MATLAB
for process control

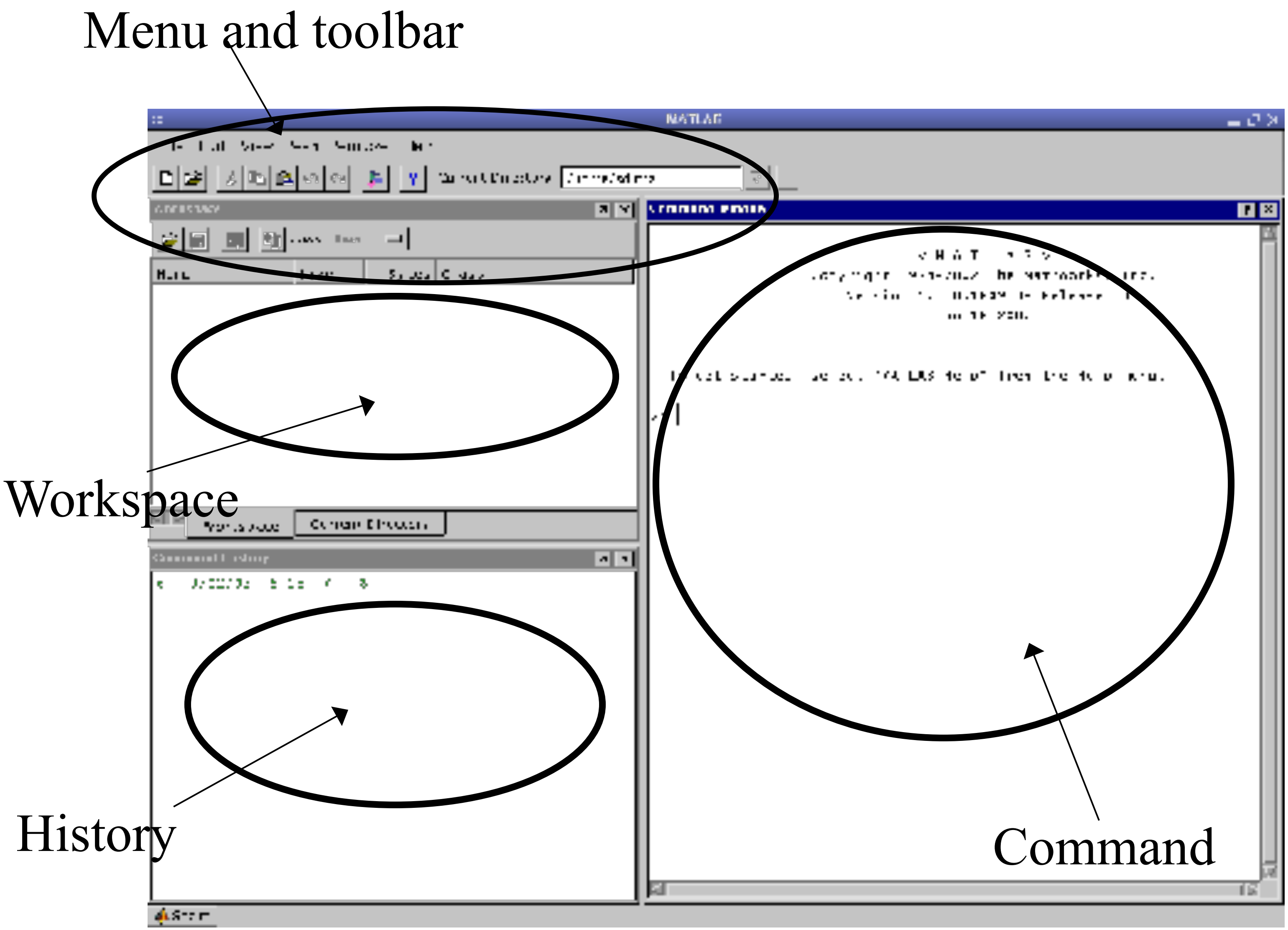
Min Huang, PhD

CheEng@TongjiU

What is MATLAB

- High level language for technical computing
- Stands for **MAT**rix **LAB**oratory
- Everything is a matrix - easy to do linear algebra

MATLAB Desktop



Matrices & Vectors

- All (almost) entities in MATLAB are matrices
- Easy to define:

```
>> A = [16 3; 5 10]
  A =     16      3
           5     10
```

- Use ‘,’ or ‘ ’ to separate row elements -- use ‘;’ to separate rows

Creating Vectors and Matrices

- Define

```
>> A = [16 3; 5 10]
  A =     16      3
           5     10
>> B = [3 4 5
  6 7 8]
  B = 3   4   5
      6   7   8
```

- Transpose

Vector :

```
>> a=[1 2 3];
>> a'
     1
     2
     3
```

Matrix:

```
>> A=[1 2; 3 4];
>> A'
ans =
     1     3
     2     4
```

Creating Matrices

- `zeros(m, n)`: matrix with all zeros
- `ones(m, n)`: matrix with all ones.
- `eye(m, n)`: the identity matrix
- `rand(m, n)`: uniformly distributed random
- `randn(m, n)`: normally distributed random
- `magic(m)`: square matrix whose elements have the same sum, along the row, column and diagonal.
- `pascal(m)` : Pascal matrix.

Matrix operations

- `^`: exponentiation
- `*`: multiplication
- `/`: division
- `\`: left division. The operation `A\B` is effectively the same as `INV(A)*B`, although left division is calculated differently and is much quicker.
- `+`: addition
- `-`: subtraction

Indexing Matrices

Given the matrix:

$$
\begin{array}{cc}
 & n \\
m & \begin{array}{ccc} 0.9501 & 0.6068 & 0.4231 \\ 0.2311 & 0.4860 & 0.2774 \end{array}
\end{array}
$$

```
A =
    0.9501    0.6068    0.4231
    0.2311    0.4860    0.2774
```

Then:

```
A(1,2) = 0.6068

A(3) = 0.6068

A(:,1) = [0.9501
          0.2311 ]
```

1:m

```
A(1,2:3)=[0.6068   0.4231]
```

Workspace

- Matlab remembers old commands
- **And** variables as well
- Each Function maintains its own scope
- The keyword `clear` removes all variables from workspace
- The keyword `who` lists the variables

MatLab for Control

Obtaining the Partial Fraction Expansion

- Example

$$Y(s) = \frac{4s^2 + 24s + 12}{s^3 + 5s^2 + 6s}$$

- num=[4 24 12] , den =[1 5 6 0]
- **[r,p,k]=residue (num, den)** provides
- r = 2.00 p = 0 k=0
- = -8.0 = -3.0
- = 10.0 = -2.0

residue ploes direct term

- Therefore, the partial fraction expansion of the transfer function Y(s) is given,

$$Y(s)=\frac{2}{s}-\frac{8}{s+3}+\frac{10}{s+2}+0 \ \text{(direct term)}$$

- Therefore the inverse Laplace transform is,

$$y(t)=(2-8e^{-3t}+10e^{-2t})$$

- Command [num,den]=residue(r,p,k)
  - Num= 4.0 24.0 12.0
  - Den= 1.0 5.0 6.0 0.0

$$Y(s)=\frac{4s^2+24s+12}{s^3+5s^2+6s}$$

Array Operations

- Evaluated element by element (same size)
  - `.'` : array transpose (non-conjugated transpose)
  - `.^` : array power
  - `.*` : array multiplication
  - `./` : array division
- Very different from Matrix operations

```
>> A=[1 2;3 4];
>> B=[5 6;7 8];
>> A*B
    19    22
    43    50
```

But:

```
>> A.*B
     5    12
    21    32
```

Some Built-in functions

- `mean(A)`: mean value of a vector
- `max(A), min (A)`: maximum and minimum.
- `sum(A)`: summation.
- `sort(A)`: sorted vector
- `median(A)`: median value
- `std(A)`: standard deviation.
- `det(A)` : determinant of a square matrix
- `dot(a,b)`: dot product of two vectors
- `Cross(a,b)`: cross product of two vectors
- `Inv(A)`: Inverse of a matrix A

Transformation between the state-space form and the transfer function form

Example

$$\frac{C(s)}{U(s)} = \frac{s^2 + 4s + 1}{s^3 + 9s^2 + 8s}$$

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

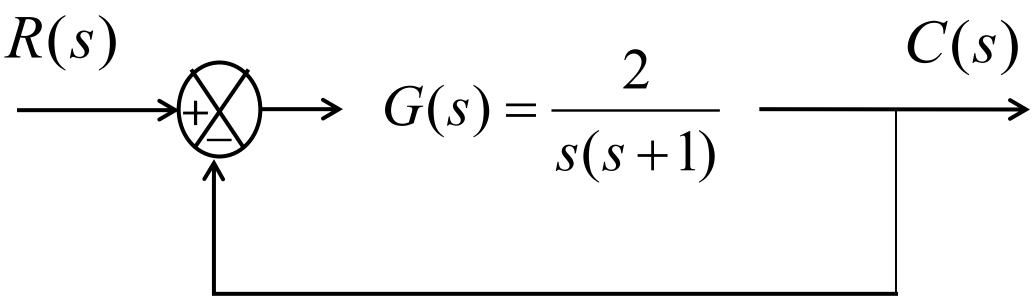
[num,den]=ss2tf(A,B,C,D,iu)

Example

```
A=[-9  -8  0;1  0  0;0  1  0]
B=[1;0;0]
C=[1  4  1]
D=[0]
[num,den]=ss2tf(A,B,C,D)
num = 0  1.0  4.0  1.0
den =   1   9    8    0
```

Obtaining transient response of a system

Example



$$\frac{C(s)}{R(s)} = \frac{G}{1+GH} = \frac{\frac{2}{s(s+1)}}{1+\frac{2}{s(s+1)}} = \frac{2}{s^2+s+2}$$

num=[2],   den=[1 1 2],   **step(num, den)**

Example

```
A=[0  1;-2  -1]
B=[0;2]
C=[1  0]
D=[0 ]
step(A,B,C,D)
grid
title(    )
```