_y, F_z, m_x, m_y, m_z]^T$$



Để điều khiển được tất cả 6 dof của ROV

$$n_p = \dim(u) \geq \dim(\tau) = 6 \quad \& \quad \text{rank}(B) = 6$$

Yêu cầu bố trí các chân vịt sao cho ma trận B có hạng đầy đủ (full rank)

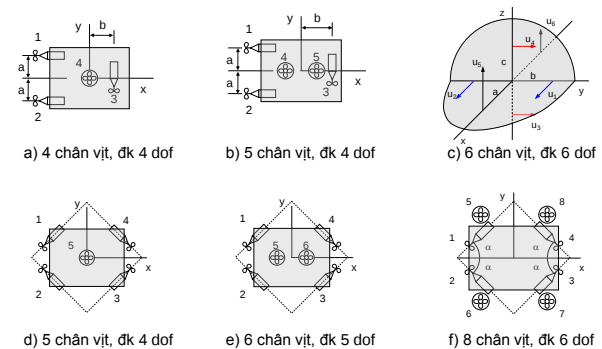
- $n_p < n_o$ tàu lặn thiếu hụt dẫn động
- $n_p = n_o$ tàu lặn có dẫn động đầy đủ, và
- $n_p > n_o$ tàu lặn có hệ dẫn động dư (có tính chất dự phòng)

Application of pseudo-inverse: Hệ thống chân vịt đẩy

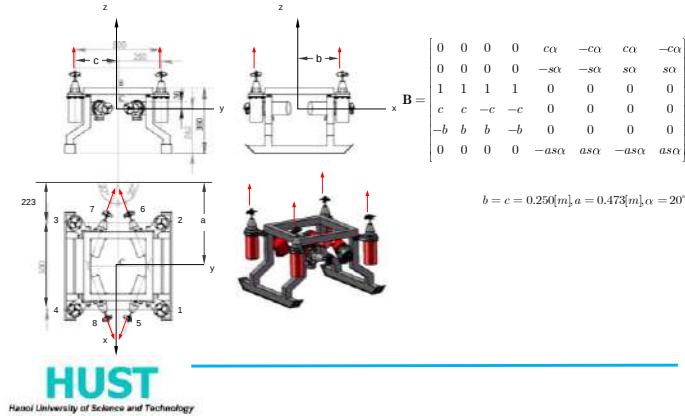
Hệ thống chân vịt đóng một vai trò quan trọng trong việc điều khiển tàu lặn, đặc biệt trong các trường hợp yêu cầu độ chính xác định vị cao. Do đó cần thiết phải xây dựng mô hình chính xác nhất có thể cho hệ thống này. Trong phần này trình bày việc đưa ra mối quan hệ giữa lực điều khiển và các lực đẩy của chân vịt và mô hình động lực của hệ thống lực đẩy chân vịt. Việc điều khiển chuyển động tàu lặn phụ thuộc vào kiểu dáng của từng loại tàu và cách bố trí các động cơ – chân vịt đẩy. Phần này trình bày việc lựa chọn số lượng và cách bố trí hệ thống đẩy cho tàu lặn cỡ nhỏ dạng ROV. Các ưu nhược điểm của các phương pháp bố trí được phân tích. Phần này cũng đề xuất một phương án phân bố tối ưu lực điều khiển cho các động cơ. Ngoài ra, sự phân bố lại được thực hiện khi một vài động cơ ngừng hoạt động cũng được chú ý nhằm đảm bảo để tàu làm việc liên tục.

Application of pseudo-inverse: Hệ thống chân vịt đẩy

Một số trường hợp bố trí hệ thống chân vịt



Hệ thống chân vịt đẩy



Application of pseudo-inverse: Hệ thống chân vịt đẩy

Quan hệ lực điều khiển – lực đẩy

$$\tau = Bu$$

Tiêu chuẩn tối ưu

$$J = \frac{1}{2} u^T u = \frac{1}{2} \|u\|_2^2 \rightarrow \min$$

Nghiệm tối ưu

$$u = B^\# \tau = B^T [BB^T]^{-1} \tau$$

Ma trận trạng thái động cơ

$$\Psi = \text{diag}\{a_1, a_2, \dots, a_n\}$$

$a_i = 1$ - động cơ i không hỏng

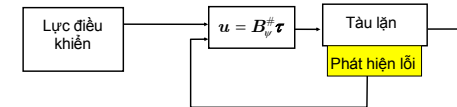
$a_i = 0$ - động cơ i hỏng (lỗi)

Phân bổ lại lực điều khiển

$$\tau = B\Psi u = B_\Psi u$$

$$u = \Psi B^T [B\Psi B^T]^{-1} \tau$$

Sơ đồ phân bổ lực



Application of pseudo-inverse: Hệ thống chân vịt đẩy

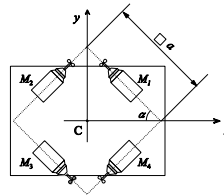
Ví dụ

$$B = \begin{bmatrix} -\cos \alpha & \cos \alpha & \cos \alpha & -\cos \alpha \\ \sin \alpha & \sin \alpha & -\sin \alpha & -\sin \alpha \\ a/2 & -a/2 & a/2 & -a/2 \end{bmatrix} \quad \tau = [X, Y, M]^T$$

$$u = [u_1, u_2, u_3, u_4]^T$$

Bốn động cơ-chân vịt làm việc bình thường

$$u = B^\# \tau = \frac{1}{2a \sin 2\alpha} \begin{bmatrix} -a \sin \alpha & a \cos \alpha & \sin 2\alpha \\ a \sin \alpha & a \cos \alpha & -\sin 2\alpha \\ a \sin \alpha & -a \cos \alpha & \sin 2\alpha \\ -a \sin \alpha & -a \cos \alpha & -\sin 2\alpha \end{bmatrix} \begin{bmatrix} X \\ Y \\ M \end{bmatrix}$$



Khi động cơ số 4 lỗi, ta có ma trận phát hiện lỗi như sau

$$\Psi = \text{diag}\{1, 1, 1, 0\}$$

$$B_\Psi = \frac{1}{a \sin 2\alpha} \begin{bmatrix} 0 & a \cos \alpha & \sin 2\alpha \\ a \sin \alpha & a \cos \alpha & 0 \\ a \sin \alpha & 0 & \sin 2\alpha \\ 0 & 0 & 0 \end{bmatrix}$$

$$u = B_\Psi^\# \tau$$

$$u = \begin{bmatrix} \frac{Y}{2 \cos \alpha} + \frac{M}{a} \\ \frac{X}{2 \cos \alpha} + \frac{Y}{2 \sin \alpha} \\ \frac{X}{2 \cos \alpha} + \frac{M}{a} \\ 0 \end{bmatrix}$$

Lecture 3

Lập trình trong Matlab

Nguyen Q. Hoang
Department of Applied Mechanics
Hanoi University of Science and Technology

HUST

Hanoi University of Science and Technology

1

Các kiểu dữ liệu

- Trong Matlab có nhiều kiểu dữ liệu, sau đây chỉ trình bày một số kiểu liên quan đến dạng thể hiện và lưu trữ. Để có thể nhận được sự trợ giúp trực tiếp từ Matlab, bạn cần tận dụng câu lệnh *help*.
 - Kiểu véc-tơ và ma trận
 - Chuỗi ký tự (ký tự, xâu - string)
 - Kiểu ô, mảng (Cell-Array)
 - Kiểu cấu trúc
- Kiểu véc-tơ và ma trận**
Biểu diễn số các ma trận hai hay nhiều chiều cũng như các véc-tơ trong chương 2 là một kiểu dữ liệu đặc biệt. Mỗi phần tử của véc-tơ hay ma trận theo chuẩn cần ô nhớ 8 Byte (class double) hoặc 4 Byte (class single). Các đại lượng phức cần một ô nhớ gấp đôi, phần thực và ảo được ghi riêng rẽ.
- Chuỗi ký tự (ký tự, xâu - string)**
Chuỗi ký tự được đặt trong cặp dấu nháy đơn trên
>> 'Day la chuoai ky tu'
Mỗi phần tử của chuỗi ký tự (Ma trận ký tự) cần ô nhớ 2 Byte.

HUST

Hanoi University of Science and Technology

3

Chương 3. Lập trình trong Matlab

3.1 Các kiểu dữ liệu

Kiểu véc-tơ và ma trận
Chuỗi ký tự (ký tự, xâu - string)
Kiểu ô, mảng (Cell-Array)
Kiểu cấu trúc

3.2 Soạn thảo Script file trong Matlab

Scripts
Các hàm, MATLAB - Function

3.3 Các vòng lặp và rẽ nhánh

Vòng lặp FOR
Vòng lặp WHILE
Lệnh if, cấu trúc if - else - end
Cấu trúc switch-case

3.4 Các Mat-File

HUST

Hanoi University of Science and Technology

2

Các kiểu dữ liệu

• Kiểu ô, mảng (Cell-Array)

- Các dữ liệu của các kiểu khác nhau, ví dụ chuỗi ký tự, các ma trận có cỡ khác nhau, có thể được sử dụng như các ô của một mảng. Các phần tử ô được gọi đến thông qua chỉ số của nó. Các phần tử của một ô được đặt trong dấu ngoặc nhọn {...}. Ví dụ

```
>> A(1,1)=[1 2 3; 4 2 6; 1 7 9];
>> A(1,2)='Ma tran thu';
>> A(2,1)=[3+8i];
>> A(2,2)=[0:pi/20:2*pi];
```

```
% gọi đến các phần tử
>> A(1,1)
ans = [3x3 double]
```

```
>> A{1,1} % ô (1,1)
ans =
    1     2     3
    4     2     6
    1     7     9
>> A{1,2} % ô (1,2)
ans = Ma tran thu
>> A(1,2)
ans = 'Ma tran thu'
```

HUST

Hanoi University of Science and Technology

4

Các kiểu dữ liệu

• Kiểu cấu trúc

- Cấu trúc được sử dụng để làm việc với các kiểu dữ liệu khác nhau. Tên của một cấu trúc gồm hai phần, tên cấu trúc trước dấu chấm (.) và tên trường trong cấu trúc sau dấu chấm, (struct_name.field_name). Các phần tử của cấu trúc được gọi đến qua tên và chỉ số.
- Cú pháp % tạo một cấu trúc
`struct=struct('name_1', value1, 'name_2', value2,...)`
 % tạo ra phần tử của cấu trúc
`struct.name`
 hoặc
`struct.name_1 = value_1;`
`struct.name_2 = value_2;`

Các kiểu dữ liệu

• Kiểu cấu trúc (ví dụ)

```
>> A=[1 2 3; 4 2 6; 1 7 9];
>> my_structur = struct('data', A, 'dimension',[3 3])
my_structur =
    data: [3x3 double]
    dimension: [3 3]

>> my_structur.dimension
ans =
     3     3

>> my_structur.data
ans =
     1     2     3
     4     2     6
     1     7     9

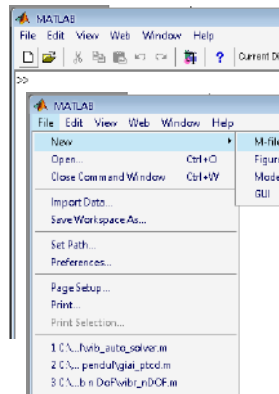
>> my_structur(2).data=inv(A) % mở rộng trường data
my_structur =
1x2 struct array with fields:
    data
    dimension
```

```
>> my_structur.data % in ra
ma tran A va inv(A)
ans =
     1     2     3
     4     2     6
     1     7     9

ans =
    4.0000   -0.5000   -1.0000
    5.0000   -1.0000   -1.0000
   -4.3333    0.8333    1.0000
```

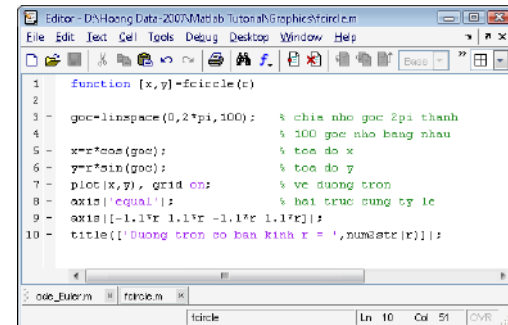
Soạn thảo Script file trong Matlab

- Matlab Script files là một chuỗi các lệnh được gõ vào trong cửa sổ soạn thảo và được ghi vào tệp có phần mở rộng là m (filename.m, hay được gọi là m-file). Để tạo m-file, trong menu chính bạn chọn New ==> M-file hoặc bạn nhấp chuột vào biểu tượng "tờ giấy trắng" trên thanh công cụ (hình 3-1).
- Việc chạy m-file tương đương với việc đánh toàn bộ các dòng lệnh trên cửa sổ lệnh tại dấu nhắc '>>' của Matlab. Các biến sử dụng trong m-file được đặt vào trong không gian làm việc của Matlab. Không gian làm việc này là trống khi khởi động Matlab, và nó sẽ chứa tất cả các biến được định nghĩa trong phiên làm việc. Muốn xóa tất cả các biến ta sử dụng lệnh `clear all`



Soạn thảo Script file trong Matlab

- Ví dụ soạn m-file vẽ đường tròn là hàm của bán kính r.



Soạn thảo Script file trong Matlab

Ví dụ về chương trình không có biến vào

```
function V = volume(r)

%r = input('Hay cho ban kinh cua hinh cau')
vol = (4/3)*pi*r^3;
disp('The tích hinh cau nay la:')
disp(vol)
```

Save với tên file **volume.m**, và gọi hàm trong command windows

```
>> volume
```

Soạn thảo Script file trong Matlab

Ví dụ về chương trình không có biến vào

```
function vol = volumef(r)
vol = (4/3)*pi*r^3;
```

Save với tên file **volumef.m**, và gọi hàm trong command windows

```
>> volumef(2)

function V = Tong_volume(r1, r2)
V1 = volumef(r1);
V2 = volumef(r2);
V = V1 + V2;

>> Tong_volume(2, 5)
```

Soạn thảo Script file trong Matlab

Ví dụ về script file

```
r1 = input('Hay cho ban kinh cua hinh cau')
r2 = input('Hay cho ban kinh cua hinh cau')
disp('Tong the tích hai hinh cau')
Tong_volume(r1, r2)
```

Save với tên file **TinhV.m**, và gọi hàm trong command windows

```
>> tinhV
```

Soạn thảo Script file trong Matlab

- Các m-file dạng hàm

```
function [output 1, output 2] = function_name(input1,input2)
% Some comments that explain what the function does go here.
```

```
global      x y z
persistent a b c
```

```
MATLAB command 1;
MATLAB command 2;
MATLAB command 3;
```

```
output 1 = ....;
output 2 = ....;
```

Soạn thảo Script file trong Matlab

- Biến toàn cục (**global variable**) và biến cục bộ hay biến địa phương (*local variable*). Biến toàn cục được khai báo theo cú pháp
`global var_1 var_2`
- Các biến toàn cục được xóa khỏi môi trường tính toán bằng lệnh *clear* [tên biến]. Biến cục bộ chỉ được sử dụng trong một hàm, biến này không được liệt kê trong cửa sổ *workspace*, và do đó không thể tác động đến được trong môi trường làm việc.
- Biến **persistent** trong một hàm có thể được thống nhất với cách khai báo
`persistent var_1 var_2`
- Trái với các đại lượng đã được khai báo bằng **global**, các đại lượng với khai báo **persistent** chỉ được biết đến trong hàm mà nó được khai báo. Do đó các hàm khác không thể tác động được đến biến này. Các biến **persistent** chỉ được xóa khi hàm chứa nó thoát ra khỏi bộ nhớ (**clear function_name**) hoặc khi hàm được thay đổi sau đó được ghi lại.

HUST

Hanoi University of Science and Technology

13

Các vòng lặp và rẽ nhánh

Ví dụ về lệnh for. Viết một hàm sử dụng vòng lặp for để tính tổng các phần tử của một vectơ v. Trên thanh công cụ, ta lựa chọn New → m-file để mở một cửa sổ soạn thảo, trong cửa sổ này ta soạn nội dung sau

```
function tong = tongvector(v)
% ham tinh tong cac phan tu cua mot vector
n=length(v); % tra lai so phan tu, chieu dai cua vector
s = 0; % khoi gan cho tong
for i = 1:n % hoac for i = 1 : 1 : n
    s = s + v(i);
end;
tong = s;
% save with file name tongvector.m
```

Sử dụng hàm

```
>> x=[1:2:50];
>> s=tongvector(x)
s = 625
```

HUST

Hanoi University of Science and Technology

15

Các vòng lặp và rẽ nhánh

Lệnh for	<code>for var = bien_chay các lệnh end</code>
Lệnh while	<code>while dieukien, các lệnh end</code>
Lệnh if (nếu .. thì ..)	<code>if dieukien ... <elseif> <else> end</code>
Lệnh switch (chuyển)	<code>switch .. case .. <otherwise ..> end</code>
Lệnh ngắt vòng lặp trong for và while	<code>break;</code>

HUST

Hanoi University of Science and Technology

14

Các vòng lặp và rẽ nhánh

Ví dụ về lệnh while. Viết hàm tính tổng $S = 1 + 1/2 + 1/3 + \dots + 1/100$

```
% su dung vong lap while
function tongS
n=100;
S=0; i=1;
while i<=n
    S = S+1/i;
    i = i+1;
end
tong=S;
disp('Voi n = 100, tong la :')
disp(tong)
```

```
>> >> tong_S
Voi n = 100, tong la :
5.1874
```

```
% su dung vong lap for
function tong_Sn
n=100;
S=0;
for i=1:n
    S=S+1/i;
end
tong=S;
disp('Voi n = 100, tong la :')
disp(tong)
```

```
>> >> tong_S
Voi n = 100, tong la :
5.1874
```

HUST

Hanoi University of Science and Technology

16

Các vòng lặp và rẽ nhánh

Sử dụng lệnh **while** giải bài toán. Ví dụ tìm số n lớn nhất sao cho tổng sau nhỏ hơn 27

$$S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} = \sum_{i=1}^n \frac{1}{i} < 27$$

```
function n = timn(S)
% tìm số n lớn nhất thỏa mãn
% tongS(n) = 1+1/2+1/3+...+1/n
< S
% giá trị S > 1 vào tu bàn phím
T=1;
while T
disp('Vào số dương S')
S = input('S = ');
if (S>1 & S<10) T = 0; end
end
format long
```

```
format long
tong=0; i=0;
while(tong<S)
i = i+1;
tong = tong+1/i;
end
n=i-1;
tong=tong-1/i;
disp('Số n tìm được là: ')
disp(n)
disp('Giá trị của tong S(n) là: ')
disp(tong)
```

HUST

Hanoi University of Science and Technology

17

Các vòng lặp và rẽ nhánh

Lệnh if, cấu trúc if - else - end

Lệnh điều kiện nếu - thì. Cú pháp

```
IF expression
statements 1
ELSEIF expression
statements 2
ELSE
statements 3
END
```

HUST

Hanoi University of Science and Technology

18

Các vòng lặp và rẽ nhánh

	% kết quả chương trình				
k=5;	>> A				
for i=1:k	A =				
for j=1:k	4	0	1	0	0
if i==j	0	4	0	1	0
A(i,j)=4;	1	0	4	0	1
elseif (abs(j-i)==2)	0	1	0	4	0
A(i,j)=1;	0	0	1	0	4
else					
A(i,j)=0;					
end					
end					
end					

HUST

Hanoi University of Science and Technology

19

Các vòng lặp và rẽ nhánh

Cấu trúc switch-case

Lệnh chuyển trong nhiều trường hợp dựa trên biểu thức và có dạng tổng quát như sau

```
SWITCH switch_expr
CASE case_expr,
statement, ..., statement
CASE {case_expr1, case_expr2, case_expr3,...}
statement, ..., statement
...
OTHERWISE,
statement, ..., statement
END
```

HUST

Hanoi University of Science and Technology

20

Các vòng lặp và rẽ nhánh

```
diem = 1; % 2, 3, 4, 5, 6
switch diem
case 1
    disp('Diem gioi')
case 2
    disp('Diem tot')
case 3
    disp('Diem kha')
case 4
    disp('Diem trung binh')
otherwise
    disp('Diem kem.')
end
```

Bài tập

- Viết một hàm Matlab hỏi người sử dụng bán kính và chiều cao của hình trụ, sau đó tính toán và đưa ra màn hình diện tích toàn phần của hình trụ và thể tích của hình trụ.
- Viết một chương trình con có sử dụng lệnh while để tính tổng $S(x,n)$

$$S(x,n) = 1 + x + x^2 + \dots + x^n$$

- Viết một chương trình con có sử dụng lệnh while để tính căn của $a > 0$ theo phương pháp lặp Newton-Raphson, theo công thức [heron]?

$$x_{n+1} = \frac{x_n^2 + a}{2x_n} = \frac{x_n}{2} + \frac{a}{2x_n} \quad \begin{array}{l} \text{Với } x(1) = 1, \text{ độ chính xác yêu cầu} \\ \text{là } 0.0000001 \end{array} \quad |x_{n+1} - x_n| < 10^{-7}$$

- Hãy sử dụng vòng lặp while để tìm thương và số dư khi chia hai số nguyên a cho b , $a > b$.

Các Mat-File

Mat-file là tệp có phần mở rộng là mat và do đó được gọi là tệp chấm mat (filename.mat). Mat-file là những tệp nhị phân được nén để lưu trữ kết quả số. Các file này được sử dụng để ghi các kết quả mà đã được tạo ra bởi chuỗi các lệnh, chỉ thị của Matlab. Chẳng hạn, để ghi giá trị của hai biến x_1 và x_2 trong file có tên dulieu.mat ta đánh dòng lệnh

```
>> x1=5;
>> x2=10;
>> save dulieu.mat x1 x2
```

Muốn tải (đọc) một mat-file vào MATLAB ta gõ dòng lệnh

```
>> load filename (or load filename.mat)
```

Viết thuật toán (thuật giải)

$$f(x) = x^2 - a = 0, \quad a > 0, \quad x = \sqrt{a}$$

$$k = 1; \quad x(k) = 1;$$

$$\begin{aligned} x_{k+1} &= x_k - \frac{f(x_k)}{f'(x_k)} \\ &= x_k - \frac{x_k^2 - a}{2x_k} = \frac{x_k^2 + a}{2x_k} = \frac{x_k}{2} + \frac{a}{2x_k} \\ &= \frac{1}{2} \left(x_k + \frac{a}{x_k} \right) \end{aligned}$$

Viết thuật toán (thuật giải)

1. Nhập $a > 0$;
2. Khởi động (khởi gán), $k = 1$, $x(1) = 1$; $ss = 1$; $eps = 1e-5$;
3. While $ss > eps$
 $x(k+1) = x(k)/2 + a/2/x(k)$;
 $ss = abs(x(k+1) - x(k))$;
 $k = k + 1$;
end
4. $x = x(k - 1)$

Lecture 4

Đồ họa trong Matlab

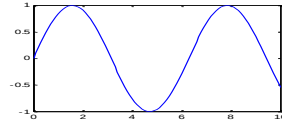
Nguyen Q. Hoang
Department of Applied Mechanics
Hanoi University of Science and Technology

Đồ thị trong mặt phẳng – 2D

- Trước hết ta bắt đầu với việc vẽ đồ thị hàm một biến $y = f(x)$. Công việc này trong Matlab bao gồm ba bước:
 - Định nghĩa hàm cần vẽ, $y = f(x)$
 - Xác định miền giá trị của biến, x thuộc miền $[a, \dots, b]$
 - Gọi hàm `plot(x,y)` của Matlab.
- Ví dụ cần vẽ đồ thị hàm $y = \sin(x)$ trong khoảng từ 0 đến 10.
 - Chia khoảng cần vẽ $[0, 10]$ bằng các điểm chia cách đều, sử dụng toán tử `(:)`, chẳng hạn $x = [a:h:b]$, với h là bước chia.
 - Tính giá trị của hàm $f(x)$ tại các điểm chia tương ứng, $y(i) = f(x(i))$; kết quả được hai véc tơ x và y có cùng số các phần tử.
 - Gọi hàm `plot(x,y)`.

Cụ thể:

```
>> x = [0:0.1:10];
>> y = sin(x);
>> plot(x,y)
```



Hãy thực hiện với $h = 0.01$; $h = 0.5$; $h = 1$. Nhận xét đồ thị.

Chương 4. Đồ họa trong Matlab

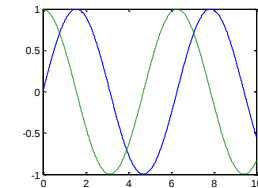
4.1 Đồ thị trong mặt phẳng - 2D

Đặt màu và kiểu đường cho đồ thị
Một số tùy chọn khi vẽ đồ thị 2D
Vẽ nhiều đồ thị trên cùng một hệ trục
Các lệnh axis
Đặt giới hạn miền vẽ với lệnh axis
Lệnh Subplots
Vẽ các đồ thị xếp chồng và lệnh `linspace`
Vẽ biểu đồ với lệnh `contour` – vẽ đường đồng mức
Thêm chú thích trên đồ thị
Vẽ đồ thị các hàm có điểm không xác định

4.2 Các lệnh vẽ trong không gian – 3D

Đồ thị trong mặt phẳng – 2D

- **Đặt màu và kiểu đường cho đồ thị**
 - Khi muốn thể hiện nhiều đồ thị trong một hình người ta thường phân biệt các đồ thị bằng các màu, và kiểu đường khác nhau (nét liền, nét đứt)
 - Trong Matlab đã định nghĩa sẵn các lựa chọn
 - Ví dụ:
 - `>> x=[0:0.01:10]; y1=sin(x);`
 - `>> y2=cos(x); plot(x,y1, x,y2,'--')`



Đồ thị trong mặt phẳng – 2D

Kiểu đường		Ký hiệu	Màu
- solid line	nét liền	k	black - đen
-- dashed line	nét đứt	b	blue - xanh da trời
-. dashdot line	nét chấm gạch	c	cyan - lục lam
: dotted line	nét chấm	g	green - xanh lá cây
(none) no line	(không hiện)	r	red - đỏ
		m	magenta- đỏ tươi
		y	yellow - vàng
		w	white - trắng
Kiểu đánh dấu (marker)			
. point	điểm		
o circle	vòng tròn		
x x-mark	chữ x		
+ plus	dấu cộng		
* star	dấu sao		
s square	dấu vuông		
.....			

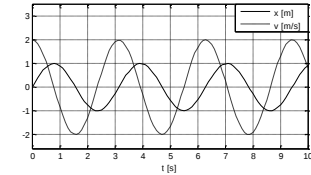
Sử dụng lệnh trợ giúp:
>> help plot

Sử dụng lệnh trợ giúp:
 >> help plot

Đồ thị trong mặt phẳng – 2D

- lệnh **legend** cho phép ta điền các chú giải lên hình vẽ

Ví dụ:
 >> t = [0:0.1:10];
 >> x = sin(2*t); v = 2*cos(2*t);
 >> plot(t,x,'k-', t,v,'k--');
 >> grid on, xlabel('t [s]')
 >> legend('x [m]', 'v [m/s]')
 >> axis([0 10 -2.6 3.5])

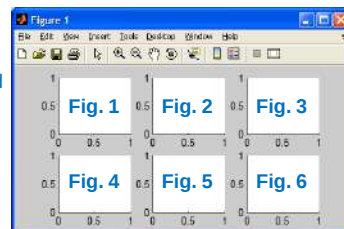


Đồ thị trong mặt phẳng – 2D

- Lệnh **subplot** - cho phép ta vẽ nhiều đồ thị trong nhiều hệ trục tọa độ vào cùng một hình vẽ. Lệnh này được gọi với cú pháp `subplot(m, n, p)`, trong đó m và n cho biết số hàng và số cột trong hình vẽ như là một ma trận hay mảng. Số p cho biết đồ thị sẽ được vẽ vào ô thứ mấy trong mảng. Thứ tự các ô trước hết theo chỉ số hàng.

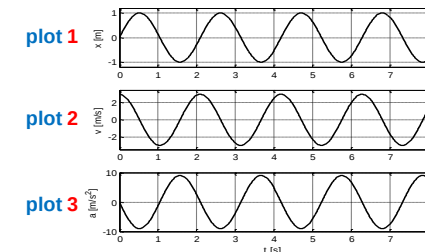
`subplot(m,n,p)` % p = 1,2,...mxn

>> for i = 1:6 subplot(2,3,i), end



Đồ thị trong mặt phẳng – 2D

Ví dụ:
 t = [0:0.01:8]; x = sin(3*t); v = 3*cos(3*t); a = -9*sin(3*t);
 subplot(3,1,1), plot(t,x,'k-', 'Linewidth',1), grid on, ylabel('x [m]'),
 subplot(3,1,2), plot(t,v,'k-', 'Linewidth',1), grid on, ylabel('v [m/s]'),
 subplot(3,1,3), plot(t,a,'k-', 'Linewidth',2), grid on, ylabel('a [m/s^2]'),
 xlabel('t [s]')



Vẽ hàm $y=y(x)$, biết $x = x(t)$, $y=y(t)$

$$x = a \sin(\omega t)$$

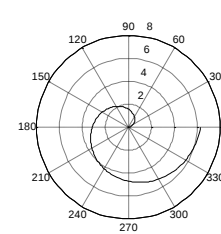
$$y = a\omega \cos(\omega t) \Rightarrow y = y(x)$$

Đồ thị trong mặt phẳng – 2D

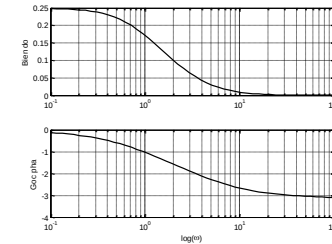
Các lệnh plot 2D	
<code>plot(x,y)</code>	Vẽ đồ thị bằng cách nối thẳng các điểm $[x(i),y(i)]$
<code>comet(x,y)</code>	Vẽ hoạt hình một quỹ đạo
<code>stair(x)</code> , <code>stairs(x,y,...)</code>	Vẽ đồ thị kiểu bậc thang
<code>stem(x,y)</code>	
<code>spy(matrix)</code>	hiển thị một ma trận thưa
<code>semilogx(x,y)</code>	Hiện thị $y(x)$, sử dụng thang logarith cho trục x
<code>semilogy(x,y)</code>	Hiện thị $y(x)$, sử dụng thang logarith cho trục y
<code>loglog(x,y)</code>	Hiện thị $y(x)$, sử dụng thang logarith cho trục x và y
<code>fplot(ham,mien)</code>	Vẽ hàm số trong miền
<code>hold <on off></code>	giữ đối tượng sẵn có, đóng lại

Đồ thị trong mặt phẳng – 2D

- Đồ thị theo tọa độ cực và Đồ thị với thang chia Logarith



Lệnh `polar(phi,r)`
`a = 1;`
`phi=[0:pi/90:2*pi];`
`r = a*phi;`
`polar(phi, r, '-k');`

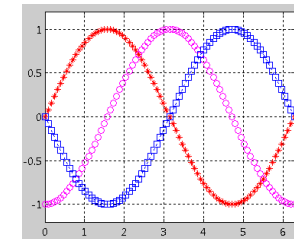


Ngoài ra ta có thể vẽ các đồ thị với lệnh ***semilogx*** hoặc ***semilogy***, với các lệnh này chỉ có một trục x hoặc y tương ứng sử dụng thang logarith còn trục kia sử dụng thang thập phân

Đồ thị trong mặt phẳng – 2D

Để vẽ nhiều đồ thị trong một cửa sổ đồ họa, ta sử dụng lệnh **hold on** ngay sau lệnh vẽ đồ thị đầu tiên. Lựa chọn này được dỡ bỏ nếu sau lệnh vẽ plot cuối cùng ta sử dụng lệnh **hold off**. Các giải thích trên về kiểu đường, màu sắc, đánh dấu (line style, colour, marker) được làm sáng tỏ trong thí dụ sau.

```
figure(1)
clf % xóa tất cả các đồ thị hiện có
t = [0:pi/40:2*pi];
plot(t,sin(t),'-.*'), % vẽ đồ thị 1
hold on
plot(t,sin(t-pi/2),'mo') % vẽ đồ thị 2
plot(t,sin(t-pi),'bs') % vẽ đồ thị 3
axis([0, 2*pi, -1.2, 1.2])
hold off
grid on
```



Đồ thị trong mặt phẳng – 2D

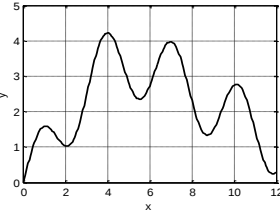
Khi vẽ đồ thị hàm phức tạp $f(x)$, ta có thể sử dụng các phép tính phần tử:

.*, ./, .^

Ví dụ:

$$y = e^{-0.40x} x^2 + \sin 2x$$

```
clc,
clear all
x=[0:0.05:12];
y=exp(-0.4*x).*x.^2+sin(2*x);
plot(x,y,'k-','linewidth',2),
grid on
xlabel('x'), ylabel('y')
```



Đồ thị trong mặt phẳng – 2D

Các trục tọa độ và các chú thích trên đồ thị

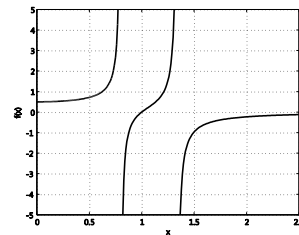
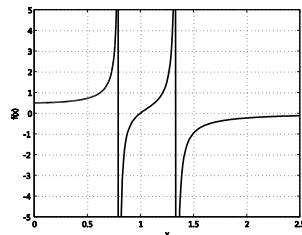
axis([xmin, xmax, ymin, ymax])	Điều chỉnh phạm vi các trục tọa độ
axis([-inf, xmax, ymin, ymax])	Tự động chọn xmin
grid <on off >	Sử dụng chia lưới hay không chia lưới
axis <off on auto equal square>	Nhập lựa chọn đối với các trục
xlabel(String)	Chú thích cho trục x
ylabel(String)	Chú thích cho trục y
title(String)	Hiện thị tên đồ thị
text(String)	Chèn chữ vào đồ thị
legend(string_1, string_2,... <position>)	Hiện thị các chú giải

Đồ thị trong mặt phẳng – 2D

Lưu ý khi vẽ đồ thị của hàm $f(x)$ có điểm không xác định. Ví dụ

$$f(x) = \frac{4(1-x^2)}{(10-7x^2)(1-x^2)-2}$$

$$x_1 = 0.7990, \quad x_2 = 1.3380$$



Đồ thị trong không gian – 3D

Các lệnh vẽ trong không gian – 3D

plot3(x,y,z <,plotstil>)	surfl(x,y,z <,color>)
comet3(x,y,z <,comet length>)	patch(x,y <,z> ,color)
mesh(x,y,z <,color>)	waterfall(x,y,z...<,...>,...)
surf(x,y,z <,color>)	contour3(x,y,z... <,...>,...)
surfc(x,y,z <,color>)	contour(x,y,z... <,...>,...)

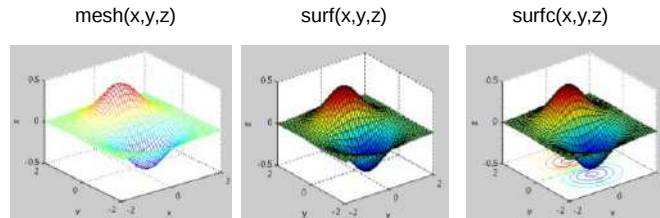
Ví dụ

Cần thể hiện đồ thị hàm $z = ye^{-x^2-y^2}$

trong miền $-2 \leq x, y \leq 2$

Đồ thị trong không gian – 3D

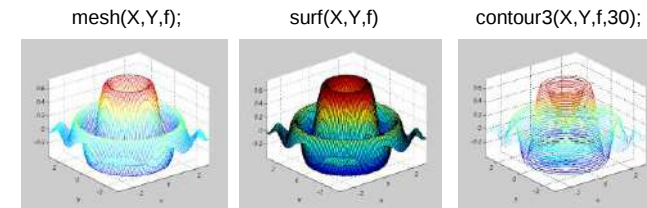
- `[x,y] = meshgrid(-2:0.1:2);`
- `z = y.*exp(-x.^2-y.^2);`
- `mesh(x,y,z), xlabel('x'), ylabel('y'), zlabel('z')`



Đồ thị trong không gian – 3D

- Ví dụ: $z = f(x, y) = e^{-0.2(x^2+y^2)} \cdot \sin(x^2 + y^2), \quad -3 \leq x, y \leq 3$

```
x = [-3 : 0.1 : 3]; y = [-3 : 0.1 : 3];
[X, Y] = meshgrid(x,y);
f = exp(-0.2*(X.^2+Y.^2)).*sin(X.^2+Y.^2);
```



Lecture 5

Nội suy và làm mịn đường cong

Nguyen Q. Hoang
Department of Applied Mechanics
Hanoi University of Science and Technology

Nêu bài toán

- Trong kỹ thuật khi đo đạc ta nhận được các số liệu rời rạc, từ các số liệu đo này ta cần biểu diễn các đường để biết được đặc tính của hệ khảo sát và có thể suy ra được những giá trị lân cận các giá trị đã đo. Matlab đã cung cấp sẵn các hàm cho phép đưa ra được các đường biểu diễn phù hợp nhất với bộ số liệu đã có. Trong chương này ta sẽ trình bày các kỹ thuật đơn giản thực hiện công việc đó.

- Mịn hóa bằng đa thức**

Lệnh **polyfit**(x,y,n) cho ta một đa thức bậc n dạng

$$p_n(x) = p_1 x^n + p_2 x^{n-1} + \dots + p_{n-1} x^2 + p_n x^1 + p_{n+1}$$

các hệ số của đa thức này được suy ra hai véc-tơ số liệu x và y, nhờ phương pháp sai số bình phương bé nhất (least mean square).

$$\mathbf{x} = [x_1 \ x_2 \ x_3 \ \dots \ x_m], \ \mathbf{y} = [y_1 \ y_2 \ y_3 \ \dots \ y_m] \quad m > n$$

Chương 5. Lập trình trong Matlab

5.1 Mịn hóa đường cong bằng đa thức

5.2 Nội suy đa thức

5.3 Bài tập thực hành

Phương pháp sai số bình phương bé nhất

Trường hợp đơn giản nhất, $n = 1$, đa thức có dạng một phương trình đường thẳng

$$y = ax + b$$

Theo phương pháp sai số bình phương bé nhất hai hệ số a và b được tìm như sau. Sai số giữa giá trị của số liệu đo $y(i)$ và giá trị tính theo đường thẳng vừa đưa ra được tính theo công thức

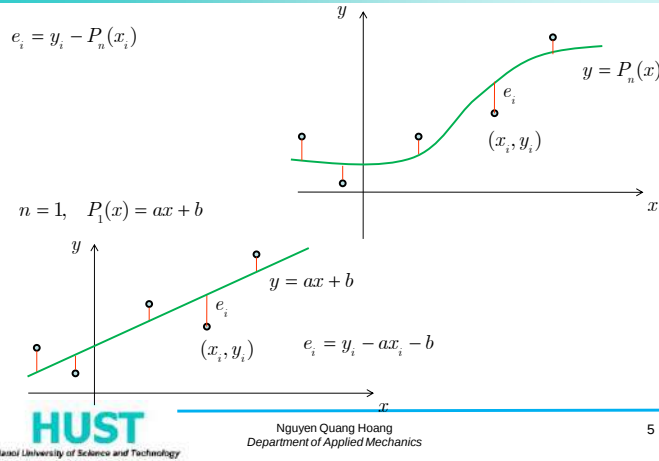
$$e_i = (ax_i + b) - y_i, \quad i = 1, 2, \dots, m$$

Như thế tổng bình phương các sai số là

$$S(a, b) = \sum_{i=1}^m e_i^2 = \sum_{i=1}^m (ax_i + b - y_i)^2$$

Bài toán đặt ra ở đây là tìm a, b sao cho hàm S đạt cực tiểu.

Phương pháp sai số bình phương bé nhất



Phương pháp sai số bình phương bé nhất

Đạo hàm S theo các biến a và b ta nhận được

$$\frac{\partial S}{\partial a} = \sum_{i=1}^m 2(ax_i + b - y_i)x_i = 0, \quad \frac{\partial S}{\partial b} = \sum_{i=1}^m 2(ax_i + b - y_i) = 0$$

$$a \sum_{i=1}^m x_i x_i + b \sum_{i=1}^m x_i - \sum_{i=1}^m x_i y_i = 0, \quad a \sum_{i=1}^m x_i + mb - \sum_{i=1}^m y_i = 0$$

Từ đây giải được

$$b = \frac{\bar{y} \sum_{i=1}^m x_i x_i - \bar{x} \sum_{i=1}^m x_i y_i}{\sum_{i=1}^m x_i x_i - m \bar{x}^2}, \quad a = \frac{\sum_{i=1}^m x_i y_i - m \bar{x} \bar{y}}{\sum_{i=1}^m x_i x_i - m \bar{x}^2}$$

với các giá trị trung bình

$$\bar{x} = \frac{1}{m} \sum_{i=1}^m x_i, \quad \bar{y} = \frac{1}{m} \sum_{i=1}^m y_i$$

$$a = \frac{\sum_{i=1}^m y_i (x_i - \bar{x})}{\sum_{i=1}^m x_i (x_i - \bar{x})}, \quad b = \bar{y} - \bar{x}a$$

Phương pháp sai số bình phương bé nhất

Sử dụng ma trận tựa nghịch đảo (nghịch đảo trái)

Quan hệ tuyến tính

$$x_i a + b = y_i, \quad i = 1, 2, \dots, m$$

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \dots & \dots \\ x_m & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_m \end{bmatrix} \Rightarrow \mathbf{A} \mathbf{p} = \mathbf{y}$$

$$\mathbf{p} = [\mathbf{A}^T \mathbf{A}]^{-1} \mathbf{A}^T \mathbf{y}$$

Phương pháp sai số bình phương bé nhất

Sử dụng ma trận tựa nghịch đảo (nghịch đảo trái)

Quan hệ bậc 2

$$x_i^2 p_1 + x_i p_2 + p_3 = y_i, \quad i = 1, 2, \dots, m$$

$$\begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ \dots & \dots & \dots \\ x_m^2 & x_m & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_m \end{bmatrix} \Rightarrow \mathbf{A} \mathbf{p} = \mathbf{y}$$

$$\mathbf{p} = [\mathbf{A}^T \mathbf{A}]^{-1} \mathbf{A}^T \mathbf{y}$$

Phương pháp sai số bình phương bé nhất

Ví dụ 1. Cho biết các số liệu đo được như sau

x	6	8	10	12	14	16	18	20	22	24
y	3.94	3.8	4.1	3.87	4.45	4.33	4.12	4.43	4.6	4.5

Nếu biểu diễn các số liệu trên bằng một đường thẳng, $y = a x + b$. Hãy tìm các giá trị của a , b và tìm giá trị của y khi $x = 11$.

Trước hết ta cần nhập hai vectơ số liệu x và y vào Matlab.

```
>> x = [6: 2 : 24]
```

```
x =
```

```
6 8 10 12 14 16 18 20 22 24
```

```
>> y = [3.94 3.8 4.1 3.87 4.45 4.33 4.12 4.43 4.6 4.5]
```

```
y =
```

```
3.94 3.80 4.10 3.87 4.45 4.33 4.12 4.43 4.60 4.50
```

Tiếp theo sử dụng lệnh `polyfit` để tìm các hệ số của đa thức

```
>> p = polyfit(x,y,1)
```

```
p = 0.0392 3.6267
```

Phương pháp sai số bình phương bé nhất

Theo cách biểu diễn đa thức bằng một vectơ bắt đầu từ hệ số ứng với số hạng có bậc lũy thừa lớn nhất, ta có

```
>> a=p(1)
```

```
a = 0.0392
```

```
>> b=p(2)
```

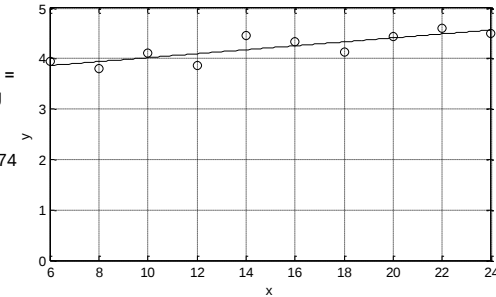
```
b = 3.6267
```

Giá trị cần tìm tại $x = 11$ được tính bằng

```
lệnh polyval(p,x)
```

```
>> polyval(p,11)
```

```
ans = 4.0574
```



Phương pháp sai số bình phương bé nhất

Xét tổng bình phương của sai số là

```
>> w = a*x+b;
```

```
>> e = w-y;
```

```
>> S1 = sum(e.*e)
```

```
S1 = 0.2274
```

Nếu các điểm trên được làm mịn bằng đường bậc 2:

```
>> n=2;
```

```
>> p = polyfit(x,y,n);
```

```
p =
```

```
-0.0002 0.0443 3.5940
```

```
>> a=p(1); b=p(2); c = p(3);
```

```
>> w = a*x.^2+b*x+c;
```

```
>> e = w-y;
```

```
>> S2 = sum(e.*e)
```

```
S2 = 0.2272
```

Ta thấy sai số có nhỏ hơn ($S2 < S1$) nhưng không đáng kể.

Mịn hóa bằng đa thức

Ví dụ 2. Cho bộ số liệu sau

x	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	5.0	6.0	6.1	7.0
y	300	281	261	244	228	214	202	191	181	164	151	149	141

Hãy tìm một đa thức bậc 3 phù hợp với số liệu cho. So sánh sai số so với trường hợp nếu sử dụng đa thức bậc hai.

Trước hết nhập số liệu vào hai vectơ và nối các điểm ta được

```
>> x = [0, 0.5, 1.0, 1.5, 2, 2.5, 3, 3.5, 4, 5, 6, 6.1, 7];
```

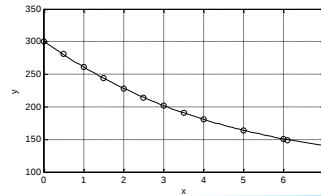
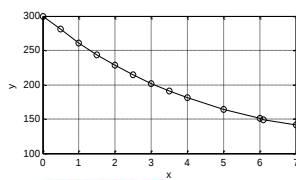
```
>> y = [300, 281, 261, 244, 228, 214, 202, 191, 181, 164, 151, 149, 141];
```

```
>> plot(x,y,'ko', x,y, 'k-'), grid on, xlabel('x'),  
ylabel('y')
```

Mịn hóa bằng đa thức

```
>> n=2; % Xấp xỉ bậc 2
>> p = polyfit(x,y,n)
>> a=p(1); b=p(2);
>> c = p(3);
>> w = a*x.^2+b*x+c;
>> e = w-y;
>> S2= sum(e.*e)
S2 = 15.4079
```

```
>> n=3; % Xấp xỉ đa thức bậc 3
>> p = polyfit(x,y,n)
>> a=p(1); b=p(2);
>> c = p(3); d = p(4);
>> w = a*x.^3+b*x.^2+c*x+d;
>> e = w-y;
>> S3= sum(e.*e)
S3 = 2.7933
```



Mịn hóa bằng hàm e mũ

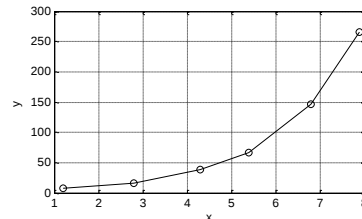
Ví dụ xét bộ số liệu sau

x	1.2	2.8	4.3	5.4	6.8	7.9
y	7.5	16.1	38.9	67.0	146.6	266.2

Nhập số liệu vào Matlab

```
>> x = [1.2 2.8 4.3 5.4 6.8 7.9];
>> y = [7.5 16.1 38.9 67.0 146.6 266.2];
```

```
>> plot(x,y,'ko', x,y,
'k-'),
grid on, xlabel('x'),
ylabel('y')
```



Mịn hóa bằng hàm e mũ

Trong nhiều trường hợp khi mịn hóa bằng đa thức không đáp ứng được yêu cầu đặt ra, ta cần phải tìm một phương án khác. Biểu diễn các số liệu thông qua hàm e mũ là một phương án nên được xem xét. Giả sử bộ số liệu (x,y) có thể lấy xấp xỉ bằng hàm

$$y = be^{ax}$$

Bây giờ ta cần tìm hai hệ số a và b để các điểm dữ liệu nằm gần đường cong nhất. Nếu áp dụng trực tiếp phương pháp sai số bình phương bé nhất ta sẽ được một hệ phương trình phi tuyến đối với các ẩn a và b. Để tránh điều đó, ta lấy logarit cơ số tự nhiên hai vế được

$$\ln y = ax + \ln b$$

Đặt $w = \ln(y)$, $z = x$ và $p_1 = a$ và $p_2 = \ln(b)$ bài toán trở thành tìm các hệ số p_1 , p_2 để xấp xỉ các số liệu $(z, w) = (x, \ln(y))$ bằng đường thẳng

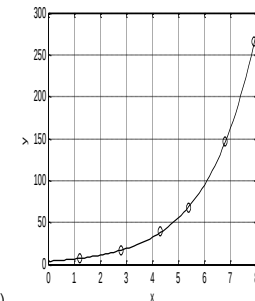
$$w = p_1 z + p_2$$

Mịn hóa bằng hàm e mũ

Xấp xỉ bằng hàm e mũ

```
>> p = polyfit(x,log(y),1)
p =
0.5366 1.3321
```

```
>> a = p(1); b=exp(p(2));
>> w = b*exp(a*x);
>> err = w-y;
>> S3= sum(err.*err)
Se = 17.6259
>> x3 = [0:0.1:8];
>> y3 = b*exp(a*x3);
>> plot(x,y,'ko', x3,y3, 'k-'),
grid on, xlabel('x'),ylabel('y')
```



Nội suy bằng đa thức

Lệnh **polyfit(x,y,n)** cho ta một đa thức bậc n mà đồ thị của nó là đường gần với đường tạo ra khi nối các điểm cho bởi bộ số liệu, còn nội suy đa thức cũng cho ta một đường cong nhưng đi qua các điểm số liệu đó. Có nhiều phương pháp để tạo ra đa thức đi qua các điểm cho như phương pháp nội suy theo công thức Lagrange, phương pháp Newton, phương pháp Neville, phương pháp nội suy với đường spline bậc 3, Chi tiết về các phương pháp này có thể xem trong các tài liệu về giải tích số và phương pháp số.

Trong phần này chỉ trình bày việc sử dụng các lệnh **interp** của Matlab để thực hiện tìm các giá trị nội suy từ các số liệu sẵn có. Bài toán đặt ra là cho biết các giá trị tương ứng của hai đại lượng x và y như trong bảng số liệu

x	x_1	x_2	x_3	...	x_n
y	y_1	y_2	y_3	...	y_n

Nội suy bằng đa thức

Nếu giá trị x^* nằm ngoài khoảng $[x_1, x_n]$, khi đó ta gọi phép tính giá trị $y^* = y(x^*)$ là phép tính ngoại suy và sử dụng lệnh với cú pháp:

y_star = interp1(x, y, x_star, method, 'extrap')

Sau đây xét ví dụ:

```
>> x = 1:20; y = sin(x);
>> x_star = 1.5;
>> y_star = interp1(x,y, x_star)
y_star = 0.8754
>> y_star = interp1(x,y, x_star, 'linear')
y_star = 0.8754
>> y_star = interp1(x,y, x_star, 'cubic')
y_star = 0.9008
>> y_star = interp1(x,y, x_star, 'nearest')
y_star = 0.9093
>> y_star = interp1(x,y, x_star, 'spline')
y_star = 1.0200
```

Nội suy bằng đa thức

Hãy tìm giá trị của biến y^* ứng với giá trị nào đó của biến x^* với $x_1 \leq x^* \leq x_n$. Bài toán này trong Matlab được giải quyết dễ dàng với lệnh **interp**. Cách gọi lệnh này như sau:

y_star = interp1(x,y,x_star)

Nếu x là véc-tơ gồm các phần tử nguyên từ 1 đến n ($x = 1:n$), với $n = \text{length}(y)$, thì ta có thể gọi lệnh

y_star = interp1(y,x_star)

Để nói rõ hơn phương pháp nào được sử dụng trong việc nội suy này ta sử dụng cú pháp

y_star = interp1(x, y, x_star, method)

với **method** được chọn là một trong các phương pháp sau

'nearest' - nội suy theo lân cận gần nhất

'linear' - nội suy tuyến tính

'spline' - nội suy sử dụng đường spline bậc 3

'pchip' - nội suy bậc 3 từng đoạn

'cubic' - tương tự như 'pchip'

Nội suy bằng đa thức

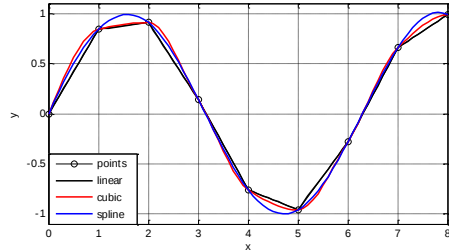
Như thế các phương pháp nội suy khác nhau sẽ cho ta các kết quả khác nhau. Để thấy rõ hơn kết quả của các phương pháp nội suy, ta xét ví dụ sau

```
x = 0:8; y = sin(x); plot(x,y,'-ko'), hold on
xi = 0:0.1:8;
y1 = interp1(x,y, xi, 'linear');
y2 = interp1(x,y, xi, 'cubic');
y3 = interp1(x,y, xi, 'spline');
```

```
plot(xi,y1,'-k','linewidth',1.5)
plot(xi,y2,'-r','linewidth',1.5)
plot(xi,y3,'-b','linewidth',1.5), grid on
```

```
xlabel('x'), ylabel('y'), axis([0 8, -1.1 1.1])
legend('points','linear','cubic','spline')
```

Nội suy bằng đa thức



Bài tập

3. Bảng số liệu dưới đây cho biết khối lượng m (kg) và suất tiêu hao nhiên liệu y (km/lit) của một số xe ô tô sản xuất bởi Ford và Honda năm 1999. Tìm phương trình đường thẳng $y = a.m + b$ thể hiện quan hệ tuyến tính giữa khối lượng và suất tiêu hao nhiên liệu.

Model	m (kg)	y (km/lit)
Contour	1310	10.2
Crown	1810	8.1
Escort	1175	11.9
Expedition	2360	5.5
Explorer	1960	6.8
F-150	2020	6.8
Ranger	1755	7.7
Taurus	1595	8.9
Accord	1470	9.8
CR-V	1430	10.2
Civic	1110	13.2
Passport	1785	7.7

Bài tập

2. Tìm phương trình đường thẳng $y = a.x + b$ thể hiện quan hệ tuyến tính của bộ số liệu cho trong bảng

x	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
y	3.076	2.810	2.588	2.297	1.981	1.912	1.653	1.478	1.399	1.018	0.794

4. Tỷ trọng tương đối của không khí thay đổi theo chiều cao đo được như trong bảng.

h (km)	0	1.525	3.050	4.575	6.10	7.625	9.150
ρ	1	0.8617	0.7385	0.6292	0.5328	0.4481	0.3741

Tìm phương trình bậc hai thể hiện quan hệ giữa độ cao và tỷ trọng không khí, từ đó tính tỷ trọng của không khí ở độ cao $h = 10.5$ km.

$$\rho = a + bh + ch^2$$

Bài tập

5. Độ nhớt động học μ_k của nước thay đổi theo nhiệt độ T được cho như trong bảng dưới đây

Xác định đường bậc ba thể hiện quan hệ này, từ đó tính độ nhớt động học tại các nhiệt độ $T = 10^\circ, 30^\circ, 60^\circ, \text{ và } 90^\circ \text{ C}$

T ($^\circ\text{C}$)	0	21.1	37.8	54.4	71.1	87.8	100
μ_k ($10^{-3} \text{ m}^2/\text{s}$)	1.79	1.13	0.696	0.519	0.338	0.321	0.296

Lecture 6

Giải phương trình vi phân trong Matlab

Nguyen Q.Hoang
Department of Applied Mechanics
Hanoi University of Science and Technology



Nguyen Quang Hoang
Department of Applied Mechanics

1

Chương 6. Lập trình trong Matlab

6.1 Giải phương trình vi phân bậc nhất với ode23 và ode45

6.2 Giải các phương trình vi phân bậc hai

6.3 Bài tập thực hành



Nguyen Quang Hoang
Department of Applied Mechanics

2

Ví dụ giải phương trình vi phân thường

Tìm hàm $y(t)$ thỏa mãn phương trình vi phân

$$\frac{dy}{dt} = \cos(t) \quad \text{với điều kiện đầu} \quad y(0) = 2$$

Với phương trình đơn giản này ta có thể viết ra được nghiệm chính xác (nghiệm giải tích),

$$y(t) = \sin(t) + C$$

với C là hằng số tích phân phụ thuộc vào điều kiện đầu. Từ điều kiện đầu nhận được hằng số C :

$$y(0) = \sin(0) + C = 2 \Rightarrow C = 2$$

$$y(t) = \sin(t) + 2$$



Nguyen Quang Hoang
Department of Applied Mechanics

3

Giải số bằng lệnh ode23

Cú pháp lệnh ode23

`[t, y] = ode23('funct_name', [t0, t_end], y0)`

Tên m-file chứa phương trình vi phân cần giải

Khoảng thời gian cần giải t từ t_0 đến t_{end}

Kết quả tính được ghi vào hai vectơ t và y chứa các cặp $[t(i), y(t_i)]$

Giá trị điều kiện đầu



Nguyen Quang Hoang
Department of Applied Mechanics

4

Giải số bằng lệnh ode23

Cú pháp lệnh ode23

```
[t, y] = ode23('func_name', [t0:h:t_end], y0)
```

Tên m-file chứa phương trình vi phân cần giải

Khoảng thời gian cần giải t từ t0 đến t_end, với h là bước thời gian

Kết quả tính được ghi vào hai vectơ t và y chứa các cặp [t(i), y(t_i)]

Giá trị điều kiện đầu

Ví dụ giải phương trình vi phân thường

Ví dụ: sử dụng ode23 giải phương trình vi phân

$$\frac{dy}{dt} = \cos(t) \quad \text{với điều kiện đầu} \quad y(0) = 2$$

Trong khoảng thời gian t = [0 2*pi]

Trước hết tạo m-file chứa phương trình vi phân cần giải:

```
function ydot = eq1(t,y)
    ydot = cos(t);
end
```

Ghi lại với tên file là tên hàm eq1.m

Sử dụng lệnh ode23:

```
>> [t, y] = ode23('eq1', [0, 2*pi], 2)
```

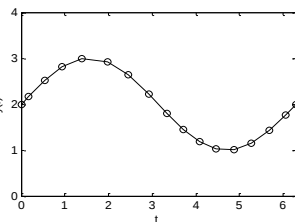
Ví dụ giải phương trình vi phân thường

- Vẽ kết quả và so sánh với nghiệm chính xác

```
>> f = sin(t) + 2;
>> plot(t, f, 'k-'), hold on
>> axis([0 2*pi 0 4])

>> plot(t, y, 'ko')

>> xlabel('t'),
>> ylabel('y(t)')
```



Giải số bằng lệnh ode45

Cú pháp lệnh ode45

```
[t, y] = ode45('func_name', [t0, t_end], y0)
```

Tên m-file chứa phương trình vi phân cần giải

Khoảng thời gian cần giải t từ t0 đến t_end

Kết quả tính được ghi vào hai vectơ t và y chứa các cặp [t(i), y(t_i)]

Giá trị điều kiện đầu

Giải số bằng lệnh ode45

Ví dụ: sử dụng ode45 giải phương trình vi phân

$$\frac{dy}{dt} = -5y + 5F(t) \quad \text{với điều kiện đầu} \quad y(0) = 0$$

$$F(t) = te^{-t/tc} \cos(\omega t) \quad tc = 0.1, \quad \omega = 10$$

Trong khoảng thời gian $t = [0 \quad 30]$

```
function ydot = eq2(t,y)
    tc = 0.1; w = 10;
    Ft = t*exp(-t/tc)*cos(w*t);
    ydot = -5*y + 5*Ft;
end
```

Ghi lại với tên
file là tên
hàm eq2.m

```
>> [t, y] = ode45('eq2', [0, 30], 0);
>> % [t, y] = ode45('eq2', [0:0.01:30], 0);
>> plot(t,y)
```

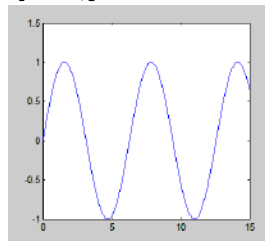
HUST

Nguyen Quang Hoang
Department of Applied Mechanics

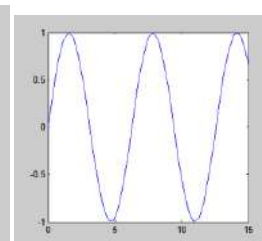
9

Giải số bằng lệnh ode23

```
>> [t, y2] = ode45('eq1', [0:0.01:30], 0);
>> plot(t,y2)
```



y2(t)



y(t)

HUST

Nguyen Quang Hoang
Department of Applied Mechanics

10

Giải số bằng lệnh ode45

Ví dụ: sử dụng ode45 giải phương trình vi phân

$$\frac{dy}{dt} = -5y + 5F(t) \quad \text{với điều kiện đầu} \quad y(0) = 0$$

$$F(t) = te^{-t/tc} \cos(\omega t) \quad tc = 0.1, \quad \omega = 10$$

Trong khoảng thời gian $t = [0 \quad 30]$

```
function ydot = odevidu3(t,y, tc, w)
    Ft = t*exp(-t/tc)*cos(w*t);
    ydot = -5*y + 5*Ft;
end
```

Ghi lại với tên
file là tên hàm
odevidu3.m

```
>> [t,y] = ode45(@odevidu3,t,y,0.1,10), [0:0.01:30], 0);
>> plot(t,y)
```

HUST

Nguyen Quang Hoang
Department of Applied Mechanics

11

Hệ phương trình vi phân cấp 1

Hệ n phương trình vi phân cấp 1.

$$\dot{y}_1 = \frac{dy_1}{dt} = f_1(t, y_1, \dots, y_n)$$

$$\dot{y}_2 = \frac{dy_2}{dt} = f_2(t, y_1, \dots, y_n)$$

$$\dots$$

$$\dot{y}_n = \frac{dy_n}{dt} = f_n(t, y_1, \dots, y_n)$$

Các điều kiện đầu

$$y_1(0) = y_{10}$$

$$y_2(0) = y_{20}$$

$$\dots$$

$$y_n(0) = y_{n0}$$

Hay viết ở dạng véc tơ

$$\dot{\mathbf{y}} = \frac{d\mathbf{y}}{dt} = \mathbf{f}(t, \mathbf{y}), \quad \mathbf{y}(0) = \mathbf{y}_0$$

$$\mathbf{y} = [y_1, y_2, \dots, y_n]^T, \quad \mathbf{f} = [f_1, f_2, \dots, f_n]^T$$

HUST

Nguyen Quang Hoang
Department of Applied Mechanics

12

Phương trình vi phân cấp cao

Xét 1 phương trình vi phân cấp n.

$$x^{(n)} = \frac{d^n x}{dt^n} = f(t, x, \dot{x}, \ddot{x}, \dots, x^{(n-1)}) \quad \text{Các điều kiện đầu}$$

$$\begin{aligned} x(0) &= x_0 \\ \dot{x}(0) &= \dot{x}_0 \\ &\dots \\ x^{(n-1)}(0) &= x_0^{(n-1)} \end{aligned}$$

Đặt các biến mới và hạ bậc

$$\begin{aligned} y_1 &= x, & \dot{y}_1 &= \dot{x} = y_2 \\ y_2 &= \dot{x}, & \dot{y}_2 &= \ddot{x} = y_3 \\ &\dots & &\dots \\ y_{n-1} &= x^{(n-2)}, & \dot{y}_{n-1} &= x^{(n-1)} = y_n \\ y_n &= x^{(n-1)}, & \dot{y}_n &= x^{(n)} = f(t, y_1, \dots, y_n) \end{aligned}$$

Hay viết gọn lại dạng véc tơ

$$\begin{aligned} \dot{\mathbf{y}} &= \frac{d\mathbf{y}}{dt} = \mathbf{f}(t, \mathbf{y}), & \mathbf{y}(0) &= \mathbf{y}_0 \\ \mathbf{y} &= [y_1, y_2, \dots, y_n]^T, & \mathbf{f} &= [y_2, y_3, \dots, y_n, f(t, \mathbf{y})]^T \end{aligned}$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

13

Phương trình vi phân cấp cao

Ví dụ hạ bậc phương trình vi phân cấp 2 sau

$$\ddot{x} = f(t, x, \dot{x}), \quad f(t, x, \dot{x}) = -b\dot{x} - cx + F_0 \sin \Omega t$$

Đặt các biến mới và hạ bậc

$$\begin{aligned} y_1 &= x, & \dot{y}_1 &= \dot{x} = y_2 \\ y_2 &= \dot{x}, & \dot{y}_2 &= \ddot{x} = f(t, y_1, y_2) \end{aligned}$$

Cụ thể trong trường hợp trên

$$\begin{aligned} \dot{y}_1 &= y_2 \\ \dot{y}_2 &= -by_2 - cy_1 + F_0 \sin \Omega t \end{aligned} \quad \Rightarrow \quad \dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y})$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \quad \mathbf{f}(t, \mathbf{y}) = \begin{bmatrix} y_2 \\ -by_2 - cy_1 + F_0 \sin \Omega t \end{bmatrix}$$

HUST

Hanoi University of Science and Technology

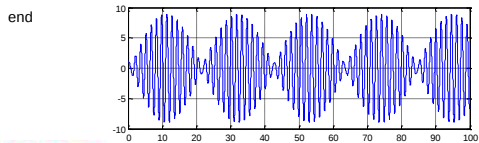
Nguyen Quang Hoang
Department of Applied Mechanics

14

```
function ydot = daodong1dof(t,y)
```

```
% t=1; y = [1; 2]
m = 1; b = 0.0; c = 16; >> [t, y] = ode45('daodong1dof',
F0 = 10; Ome = 4.1; >> [0:0.01:100], [1, 2])
>> plot(t,y(:,1))

ydot = zeros(2,1);
Ft = F0*sin(Ome*t);
ydot(1) = y(2);
ydot(2) = 1/m*(Ft - b*y(2) - c*y(1));
```



HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

15

Ví dụ hệ dao động 1dof

$$m\ddot{x} + c\dot{x} + kx = F_0 \sin(\Omega t) \quad \Rightarrow \quad \begin{aligned} \dot{y}_1 &= \dot{x} = y_2 \\ \dot{y}_2 &= \ddot{x} = \frac{1}{m} F_0 \sin(\Omega t) - cy_2 - ky_1 \end{aligned}$$

```
function ydot = vibr1dof(t,y, m,c,k,F0, Ome)
ydot = zeros(2,1);
```

```
Ft = F0*sin(Ome*t);
ydot(1) = y(2);
ydot(2) = 1/m*(Ft - c*y(2) - k*y(1));
```

```
end
```

```
>> [t, y] = ode45(@(t,y)vibr1dof(t,y, m,c,k,F0, Ome), [0:0.01:100], [1, 2])
>> [t, y] = ode45(@(t,y)vibr1dof(t,y, 1, 0.1,16,10, 4.1), [0:0.01:100], [1, 2])
>> plot(t,y(:,1))
```

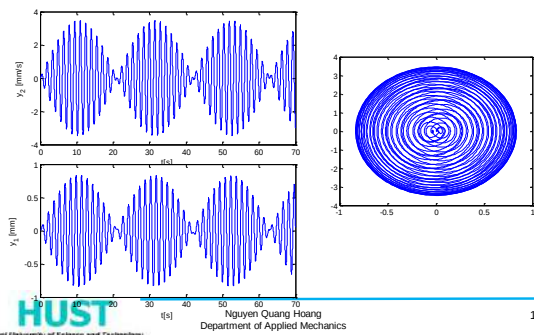
HUST

Hanoi University of Science and Technology

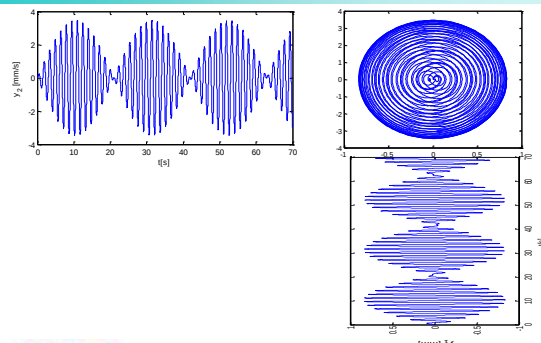
Nguyen Quang Hoang
Department of Applied Mechanics

16

```
>> [t, y] = ode45(@vibr1dof(t,y), 1, 0.0, 16, 1, 4.3), [0:0.01:70], [0,0]);
>> plot(t,y(:,1)), ylabel('y_1 [mm]'), xlabel('t[s]')
```



17



18

Giải hệ phương trình vi phân cấp 1

Trước hết xét việc giải hệ hai phương trình vi phân cấp một sau

$$\frac{dx}{dt} = -x^2 + y, \quad \frac{dy}{dt} = -x - xy \quad \text{điều kiện đầu} \quad x(0) = 0, y(0) = 1$$

Viết lại hệ dạng véc tơ như sau:

$$x_1 = x, \quad x_2 = y \Rightarrow \frac{dx_1}{dt} = -x_1^2 + x_2, \quad \frac{dx_2}{dt} = -x_1 - x_1 x_2$$

Để giải hệ trên bằng ode45, ta viết một m-file thể hiện về phải của hệ trên

```
function xdot = eqx(t,x);
xdot = zeros(2,1);
xdot(1) = -x(1)^2 + x(2);
xdot(2) = -x(1) - x(1)*x(2);
end % save with file name eqx.m
```

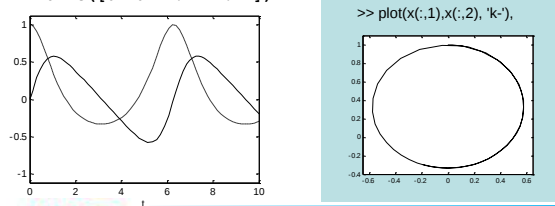
19

Giải hệ phương trình vi phân cấp 1

```
>> [t,x] = ode45('eqx', [0 10], [0,1]);
```

Nhiệm cần tìm x được ghi lại bằng hai vectơ cột, cột thứ nhất x1 = x(:,1) và cột thứ hai x2 = x(:,2). Với lệnh vẽ

```
>> plot(t,x(:,1),t,x(:,2),'--'), xlabel('t'),
>> axis([0 10 -1.12 1.12])
```



20

Giải hệ phương trình vi phân cấp 1

Ví dụ dưới đây trình bày việc giải phương trình vi phân bậc hai. Hãy tìm nghiệm của phương trình vi phân sau

$$\ddot{y} + 16y = \sin(4.3t) \quad \text{điều kiện đầu} \quad y(0) = 0, \dot{y}(0) = 0$$

Bằng cách đặt $x_1 = y$, $x_2 = \dot{y}$ ta nhận được hệ hai ptvp cấp một

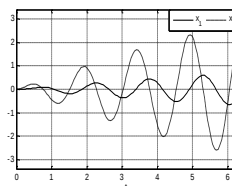
$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \sin(4.3t) - 16x_1 \end{aligned} \quad \longrightarrow \quad \dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x})$$

Về phải của hệ trên được thể hiện trong một m-file như sau:

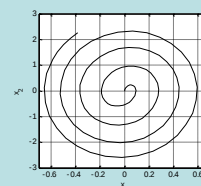
```
function xdot = eqx2(t,x);
xdot = zeros(2,1);
xdot(1) = x(2);
xdot(2) = sin(4.3*t)-16*x(1);
end
% save with file name eqx2.m
```

Giải hệ phương trình vi phân cấp 1

```
>> [t,x] = ode45('eqx2',[0 2*pi],[0,0]);
>> plot(t,x(:,1),t,x(:,2),'--'), xlabel('t'),
>> axis([0 2*pi -3 3])
```

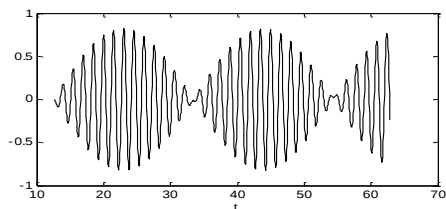


```
>> plot(x(:,1), x(:,2),'k-'),
>> xlabel('x_1'), ylabel('x_2')
```



Giải hệ phương trình vi phân cấp 1

```
>> [t,x] = ode45('eqx2',[4*pi 20*pi],[0,0]);
>> plot(t,x(:,1)), xlabel('t')
```



Giải hệ phương trình vi phân cấp 1

TT	Lệnh solver	Dạng bài toán	Độ chính xác	Trường hợp áp dụng
1	ode45	không cứng (nonstiff)	trung bình	Hay được sử dụng, là sự lựa chọn đầu tiên (sử dụng công thức R-K45)
2	ode23	không cứng (nonstiff)	thấp	Sử dụng khi không cần độ chính xác cao, hoặc khi giải bài toán cứng vừa phải. (sử dụng công thức R-K23)
3	ode113	không cứng (nonstiff)	thấp đến cao	(sử dụng công thức Adams-Bashforth-Moulton)
4	ode15s	cứng (stiff)	thấp đến trung bình	sử dụng khi ode45 cho độ chính xác không cao do bài toán cứng

Giải hệ phương trình vi phân cấp 1

TT	Lệnh solver	Dạng bài toán	Độ chính xác	Trường hợp áp dụng
5	ode23s	cứng (stiff)	thấp	Khi độ chính xác không của hệ cứng và khi ma trận khối lượng là hằng số
6	ode23t	cứng vừa phải (moderately stiff)	thấp	Giải phương trình có độ cứng vừa phải, nghiệm không giảm dần, có thể giải được hệ DAEs
7	ode23tb	cứng (stiff)	thấp	giải hệ cứng với độ chính xác không cao

Ví dụ về hệ cứng (stiffness system). Ref. Chapter 5. Solving Diff. Equations

$$\dot{\mathbf{y}} + \mathbf{A}\mathbf{y} = 0, \quad \mathbf{y}(0) = \mathbf{y}_0$$

Ma trận $\mathbf{A}$ có các trị riêng phân biệt.

$$\mathbf{y}(t) = \sum_i C_i \mathbf{v}_i \exp(-\lambda_i t)$$

Ví dụ

$$\ddot{y} + 1001\dot{y} + 1000y = 0 \quad \begin{cases} \dot{y}_1 = y_2 \\ \dot{y}_2 = -1000y_1 - 1001y_2 \end{cases} \quad \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = - \begin{bmatrix} 0 & -1 \\ 1000 & 1001 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 0 & -1 \\ 1000 & 1001 \end{bmatrix} \quad |\mathbf{A} - \lambda \mathbf{I}| = \begin{vmatrix} -\lambda & -1 \\ 1000 & 1001 - \lambda \end{vmatrix} = 0$$

$$\lambda_1 = 1, \quad \lambda_2 = 1000$$

$$h < 2 / \lambda_{\max}$$

Bài tập

Kết hợp m-file và lệnh ode45 giải các phương trình vi phân sau:

eqbt1.m	$\frac{dy}{dt} = -2.3y, \quad y(0) = 0, \quad t_{\text{end}} = 10, \quad \Delta t = 0.01$
eqbt2.m	$\frac{dx_1}{dt} = 2x_1x_2, \quad \frac{dx_2}{dt} = -x_1^2, \quad x_1(0) = x_2(0) = 0, \quad t_{\text{end}} = 10$
eqbt3.m	$\frac{dx_1}{dt} = x_2, \quad \frac{dx_2}{dt} = -x_1, \quad x_1(0) = x_2(0) = 1, \quad t_{\text{end}} = 10$
eqbt4.m	$\frac{dy}{dt} = -ty + 1, \quad y(0) = 1, \quad t_{\text{end}} = 10, \quad \Delta t = 0.01$
eqbt5.m	$\frac{d^2y}{dt^2} - 2\frac{dy}{dt} + y = e^{-t}, \quad y(0) = 2, \quad \dot{y}(0) = 0, \quad t_{\text{end}} = 20$

Bài tập

Phương trình vi phân dao động cưỡng bức của hệ tuyến tính n dof

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{f}(t), \quad \mathbf{q}(0) = \mathbf{q}_0, \quad \dot{\mathbf{q}}(0) = \dot{\mathbf{q}}_0 \quad \text{eqbt7.m}$$

Bằng cách hạ bậc ta nhận được

$$\dot{\mathbf{q}} = \mathbf{v}, \quad \dot{\mathbf{v}} = \mathbf{M}^{-1}(\mathbf{f}(t) - \mathbf{C}\mathbf{v} - \mathbf{K}\mathbf{q}) \quad \mathbf{y} = \begin{bmatrix} \mathbf{q} \\ \mathbf{v} \end{bmatrix}, \quad \dot{\mathbf{y}} = \begin{bmatrix} \mathbf{v} \\ \mathbf{M}^{-1}(\mathbf{f}(t) - \mathbf{C}\mathbf{v} - \mathbf{K}\mathbf{q}) \end{bmatrix}$$

Hãy viết các m-file thể hiện các phương trình vi phân và thực hiện việc giải số tìm dao động của hệ, với

$$\mathbf{M} = \begin{bmatrix} 12 & 0 & 0 \\ 0 & 15 & 0 \\ 0 & 0 & 20 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 10 & -10 & 0 \\ -10 & 20 & -10 \\ 0 & -10 & 10 \end{bmatrix},$$

$$\mathbf{K} = \begin{bmatrix} 2000 & -1000 & 0 \\ -1000 & 2000 & -1000 \\ 0 & -1000 & 1000 \end{bmatrix}, \quad \mathbf{f}(t) = \begin{bmatrix} 5 \sin 10t \\ 0 \\ 0 \end{bmatrix}$$

Bài tập

```
function ydot = daodong3dof(t,y);
    ydot = zeros(6,1);
    q = y(1:3);    v = y(4:6);

    M = [12, 0, 0; 0, 15, 0; 0, 0, 20];
    C = [10 -10 0; -10 20 -10; 0 -10 10];
    K = [2, -1 0; -1 2 -1; 0 -1 1]*1000;
    ft = [5*sin(10*t); 0; 0];

    qdot = v;
    vdot = inv(M)*(ft-C*v-K*q);
    ydot = [qdot; vdot];

end % save with file name daodong3dof.m

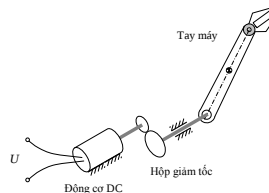
>> [t,y] = ode45('daodong3dof',[0 20],[0,0,0,0,0,0]);
>> plot(t,y(:,1)),xlabel('t')
```

Bài tập

Phương trình vi phân chuyển động của tay máy 1 dof:

$$(J_o + J_m r^2)\ddot{\phi} + (K_m K_e r^2 / R_a + b r^2)\dot{\phi} + mgl \cos(\phi) = (K_m r / R_a)U$$

Hãy viết các m-file thể hiện các phương trình vi phân và thực hiện việc giải số tìm chuyển động của tay máy: taymay1dof.m



Bài tập

Xét hệ dao động n dof, cho $M, C, K, f(t)$: $M\ddot{q} + C\dot{q} + Kq = f(t)$,

- Thiết lập phương trình mô tả hệ
- Xét dao động tự do không cản
 - Tần số dao động riêng, các dạng dao động riêng
 - Vẽ các dạng dao động riêng của hệ
 - Kiểm tra tính trực giao của ma trận dạng riêng thông qua M và K
- Xét dao động cưỡng bức có cản
 - Đồ thị biên độ tần số
 - Tính dao động cưỡng bức [thành phần dao động cùng tần số cưỡng bức]
 - Tính dao động của hệ theo thời gian $q(t)$
 - Vẽ quỹ đạo pha của từng tọa độ suy rộng
- Mô phỏng tìm $q(t)$ với các kích động tùy ý [xung, bước nhảy, $f(t)$ – tùy ý]

Bài tập

Động lực học robot dạng chuỗi hở

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + D\dot{q} + F_s \operatorname{sgn}(\dot{q}) + g(q) = u(t),$$

- Thiết lập phương trình mô tả hệ
- Cho $q(t)$ tìm lực/mô men $u(t)$
 - Vẽ đồ thị $q(t)$, $\dot{q}(t)$, $\ddot{q}(t)$ và $u(t)$
- Cho lực/mô men $u(t)$ tìm chuyển động $q(t)$
 - Vẽ đồ thị $q(t)$, $\dot{q}(t)$, và $u(t)$

$$u(t) = g(q) - K_p(q - q_0) - K_d\dot{q}$$

Bài tập

Động lực học robot song song

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{D}\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{u}(t) + \Phi^T(\mathbf{q})\boldsymbol{\lambda},$$

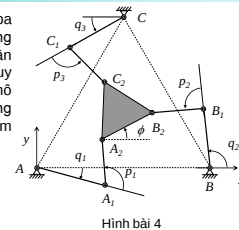
$$\Phi(\mathbf{q}) = \mathbf{0}$$

$$\mathbf{q}^T = [\mathbf{q}_a^T, \mathbf{q}_p^T, \mathbf{x}^T]$$

1. Thiết lập phương trình mô tả hệ
2. Động học thuận/ngược robot song song
3. Cho chuyển động của bàn máy [khâu cuối], $\mathbf{x}(t)$, tìm lực/mô men $\mathbf{u}(t)$
4. Cho lực/mô men $\mathbf{u}(t)$ tìm chuyển động bàn máy $\mathbf{x}(t)$

Bài 4. Cho một rôbốt song song phẳng ba bậc tự do, chuyển động trong mặt phẳng ngang. Hãy thiết lập phương trình vi phân chuyển động cho rôbốt với các tọa độ suy rộng dư. Viết chương trình Matlab để mô phỏng động lực học cho rôbốt. Biết rằng các tam giác ABC và $(ABC)_2$ là các tam giác đều.

$$\mathbf{q} = [q_1, q_2, q_3, p_1, p_2, p_3, x_{A2}, y_{A2}, \phi]^T$$



Hình bài 4

Lecture 8

Giới thiệu về Simulink

Nguyen Q. Hoang
Department of Applied Mechanics
Hanoi University of Science and Technology

HUST
Hanoi University of Science and Technology

1

Khái niệm về simulink

- Công cụ simulink trong Matlab cho phép mô phỏng các hệ động lực với một giao diện đồ họa đặc biệt bằng các khối kết nối với nhau. Mục đích của chương này là giới thiệu về simulink thông qua các ví dụ. Sau khi thiết lập các phương trình vi phân mô tả hệ, một sơ đồ khối tương ứng được đưa ra và từ đó sử dụng các khối của simulink để mô phỏng hệ.
- Simulink là một phần mở rộng của MATLAB được phát triển bởi Mathworks Inc. Simulink là một gói chương trình để mô hình hóa, mô phỏng và phân tích các hệ động lực. Với simulink ta có thể mô hình hóa các hệ thống tuyến tính và phi tuyến liên tục theo thời gian, rời rạc theo thời gian, hoặc hỗn hợp cả liên tục và rời rạc theo thời gian. Các hệ cũng có thể là đa tốc độ khác nhau, nghĩa là các phần khác nhau có thể được lấy mẫu hoặc cập nhật số liệu ở các tốc độ khác nhau.

HUST
Hanoi University of Science and Technology

3

Chương 8. Giới thiệu về Simulink

8.1 Khái niệm về simulink

8.2 Nguyên lý hoạt động và việc thực hành trong simulink

Khởi động simulink

Các khối trong simulink

8.3 Một số ví dụ đơn giản

a) Mô hình hóa một phương trình bằng sơ đồ khối

b) Mô phỏng một quá trình động học

c) Mô hình hóa một hệ động lực liên tục đơn giản

d) Mô tả hệ dao động một bậc tự do

e) Mô phỏng số tay máy một bậc tự do

8.4 Đơn giản sơ đồ simulink

Đơn giản sơ đồ simulink bằng khối Fcn

Đơn giản sơ đồ simulink bằng khối subsystem

Kết hợp simulink và script file (m-file)

8.5 Xử lý kết quả mô phỏng

8.6 Bài tập thực hành

HUST
Hanoi University of Science and Technology

2

Biểu diễn sơ đồ khối của biểu thức

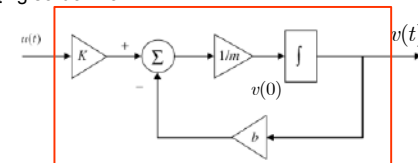
Xét phương trình vi phân sau

$$m\dot{v} + bv = Ku(t) \Rightarrow \dot{v} = \frac{1}{m}[Ku(t) - bv]$$

Biểu thức nghiệm tìm được dạng

$$v(t) = v(0) + \int_0^t \dot{v}(\tau) d\tau = v(0) + \int_0^t \frac{1}{m}[Ku(\tau) - bv] d\tau$$

Biểu diễn dạng sơ đồ khối



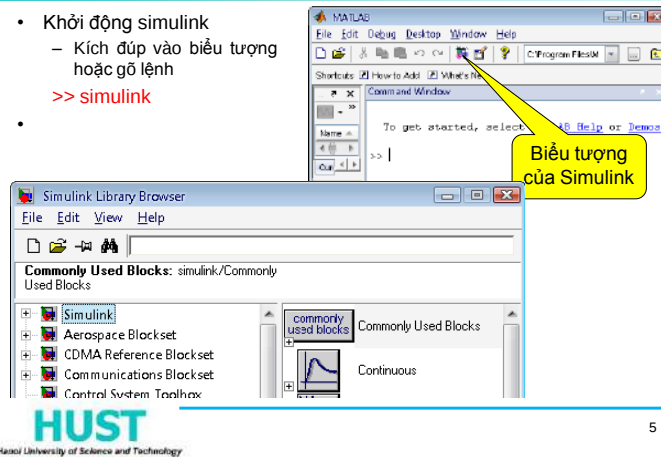
HUST
Hanoi University of Science and Technology

4

Khái niệm về simulink

- Khởi động simulink
 - Kích đúp vào biểu tượng hoặc gõ lệnh

>> simulink



5

Các khối trong simulink

Các phần tử trong source (nguồn)					
	Constant	Cấp một hằng số, hay một vectơ hằng		Step	Cấp một tín hiệu dạng bước nhảy
	Ramp	Cấp một tín hiệu theo đường xiên (đường dốc)		Sine Wave	Cấp một tín hiệu hình sin
	Pulse Generator	Cấp một tín hiệu dạng xung		Repeating Sequence Stair	Cấp một tín hiệu bậc thang

HUST
Hanoi University of Science and Technology

6

Các khối trong simulink

	Clock	Đồng hồ cấp thời gian		Repeating Sequence	Cấp một tín hiệu răng cưa
	Random Number	Cấp một số ngẫu nhiên		Band-Limited White Noise	Cấp một tín hiệu nhiễu

HUST
Hanoi University of Science and Technology

7

Các khối kết trong simulink

Các khối kết trong sink					
	Scope	Vẽ ra đồ thị tín hiệu theo thời gian		XY Graph	Cho phép biểu diễn hai tín hiệu dạng đồ thị, $y(x)$
	Display	Hiện thị giá trị của tín hiệu		untitled.mat To File	Để ghi lại tín hiệu vào một file.mat
	Out1	Một đầu tín hiệu ra, sử dụng cho hệ con		simout To Workspace	Để chuyển tín hiệu vào workspace
	Terminator	Sử dụng để kết thúc các tín hiệu ra, (tránh để chương trình không cảnh báo công tín hiệu không được)		Stop Simulation	Sử dụng để dừng chương trình mô phỏng (khi tín hiệu bằng 0, chương trình sẽ dừng lại).

HUST
Hanoi University of Science and Technology

8

Các khối trong simulink

Các khối hàm toán học, Math Operations

	Tính tổng hai tín hiệu		Tính tổng hai tín hiệu
	Nhân hai tín hiệu		Tính hiệu hai tín hiệu
	Cho giá trị tuyệt đối của tín hiệu		Nhân/chia hai tín hiệu
	Tính tích vô hướng hai vectơ		Nhân tín hiệu với một số (khuếch đại tín hiệu)

HUST

Hanoi University of Science and Technology

9

Các khối trong simulink

	Tính tổng các phần tử		Hàm dấu (cho dấu của biến x)
	Tính tích các phần tử		Hàm sin
	Một hàm toán học exp, sin, cos,...		Hàm lượng giác

HUST

Hanoi University of Science and Technology

10

Các khối trong simulink

Các khối hàm toán học liên tục và không liên tục

	đạo hàm một tín hiệu theo thời gian		khối hàm truyền bậc nhất hệ rời rạc
	toán tử tích phân		khối giữ tín hiệu trích mẫu bậc 0
	khối hàm truyền bậc nhất hệ liên tục		khối giữ tín hiệu trích mẫu bậc 1
	khối mô tả phương trình trạng thái hệ tuyến tính		Giữ tín hiệu bão hòa

HUST

Hanoi University of Science and Technology

11

Các khối trong simulink

	tạo một hàm định nghĩa bởi người sử dụng		Ghi nhớ tín hiệu của bước trước
	tạo một hàm những định nghĩa bởi người sử dụng		Khối mô tả ma sát Coulomb và ma sát nhớt
	f(u) là một hàm nào đó của Matlab		Khối làm trễ tín hiệu

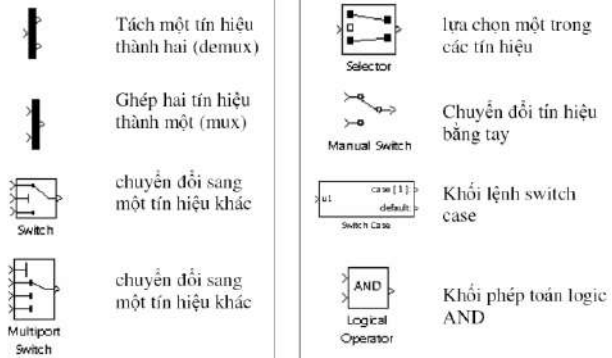
HUST

Hanoi University of Science and Technology

12

Các khối trong simulink

Một số khối liên quan tách nhập, lựa chọn tín hiệu, subsystem



HUST

Hanoi University of Science and Technology

13

Một số ví dụ đơn giản

Mô hình hóa một phương trình bằng sơ đồ khối

Ví dụ ta cần chuyển đổi từ thang nhiệt độ **C** sang thang nhiệt độ **Fahrenheit**, phương trình mô tả quan hệ nhiệt độ giữa hai thang đo này như sau

$$T_F = \frac{9}{5} T_C + 32$$

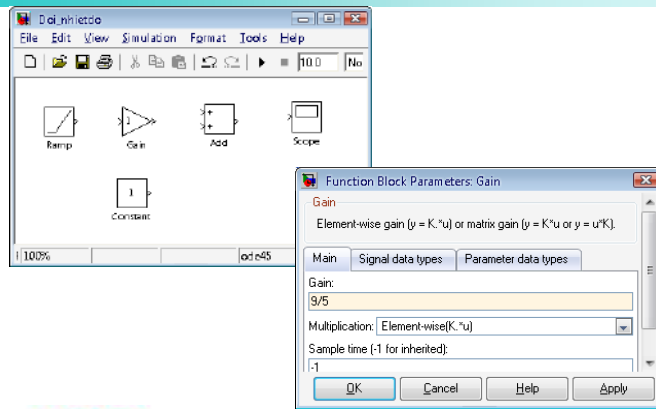
Các khối cần được kéo vào model window	Vị trí các khối trong thư viện Simulink
Ramp	Sources
Constant	Sources
Gain	Math Operation
Add hoặc Sum	Math Operation
Scope	Sinks

HUST

Hanoi University of Science and Technology

14

Một số ví dụ đơn giản

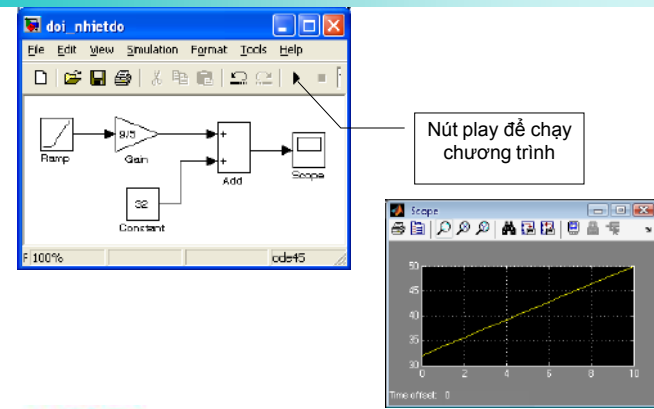


HUST

Hanoi University of Science and Technology

15

Một số ví dụ đơn giản



HUST

Hanoi University of Science and Technology

16

Mô phỏng một quá trình động học

Trong ví dụ này ta sẽ sử dụng Simulink để mô tả một phương trình động học sau

$$x(t) = A \sin(\omega t + \varphi)$$

Trong ví dụ này các thông số được nhận các giá trị

$$A = 2, \omega = 4, \varphi = \pi/3$$

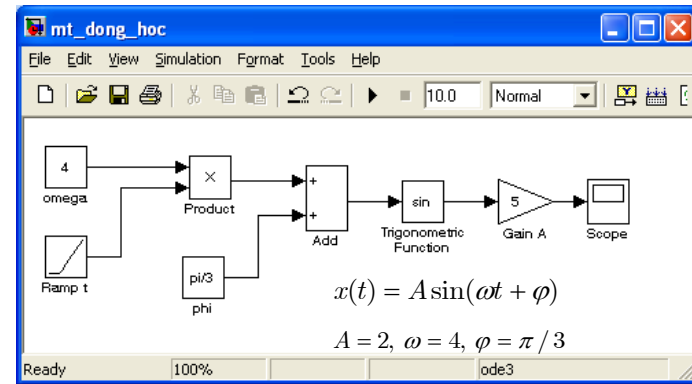
Các khối cần được kéo vào model window	Vị trí các khối trong thư viện Simulink
Constant	Sources
Ramp	Sources
Gain	Math Operation
Sum	Math Operation
Product	Math Operation
Trigonometry Function	Math Operation
Mux	Signal Routing
Scope	Sinks

HUST

Hanoi University of Science and Technology

17

Mô phỏng một quá trình động học

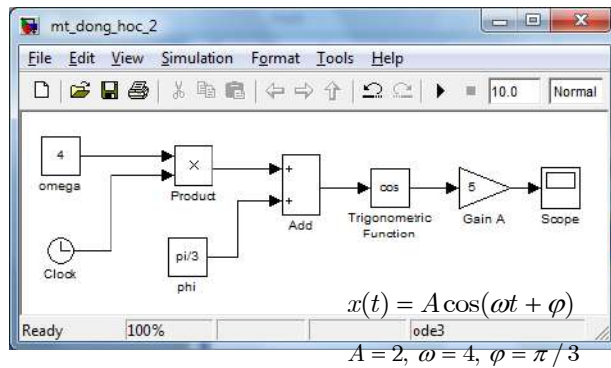


HUST

Hanoi University of Science and Technology

18

Mô phỏng một quá trình động học



HUST

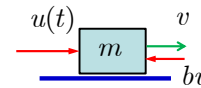
Hanoi University of Science and Technology

19

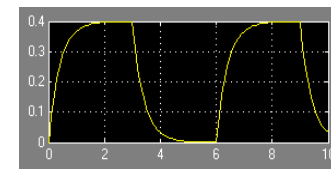
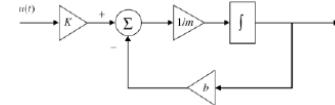
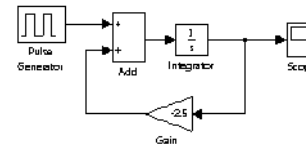
Giải phương trình vi phân

Xét phương trình vi phân mô tả một hệ động lực liên tục như sau

$$\Rightarrow \dot{v} = \frac{1}{m} [Ku(t) - bv]$$



$$\dot{x}(t) = -2.5x(t) + u(t)$$

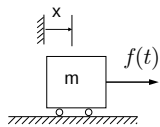


HUST

Hanoi University of Science and Technology

20

Mô tả hệ dao động một bậc tự do

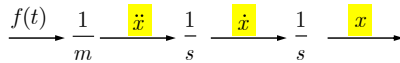


Phương trình vi phân chuyển động

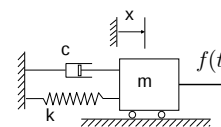
$$m\ddot{x}(t) = f(t)$$

Viết lại phương trình vi phân chuyển động trên dưới dạng sau

$$\ddot{x} = \frac{1}{m}[f(t)]$$



Mô tả hệ dao động một bậc tự do



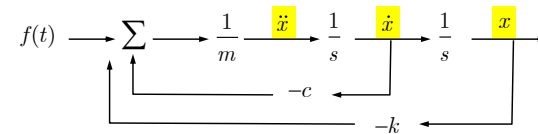
Phương trình vi phân chuyển động

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = f(t)$$

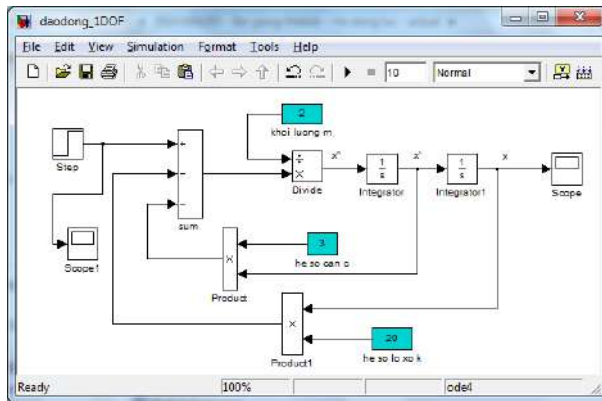
Viết lại phương trình vi phân chuyển động trên dưới dạng sau

$$\ddot{x} = \frac{1}{m}[f(t) - c\dot{x} - kx]$$

Hình 8-12.

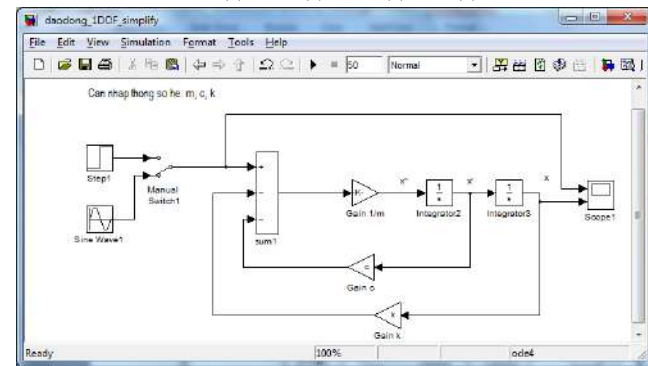


Mô tả hệ dao động một bậc tự do



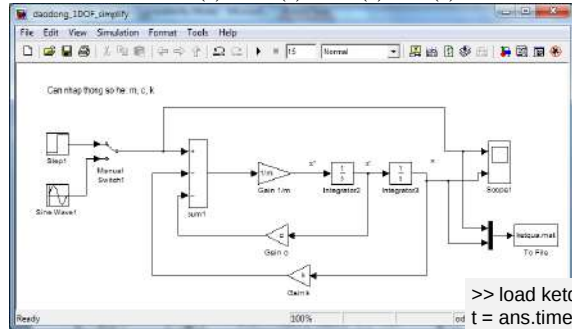
Mô tả hệ dao động một bậc tự do

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = f(t)$$



Mô tả hệ dao động một bậc tự do

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = f(t)$$



```
>> load ketqua.mat
t = ans.time(:,1)
u = ans.data(:,1)
x = ans.data(:,2)
plot(t,u),
```

HUST

Hanoi University of Science and Technology

Mô hình hóa và mô phỏng hệ cơ điện

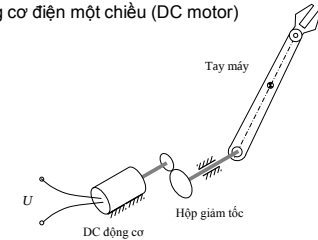
Xét tay máy 1 dof được dẫn động bởi động cơ điện một chiều (DC motor)

Các thông số của **động cơ điện một chiều**:

- điện trở, R_a
- cuộn cảm có hệ số tự cảm, L_a
- hằng số phản sức điện động, K_e
- hằng số mômen, K_m
- mô men quán tính của rô-to, J_m
- Hệ số cản ô trục, b_1

Các thông số **tay máy**:

- khối lượng m , khối tâm C , $OC = L$.
- mômen quán tính khối đối với trục quay O , J_o ;
- hệ số cản tại ô trục b_2



Hộp giảm tốc:

- tỷ số truyền của

$$r = \omega_1 / \omega_2 = r_2 / r_1 > 1$$

Tương tác cơ – điện:

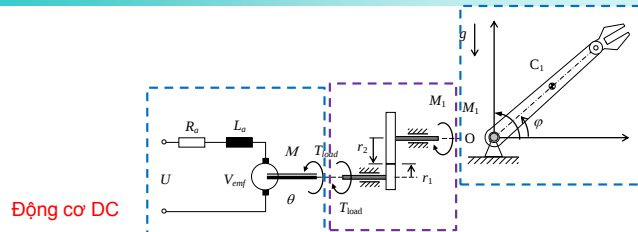
$$V_{emf} = K_e \dot{\theta}, \quad M = K_m i$$

HUST

Hanoi University of Science and Technology

26

Mô hình hóa và mô phỏng hệ cơ điện



Động cơ DC

Phương trình điện (từ phương trình cân bằng điện áp của một mạch vòng)

$$U = U_R + U_L + V_{emf} \Rightarrow L_a \frac{di}{dt} + R_a i = U - K_e \omega \quad i = i_e$$

Phương trình cơ (phương trình chuyển động quay của rôto)

$$\frac{d}{dt} \bar{L}_z = \Sigma m_z(\bar{F}_k) \Rightarrow J_m \dot{\omega} = K_m i - b_1 \omega - T_{load}, \quad \omega = \dot{\theta}$$

HUST

Hanoi University of Science and Technology

27

Mô hình hóa và mô phỏng hệ cơ điện

Đối với hộp giảm tốc (không khối lượng, không ma sát) ta có công suất vào bằng công suất ra

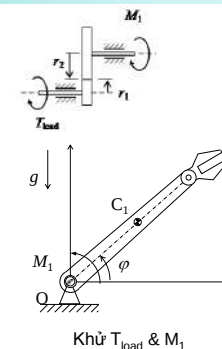
$$T_{load} \dot{\theta} = M_1 \dot{\phi}$$

$$r_1 \dot{\theta} = r_2 \dot{\phi} \Rightarrow \dot{\theta} = \dot{\phi} r_2 / r_1 = r \dot{\phi}$$

$$T_{load} = M_1 \dot{\phi} / \dot{\theta} = M_1 / r$$

Phương trình chuyển động của tay máy do tác dụng của mômen

$$J_O \ddot{\phi} = M_1 - b_2 \dot{\phi} - mgl \cos(\phi)$$



Khử T_{load} & M_1

HUST

Hanoi University of Science and Technology

28

Mô hình hóa và mô phỏng hệ cơ điện

Các PTVP mô tả

$$J_m \dot{\omega} = K_m i - b_1 \omega - T_{load}, \quad \omega = \dot{\theta}$$

$$L_a \frac{di}{dt} + R_a i = U - K_e \omega, \quad i = \dot{q}_e$$

$$J_O \ddot{\phi} = M_1 - b_2 \dot{\phi} - mgl \cos(\phi)$$

Các ràng buộc

$$\dot{\theta} = \omega = r \dot{\phi}$$

Khử T_{load} & M_1

$$T_{load} \dot{\theta} = M_1 \dot{\phi}$$

$$T_{load} = M_1 \dot{\phi} / \dot{\theta} = M_1 / r$$

HUST

Hanoi University of Science and Technology

29

Mô hình hóa và mô phỏng hệ cơ điện

Hệ 3 phương trình vi phân cấp 1 mô tả qua hệ đầu vào U và đầu ra ϕ

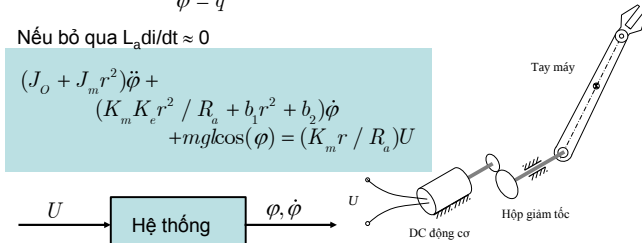
$$L_a \frac{di}{dt} = U - R_a i - K_e r q$$

$$(J_O + J_m r^2) \dot{q} = K_m r i - (b_1 r^2 + b_2) q - mgl \cos(\phi)$$

$$\dot{\phi} = q$$

Nếu bỏ qua $L_a di/dt \approx 0$

$$(J_O + J_m r^2) \ddot{\phi} + (K_m K_e r^2 / R_a + b_1 r^2 + b_2) \dot{\phi} + mgl \cos(\phi) = (K_m r / R_a) U$$



HUST

Hanoi University of Science and Technology

30

Mô hình hóa và mô phỏng hệ cơ điện

Các khối cần được kéo vào model window

Vị trí các khối trong thư viện Simulink

Constant
Sum
Product
Divide
Gain
Scope

Trigonometric functions

Integrator
Scope

Sources
Math Operation
Math Operation
Math Operation
Math Operation

Math Operation

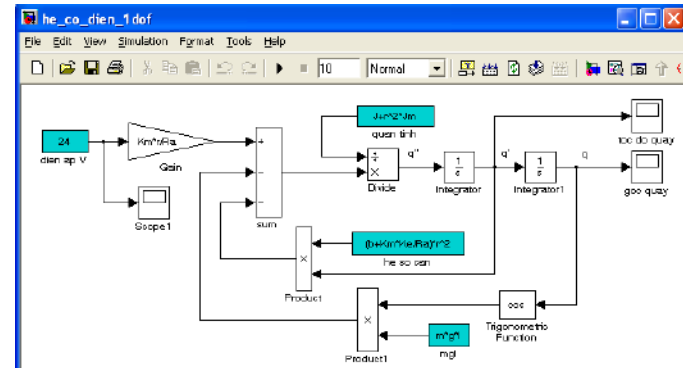
Continuous
Sinks

HUST

Hanoi University of Science and Technology

31

Mô hình hóa và mô phỏng hệ cơ điện



HUST

Hanoi University of Science and Technology

32

```
% Can nhap thong so he:
% - DC motor: Ra, La, Ke, Km, Jm, b1
% - Tay may: m, lc, g, JO, b2
% - hop giam toc: ti so truyen r

% - DC motor:
Ra=1,      La=0.0001,
Ke=1,      Km=1,
Jm=0.001,  b1=0.01;
% - Tay may:
m=5,      lc=0.5,  g=9.81,
JO=0.2,   b2=0.2;
% -hop giam toc: ti so truyen
r = 2.0
```

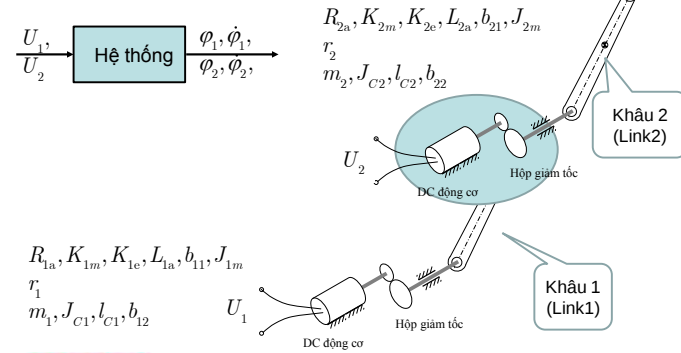
HUST

Hanoi University of Science and Technology

33

Mô hình hóa và mô phỏng hệ cơ điện

Bài tập: Ghép 2 tay máy 1dof thành 1 tay máy phẳng 2dof.


HUST

Hanoi University of Science and Technology

34

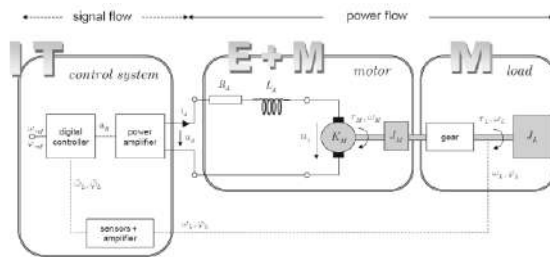


Fig. 2.54. Example of a multi-domain model: servo-controlled drive shaft (e.g. robot, machine tool)

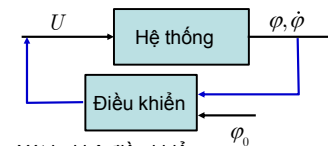
HUST

Hanoi University of Science and Technology

35

Mô hình hóa và mô phỏng hệ cơ điện

$$(J_o + J_m r^2) \ddot{\varphi} + (K_m K_c r^2 / R_a + b_1 r^2 + b_2) \dot{\varphi} + mgl \cos(\varphi) = (K_m r / R_a) U$$



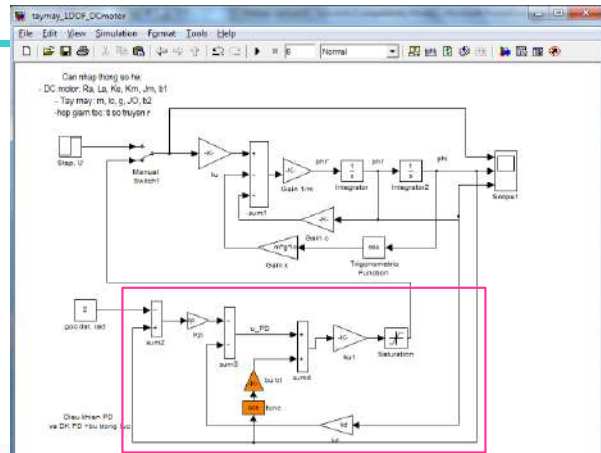
Xét hai bộ điều khiển

1. $U = (K_m r / R_a)^{-1} [-k_p (\varphi - \varphi_0) - k_d \dot{\varphi}]$ PD-controller
2. $U = (K_m r / R_a)^{-1} [-k_p (\varphi - \varphi_0) - k_d \dot{\varphi} + mgl \cos \varphi]$ PD-controller + bù trọng lực

HUST

Hanoi University of Science and Technology

36



HUST

Hanoi University of Science and Technology

37

Bài tập

Điều kiện đầu là $y(0) = 0.01$ m, $\dot{y}(0) = 0$. Thực hiện mô phỏng bằng simulink trong khoảng thời gian $t = 0 \dots 2$ s và đưa ra các kết quả dạng đồ thị $y(t)$ và $\dot{y}(t)$.

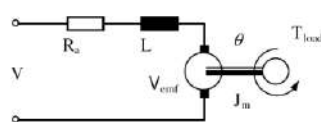
3. Cho sơ đồ động cơ điện một chiều như trên hình vẽ và phương trình vi phân mô tả như sau:

$$J_m \frac{d^2\theta}{dt^2} + b \frac{d\theta}{dt} = T(t) = K_T i(t),$$

$$L \frac{di}{dt} + Ri(t) + B_s \frac{d\theta}{dt} = V(t)$$

Hãy hạ bậc đưa hệ trên về hệ phương trình vi phân bậc nhất với các biến trạng thái

$$\mathbf{x} = [\theta, \dot{\theta}, i]^T$$



Sơ đồ động cơ điện một chiều

HUST

Hanoi University of Science and Technology

39

Bài tập

Xây dựng mô hình simulink để mô phỏng các hệ cho bởi các phương trình vi phân sau :

1. Phương trình vi phân chuyển động của con lắc vật lý

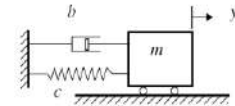
$$J_o \ddot{\varphi} + b\dot{\varphi} + mg l \sin \varphi = 0, \text{ với } J_o = 1, b = 0.1, mgl = 10,$$

Lấy các điều kiện đầu $\varphi_o = 0.30$, $\dot{\varphi}_o = 0$.

2. Hệ dao động gồm khối lượng - lò xo - cản nhớt có phương trình vi phân mô tả như sau:

$$\ddot{y} + \frac{b}{m} \dot{y} + \frac{c}{m} y = 0$$

với $m = 2$ kg, $b = 40$ N.s/m, $c = 450$ N/m



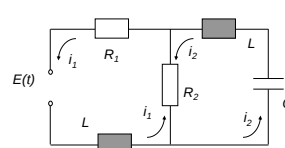
HUST

Hanoi University of Science and Technology

38

Bài tập

6. Cho sơ đồ mạch điện như trên hình vẽ. Áp dụng định luật Kirchhoff ta nhận được phương trình vi phân của mạch điện như sau:



$$L \frac{di_1}{dt} + R_1 i_1 + R_2 (i_1 - i_2) = E(t)$$

$$L \frac{di_2}{dt} - R_2 (i_1 - i_2) + \frac{q_2}{C} = 0$$

$$\frac{dq_2}{dt} = i_2$$

Cho biết các số liệu $R_1 = 4 \Omega$, $R_2 = 10 \Omega$, $L = 0.032$ H, $C = 0.53$ F

và điện áp đặt lên mạch có dạng

$$E(t) = \begin{cases} 20 \text{ V} & \text{khí } 0 < t < 0.005 \text{ s} \\ 0, & \text{khí } t \geq 0.005 \text{ s} \end{cases}$$

Hãy giải và vẽ ra đồ thị các dòng điện

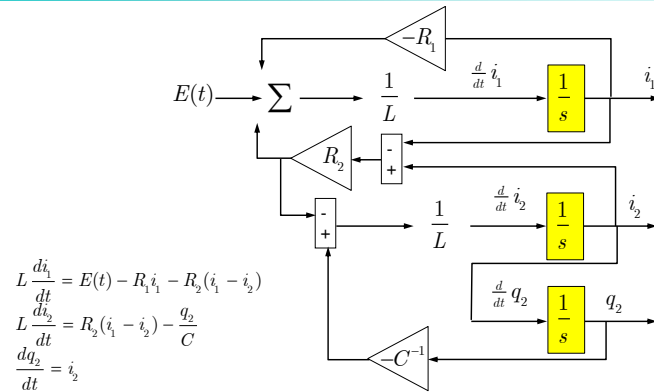
$$i_1(t), i_2(t) \quad t = 0 \dots 0.05 \text{ s}$$

HUST

Hanoi University of Science and Technology

40

Bài tập

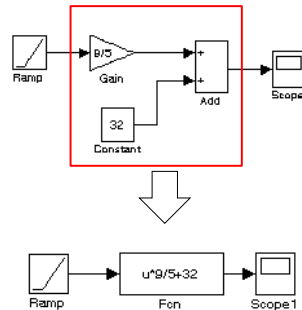


Đơn giản hoá sơ đồ simulink

1. Đơn giản sơ đồ simulink bằng khối Fcn
2. Đơn giản sơ đồ simulink bằng khối subsystem
3. Kết hợp simulink và script file (m-file)

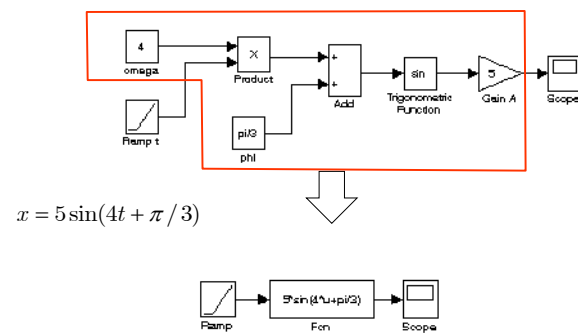
Đơn giản hoá sơ đồ simulink

1. Đơn giản sơ đồ simulink bằng khối Fcn



Đơn giản hoá sơ đồ simulink

1. Đơn giản sơ đồ simulink bằng khối Fcn



Đơn giản hoá sơ đồ simulink

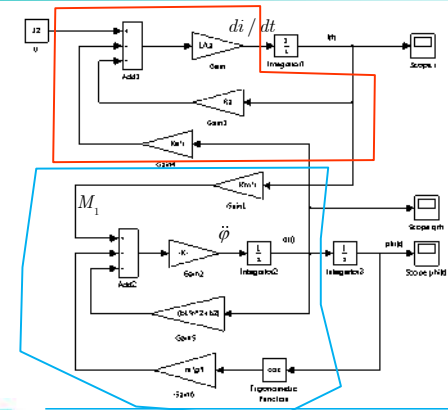
2. Đơn giản sơ đồ simulink bằng khối subsystem

$$\frac{di}{dt} = \frac{1}{L_a} (U - R_a i - K_e r q)$$

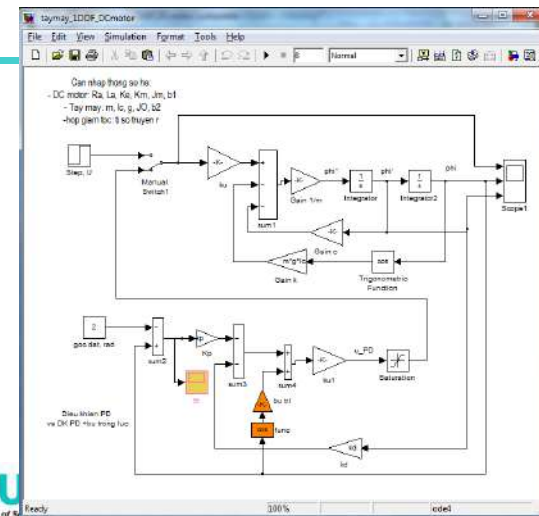
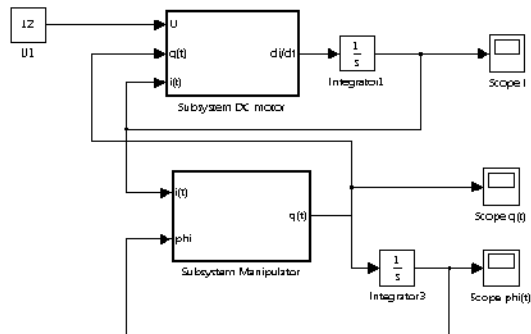
$$\frac{dq}{dt} = \frac{1}{(J_o + J_m r^2)} [K_m r i - (b_1 r^2 + b_2) q - mgl \cos(\varphi)]$$

$$\frac{d\varphi}{dt} = q$$

Đơn giản hoá sơ đồ simulink



Đơn giản hoá sơ đồ simulink



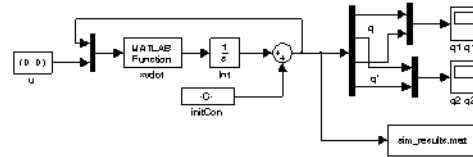
51

50

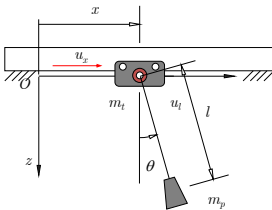
52

Đơn giản hoá sơ đồ simulink

Ví dụ: kết hợp simulink và m-file



Con lắc elliptic



Đơn giản hoá sơ đồ simulink

```
function xdot = xdot_elliptic_pendul(ins)
```

```
% cac tham so he
m1 = 5; % kg
m2 = 2; % kg
l = 0.5; % m
gr = 9.81; % m/s^2
d1 = 6.2; d2 = 0; % damping coeffs
```

```
% ins = [1 2 3 4 5]'; % for test
% viet ra cac phuong trinh vi phan
q = ins(1:2);
% bien chua toa do suy rong
qdot = ins(3:4);
% bien chua van toc suy rong
u = ins(5);
% bien vao (bien dieu khien)
```

$$\mathbf{M} = \begin{bmatrix} m_1 + m_2 & m_2 l \cos(q(2)) \\ m_2 l \cos(q(2)) & m_2 l^2 \end{bmatrix};$$

```
C = [0, -m2*I*qdot(2)*sin(q(2));  
     0, 0];
```

```
D = diag([d1, d2]); % damping matrix
```

$$\mathbf{g} = [0; m_2 l \cdot g \cdot \sin(q_2)];$$

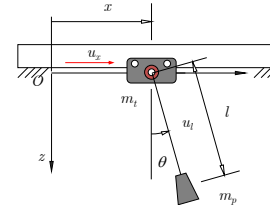
```
B = [1; 0];    % he hut dan dong
% B = [1, 0; 0, 1]; % he đủ dan dong
```

$$\dot{v} = \text{inv}(M) * (B * u - (C + D) * \dot{q} - g);$$

$$\dot{x} = [\dot{q}; \dot{v}];$$

Đơn giản hoá sơ đồ simulink

Ví dụ: kết hợp simulink và m-file



$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{B}\mathbf{u}$$

$$\dot{\mathbf{q}} = \mathbf{v}$$

$$\dot{\mathbf{v}} = \mathbf{M}(\mathbf{q})^{-1}[\mathbf{u} - \mathbf{C}(\mathbf{q}, \mathbf{v})\mathbf{v} - \mathbf{g}(\mathbf{q})]$$

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} m_t + m_p & m_p l \cos \theta \\ m_p l \cos \theta & m_p l^2 \end{bmatrix}$$

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} 0 & -m_p l \dot{\theta} \sin \theta \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} 0 \\ m_p g l \sin \theta \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

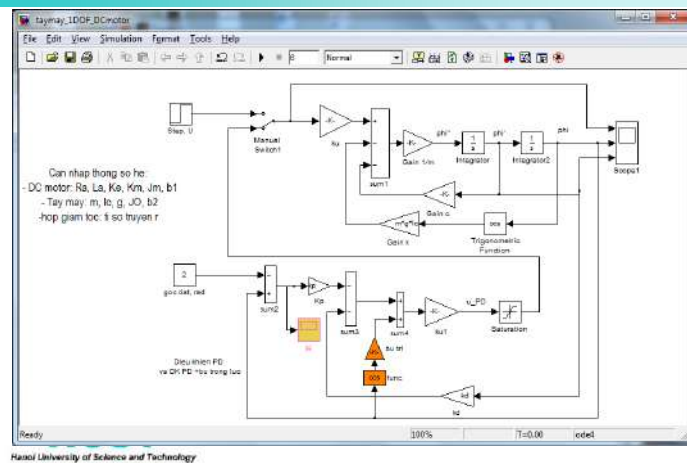
Đơn giản hoá sơ đồ simulink

Kết hợp simulink-model với lệnh **sim**

- Thực hiện mô phỏng mà không cần mở simulink
- Sử dụng được mô hình simulink trong một m-file
- Thuận tiện cho các bài toán lặp như tìm tham số tối ưu cho bộ điều khiển bằng GA, PSO, Nelder-Mead,....
- Cú pháp: xem >> help **sim**

5. Mask and unmask ??

Đơn giản hoá sơ đồ simulink



Đơn giản hoá sơ đồ simulink

```
% Can nhap thong so he:
% - DC motor: Ra, La, Ke, Km, Jm, b1
% - Tay may: m, lc, g, JO, b2
% - hop giam toc: ty so truyen r
```

```
Ra=1, La=0.0001, % - DC motor
Ke=11, Km=1, Jm=0.001, b1=0.00000001;
m=5, lc=0.5, g=9.81, JO=0.2, b2=2e-7; % - Tay may
r = 2.0; % -hop giam toc: ti so truyen
kp = 100; kd = 50; % -tham so bo dieu khien
```

```
[t, y] = sim('taymay_1DOF_DCmotor',[0, 8])
```

```
plot(t,y(:,1), 'r-', 'linewidth',2), grid on,
xlabel('t [s]'), ylabel('\phi (t) [rad]')
```

```
% su dung neu dua mdl vao trong mot function
options = simset('SrcWorkspace','current');
[t, y] = sim('modelname',[],options)
```

HUST

Hanoi University of Science and Technology

58

Phân tích hệ động lực

Điều khiển tối ưu LQR Linear Quadratic Regulators

Nguyen Q. Hoang
Department of Applied Mechanics
Hanoi University of Science and Technology

Chương x. Xây dựng mô hình

1. Nêu bài toán điều khiển tối ưu
2. Cơ sở lý thuyết
3. Áp dụng điều khiển tối ưu hệ tuyến tính
4. Điều khiển tối ưu LQR
5. Ví dụ

- Ref. Alok Sinha : Linear Systems- Optimal and Robust Control, CRC press, 2007.

Nêu bài toán điều khiển tối ưu

Để điều khiển hệ từ trạng thái $x(0)$ về gốc tọa độ bằng điều khiển phản hồi $u = u(x)$, có thể có rất nhiều thuật toán điều khiển khác nhau. Ngay cả đối với một thuật toán điều khiển, chẳng hạn $u = -Kx$, thì có rất nhiều khả năng chọn ma trận phản hồi K .

Bài toán điều khiển tối ưu sẽ đưa ra phương thức lựa chọn bộ thông số của bộ điều khiển để một chỉ tiêu chất lượng nào đó được tối thiểu hóa.

Trong phần này sẽ nêu bài toán điều khiển tối ưu, và phương pháp thiết kế tối ưu LQR.

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

3

Nêu bài toán điều khiển tối ưu

Xét hệ động lực

$$\dot{x} = f(x, u), \quad x(0) = x_0$$

$$y = g(x)$$

Điều khiển phản hồi trạng thái

$$u = u(x)$$

Các tiêu chuẩn tối ưu cần được cực tiểu có thể là

$$J_1 = \int_0^{t_f} e^T Q e dt = \int_0^{t_f} (x - x_d)^T Q (x - x_d) dt \quad (1)$$

$$J_2 = \int_0^{t_f} (y - y_d)^T Q (y - y_d) dt \quad (2)$$

$$J_3 = \int_0^{t_f} u^T R u dt \quad (3)$$

$$J_4 = \int_0^{t_f} 1 dt \quad (4)$$

$$Q = Q^T \geq 0, \quad R = R^T > 0$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

4

Nêu bài toán điều khiển tối ưu

Để giải quyết tình trạng khó xử, chúng ta có thể thỏa hiệp giữa hai mục tiêu mâu thuẫn nhau bằng cách giảm thiểu các chỉ số thực hiện mà là một sự kết hợp lồi của J_1 và J_3 ,

$$J = (1 - \lambda)J_1 + \lambda J_3 = \int_0^{t_f} [(x - x_d)^T Q(x - x_d) + u^T R u] dt \quad (5)$$

Trong các ứng dụng nhất định, chúng ta có thể muốn trạng thái cuối cùng $x(t_f)$ là càng gần 0 càng tốt. Sau đó, một chỉ số thực hiện có thể được giảm thiểu là

$$x^T(t_f) F x(t_f), \quad F = F^T > 0 \quad (6)$$

Chúng ta có thể kết hợp các biện pháp thực hiện (1), (3), và (6) khi mục tiêu điều khiển là để giữ cho trạng thái "nhỏ", sự điều khiển "không quá lớn," và trạng thái cuối cùng càng gần 0 càng tốt. Chỉ số thực hiện kết quả là

$$J = x^T(t_f) F x(t_f) + \int_0^{t_f} [(x - x_d)^T Q(x - x_d) + u^T R u] dt \quad (7)$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

5

Nêu bài toán điều khiển tối ưu

Xét hệ động lực tuyến tính, với cặp (A, B) là điều khiển được

$$\dot{x} = Ax + Bu, \quad x(0) = x_0$$

$$y = Cx$$

Tiêu chuẩn 2 trở thành

$$J_2 = \int_0^{t_f} y^T y dt = \int_0^{t_f} x^T C^T C x dt = \int_0^{t_f} x^T Q x dt, \quad Q = C^T C$$

Bài toán tối ưu được đặt ra là: Tìm luật điều khiển $u(t)$ để làm tối thiểu một trong các chỉ số thực hiện nêu trên, hoặc tối thiểu một chỉ tiêu thỏa hiệp trong các trường hợp:

1. Cố định thời gian cuối t_f ;
2. Không cố định thời gian cuối t_f ;
3. Không có ràng buộc đặt lên $u(t)$;
4. Có ràng buộc đặt lên điều khiển, $u_i(t)$ có cận trên và cận dưới.

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

6

Cơ sở lý thuyết điều khiển tối ưu

Xét hệ động lực điều khiển được

$$\dot{x} = f(x, u), \quad x(0) = x_0$$

Và tiêu chuẩn tối ưu sau cần được cực tiểu (hoặc cực đại)

$$J = G[x(t_f)] + \int_0^{t_f} F(x, u) dt$$

Các ràng buộc đặt lên điều khiển $u_i(t)$, cận trên và cận dưới

$$u_{i,m} \leq u_i \leq u_{i,M}, \quad i = 1, 2, \dots, m$$

$$\dim(x) = \dim(f) = n, \quad \dim(u) = m$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

7

Cơ sở lý thuyết điều khiển tối ưu

Các điều kiện cần cho tối ưu:

Nếu điều khiển $u_0(t)$ là nghiệm của bài toán tối ưu và $x_0(t)$ là véc tơ trạng thái tương ứng:

$$\dot{x}_0(t) = f(x_0(t), u_0(t)),$$

Khi đó ta sẽ đánh giá hàm chất lượng điều khiển (chỉ tiêu tối ưu), khi điều khiển thay đổi một lượng nhỏ $\delta u(t)$, tức là

$$u(t) = u_0(t) + \delta u(t),$$

Tiêu chuẩn tối ưu thay đổi một lượng là

$$\begin{aligned} \delta J &= J[u_0(t) + \delta u(t)] - J[u_0(t)] \\ &= \frac{\partial G}{\partial x} \delta x(t_f) + \int_0^{t_f} \left[\frac{\partial F}{\partial x} \delta x + \frac{\partial F}{\partial u} \delta u \right] dt + \left[F(t_f) + \frac{\partial G}{\partial x} f(t_f) \right] \delta t_f \end{aligned} \quad (8)$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

8

Cơ sở lý thuyết điều khiển tối ưu

Tính gần đúng theo xấp xỉ bậc nhất nghiệm $\delta \mathbf{x}(t)$ ứng với $\delta \mathbf{u}(t)$:

$$\dot{\mathbf{x}}_0(t) + \delta \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}_0(t) + \delta \mathbf{x}(t), \mathbf{u}_0(t) + \delta \mathbf{u}(t)),$$

$$\delta \dot{\mathbf{x}}(t) = \frac{d}{dt} \delta \mathbf{x} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \delta \mathbf{x}(t) + \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \delta \mathbf{u}(t),$$

Nhân hai vế phương trình trên với $\lambda^T(t)$ và lấy tích phân từ 0 đến t_f được

$$\int_0^{t_f} \left[\lambda^T(t) \frac{d(\delta \mathbf{x})}{dt} \right] dt - \int_0^{t_f} \lambda^T(t) \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \delta \mathbf{x}(t) + \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \delta \mathbf{u}(t) \right] dt = 0 \quad (9)$$

Cộng (8) với (9), thực hiện tích phân từng phần số hạng đầu của (9) được

$$\begin{aligned} \delta J = & \left[F(t_f) + \frac{\partial G}{\partial \mathbf{x}} \mathbf{f}(t_f) \right] \delta t_f + \lambda^T(0) \delta \mathbf{x}(0) + \left[\frac{\partial G}{\partial \mathbf{x}} - \lambda^T(t_f) \right] \delta \mathbf{x}(t_f) \\ & + \int_0^{t_f} \left[\frac{\partial H}{\partial \mathbf{x}} \delta \mathbf{x}(t) + \frac{\partial H}{\partial \mathbf{u}} \delta \mathbf{u}(t) + \frac{d\lambda^T(t)}{dt} \delta \mathbf{x} \right] dt \end{aligned} \quad (10)$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

9

Cơ sở lý thuyết điều khiển tối ưu

Trong (10) H được gọi là Hamilton, định nghĩa như sau

$$H = F(\mathbf{x}, \mathbf{u}) + \lambda^T(t) \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (11)$$

Do $\lambda(t)$ là tùy ý, nên ta chọn nó thỏa mãn

$$\frac{d\lambda^T(t)}{dt} = - \frac{\partial H}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{u}, \lambda) \quad (12)$$

Số hạng ngoài dấu tích phân của (10) là các số hạng liên quan điều kiện biên, chúng sẽ không xuất hiện trong các bài toán cụ thể:

Nếu t_f và $\mathbf{x}(0)$ xác định (cố định) thì ta có $\delta t_f = 0$ và $\delta \mathbf{x}(0) = 0$; số hạng thứ 3 ngoài dấu tích phân triệt tiêu nếu chọn

$$\lambda^T(t_f) = \left(\frac{\partial G}{\partial \mathbf{x}} \right)_{t=t_f} \quad (13)$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

10

Cơ sở lý thuyết điều khiển tối ưu

Do đó, phương trình (10) được viết thành

$$\delta J = \int_0^{t_f} \left[\frac{\partial H}{\partial \mathbf{u}} \delta \mathbf{u}(t) \right] dt \quad (14)$$

Cực tiểu hàm J

Nếu $\mathbf{u}_0(t)$ là nghiệm tối ưu (J đạt cực tiểu) thì $\delta J \geq 0$, $\forall \delta \mathbf{u}(t)$

Trường hợp 1. Không có ràng buộc đặt trên $\mathbf{u}(t)$, thì

$$\frac{\partial H}{\partial \mathbf{u}} = 0, \quad \forall t \quad \text{Đây là điều kiện cần để tối ưu J.}$$

Trường hợp 2. Khi có ràng buộc đặt trên $\mathbf{u}(t)$, [bị chặn trên và dưới], thì

$$u_{i,0} = u_{i,M}, \quad \frac{\partial H}{\partial u_i} \leq 0, \quad \& \quad u_{i,0} = u_{i,m}, \quad \frac{\partial H}{\partial u_i} \geq 0$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

11

Cơ sở lý thuyết điều khiển tối ưu

Do đó, phương trình (10) được viết thành

$$\delta J = \int_0^{t_f} \left[\frac{\partial H}{\partial \mathbf{u}} \delta \mathbf{u}(t) \right] dt \quad (14)$$

Cực đại hàm J

Nếu $\mathbf{u}_0(t)$ là nghiệm tối ưu (J đạt cực đại) thì $\delta J \leq 0$, $\forall \delta \mathbf{u}(t)$

Trường hợp 1. Không có ràng buộc đặt trên $\mathbf{u}(t)$, thì

$$\frac{\partial H}{\partial \mathbf{u}} = 0, \quad \forall t \quad \text{Đây là điều kiện cần để tối ưu J.}$$

Trường hợp 2. Khi có ràng buộc đặt trên $\mathbf{u}(t)$, [bị chặn trên và dưới], thì

$$u_{i,0} = u_{i,M}, \quad \frac{\partial H}{\partial u_i} \geq 0, \quad \& \quad u_{i,0} = u_{i,m}, \quad \frac{\partial H}{\partial u_i} \leq 0$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

12

Cơ sở lý thuyết điều khiển tối ưu

Các tính chất của Hamilton đối với hệ autonomous $\dot{x} = f(x, u)$,

$$H = H(x, u, \lambda) \Rightarrow \frac{dH}{dt} = \frac{\partial H}{\partial x} \frac{dx}{dt} + \frac{\partial H}{\partial u} \frac{du}{dt} + \frac{\partial H}{\partial \lambda} \frac{d\lambda}{dt} \quad (15)$$

Nhưng $\frac{\partial H}{\partial \lambda} = f^T(x, u)$ và theo cách chọn $\frac{d\lambda^T(t)}{dt} = -\frac{\partial H}{\partial x}(x, u, \lambda)$

nên
$$\frac{dH}{dt} = \frac{\partial H}{\partial u} \frac{du}{dt} \quad (16)$$

Trường hợp 1. Không có ràng buộc đặt trên $u(t)$, thì

$$\frac{\partial H}{\partial u} = 0$$

Trường hợp 2. Khi có ràng buộc đặt trên $u(t)$, thì $u = u_{\max}$ hoặc $u = u_{\min}$ nên

$$\frac{du}{dt} = 0$$

Do đó, đối với điều khiển tối ưu ta có $\frac{dH}{dt} = 0$ hay $H = \text{const.}$ (16)

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

13

Cơ sở lý thuyết điều khiển tối ưu

Trường hợp đặc biệt: Thời điểm cuối không xác định $\delta t_f \neq 0$.

$$\begin{aligned} \delta J = & \left[F(t_f) + \frac{\partial G}{\partial x} f(t_f) \right] \delta t_f + \lambda^T(0) \delta x(0) + \left[\frac{\partial G}{\partial x} - \lambda^T(t_f) \right] \delta x(t_f) \\ & + \int_0^{t_f} \left[\frac{\partial H}{\partial x} \delta x(t) + \frac{\partial H}{\partial u} \delta u(t) + \frac{d\lambda^T(t)}{dt} \delta x \right] dt \end{aligned} \quad (10)$$

Để δJ không phụ thuộc δt_f , ta cho

$$F(t_f) + \frac{\partial G}{\partial x} f(t_f) = 0 \quad (17)$$

Nếu điều kiện cuối về trạng thái không xác định $\delta x(t_f)$, điều kiện (13) được thỏa mãn và do đó

$$\begin{aligned} H &= F(x, u) + \lambda^T(t) f(x, u) \\ H(t_f) &= F(t_f) + \lambda^T(t_f) f(t_f) = 0 \end{aligned} \quad (18)$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

14

Cơ sở lý thuyết điều khiển tối ưu

Trường hợp nếu trạng thái cuối không xác định, với xấp xỉ bậc nhất trong khai triển Taylor, ta có

$$\delta \mathbf{x}(t_f) = \mathbf{x}(t_f) - \mathbf{x}_0(t_f) = -\mathbf{f}(t_f)\delta t_f \quad (19)$$

Với các biểu thức trên, ta có

$$\delta J = \left[F(t_f) + \boldsymbol{\lambda}^T(t_f)\mathbf{f}(t_f) \right] \delta t_f + \boldsymbol{\lambda}^T(0)\delta \mathbf{x}(0) + \int_0^{t_f} \left[\frac{\partial H}{\partial \mathbf{u}} \delta \mathbf{u}(t) \right] dt \quad (20)$$

Để loại bỏ số hạng liên quan điều kiện biên khi t_f không xác định

$$H(t_f) = F(t_f) + \boldsymbol{\lambda}^T(t_f)\mathbf{f}(t_f) = 0$$

Vì $H(t) = \text{const}$, nên $H(t) = H(t_f) = 0$ dọc theo quỹ đạo tối ưu đối với hệ autonomous, khi thời điểm cuối t_f không xác định.

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

15

Điều khiển tối ưu LQR hệ tuyến tính

Xét hệ động lực tuyến tính, với cặp $(\mathbf{A}, \mathbf{B})$ là điều khiển được

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad \mathbf{x}(0) = \mathbf{x}_0 \quad (21)$$

Và tiêu chuẩn tối ưu sau cần được cực tiểu

$$J = \frac{1}{2} \mathbf{x}^T(t_f) \mathbf{F} \mathbf{x}(t_f) + \frac{1}{2} \int_0^{t_f} \left[\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u} \right] dt \quad (22)$$

$$\mathbf{F} = \mathbf{F}^T \geq 0, \quad \mathbf{Q} = \mathbf{Q}^T \geq 0, \quad \mathbf{R} = \mathbf{R}^T > 0$$

$$u_{i,m} \leq u_i \leq u_{i,M}, \quad i = 1, 2, \dots, n$$

Hàm Hamilton trong bài toán này là

$$H = \frac{1}{2} \left\{ \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u} \right\} + \boldsymbol{\lambda}^T(t) (\mathbf{A} \mathbf{x}(t) + \mathbf{B} \mathbf{u}(t)) \quad (23)$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

16

Điều khiển tối ưu LQR hệ tuyến tính

Lời giải bài toán tối ưu vòng mở: Từ điều kiện cần cho nghiệm tối ưu

$$\begin{aligned}\frac{\partial H}{\partial \mathbf{u}} = 0 &\Rightarrow \mathbf{R}\mathbf{u} + \mathbf{B}^T \boldsymbol{\lambda}(t) = 0 \\ &\Rightarrow \mathbf{u} = -\mathbf{R}^{-1} \mathbf{B}^T \boldsymbol{\lambda}(t)\end{aligned}\quad (24)$$

Động học của $\boldsymbol{\lambda}(t)$ đưa ra từ (12) cùng với điều kiện cuối (13)

$$\frac{d\boldsymbol{\lambda}(t)}{dt} = -\mathbf{Q}\mathbf{x} - \mathbf{A}^T \boldsymbol{\lambda}(t), \quad \boldsymbol{\lambda}(t_f) = \mathbf{F}\mathbf{x}(t_f) \quad (25)$$

Kết hợp (21)-(24)-(25) ta được

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{R}^{-1} \mathbf{B}^T \boldsymbol{\lambda}(t), \quad \mathbf{x}(0) = \mathbf{x}_0 \\ \dot{\boldsymbol{\lambda}}(t) &= -\mathbf{Q}\mathbf{x} - \mathbf{A}^T \boldsymbol{\lambda}(t), \quad \boldsymbol{\lambda}(t_f) = \mathbf{F}\mathbf{x}(t_f)\end{aligned}$$

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\boldsymbol{\lambda}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{B}\mathbf{R}^{-1} \mathbf{B}^T \\ -\mathbf{Q} & -\mathbf{A}^T \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \boldsymbol{\lambda}(t) \end{bmatrix} = \mathbf{M} \begin{bmatrix} \mathbf{x}(t) \\ \boldsymbol{\lambda}(t) \end{bmatrix} \quad (26)$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

17

Điều khiển tối ưu LQR hệ tuyến tính

Giải (26) tìm được nghiệm

$$\begin{bmatrix} \mathbf{x}(t) \\ \boldsymbol{\lambda}(t) \end{bmatrix} = e^{\mathbf{M}t} \begin{bmatrix} \mathbf{x}(0) \\ \boldsymbol{\lambda}(0) \end{bmatrix}$$

Xác định điều kiện đầu $\boldsymbol{\lambda}(0)$ ta viết

$$e^{\mathbf{M}t} = \begin{bmatrix} \mathbf{E}_{11}(t) & \mathbf{E}_{12}(t) \\ \mathbf{E}_{21}(t) & \mathbf{E}_{22}(t) \end{bmatrix}$$

Suy ra

$$\begin{aligned}\mathbf{x}(t) &= \mathbf{E}_{11}(t)\mathbf{x}(0) + \mathbf{E}_{12}(t)\boldsymbol{\lambda}(0) \\ \boldsymbol{\lambda}(t) &= \mathbf{E}_{21}(t)\mathbf{x}(0) + \mathbf{E}_{22}(t)\boldsymbol{\lambda}(0)\end{aligned}\quad (27)$$

Và từ điều kiện $\boldsymbol{\lambda}(t_f) = \mathbf{F}\mathbf{x}(t_f)$

$$\begin{aligned}\Rightarrow \mathbf{E}_{21}(t_f)\mathbf{x}(0) + \mathbf{E}_{22}(t_f)\boldsymbol{\lambda}(0) &= \mathbf{F}[\mathbf{E}_{11}(t_f)\mathbf{x}(0) + \mathbf{E}_{12}(t_f)\boldsymbol{\lambda}(0)] \\ \Rightarrow [\mathbf{E}_{22}(t_f) - \mathbf{F}\mathbf{E}_{12}(t_f)]\boldsymbol{\lambda}(0) &= [\mathbf{F}\mathbf{E}_{11}(t_f) - \mathbf{E}_{21}(t_f)]\mathbf{x}(0) \\ \Rightarrow \boldsymbol{\lambda}(0) &= [\mathbf{E}_{22}(t_f) - \mathbf{F}\mathbf{E}_{12}(t_f)]^{-1} [\mathbf{F}\mathbf{E}_{11}(t_f) - \mathbf{E}_{21}(t_f)]\mathbf{x}(0)\end{aligned}$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

18

Điều khiển tối ưu LQR hệ tuyến tính

Sau khi xác định được $\lambda(0)$, ta sẽ có $\lambda(t)$ và $u(t)$. Tuy nhiên, bài toán cần phải giải lại khi điều kiện đầu thay đổi. Ngoài ra, đầu vào điều khiển $u(t)$ là khả thi chỉ trong biểu diễn, vì điều khiển ở dạng này không phải là hàm của các biến trạng thái (vòng điều khiển mở).

Lời giải bài toán tối ưu vòng điều khiển kín (phản hồi trạng thái):

Sử dụng phép biến đổi sau

$$\lambda(t) = S(t)x(t), \quad S = S^T \quad (28)$$

Thay vào (25) ta được

$$\frac{d\lambda(t)}{dt} = \dot{S}x + S\dot{x} = -Qx - A^T Sx, \quad (29)$$

Cùng với (21, 24) ta được

$$(\dot{S} + SA - SBR^{-1}B^T S + Q + A^T S)x = 0 \quad (30)$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

19

Điều khiển tối ưu LQR hệ tuyến tính

Biểu thức (30) đúng với mọi $x(t)$ nên

$$\dot{S} + SA - SBR^{-1}B^T S + Q + A^T S = 0 \quad (31)$$

Từ (13) và (28) cho ta

$$\lambda^T(t_f) = \left(\frac{\partial G}{\partial x} \right)_{t=t_f} = Fx(t_f), \lambda(t_f) = S(t_f)x(t_f) \Rightarrow S(t_f) = F \quad (32)$$

Phương trình (31) được gọi là phương trình Riccati. Phương trình vi phân phi tuyến này có thể được giải quyết bằng phương pháp số để nhận được ma trận $S(t)$. Từ đó xác định được véc tơ điều khiển

$$u = -R^{-1}B^T S(t)x(t) = -K(t)x(t), \quad (33)$$

$$K(t) = R^{-1}B^T S(t)$$

Cấu trúc của (33) có dạng điều khiển phản hồi trạng thái với ma trận tối ưu $K(t)$. Vì lời giải $S(t)$ không phụ thuộc vào trạng thái hệ, nên điều khiển này là tối ưu cho tất cả các điều kiện ban đầu của véc tơ trạng thái, $x(0)$.

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

20

Điều khiển tối ưu LQR hệ tuyến tính

Cho hệ tuyến tính

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad (\mathbf{A}, \mathbf{B}) \quad \text{là điều khiển được}$$

Thuật giải

1. Chọn các ma trận $\mathbf{Q} = \mathbf{Q}^T \geq 0$, $\mathbf{R} = \mathbf{R}^T > 0$ và $\mathbf{F} = \mathbf{F}^T \geq 0$
2. Giải phương trình Riccati (31), hệ ptvp thỏa mãn điều kiện cuối

$$\dot{\mathbf{S}} + \mathbf{S}\mathbf{A} - \mathbf{S}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{S} + \mathbf{Q} + \mathbf{A}^T\mathbf{S} = 0, \quad \mathbf{S}(t_f) = \mathbf{F}$$

3. Xác định ma trận $\mathbf{K}(t)$ từ đó xác định véc tơ điều khiển

$$\mathbf{K}(t) = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{S}(t), \quad \mathbf{u} = -\mathbf{K}(t)\mathbf{x}(t)$$

4. Đánh giá (xác định) giá trị hàm mục tiêu:

$$J = \frac{1}{2}\mathbf{x}^T(t_f)\mathbf{F}\mathbf{x}(t_f) + \frac{1}{2}\int_0^{t_f} [\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u}]dt$$

Nhận xét: gặp khó khăn khi giải phương trình Riccati. Hàm mục tiêu đôi khi không có ý nghĩa vật lý, nhưng với các $\mathbf{Q}$ và $\mathbf{R}$ khác nhau sẽ cho đáp ứng khác nhau.

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

21

Điều khiển tối ưu LQR hệ tuyến tính

Trường hợp quan trọng: $t_f \rightarrow \infty$

Xét trường hợp thời gian cuối t_f tiến đến vô cùng (thời gian điều khiển tương đối lớn). Nghiệm $\mathbf{S}(t)$ hay $\mathbf{K}(t)$ tiến đến ma trận hằng. [xem tài liệu tk]. Trong trường hợp này, ta sẽ giải phương trình đại số Riccati (Algebraic Riccati Equation-ARE) thay vì giải phương trình vi phân Riccati.

1. Chọn các ma trận $\mathbf{Q} = \mathbf{Q}^T \geq 0$, $\mathbf{R} = \mathbf{R}^T > 0$ và $\mathbf{F} = \mathbf{F}^T \geq 0$

2. Xác định ma trận $\mathbf{S}$ từ phương trình đại số Riccati

$$\mathbf{S}\mathbf{A} - \mathbf{S}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{S} + \mathbf{Q} + \mathbf{A}^T\mathbf{S} = 0, \quad \text{do} \quad \dot{\mathbf{S}} = 0,$$

3. Xác định ma trận $\mathbf{K}(t)$ từ đó xác định véc tơ điều khiển

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{S}, \quad \mathbf{u} = -\mathbf{K}\mathbf{x}(t)$$

4. Kiểm tra tính ổn định của hệ kín $\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}$

5. Đánh giá (xác định) giá trị hàm mục tiêu:

$$J = \frac{1}{2}\mathbf{x}^T(t_f)\mathbf{F}\mathbf{x}(t_f) + \frac{1}{2}\int_0^{t_f} [\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u}]dt$$

Lệnh Matlab:

>> K = lqr(A,B,Q,R)

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

22

Điều khiển tối ưu LQR hệ tuyến tính

Xét trường hợp có số hạng chéo trong hàm mục tiêu

$$J = \frac{1}{2} \mathbf{x}^T(t_f) \mathbf{F} \mathbf{x}(t_f) + \frac{1}{2} \int_0^{t_f} [\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u} + 2 \mathbf{x}^T \mathbf{N} \mathbf{u}] dt$$

Khi đó sử dụng phép biến đổi

$$\mathbf{x}^T \mathbf{Q} \mathbf{x} + 2 \mathbf{x}^T \mathbf{N} \mathbf{u} + \mathbf{u}^T \mathbf{R} \mathbf{u} = \mathbf{x}^T \mathbf{Q}_m \mathbf{x} + \mathbf{v}^T \mathbf{R} \mathbf{v}$$

Với

$$\mathbf{Q}_m = \mathbf{Q} - \mathbf{N} \mathbf{R}^{-1} \mathbf{N}^T \quad \& \quad \mathbf{v} = \mathbf{u} + \mathbf{R}^{-1} \mathbf{N}^T \mathbf{x}$$

Khi đó hàm mục tiêu trở thành

$$J = \frac{1}{2} \mathbf{x}^T(t_f) \mathbf{F} \mathbf{x}(t_f) + \frac{1}{2} \int_0^{t_f} [\mathbf{x}^T \mathbf{Q}_m \mathbf{x} + \mathbf{v}^T \mathbf{R} \mathbf{v}] dt$$

Hệ ban đầu được biến đổi tương ứng thành

$$\dot{\mathbf{x}} = \mathbf{A}_m \mathbf{x} + \mathbf{B} \mathbf{v}, \quad \text{with} \quad \mathbf{A}_m = \mathbf{A} - \mathbf{B} \mathbf{R}^{-1} \mathbf{N}^T,$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

23

Điều khiển tối ưu LQR hệ tuyến tính

Giả sử rằng $\mathbf{Q}_m$ đối xứng, bán xác định dương, khi đó bài toán được biến đổi về dạng như đã trình bày. Lời giải của bài toán này có nghiệm như sau:

Thuật giải

1. Chọn các ma trận $\mathbf{Q} = \mathbf{Q}^T \geq 0$, $\mathbf{R} = \mathbf{R}^T > 0$, $\mathbf{F} = \mathbf{F}^T \geq 0$, và ma trận $\mathbf{N}$

2. Tính các ma trận $\mathbf{Q}_m$ và $\mathbf{A}_m$ (kiểm tra ma trận $\mathbf{Q}_m \geq 0$, bán x.đ.d)

$$\mathbf{Q}_m = \mathbf{Q} - \mathbf{N} \mathbf{R}^{-1} \mathbf{N}^T, \quad \mathbf{A}_m = \mathbf{A} - \mathbf{B} \mathbf{R}^{-1} \mathbf{N}^T$$

3. Giải phương trình Riccati (31), hệ ptvp thỏa mãn điều kiện cuối

$$\dot{\mathbf{S}}_m + \mathbf{S}_m \mathbf{A}_m - \mathbf{S}_m \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{S}_m + \mathbf{Q}_m + \mathbf{A}_m^T \mathbf{S}_m = 0, \quad \mathbf{S}_m(t_f) = \mathbf{F}$$

4. Xác định ma trận $\mathbf{K}_m(t)$ từ đó xác định véc tơ điều khiển

$$\mathbf{K}_m(t) = \mathbf{R}^{-1} [\mathbf{B}^T \mathbf{S}_m(t) + \mathbf{N}^T],$$

$$\mathbf{u} = -\mathbf{K}_m(t) \mathbf{x}(t)$$

Lệnh Matlab:

>> K = lqr(A,B,Q,R)

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

24

Ví dụ 1

Xét hệ động lực bậc nhất và hàm mục tiêu như sau

$$\dot{x} = x + u, \quad J = \frac{5}{2}x(t_f)^2 + \frac{1}{2}\int_0^{t_f} (2x^2 + 3u^2)dt$$

Với

$$A = 1, \quad b = 1, \quad F = 5, \quad Q = 2, \quad R = 3$$

Phương trình Riccati trở thành

$$\dot{S} = -2S + \frac{1}{3}S^2 - 2, \quad S(t_f) = F = 5$$

Với t_f cho trước, giải phương trình vi phân trên ta nhận được $S(t)$, $0 \leq t \leq t_f$

Luật điều khiển tối ưu sẽ là

$$u(t) = -\frac{1}{3}S(t)x(t), \quad 0 \leq t \leq t_f$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

25

Ví dụ 1

Nếu cho $t_f \rightarrow \infty$

$$\dot{x} = x + u, \quad J = \frac{5}{2}x(t_f)^2 + \frac{1}{2}\int_0^\infty (2x^2 + 3u^2)dt$$

Với

$$A = 1, \quad b = 1, \quad F = 5, \quad Q = 2, \quad R = 3$$

Phương trình đại số Riccati trở thành

$$0 = -2S + \frac{1}{3}S^2 - 2 \Rightarrow S_{1,2} = 3 \pm \sqrt{15}$$

Để hệ kín ổn định ta chọn

$$S = S_1 = 3 + \sqrt{15} \quad \Rightarrow \quad K = \frac{3 + \sqrt{15}}{3}$$

Luật điều khiển tối ưu sẽ là

$$u = -Kx, \quad \Rightarrow \quad \dot{x} = -\frac{\sqrt{15}}{3}x$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

26

Ví dụ 2

Xét hệ tuyến tính sau:

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

Tìm luật điều khiển $u = -k_1(x_1 - r) - k_2 x_2$ để duy trì $x_1 = r, x_2 = 0$

cực tiểu chỉ tiêu $J = \frac{1}{2} \int_0^\infty ((x_1 - r)^2 + u^2) dt$ (tức cho $t_f \rightarrow \infty$)

Đặt biến mới

$$\begin{aligned} \tilde{x}_1 &= x_1 - r \\ \tilde{x}_2 &= x_2 \end{aligned} \Rightarrow \begin{bmatrix} \dot{\tilde{x}}_1 \\ \dot{\tilde{x}}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \quad J = \frac{1}{2} \int_0^\infty (\tilde{x}_1^2 + u^2) dt$$

$$\mathbf{Q} = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}; \quad \mathbf{R} = 2$$

Phương trình đại số Riccati trở thành

$$\mathbf{S}\mathbf{A} - \mathbf{S}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{S} + \mathbf{Q} + \mathbf{A}^T\mathbf{S} = 0$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

27

Ví dụ 2

Giải phương trình đại số Riccati

$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} \\ S_{12} & S_{22} \end{bmatrix} + \begin{bmatrix} S_{11} & S_{12} \\ S_{12} & S_{22} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} S_{11} & S_{12} \\ S_{12} & S_{22} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ S_{12} & S_{22} \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\Rightarrow -\frac{S_{12}^2}{2} + 2 = 0, \quad S_{11} - \frac{S_{12}S_{22}}{2} = 0, \quad -\frac{S_{22}^2}{2} + 2S_{12} = 0$$

Kết quả nhận được ma trận $\mathbf{S}$ và luật điều khiển phản hồi trạng thái như sau

$$\mathbf{S} = \begin{bmatrix} 2\sqrt{2} & 2 \\ 2 & 2\sqrt{2} \end{bmatrix}$$

$$u = -\mathbf{R}^{-1}\mathbf{B}^T\mathbf{S}\tilde{\mathbf{x}} = -\tilde{x}_1 - \sqrt{2}\tilde{x}_2 = -(x_1 - r) - \sqrt{2}x_2$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

28

Ví dụ 2 - Matlab

Xét hệ tuyến tính sau:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u,$$

$$J = \frac{1}{2} \int_0^\infty (\tilde{x}_1^2 + u^2) dt, \quad \tilde{x}_1 = x_1 - r$$

Hệ và bài toán tối ưu mô tả bởi:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}; \quad \mathbf{R} = 2$$

Phương trình đại số Riccati

$$\mathbf{S}\mathbf{A} - \mathbf{S}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{S} + \mathbf{Q} + \mathbf{A}^T\mathbf{S} = 0$$

Lệnh lqr của matlab

>> [K, S, e] = lqr(A,B,Q,R)

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

29

Ví dụ 2 - Matlab

Lệnh lqr của matlab

```
A = [0 1; 0 0];
B = [0; 1];
C = [1 0];
Q = [2 0; 0 0];
R = [2];
[K, S, e] = lqr(A, B, Q, R)
```

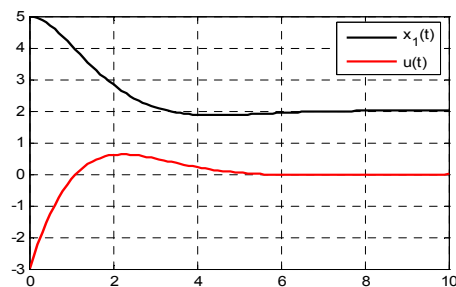
K =
1.0000 1.4142
S =
2.8284 2.0000
2.0000 2.8284
e =
-0.7071 + 0.7071i
-0.7071 - 0.7071i

e là trị riêng của ma trận (A - BK)

HUST

Hanoi University of Science and Technology

```
closed_Syst = ss(A - B*K, B*K(1,1), C, 0)
t = 0:0.1:10;
r = 2*ones(size(t));
[y, t, x] = lsim(closed_Syst, r, t, [5 0]);
u = -K*x' + K(1,1)*r;
plot(t, y, 'k', 'linewidth', 2), grid on, hold on
plot(t, u, 'r', 'linewidth', 2)
legend('x_1(t)', 'u(t)')
```



Nguyen Quang Hoang
Department of Applied Mechanics

30

Bài tập

Bài 1. Xét hệ tuyến tính sau cùng với hàm mục tiêu cần được cực tiểu hóa:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u,$$

$$J = \frac{1}{2} \times 20x_1^2(5) + \frac{2}{2} \int_0^5 (x_1^2 + 2x_2^2 + x_1x_2 + u^2)dt, \quad t_f = 5$$

- o Thiết lập các phương trình vi phân Riccati với điều kiện biên thích hợp;
- o Giải các phương trình vi phân Riccati bằng các lệnh ode23 hoặc ode45 của Matlab;
- o Tìm ma trận $\mathbf{S}(t)$, ma trận $\mathbf{K}(t)$ và đáp ứng vòng kín. Ngoài ra, tìm giá trị tối ưu của J .

HUST

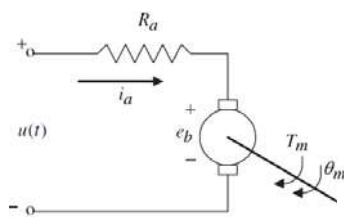
Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

31

Tối ưu năng lượng tiêu thụ động cơ DC

Bài 2. Thiết lập bài toán điều khiển vị trí cho động cơ điện một chiều, với chỉ tiêu cực tiểu hóa năng lượng.



$$\ddot{\theta}_m + \alpha \dot{\theta}_m = \beta u(t)$$

$$\alpha = \frac{B_m}{J_m} + \frac{k_i k_b}{R_a J_m} \quad \beta = \frac{k_i}{R_a J_m}$$

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t)$$

$$\mathbf{x}(t) = \begin{bmatrix} \theta_m(t) \\ \dot{\theta}_m(t) \end{bmatrix}; \quad \mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & -\alpha \end{bmatrix}; \quad \mathbf{b} = \begin{bmatrix} 0 \\ \beta \end{bmatrix}$$

Hãy đưa vào hàm cần tối ưu, từ đó tìm hàm tác động (điều khiển) $u(t)$, $0 \leq t \leq t_f$, để tiêu hao năng lượng trong khoảng thời gian xác định (cho biết t_f) bởi động cơ DC là nhỏ nhất. Biết điều kiện đầu và điều kiện cuối như sau

$$\mathbf{x}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{x}(t_f) = \begin{bmatrix} \theta_f \\ 0 \end{bmatrix}$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

32

Tối ưu năng lượng tiêu thụ động cơ DC

Năng lượng tiêu thụ của động cơ DC trong một thời gian cố định (từ 0 đến t_f) được cho bởi tích phân sau:

$$I_d = \int_0^{t_f} T_m(t) d\theta_m = \int_0^{t_f} T_m \dot{\theta}_m dt \quad \Rightarrow \quad I_d = \int_0^{t_f} \frac{k_i}{R_a} (u x_2 - k_b x_2^2) dt$$

Để thỏa mãn các yêu cầu đối với sự tồn tại nghiệm của bài toán điều khiển LQR, hàm mục tiêu được sửa đổi để được

$$I = \int_0^{t_f} \frac{k_i}{R_a} (u x_2 - k_b x_2^2) + \gamma x_2^2 + \rho u^2 dt; \quad \gamma > 0 \quad \text{and} \quad \rho > 0$$

$$I = \frac{1}{2} \int_0^{t_f} \mathbf{x}^T \mathbf{Q} \mathbf{x} + 2 \mathbf{x}^T \mathbf{N} \mathbf{u} + \mathbf{u}^T \mathbf{R} \mathbf{u} dt$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

33

Tối ưu năng lượng tiêu thụ động cơ DC

Hàm mục tiêu cần tối thiểu

$$I = \frac{1}{2} \int_0^{t_f} \mathbf{x}^T \mathbf{Q} \mathbf{x} + 2 \mathbf{x}^T \mathbf{N} \mathbf{u} + \mathbf{u}^T \mathbf{R} \mathbf{u} dt$$

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 \\ 0 & 2(\rho - \frac{k_i k_b}{R_a}) \end{bmatrix}; \quad \mathbf{N} = \begin{bmatrix} 0 \\ \frac{k_i}{R_a} \end{bmatrix}; \quad \mathbf{R} = 2\rho$$

Đưa biểu thức trong dấu tích phân về dạng của bài toán điều khiển LQR

$$\mathbf{x}^T \mathbf{Q} \mathbf{x} + 2 \mathbf{x}^T \mathbf{N} \mathbf{u} + \mathbf{u}^T \mathbf{R} \mathbf{u} = \mathbf{x}^T \mathbf{Q}_m \mathbf{x} + \mathbf{v}^T \mathbf{R} \mathbf{v}$$

ta có được (xem trường hợp có số hạng chéo, $\mathbf{x}^T \mathbf{N} \mathbf{u}$)

$$\mathbf{Q}_m = \mathbf{Q} - \mathbf{N} \mathbf{R}^{-1} \mathbf{N}^T$$

$$\mathbf{v} = \mathbf{u} + \mathbf{R}^{-1} \mathbf{N}^T \mathbf{x}$$

$$\mathbf{Q}_m = \begin{bmatrix} 0 & 0 \\ 0 & (2\gamma - \frac{2k_i k_b}{R_a} - \frac{k_i^2}{2R_a^2 \rho}) \end{bmatrix}$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

34

Tối ưu năng lượng tiêu thụ động cơ DC

Để ma trận $\mathbf{Q}_m \geq 0$, các tham số $\rho > 0$ và $\gamma > 0$ cần được chọn sao cho

$$2\gamma - \frac{2k_i k_b}{R_a} - \frac{k_i^2}{2R_a^2 \rho} \geq 0$$

Để có điều kiện trạng thái cuối bằng 0, ta sử dụng phép biến đổi

$$\tilde{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}(t_f)$$



$$\tilde{\mathbf{x}}(0) = -\mathbf{x}(t_f) \quad \text{and} \quad \tilde{\mathbf{x}}(t_f) = 0$$

Khi đó phương trình trạng thái với biến mới sẽ là

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}\tilde{\mathbf{x}}(t) + \mathbf{b}u(t)$$

Do hàm mục tiêu không chứa các biến x_1 và $x_2 = x_2^*$, hàm mục tiêu được viết lại thành

$$I = \frac{1}{2} \int_0^{t_f} \tilde{\mathbf{x}}^T \mathbf{Q} \tilde{\mathbf{x}} + 2\tilde{\mathbf{x}}^T \mathbf{N} u + u^T \mathbf{R} u dt$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

35

Tối ưu năng lượng tiêu thụ động cơ DC

Trong điều khiển LQ, không có ràng buộc về điều kiện cuối cùng (trạng thái cuối). Do đó, hàm mục tiêu được chỉnh sửa để chứa các vector trạng thái cuối:

$$I = \frac{1}{2} \tilde{\mathbf{x}}^T(t_f) \mathbf{S}_f \tilde{\mathbf{x}}(t_f) + \frac{1}{2} \int_0^{t_f} \tilde{\mathbf{x}}^T \mathbf{Q} \tilde{\mathbf{x}} + 2\tilde{\mathbf{x}}^T \mathbf{N} u + u^T \mathbf{R} u dt$$

Với ma trận $\mathbf{S}_f$

$$\mathbf{S}_f = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}; \quad a > 0 \text{ and } b > 0$$

Bỏ qua ràng buộc về trạng thái cuối, bài toán LQ có thể được giải để tối thiểu hàm mục tiêu của hệ tuyến tính với điều kiện đầu khác không. Bằng cách chọn các số a và b đủ lớn, sẽ giảm ảnh hưởng của trạng thái cuối đến lời giải [giảm vai trò của của $\tilde{\mathbf{x}}(t_f)$].

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

36

Phân tích hệ động lực

Hệ LTI Ổn định của hệ tuyến tính bất biến theo thời gian

Nguyen Q. Hoang
Department of Applied Mechanics
Hanoi University of Science and Technology

Tổng quan về tính ổn định

Về mặt tổng quát, rất khó có thể nhận được nghiệm dạng hiển của phương trình vi phân phi tuyến. Tuy nhiên, thông thường ta không cần biết nghiệm dạng hiển của hệ phương trình mô tả hệ. Nhưng chúng ta cần quan tâm đáp ứng của hệ khi thời gian tiến về vô cùng [sau khoảng thời gian đủ dài].

Hơn nữa, trong nhiều mô hình của hệ thống vật lý một sự thay đổi "nhỏ" các điều kiện đầu dẫn đến một sự thay đổi "nhỏ" của các nghiệm. Đặc biệt, nếu một hệ thống bị nhiễu từ trạng thái nghỉ của nó, hay vị trí cân bằng, sau đó nó bắt đầu di chuyển. Chúng ta gần như có thể nói rằng vị trí cân bằng là ổn định nếu hệ thống không đi xa khỏi vị trí này đối với nhiễu loạn nhỏ ban đầu.

Tuy nhiên, có những hệ mà sự **thay đổi nhỏ** của các điều kiện đầu dẫn đến sự **thay đổi lớn** trong ứng xử của hệ.

Tổng quan về tính ổn định

ỔN ĐỊNH

- Đặc điểm kỹ thuật hết sức quan trọng trong thiết kế điều khiển
- Hệ thống không ổn định phải được ổn định bởi thông tin phản hồi.
- Hệ thống vòng kín không ổn định là vô ích.
- Điều gì xảy ra nếu một hệ thống không ổn định?
 - có thể đạt các giới hạn cơ khí / hay bão hòa điện
 - có thể phá vỡ hoặc gây cháy máy, ...



HUST

Hanoi University of Science and Technology

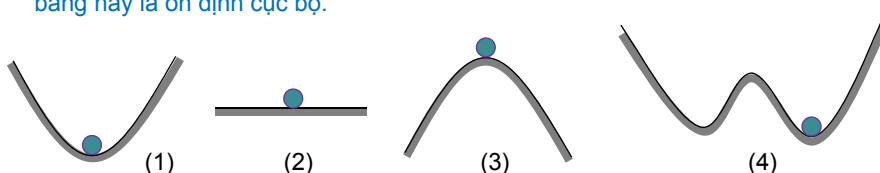
Nguyen Quang Hoang
Department of Applied Mechanics

3

Tổng quan về tính ổn định

Xét quả cầu nhỏ đặt tại bốn vị trí như trên hình vẽ. Nếu bỏ qua lực ma sát, một sự thay đổi nhỏ từ trạng thái cân bằng dẫn đến những điều sau đây:

1. Chuyển động dao động quanh trạng thái cân bằng không phụ thuộc vào độ lớn của nhiễu loạn ban đầu. Trong trường hợp này, vị trí cân bằng ổn định toàn cục.
2. Vị trí mới cũng là một trạng thái cân bằng, và quả bóng nhỏ sẽ ở lại trong nó. Đây là vị trí cân bằng ổn định (phiếm định).
3. Quả bóng di chuyển đi mà không trở về trạng thái cân bằng. Vị trí cân bằng này là không ổn định.
4. Chuyển động dao động về trạng thái cân bằng, trừ khi các nhiễu kích động ban đầu quá lớn làm quả cầu buộc phải chuyển đến vị trí cân bằng mới. Vị trí cân bằng này là ổn định cục bộ.



HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

4

Tổng quan về tính ổn định

Một tiêu chuẩn quan trọng để đánh giá một hệ động lực là đáp ứng ổn định của nó. Tổng quan về tính ổn định được thể hiện như sau:

Một hệ tuyến tính là ổn định nếu đáp ứng đầu ra bị chặn khi tác động đầu vào bị chặn (ổn định BIBO), hoặc nó trở về trạng thái gốc ban đầu (ổn định tiệm cận). Nếu xuất hiện đáp ứng khác với hai điều nêu trên thì hệ tuyến tính là không ổn định.

Kích động ngoài bao gồm cả điều kiện đầu và tác động điều khiển, cũng như nhiễu tác động.

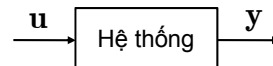
HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

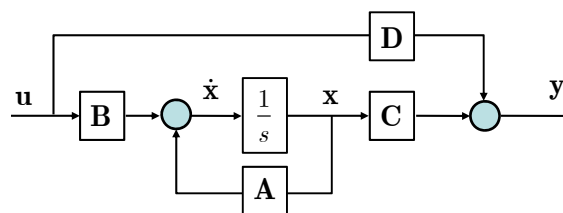
5

Ổn định BIBO



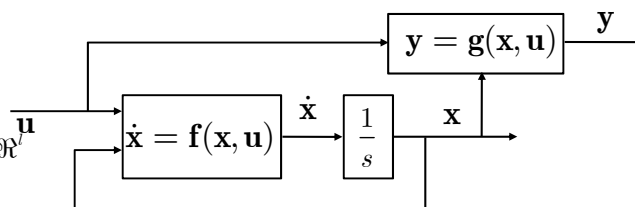
Hệ tuyến tính LTI

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu}, \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du}, \\ \mathbf{x} &\in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^m, \mathbf{y} \in \mathbb{R}^l\end{aligned}$$



Hệ phi tuyến

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}), \\ \mathbf{y} &= \mathbf{g}(\mathbf{x}, \mathbf{u}), \\ \mathbf{x} &\in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^m, \mathbf{y} \in \mathbb{R}^l\end{aligned}$$



HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

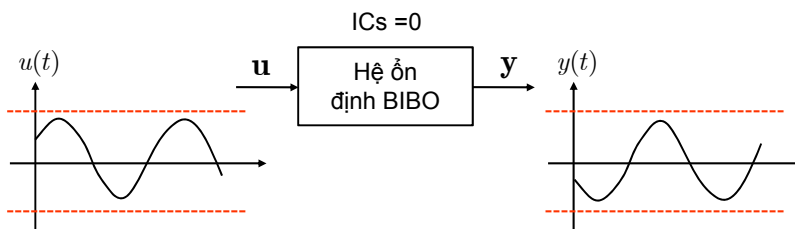
6

Ổn định BIBO

Định nghĩa ổn định BIBO: Một hệ thống được gọi là ổn định BIBO (Bounded Input – Bounded Output], nếu khi kích động đầu vào $u(t)$ bị chặn thì đáp ứng của hệ (đầu ra) $y(t)$ cũng bị chặn.

Nếu với mọi số dương N, M . Mọi đầu vào bị chặn sẽ tạo ra đầu ra tương ứng cũng bị chặn.

$$|u(t)| \leq N < \infty \Rightarrow |y(t)| \leq M < \infty$$



HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

7

Ổn định BIBO

Đối với hệ tuyến tính (SISO) $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}u, \quad \mathbf{x} \in \mathbb{R}^n, u \in \mathbb{R}$
 $y = \mathbf{c}\mathbf{x}, \quad y \in \mathbb{R}$

Tính được hàm truyền

$$G(s) = \frac{p(s)}{q(s)} = \mathbf{c}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}$$

BIBO Stability $\Leftrightarrow$ Tất cả các điểm cực của hàm truyền nằm bên trái trục ảo trong mặt phẳng phức.

Đưa ra phương trình đặc trưng, sau đó giải tìm nghiệm của phương trình đặc trưng.
 $q(s) = 0$

Nghiệm của phương trình đặc trưng chính là các điểm cực của hàm truyền $G(s)$.

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

8

Ổn định tiệm cận hệ tuyến tính

Ổn định tiệm cận [Asymptotic stability]

Khi $u(t) = 0$, hệ trở thành

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}, \quad \mathbf{y} = \mathbf{C}\mathbf{x},$$

$$\mathbf{x}(t) \rightarrow 0 \text{ khi } t \rightarrow \infty, \text{ hay } \mathbf{y}(t) \rightarrow 0$$

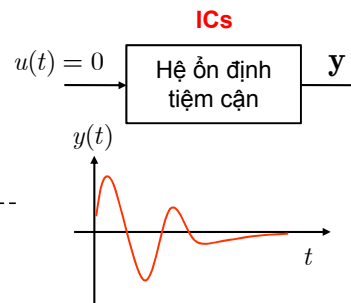
Đối với hệ tuyến tính: $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$,
 $\mathbf{y} = \mathbf{C}\mathbf{x}$,

- Hệ ổn định tiệm cận $\Leftrightarrow$ tất cả các trị riêng của ma trận $\mathbf{A}$ có phần thực âm (nằm bên trái trục ảo)

$$\mathbf{G}(s) = \frac{\mathbf{P}(s)}{q(s)} = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} = \frac{\mathbf{C} \operatorname{adj}[s\mathbf{I} - \mathbf{A}]\mathbf{B}}{|s\mathbf{I} - \mathbf{A}|}$$

$|s\mathbf{I} - \mathbf{A}| = 0 \Rightarrow$ Giải tìm các trị riêng của $\mathbf{A}$.

Lưu ý: Ổn định tiệm cận không phụ thuộc ma trận $\mathbf{B}$ và $\mathbf{C}$



HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

9

Ổn định tiệm cận hệ tuyến tính

Ổn định tiệm cận từ việc tách mô hình

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u},$$

Giả sử $\mathbf{A}$ có **tất cả các trị riêng phân biệt** (chéo hóa được $\mathbf{A}$).

$$\mathbf{y} = \mathbf{C}\mathbf{x},$$

$\lambda_i, \mathbf{v}_i$ là cặp trị riêng và véc tơ riêng của $\mathbf{A}$ $\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i, \quad i = 1, \dots, n$

Xây dựng ma trận chuyển đổi $\mathbf{T} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n]$

Biến đổi tọa độ $\mathbf{x} = \mathbf{T}\xi \Rightarrow \dot{\mathbf{x}} = \mathbf{T}\dot{\xi}$

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} \end{aligned} \Rightarrow \begin{aligned} \dot{\xi} &= \mathbf{T}^{-1}\mathbf{A}\mathbf{T}\xi + \mathbf{T}^{-1}\mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{T}\xi \end{aligned}$$

$$\bar{\mathbf{A}} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T},$$

$$\bar{\mathbf{B}} = \mathbf{T}^{-1}\mathbf{B}, \quad \bar{\mathbf{C}} = \mathbf{C}\mathbf{T}$$

$$\begin{aligned} \dot{\xi} &= \bar{\mathbf{A}}\xi + \bar{\mathbf{B}}\mathbf{u}, \\ \mathbf{y} &= \bar{\mathbf{C}}\xi \end{aligned}$$

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \end{aligned}$$

$$\mathbf{x} = \mathbf{T}\mathbf{z}$$

$$\begin{aligned} \dot{\mathbf{z}} &= \mathbf{T}^{-1}\mathbf{A}\mathbf{T}\mathbf{z} + \mathbf{T}^{-1}\mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{T}\mathbf{z} + \mathbf{D}\mathbf{u} \end{aligned}$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

10

Ổn định tiệm cận hệ tuyến tính

Lưu ý rằng

$$\bar{\mathbf{A}} = \mathbf{T}^{-1} \mathbf{A} \mathbf{T} = \begin{bmatrix} \lambda_1 & 0 & 0 \cdots & 0 \\ 0 & \lambda_2 & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}$$

Vậy nên, khi $u = 0$, ta có

$$\dot{\xi} = \bar{\mathbf{A}} \xi \Rightarrow \begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \vdots \\ \dot{\xi}_n \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 & 0 \cdots & 0 \\ 0 & \lambda_2 & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{bmatrix} \Rightarrow \begin{aligned} \dot{\xi}_i &= \lambda_i \xi_i, \quad i = 1, 2, \dots, n \\ \Rightarrow \xi_i(t) &= \xi_i(0) \exp(\lambda_i t) \end{aligned}$$

Với điều kiện đầu $\xi(0) = \mathbf{T}^{-1} \mathbf{x}(0)$

Tìm được $\mathbf{x}(t)$

$$\mathbf{x}(t) = \mathbf{T} \xi(t) = \mathbf{v}_1 \xi_1(0) e^{\lambda_1 t} + \mathbf{v}_2 \xi_2(0) e^{\lambda_2 t} + \cdots + \mathbf{v}_n \xi_n(0) e^{\lambda_n t},$$

Như vậy,

hệ ổn định tiệm cận $\Leftrightarrow$ tất cả các trị riêng của ma trận $\mathbf{A}$ nằm bên trái trục ảo

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

11

Hệ tuyến tính: Ổn định tiệm cận và ổn định BIBO

Trong trường hợp không có sự loại trừ điểm không và điểm cực (hay không có điểm không nào trùng với các điểm cực), các điểm cực của hàm truyền chính là các trị riêng của ma trận $\mathbf{A}$. Khi đó, ổn định BIBO là tương đương với ổn định tiệm cận.

Kết luận: Nếu hệ vừa điều khiển được vừa quan sát được, thì

ổn định BIBO tương đương với ổn định tiệm cận.

Conclusion: If the system is both controllable and observable, then

BIBO Stability $\Leftrightarrow$ Asymptotical Stability

Một số phương pháp kiểm tra tính ổn định hệ tuyến tính

- Ổn định tiệm cận [Asymptotically stable]
 - Nếu tất cả các trị riêng của ma trận $\mathbf{A}$ nằm bên trái trục ảo [All the eigenvalues of $\mathbf{A}$ lie in the LHP]
- Ổn định BIBO
 - Tiêu chuẩn Routh-Hurwitz [Routh-Hurwitz criterion]
 - Phương pháp quỹ tích nghiệm [Root locus method]
 - Tiêu chuẩn Nyquist, [Nyquist criterion],

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

12

Ổn định tiệm cận hệ tuyến tính

Nhắc lại một số thuật ngữ

Đối với hệ tuyến tính (SISO)

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{b}u, & \mathbf{x} &\in \mathbb{R}^n, u \in \mathbb{R} \\ y &= \mathbf{c}\mathbf{x}, & y &\in \mathbb{R}\end{aligned}$$

Tính được hàm truyền

$$G(s) = \frac{n(s)}{d(s)} = \mathbf{c}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}$$

Điểm không: là nghiệm của $n(s)=0$

Điểm cực: là nghiệm của $d(s)=0$

Đa thức đặc trưng: $d(s)=0$

Phương trình đặc trưng: $d(s)=0$

$$G(s) = \frac{n(s)}{d(s)} = \frac{(s-1)(s+1)}{(s+2)(s^2+1)}$$

Điểm không của $G(s)$: ± 1

Điểm cực của $G(s)$: $-2, \pm j$

Điều kiện ổn định trong miền phức (s)

Với một hệ được thể hiện bằng hàm truyền $G(s)$,

Hệ là ổn định BIBO



Tất cả các điểm cực của $G(s)$ nằm ở nửa trái của mặt phẳng phức



Hệ là ổn định tiệm cận

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

13

Ổn định của hệ tuyến tính

Ý tưởng của điều kiện ổn định

Xét ví dụ: $\dot{y} + \alpha y = u(t), \quad y(0) = y_0$

$$\Rightarrow sY(s) - y(0) + \alpha Y(s) = U(s) \Rightarrow Y(s) = \frac{1}{s + \alpha} (U(s) + y(0))$$

Xét ổn định tiệm cận
 $u(t)=0$ hay $U(s)=0$

$$\Rightarrow Y(s) = \frac{1}{s + \alpha} y(0)$$

$$\Rightarrow y(t) = \mathcal{L}^{-1}[Y(s)] = y(0)e^{-\alpha t} \rightarrow 0 \Leftrightarrow \text{Re}(\alpha) > 0$$

Xét ổn định BIBO
 $y(0)=0$

$$\Rightarrow Y(s) = \frac{1}{s + \alpha} U(s) = G(s)U(s)$$

$$\Rightarrow y(t) = \mathcal{L}^{-1}[Y(s)] = \mathcal{L}^{-1}[G(s)U(s)]$$

$$= \int_0^t g(\tau)u(t-\tau)d\tau = \int_0^t e^{-\alpha\tau}u(t-\tau)d\tau$$

$$|y(t)| \leq \int_0^t |e^{-\alpha\tau}| \cdot |u(t-\tau)| d\tau \leq \int_0^t |e^{-\alpha\tau}| d\tau \cdot u_{\max}$$

Bị chặn nếu $\text{Re}(\alpha) > 0$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

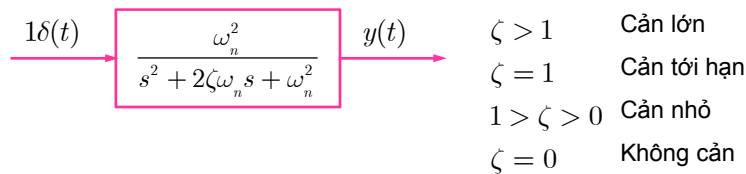
14

Ổn định tiệm cận hệ tuyến tính

Xét hệ bậc hai với kích động xung và chạm

$$\ddot{y} + 2\zeta\omega_n \dot{y} + \omega_n^2 y = \omega_n^2 \delta(t), \quad y(0) = 0, \quad \dot{y}(0) = 0$$

$$\Rightarrow s^2 Y(s) + 2\zeta\omega_n s Y(s) + \omega_n^2 Y(s) = U(s) \Rightarrow Y(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} U(s)$$



$$1 > \zeta > 0$$

$$p_{1,2} = -\zeta\omega_n \pm j\omega_n \sqrt{1 - \zeta^2}$$

$$h(t) = \frac{\omega_n}{\sqrt{1 - \zeta^2}} e^{-\zeta\omega_n t} \sin\left(\sqrt{1 - \zeta^2} \omega_n t\right)$$

$$\zeta = 0$$

$$p_{1,2} = \pm j\omega_n$$

$$h(t) = C_1 \sin(\omega_n t) + C \cos(\omega_n t)$$

HUST

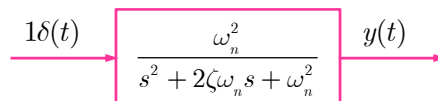
Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

15

Ổn định tiệm cận hệ tuyến tính

Xét hệ bậc hai với kích động xung và chạm



Cản nhỏ

$$1 > \zeta > 0$$

$$p_{1,2} = -\zeta\omega_n \pm j\omega_n \sqrt{1 - \zeta^2}$$

$$h(t) = \frac{\omega_n}{\sqrt{1 - \zeta^2}} e^{-\zeta\omega_n t} \sin\left(\sqrt{1 - \zeta^2} \omega_n t\right)$$

Không cản

$$\zeta = 0$$

$$p_{1,2} = \pm j\omega_n$$

$$h(t) = C_1 \sin(\omega_n t) + C \cos(\omega_n t)$$

$$\lim_{t \rightarrow \infty} h(t) = \lim_{t \rightarrow \infty} C_1 e^{-\omega_n t} \sin(\omega_n t + \alpha) = 0$$

Ổn định tiệm cận

$$\lim_{t \rightarrow \infty} h(t) = \lim_{t \rightarrow \infty} C \sin(\omega_n t + \alpha) = ???$$

$$|h(t)| \leq C$$

Ổn định biên

HUST

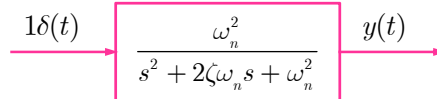
Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

16

Ổn định tiệm cận hệ tuyến tính

Xét hệ bậc hai với kích động xung và chạm



Cần tới hạn

$$\zeta = 1$$

$$p_{1,2} = -\omega_n = \lambda_{1,2}$$

$$h(t) = e^{-\omega_n t} (C_1 + tC_2)$$

$$\lim_{t \rightarrow \infty} h(t) = \lim_{t \rightarrow \infty} e^{-\omega_n t} (C_1 + tC_2) = 0$$

Cần lớn

$$\xi > 1$$

$$p_{1,2} = \omega_n \left(-\zeta \pm \sqrt{\zeta^2 - 1} \right) < 0$$

$$h(t) = C_1 e^{p_1 t} + C_2 e^{p_2 t}$$

$$\lim_{t \rightarrow \infty} h(t) = \lim_{t \rightarrow \infty} (C_1 e^{p_1 t} + C_2 e^{p_2 t}) = 0$$

Ổn định tiệm cận

Ổn định tiệm cận

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

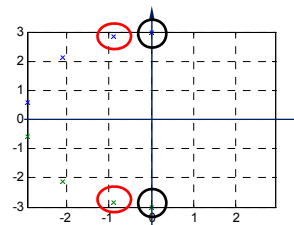
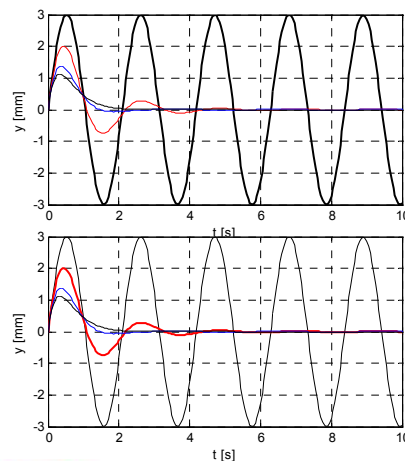
17

Ổn định tiệm cận hệ tuyến tính

$$h(t) = \frac{\omega_n}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin(\sqrt{1-\zeta^2}\omega_n t)$$

$$\omega_n = 3;$$

$$\zeta = [0, 0.3, 0.7, 0.98];$$



Xem: [he_bac_2.m](#)

HUST

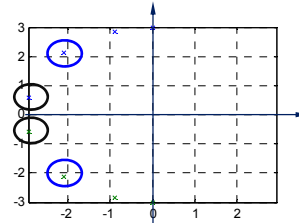
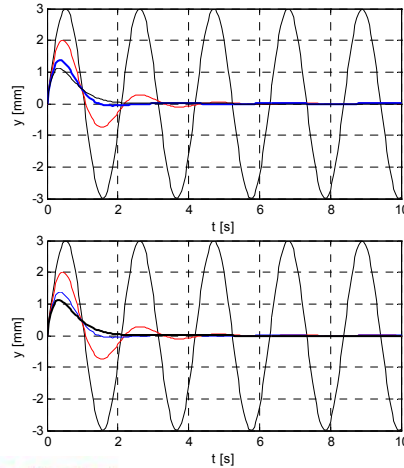
Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

18

Ổn định của hệ tuyến tính

$$h(t) = \frac{\omega_n}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin(\sqrt{1-\zeta^2}\omega_n t) \quad \omega_n = 3; \quad \zeta = [0, 0.3, 0.7, 0.98];$$



HUST

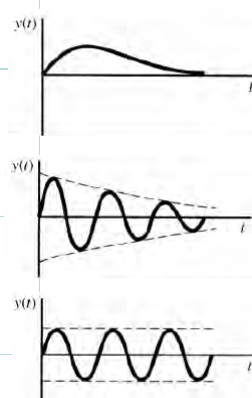
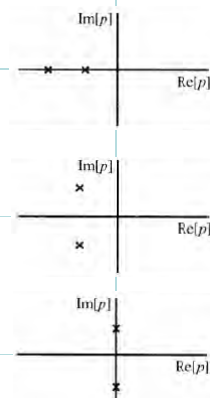
Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

19

Ổn định của hệ tuyến tính

$$1\delta(t) \rightarrow \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} y(t) \rightarrow p_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$$



ổn định tiệm cận

ổn định tiệm cận

ổn định biên

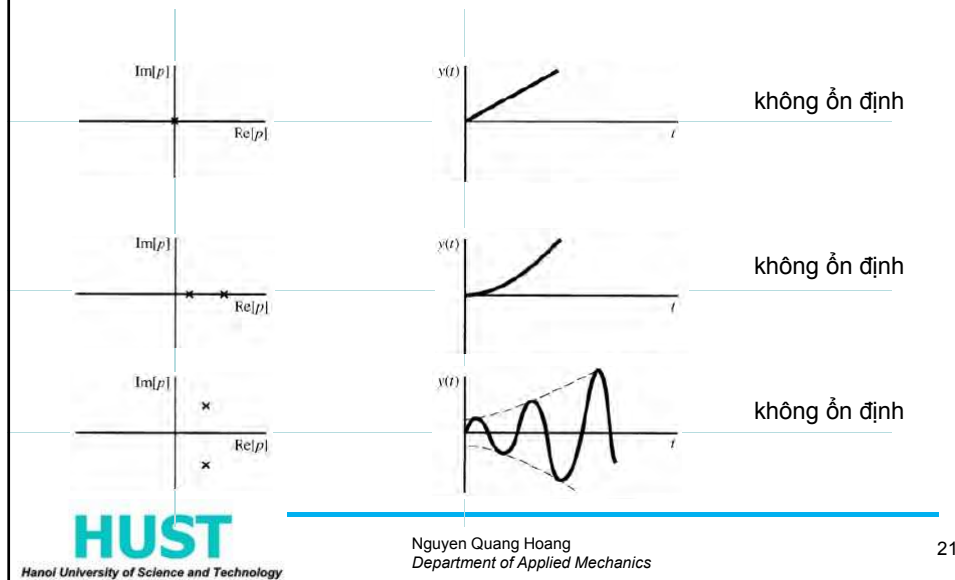
HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

20

Ổn định của hệ tuyến tính



Ổn định của hệ tuyến tính

Một số lưu ý

- ❑ Đối với một hệ tổng quát (phi tuyến), điều kiện **ổn định BIBO** và điều kiện **ổn định tiệm cận** là khác nhau.
- ❑ Đối với hệ thống **tuyến tính bất biến thời gian (LTI)** (mà chúng ta có thể sử dụng biến đổi Laplace và nhận được hàm truyền), các **điều kiện ổn định** như nhau.
- ❑ Trong khóa học này, chúng ta chỉ quan tâm hệ thống LTI, khái niệm "**ổn định**" sử dụng cho cả hai: ổn định BIBO và ổn định tiệm cận.
- ❑ **Ổn định biên nếu (hệ dao động):**
 - o $G(s)$ không có điểm cực nằm trong nửa trái mặt phẳng phức, và
 - o $G(s)$ có một điểm cực đơn nằm trên trục ảo, và
 - o $G(s)$ không có điểm cực bội (nghiệm bội) nằm trên trục ảo.

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

Ổn định biên

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

Không ổn định biên

- ❑ **Unstable** if a system is neither stable nor marginally stable.

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

22

Ổn định tiệm cận hệ tuyến tính

□ Hệ có điểm cực kép (nghiệm kép)

$$\begin{aligned} \mathcal{L}^{-1} \left[\frac{2\omega s}{(s^2 + \omega^2)^2} \right] &= t \sin \omega t & \mathcal{L}^{-1} \left[\frac{s^2 - \omega^2}{(s^2 + \omega^2)^2} \right] &= t \cos \omega t \\ \mathcal{L}^{-1} \left[\frac{\dots}{(s^2 + \omega^2)^3} \right] &= t^2 \sin \omega t & \mathcal{L}^{-1} \left[\frac{\dots}{(s^2 + \omega^2)^2} \right] &= t^2 \cos \omega t \end{aligned}$$

Ổn định biên có dẫn đến ổn định BIBO ?

❖ Xét ví dụ $G(s) = \frac{2s}{(s^2 + 1)}$

❖ Đầu ra của hệ

$$\begin{aligned} Y(s) &= G(s)U(s) = \frac{2s}{(s^2 + 1)} \frac{1}{(s^2 + 1)} \\ \Rightarrow y(t) &= \mathcal{L}^{-1} \left[\frac{2s}{(s^2 + 1)^2} \right] = t \sin t \end{aligned}$$

❖ Với đầu vào tác động

$$u(t) = \sin t \Rightarrow U(s) = \frac{1}{(s^2 + 1)}$$

Thực chất đây là trường hợp hệ ổn định biên (hệ dao động), nhưng do kích động điều hòa có tần số kích động trùng tần số dao động riêng. Xây ra hiện tượng công hưởng.

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

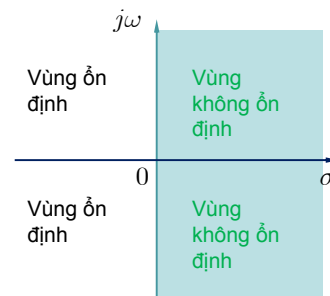
23

Ổn định của hệ tuyến tính

Tóm lại

Với s_i là điểm cực của hàm truyền $G(s)$, hệ $G(s)$ là:

- **ổn định BIBO, ổn định tiệm cận** nếu $\text{Re}(s_i) < 0$, với mọi i .
- **ổn định biên** nếu
 - o $\text{Re}(s_i) \leq 0$ với mọi i , và
 - o có một nghiệm đơn $\text{Re}(s_i) = 0$
- **không ổn định** nếu hệ không nằm trong hai trường hợp trên (ổn định và ổn định biên).



HUST

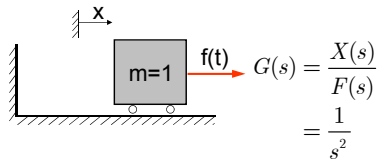
Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

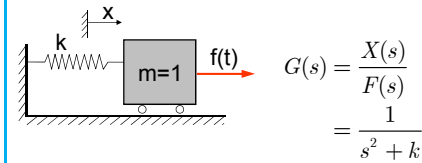
24

Ổn định của hệ tuyến tính

Xét ổn định của hệ bậc 2 (hệ cơ)

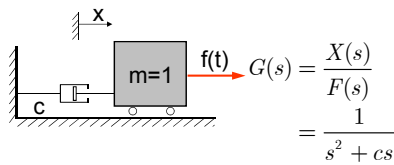


Các điểm cực =
Ổn định?

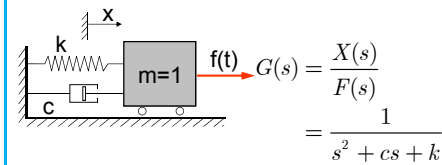


Các điểm cực =
Ổn định?

Ngoại trừ $F(s) = \frac{\sqrt{k}}{s^2 + k}$, $F(s) = \frac{s}{s^2 + k}$



Các điểm cực =
Ổn định?



Các điểm cực =
Ổn định?

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

25

Ổn định của hệ tuyến tính

Tóm lại

- ❑ Ổn định của hệ LTI
 - o ổn định (BIBO và tiệm cận), ổn định biên, và không ổn định.
 - o Ổn định của hệ có hàm truyền $G(s)$ xác định bởi các điểm cực của $G(s)$.
- ❑ Tiêu chuẩn ổn định Routh-Hurwitz xác định tính chất ổn định mà không cần tính các điểm cực của hệ.

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

26

Ổn định của hệ tuyến tính

Bài tập

$G(s)$	Các điểm không z và điểm cực p	Ổn định/ổn định biên/không ổn định
$\frac{5(s+2)}{(s+1)(s^2+s+1)}$		
$\frac{5(-s+2)}{(s+1)(s^2+s+1)}$		
$\frac{5}{(s-2)(s^2+3)}$		
$\frac{(s^2+3)}{(s+1)(s^2-s+1)}$		
$\frac{5(-s+2)}{(s+1)(s^2+1)^2}$		
$\frac{1}{(s^2-1)(s+1)}$		

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

27

Phương pháp Lyapunov

Ổn định của hệ tuyến tính (LTI): Phương pháp Lyapunov

Tiêu chuẩn ổn định Lyapunov xét tính ổn định của hệ trực tiếp trong không gian trạng thái, thích hợp đối với các hệ được mô tả bởi mô hình trạng thái.

phương pháp này áp dụng được cho cả hệ tuyến tính và phi tuyến.

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

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

28

Phương pháp Lyapunov

Định lý: Hệ động lực

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}$$

Ổn định BIBO khi và chỉ khi nó **ổn định tiệm cận Lyapunov**, tức là khi và chỉ khi các quỹ đạo trạng thái tự do ($\mathbf{u} = 0$) có hướng tiến về gốc tọa độ và kết thúc tại đó.

Chứng minh: Hệ động lực trên ổn định BIBO, tức là tất cả các trị riêng của ma trận hệ thống $\mathbf{A}$ có phần thực âm.

Quỹ đạo trạng thái tự do ($\mathbf{u} = 0$), xác định bởi

Do tất cả các trị riêng của ma trận hệ thống $\mathbf{A}$ có phần thực âm, nên

$$\lim_{t \rightarrow \infty} \mathbf{x}(t) = \lim_{t \rightarrow \infty} e^{\mathbf{A}t} \mathbf{x}(0) = \mathbf{0}$$

Do đó, để kiểm tra tính ổn định tiệm cận Lyapunov (cũng là tính ổn định BIBO), ta chỉ cần kiểm tra xem quỹ đạo trạng thái của hệ trong quá trình tự do có hướng tiến về gốc tọa độ và kết thúc tại đó không.

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

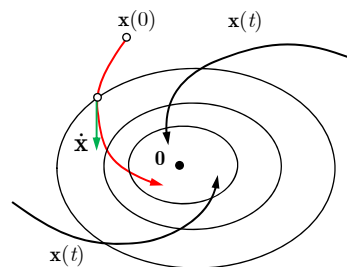
29

Phương pháp Lyapunov

Tư tưởng của phương pháp Lyapunov

Dựa trên phương diện năng lượng: Năng lượng còn tồn tại bên trong hệ do tác động tức thời bên ngoài đưa vào được mô bởi một hàm không âm. Hệ sẽ ổn định (tiệm cận) ở trạng thái cân bằng của nó nếu trong lân cận điểm cân bằng đó hàm năng lượng này của hệ luôn có xu hướng giảm dần về 0.

Giả sử bao quanh điểm cân bằng ($\mathbf{x} = \mathbf{0}$) có họ các đường cong khép kín γ . Các đường cong này được xem như các biên lân cận điểm gốc 0. Để xem quỹ đạo trạng thái $\mathbf{x}(t)$ tự do ($\mathbf{u} = 0$) có tiến về gốc tọa độ hay không, ta chỉ cần xem quỹ đạo $\mathbf{x}(t)$ có cắt tất cả các đường cong thuộc họ γ từ bên ngoài vào trong hay không và nếu điều đó xảy ra thì chắc chắn $\mathbf{x}(t)$ phải có hướng tiến về gốc tọa độ và kết thúc tại đó.



HUST

Hanoi University of Science and Technology

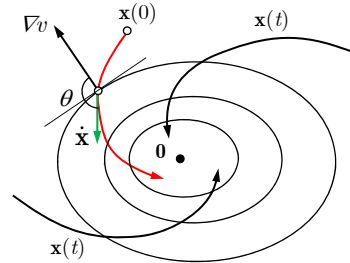
Nguyen Quang Hoang
Department of Applied Mechanics

30

Phương pháp Lyapunov

Phương pháp Lyapunov gồm hai bước:

1. Xây dựng họ các đường cong v khép kín chứa điểm cân bằng (gốc tọa độ) bên trong.
2. Kiểm tra xem quỹ đạo trạng thái $\mathbf{x}(t)$ tự do ($\mathbf{u}=0$) có cắt tất cả các đường cong thuộc họ v từ bên ngoài vào trong hay không. Hiển nhiên, để $\mathbf{x}(t)$ cắt đường cong thuộc họ v từ ngoài vào trong là tại điểm cắt đó, tiếp tuyến với quỹ đạo tại điểm cắt đó phải tạo với véc tơ ∇v vuông góc với đường cong theo hướng từ trong ra ngoài một góc lớn hơn 90° .



Định lý Lyapunov (về ổn định tiệm cận): Nếu tồn tại hàm $V(\mathbf{x})$ thỏa mãn điều kiện:

□ Khả vi, xác định dương, tức $V(\mathbf{x}) > 0 \quad \forall \mathbf{x} \neq \mathbf{0} \quad \& \quad V(\mathbf{x}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{0}$

□ và $\frac{dV(\mathbf{x})}{dt} = \frac{\partial V}{\partial \mathbf{x}} \frac{d\mathbf{x}}{dt} < 0$ (đạo hàm $V(\mathbf{x})$ dọc theo quỹ đạo trạng thái tự do âm).

Thì hệ là ổn định tiệm cận Lyapunov tại gốc ($\mathbf{x} = \mathbf{0}$) hay ổn định BIBO.

Hàm $V(\mathbf{x})$ khi đó được gọi là hàm Lyapunov.

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

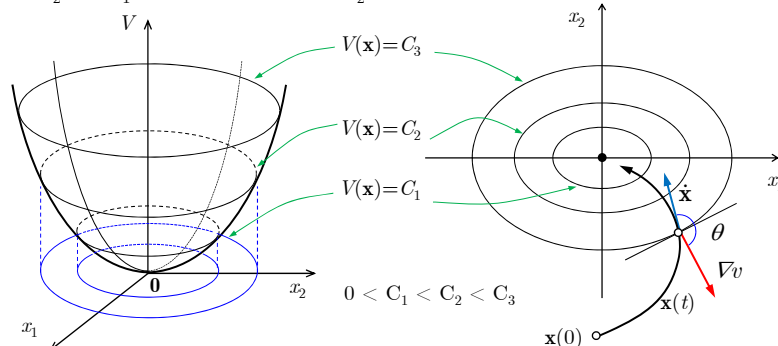
31

Phương pháp Lyapunov

Chứng minh Định lý Lyapunov (về ổn định tiệm cận):

Hàm khả vi xác định dương $V(\mathbf{x})$ có tính chất: khi ta cắt nó bằng một mặt phẳng $V = C_k$ song song với đáy (không gian trạng thái) và chiếu xuống đáy thì ta sẽ được một đường cong kín chứa gốc tọa độ. Đường cong v_k ứng với C_k nhỏ hơn thì nằm bên trong đường cong ứng với C_k lớn hơn.

$C_1 < C_2 \Leftrightarrow v_1$ nằm bên trong v_2



HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

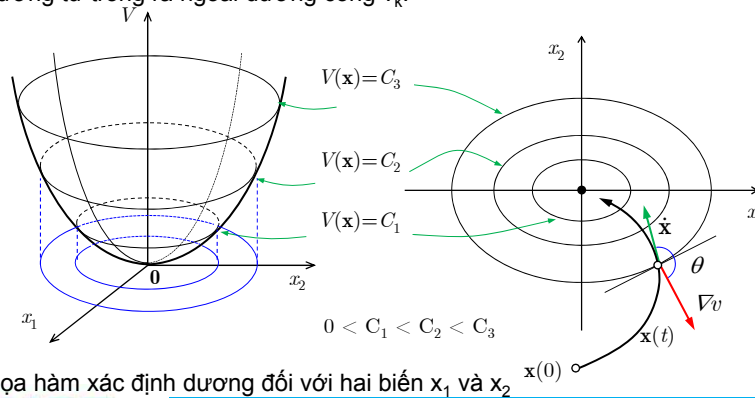
32

Phương pháp Lyapunov

Véc tơ vuông góc với tiếp tuyến của đường cong v_k và hướng theo chiều tăng C_k là

$$\nabla_v = \text{grad } V = \left[\frac{\partial V}{\partial \mathbf{x}} \right]^T = \left[\frac{\partial V}{\partial x_1}, \frac{\partial V}{\partial x_2}, \dots, \frac{\partial V}{\partial x_n} \right]^T$$

Sẽ hướng từ trong ra ngoài đường cong v_k .



Minh họa hàm xác định dương đối với hai biến x_1 và x_2

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

33

Phương pháp Lyapunov

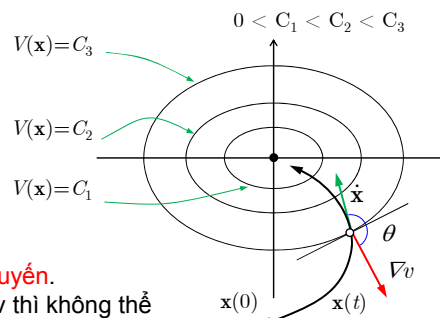
Do có

$$\begin{aligned} \frac{dV}{dt} &= \frac{\partial V}{\partial \mathbf{x}} \frac{d\mathbf{x}}{dt} = \left[\frac{\partial V}{\partial x_1}, \frac{\partial V}{\partial x_2}, \dots, \frac{\partial V}{\partial x_n} \right]^T \frac{d\mathbf{x}}{dt} \\ &= [\text{grad } V]^T \frac{d\mathbf{x}}{dt} = |\text{grad } V| \left| \frac{d\mathbf{x}}{dt} \right| \cos \theta \end{aligned}$$

mà $\dot{\mathbf{x}} = d\mathbf{x} / dt$ chính là tiếp tuyến của quỹ đạo trạng thái $\mathbf{x}(t)$, nên với điều kiện $dV/dt < 0$, góc θ tạo bởi $\text{grad } V$ và $d\mathbf{x}/dt$ phải là góc tù (lớn hơn 90°), tức là quỹ đạo trạng thái $\mathbf{x}(t)$ sẽ cắt các đường cong v_k theo hướng từ ngoài vào trong.

Lưu ý:

- Định lý này áp dụng cho cả **hệ phi tuyến**.
- Nếu không tìm được hàm Lyapunov thì không thể khẳng định về tính ổn định của hệ. Định lý cho ta **điều kiện đủ** để xác định tính ổn định của hệ.



HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

34

Phương pháp Lyapunov

Ví dụ minh họa. Xét hệ động lực mô tả bởi

$$\frac{d\mathbf{x}}{dt} = \mathbf{Ax} + \mathbf{Bu} = \begin{bmatrix} -4 & 2 & 0 \\ -2 & -5 & 1 \\ 0 & -1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

Chọn hàm khả vi, xác định dương

$$V(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 > 0 \quad \forall \mathbf{x} \neq 0$$

Đạo hàm $V(\mathbf{x})$ dọc quỹ đạo $\mathbf{x}(t)$ của quá trình tự do ($\mathbf{u} = 0$), ta được

$$\begin{aligned} \frac{dV}{dt} &= \frac{\partial V}{\partial \mathbf{x}} \dot{\mathbf{x}}_{u=0} \\ &= [2x_1 \quad 2x_2 \quad 2x_3] \mathbf{Ax} \\ &= -8x_1^2 - 10x_2^2 - 4x_3^2 < 0 \quad \forall \mathbf{x} \neq 0 \end{aligned}$$

Theo định lý Lyapunov về ổn định tiệm cận, ta kết luận hệ ổn định.

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

35

Ổn định của hệ tuyến tính (LTI) : PP Lyapunov

Khi áp dụng phương pháp trực tiếp Lyapunov đối với hệ tuyến tính $\dot{\mathbf{x}} = \mathbf{Ax}$

Người ta hay sử dụng hàm trơn xác định dương dạng $\det \mathbf{A} \neq 0$

$$V(\mathbf{x}) = \mathbf{x}^T \mathbf{Px}, \quad \mathbf{P} = \mathbf{P}^T > 0 \quad \mathbf{P} \text{ là ma trận đx xác định dương}$$

Đạo hàm $V(\mathbf{x})$ theo thời gian dọc quỹ đạo ta có

$$\begin{aligned} \dot{V}(\mathbf{x}) &= \dot{\mathbf{x}}^T \mathbf{Px} + \mathbf{x}^T \mathbf{P} \dot{\mathbf{x}} = (\mathbf{Ax})^T \mathbf{Px} + \mathbf{x}^T \mathbf{P} \mathbf{Ax} \\ &= \mathbf{x}^T \mathbf{A}^T \mathbf{Px} + \mathbf{x}^T \mathbf{PAx} = \mathbf{x}^T (\mathbf{A}^T \mathbf{P} + \mathbf{PA}) \mathbf{x} = -\mathbf{x}^T \mathbf{Qx} \end{aligned}$$

Như thế, nếu tồn tại ma trận $\mathbf{Q}$ đối xứng x.đ.d. thỏa mãn

$$(\mathbf{A}^T \mathbf{P} + \mathbf{PA}) = -\mathbf{Q}$$

thì V là hàm Lyapunov và điểm cân bằng $\mathbf{x} = 0$ là ổn định toàn cục. Phương trình trên được gọi là phương trình ma trận Lyapunov.

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

36

Ổn định của hệ tuyến tính

Xét hệ động lực tuyến tính mô tả bởi phương trình

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}, \quad \mathbf{x}(0) = \mathbf{x}_0 \quad \det \mathbf{A} \neq 0$$

Hệ trên là ổn định tiệm cận khi và chỉ khi đối với mỗi ma trận $\mathbf{Q}$ thực đối xứng xác định dương nghiệm $\mathbf{P}$ của phương trình ma trận Lyapunov sau

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} = -\mathbf{Q}$$

là (thực đối xứng) xác định dương.

Chứng minh (điều kiện cần)

1. Nếu hệ ổn định tiệm cận thì với mỗi $\mathbf{Q}$ thực đối xứng x.đ.d tồn tại $\mathbf{P}$ thực đối xứng xác định dương thỏa mãn pt ma trận Lyapunov.

Sử dụng phương pháp phản chứng: giả sử rằng với mọi $\mathbf{x} \neq \mathbf{0}$, ta có

$$\mathbf{x}^T \mathbf{P} \mathbf{x} > 0, \quad \& \quad -\mathbf{x}^T \mathbf{Q} \mathbf{x} < 0 \quad \forall \mathbf{x} \neq \mathbf{0}$$

Mà hệ không ổn định tiệm cận hay ma trận hệ $\mathbf{A}$ không ổn định tiệm cận. Điều này có nghĩa là với điều kiện đầu $\mathbf{x}(0) \neq \mathbf{0}$, nghiệm

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)} \mathbf{x}(0) \quad \text{giới nội và không hội tụ về gốc tọa độ.}$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

37

Ổn định của hệ tuyến tính

Chứng minh (điều kiện đủ)

2. Nếu tồn tại $\mathbf{Q}$ thực đối xứng xác định dương & $\mathbf{P}$ x.đ.d. thỏa mãn phương trình ma trận Lyapunov thì hệ là ổn định tiệm cận. Thực vậy, ta chọn hàm Lyapunov và tính đạo hàm của nó

$$V(\mathbf{x}) = \mathbf{x}^T \mathbf{P} \mathbf{x}, \quad p.d.$$

$$\dot{V}(\mathbf{x}) = \mathbf{x}^T \mathbf{P} \dot{\mathbf{x}} + \dot{\mathbf{x}}^T \mathbf{P} \mathbf{x} = \mathbf{x}^T \mathbf{P} \mathbf{A} \mathbf{x} + \mathbf{x}^T \mathbf{A}^T \mathbf{P} \mathbf{x}$$

$$= \mathbf{x}^T (\mathbf{P} \mathbf{A} + \mathbf{A}^T \mathbf{P}) \mathbf{x} = -\mathbf{x}^T \mathbf{Q} \mathbf{x} \quad n.d.$$

Kết luận: hệ
ổn định tiệm
cận

Định lý về sự tồn tại $\mathbf{P}$ thỏa mãn phương trình ma trận Lyapunov. Nếu ma trận $\mathbf{A}$ là Hurwitz (ma trận bền) thì với mọi $\mathbf{Q}$ đối xứng x.đ.d sẽ tồn tại $\mathbf{P}$ đối xứng x.đ.d thỏa mãn phương trình ma trận Lyapunov.

Định nghĩa: Ma trận $\mathbf{A}$ được gọi là Hurwitz nếu các trị riêng của nó có phần thực âm, $\text{Re}(\lambda_i) < 0, i = 1, 2, \dots, n$.

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

38

Ổn định của hệ tuyến tính

Chứng minh (We make a constructive proof)

Với ma trận $\mathbf{Q}$ thực đối xứng x.đ.d, ta xét nghiệm $\mathbf{P}$

$$\mathbf{P} = \int_0^{\infty} e^{\mathbf{A}^T t} \mathbf{Q} e^{\mathbf{A} t} dt \quad \text{Từ } \mathbf{Q} = \mathbf{Q}^T > 0 \Rightarrow \mathbf{P} = \mathbf{P}^T > 0$$

Chú ý rằng, tích phân trên chỉ hội tụ khi ma trận $\mathbf{A}$ là Hurwitz (các trị riêng của $\mathbf{A}$ có phần thực âm).

Ta sẽ chỉ ra, ma trận $\mathbf{P}$ thỏa mãn phương trình ma trận Lyapunov

$$(\mathbf{P}\mathbf{A} + \mathbf{A}^T \mathbf{P}) = -\mathbf{Q}$$

Ta có

$$\begin{aligned} \mathbf{P}\mathbf{A} + \mathbf{A}^T \mathbf{P} &= \int_0^{\infty} [\mathbf{A}^T e^{\mathbf{A}^T t} \mathbf{Q} e^{\mathbf{A} t} + e^{\mathbf{A}^T t} \mathbf{Q} e^{\mathbf{A} t} \mathbf{A}] dt \\ &= \int_0^{\infty} \frac{d}{dt} [e^{\mathbf{A}^T t} \mathbf{Q} e^{\mathbf{A} t}] dt = e^{\mathbf{A}^T t} \mathbf{Q} e^{\mathbf{A} t} \Big|_0^{\infty} = -\mathbf{Q} \end{aligned}$$

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

39

Ổn định của hệ tuyến tính (LTI): PP Lyapunov

Kết luận về tính ổn định tiệm cận từ trị riêng của ma trận $\mathbf{A}$

Xét điểm cân bằng $\mathbf{x} = 0$ của hệ LTI với ma trận $\mathbf{A}$ chéo hóa được bằng ma trận $\mathbf{Q}$

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} \quad \mathbf{x} = \mathbf{Q}\mathbf{y}, \quad \dot{\mathbf{x}} = \mathbf{Q}\dot{\mathbf{y}}$$

$$\dot{\mathbf{x}} = \mathbf{Q}\dot{\mathbf{y}} = \mathbf{A}\mathbf{Q}\mathbf{y} \Rightarrow \dot{\mathbf{y}} = \mathbf{Q}^{-1}\mathbf{A}\mathbf{Q}\mathbf{y}, \quad \mathbf{y}(0) = \mathbf{Q}^{-1}\mathbf{x}(0)$$

Với $\mathbf{A}$ chéo hóa được, tức tồn tại ma trận $\mathbf{Q}$:

$$\mathbf{\Lambda} = \mathbf{Q}^{-1}\mathbf{A}\mathbf{Q} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$$

$$\dot{y}_i = \lambda_i y_i, \quad y_i(0) = y_{i,0}, \quad i = 1, 2, \dots, n$$

$$y_i(t) = e^{\lambda_i t} y_i(0), \quad i = 1, 2, \dots, n$$

$$\mathbf{y}(t) = \text{diag}(e^{\lambda_1 t}, e^{\lambda_2 t}, \dots, e^{\lambda_n t}) \mathbf{y}(0)$$

$$\mathbf{x}(t) = \mathbf{Q}\mathbf{y}(t) = \mathbf{Q} e^{\mathbf{\Lambda} t} \mathbf{Q}^{-1} \mathbf{x}(0)$$

$$e^{\mathbf{\Lambda} t} = \text{diag}(e^{\lambda_1 t}, e^{\lambda_2 t}, \dots, e^{\lambda_n t})$$

Như thế, hệ tuyến tính với $\mathbf{A}$ chéo hóa được là ổn định tiệm cận nếu

$$\text{Re}(\lambda_i) < 0, \quad \forall i = 1, 2, \dots, n$$

Nếu có $\text{Re}(\lambda_i) = 0$, thì hệ không ổn định tiệm cận, nhưng ổn định biên.

HUST

Hanoi University of Science and Technology

Nguyen Quang Hoang
Department of Applied Mechanics

40