

## Digital Signal Processing

Prof. Nizamettin AYDIN

[naydin@yildiz.edu.tr](mailto:naydin@yildiz.edu.tr)

[naydin@ieee.org](mailto:naydin@ieee.org)

<http://www.yildiz.edu.tr/~naydin>

1

## Course Details

- Course Code : 0113620
- Course Name: Digital Signal Processing  
(Sayısal İşaret İşleme)
- Instructor : Nizamettin AYDIN

2

## Digital Signal Processing (DSP)

### MATLAB Tutorial

3

## Topics

- Introduction
- Running MATLAB and MATLAB Environment
- Getting help
- Variables
- Vectors, Matrices, and linear Algebra
- Mathematical Functions and Applications
- Plotting
- Programming
- M-files
- User Defined Functions

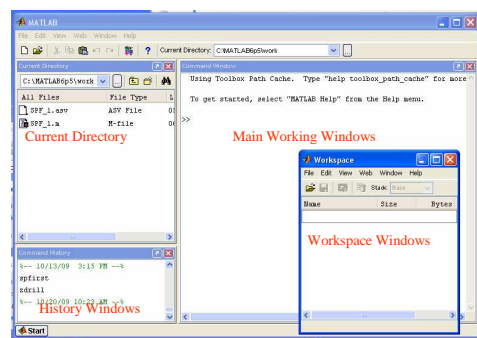
4

## Introduction

- What is **MATLAB**:
  - **MATLAB** stands for **MATrix LABoratory**
  - A tool for **Computation** and **Visualization** in a wide range on science application
- **MATLAB** abilities:
  - Variable management
  - Data import and export
  - Calculations (Based on Matrix)
  - Generation of Plots and Graphs

5

## MATLAB Environment



6

## Getting Help

- Help Windows
- Type one of the following commands in the command window:
  - **help** – lists all the help topics
  - **help topic** – provides help for the specified topic
  - **help command** – provides help for the specified command
  - **helpwin** – opens a separate help window for navigation
    - **helpwin command**
  - **doc** – Display HTML documentation in the Help browser
    - **doc command**
  - **Lookfor keyword** – search all M-files for *keyword*

7

## Variables

- Variable names:
  - Must start with a letter (*x, y, MyVar,...*)
  - May contain only letters, digits, and the underscore “\_”.
  - MATLAB is case sensitive, for example one & ONE are different variables.
  - MATLAB only recognizes the first 31 characters in a variable name.
- Assignment statement:
  - *Variable* = *number*; (*x=123;*)
  - *Variable* = *expression*; (*x=y+123;*)
- Effect of **;**
  - End of statement

8

## Variables

- Special variables:
  - **ans**: default variable name for the result.
  - **pi**:  $\pi = 3.1415926 \dots$
  - **eps**:  $\epsilon = 2.2204e-016$ , smallest value by which two numbers can differ
  - **inf**:  $\infty$ , infinity
  - **NAN** or **nan**: not-a-number
  - **i** and **j**: imaginary number  $\sqrt{-1}$
- Commands involving variables:
  - **who**: lists the names of the defined variables
  - **whos**: lists the names and sizes of defined variables
  - **clear**: clears all variables
  - **clear varname**: clears the variable *name*
  - **clc**: clears the command window
  - **clf**: clears the current figure and the graph window

9

## Vectors

- MATLAB's calculation is based on matrix/vector.
  - row vector:
    - *A*=[12 13 pi -pi]; or *A*=[12,13,pi,-pi];
  - column vector:
    - *B*=[12; 13; pi; -pi];
  - Matrix:
    - *C*=[1 2 3;4 5 6; 7 8 9];
- Use colon to create special vectors:
  - *A*=0:0.5:2;  $\leftrightarrow$  *A*=[0 0.5 1 1.5 2];
  - *A*=10:-2:2;  $\leftrightarrow$  *A*=[10 8 6 4 2];
- Vector Transpose:
  - *B*=*A'*;

10

## Vectors

- Accessing to vectors/matrices elements:
  - *A*(3), *C*(2,3), ...
- Use of colon (:) notation to access matrices elements:
  - *B*(1:3), *C*(:,1:2), *C*(2:3,:)
- Deleting a row of a matrix:
  - *B*(:,2) = [ ]; Delete 2<sup>nd</sup> column
  - *B*(3,:) = [ ]; Delete 3<sup>rd</sup> row
- Concatenation of matrices to build a larger matrix:
  - *B* = [*A A A*]; *C*=[*B B B B*];

11

## Arrays Operations

- Add(+), Subtract(-), Multiply(\*), and Divide(/) arrays and scalars.
- For A/S/M/D of an array by a scalar:
  - *B*=*A*+2; *B*=*A*-2; *B*=*A*\*2; *B*=*A*/2;
- Power n of a scalar:
  - *C*=2; *A*=*C*^2; *B*=*C*^1.3;

12

## Arrays Operations

- Element by Element Operation, dot (.) operator:
  - $C=A+B$ ;
  - $C=A-B$ ;
  - $C=A.*B$ ;
  - $C=A./B$ ;
  - $C=A.^B$
- Note: A and B Need to be same size or B be scalar!

13

13

## Matrices and Operators

- Operators:
  - $A=B+C$ ; Same Size or one of them be scalar
  - $A=B-C$ ; Same Size or one of them be scalar
  - $A=B*C$ ; Matched Size or one of them be scalar
  - $A=k*B$ ; k is scalar
  - $B=inv(A)$ ;
  - $d=det(A)$ ;
  - $B=A^2$ ;

14

14

## Matrices and Operators

- Building zero filled matrices:
  - $A= zeros(n)$ ; is an n-by-n matrix of zeros.
  - $A= zeros(n,m)$ ; is an n-by-m matrix of zeros.
- Building one filled matrices:
  - $A= ones(n)$ ; is an n-by-n matrix of ones.
  - $A= ones(n,m)$ ; is an n-by-m matrix of ones.
- Building an Identity matrix:
  - $A= eye(n)$ ;
- Getting the size of matrix:
  - $[m,n]=size(A)$ ; also  $m = size(A,1)$ ; and  $n = size(A,2)$ ;
- Getting the length of a vector:
  - $n = length(A)$ ;

15

15

## MATLAB Workspace

- MATLAB has an interactive programming capability
- format Command
  - format short: Scaled fixed point format with 5 digits.
  - format long: Scaled fixed point format with 15 digits for double and 7 digits for single.
  - format short e: Floating point format with 5 digits.
  - format long e: Floating point format with 15 digits for double and 7 digits for single.
  - format short g: Best of fixed or floating point format with 5 digits.
  - format long g: Best of fixed or floating point format with 15 digits for double and 7 digits for single.
  - format short eng: Engineering format that has at least 5 digits and a power that is a multiple of three
  - format long eng: Engineering format that has exactly 16 significant digits and a power that is a multiple of three.
- See help format for more information!

16

16

## MATLAB Workspace

- Directory Manipulation:
  - cd
  - pwd
  - dir
  - mkdir
- Load and Save command
  - save;
  - Save all variable in matlab.mat files
  - save filename a b c;
  - Save variables a, b, and c in filename.mat
  - load filename;
  - Load a pre-saved file
  - load;
  - Load matlab.mat files

17

17

## Plotting Facility

- Main command is plot
  - plot(variablename,symbol); → plot(1:10,'\*')
- Plotting curves:
  - plot(x,y); x vs. y (Both Linear)
  - semilogx(x,y); x (Logarithmic scale) vs. y (Linear scale)
  - semilogy(x,y); x (Linear scale) vs. y (Logarithmic scale)
  - loglog(x,y); x vs. y (Both logarithmic scale)
- Multiple curves
  - plot(x,y,w,z); x vs. y and z vs. w
- Multiple figures
  - figure(n); Creation a new figure plot() command
  - Close; Closes the figure n window
  - close all; Closes all the plot windows
- Subplots:
  - subplot(m,n,p); m by n grid of Plots, with p specifying the current plot as the pth window
- Line type and Color:
  - Plot(x,y,'b-',z,w,'r.',x,y,'k--')

18

18

## Plotting Facility

- axis Command

| Command                     | Description                                   |
|-----------------------------|---|
| axis([xmin xmax ymin ymax]) | Define minimum and maximum values of the axes |
| axis square                 | Produce a square plot                         |
| axis equal                  | Equal scaling factors for both axes           |
| axis normal                 | Turn off axis square, equal                   |
| axis (auto)                 | Return the axis to defaults                   |

19

19

## Plotting Facility

- Other useful Command

| Command          | Description  |
|------------------|--|
| grid on          | Add dashed grids lines at the tick marks   |
| grid off         | Removes grid lines (default)   |
| Grid             | Toggles grid status (off to on or on to off)   |
| title('text')    | Labels top of plot with text   |
| xlabel('text')   | Labels horizontal (x) axis with text   |
| ylabel('text')   | Labels vertical (y) axis with text   |
| text(x,y,'text') | Adds text to location (x,y) on the current axes, where (x,y) is in units from the current plot |

20

20

## Mathematical Functions

- Trigonometric: (sin, sind, cos, tan, asin, sinh, ...)
- Exponential: (exp, log, log10, log2, sqrt, nthroot, ...)
- Complex: (imag, real, abs, angle, conj, ...)
- Rounding and Remainder:
  - fix: Round towards zero
  - floor: Round towards minus infinity
  - ceil: Round towards plus infinity
  - round: Round towards nearest integer
  - mod: Modulus after division
  - rem: Remainder after division
- Discrete Mathematics: (gcd, lcm, factor, factorial, isprime, ...)

21

21

## Data Analysis Function

- Useful command:
  - min: Minimum element from each column of Matrix
  - max: Maximum element from each column of Matrix
  - mean: Mean of each column of Matrix
  - median: Median of each column of Matrix
  - std: Standard Deviation of each column of Matrix
  - sort: Sort a vector
  - prod: Production of a vector
  - sum: Summation of a vector

22

22

## MATLAB Programming

- Relational Operators:
  - == (equal to)      ~= (not equal)
  - < (less than)      <= (less than or equal to)
  - > (greater than)      >= (greater than or equal to)
- Logical Operators:
  - & (and)
  - | (or)
  - ~ (not)
- Logical Function:
  - and, or, xor
  - isempty: A variable is empty or not
  - all, and any: All or any of element are nonzero.

23

23

## MATLAB Programming

- Flow Control:
  - if:

|                   |                                     |
|-------------------|-------------------------------------|
| if expression     | if (a>b), disp('a>b'); end;         |
| statements        |                                     |
| elseif expression | if (a>b), disp('a>b'); else         |
| statements        | disp('a<=b'); end;                  |
| else              |                                     |
| statements        | if (a>b), disp('a>b'); elseif (a<b) |
| end               | disp('a<b'); else disp('a=b'); end; |

24

24

## MATLAB Programming

- Flow Control:

- for:

```
for variable = expr,  
statement, ...,  
statement  
end;
```

```
Cntr = 0;  
for i=1:100, Cntr = Cntr+i; end;
```

```
Cntr = 0;  
for i=100:-2:0, Cntr = Cntr+i; end;
```

```
Cntr = 0;  
for i=[1,2,5,12], Cntr = Cntr+i; end;
```

25

25

## MATLAB Programming

- Flow Control:

- while:

```
while expression  
statements  
end;
```

```
Cntr = 100;  
while (Cntr>0),  
Cntr = Cntr - 1;  
end;
```

26

26

## MATLAB Programming

- Flow Control:

- Switch-case:

```
method = 'Bilinear';  
switch lower(method)  
case {'linear','bilinear'}  
disp('Method is linear')  
case 'nearest'  
disp('Method is nearest')  
otherwise  
disp('Unknown method.')  
end;
```

27

27

## MATLAB Programming

- m files:

- script:

- A set of command in a file, which execute sequentially.

- function:

- Like functions in C

28

28

## MATLAB Programming

- Example of a script:

```
r = 100;  
theta = 12;  
x = r*cos(theta);  
y = r*sin(theta);
```

```
>>Polar2Cart;
```

29

29

## MATLAB Programming

- Example of a function:

```
function [x,y] = Polar2Cart(r,theta);  
x = r*cos(theta);  
y = r*sin(theta);
```

```
>>[x,y] = Polar2Cart(r,theta);
```

30

30

## MATLAB Programming

- Write help for your function

```
function [x,y] = Polar2Cart(r,theta);  
% This function convert Polar to Cartesian  
x = r*cos(theta); % Compute x component.  
y = r*sin(theta); % Compute y component.
```

31

31