

# 디지털신호처리



강 의 노 트

## Matlab 사용법

---

2주차 1차시

## 학습내용

- ❖ Matlab 소개
- ❖ 행렬 및 벡터 생성
- ❖ 기본적인 행렬 연산

## 학습목표

- ❖ Matlab 프로그램의 개요를 이해하고, 기본적인 기능들을 활용할 수 있다.
- ❖ Matlab의 행렬 및 벡터를 생성하는 방법을 설명할 수 있다.
- ❖ Matlab을 이용해 기본적인 행렬 연산 프로그램을 작성할 수 있다.



## Matlab 소개

### 1. Matlab 개요

#### 1) 정의

- MATrix LABoratory의 약어
- 미국의 Mathworks 사에 의해 C++언어로 개발된 컴퓨터소프트웨어
- Matlab는 기본적으로 행렬을 사용하므로 차원이 필요 없음
- 프로그래밍 언어를 사용하지 않고도 쉽게 수치 계산을 수행할 수 있음

#### 2) 응용 분야

- 수학과 관련된 다양한 계산 분야
- 알고리즘 개발
- 상황 모델링과 데이터 분석
- 여러 가지 과학과 공학적인 그래프적인 표현
- GUI에 의한 애플리케이션 개발

#### 3) Toolbox

- Matlab은 이용하고자 하는 분들의 전공에 도움을 주고자 Toolbox를 가지고 있음  
→ 신호 처리, 통계학, 영상신호 처리, 제어, Fuzzy Logic, 회계, 화학 공정에 대한 다양한 라이브러리 제공
- 해당 전공 부분의 내용을 심도 있게 지원하는 함수들의 도서관과 같은 것
- 특별히 Simulink라는 것이 있는데 주로 동적 시스템의 Simulation에 이용되고 있음
- Matlab은 외부 프로그램(C, Fortran 등)과 링크해서 이용할 수 있는 기능도 제공함

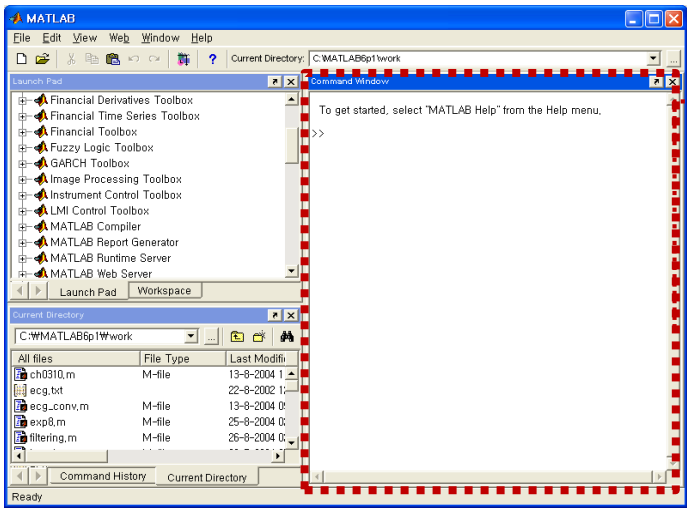


# Matlab 소개

## 2. Matlab의 구성 요소

### 1) 명령어 창

- Matlab 명령을 실행함

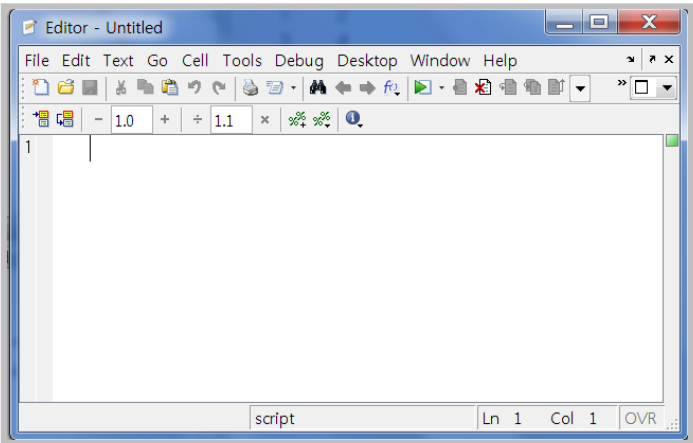


Command Window

Matlab과 대화하기 위한  
기본적인 공간

### 2) 편집기

- Matlab 프로그램을 작성함
- M-파일 프로그래밍

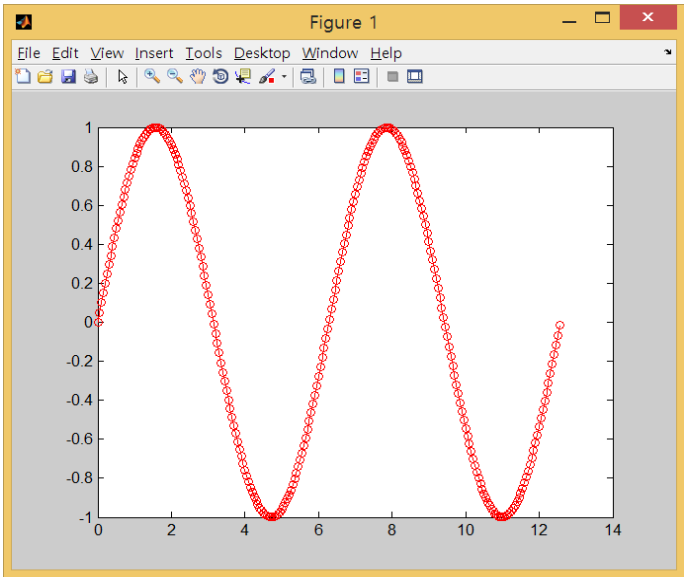




Matlab 소개

3) 그림창

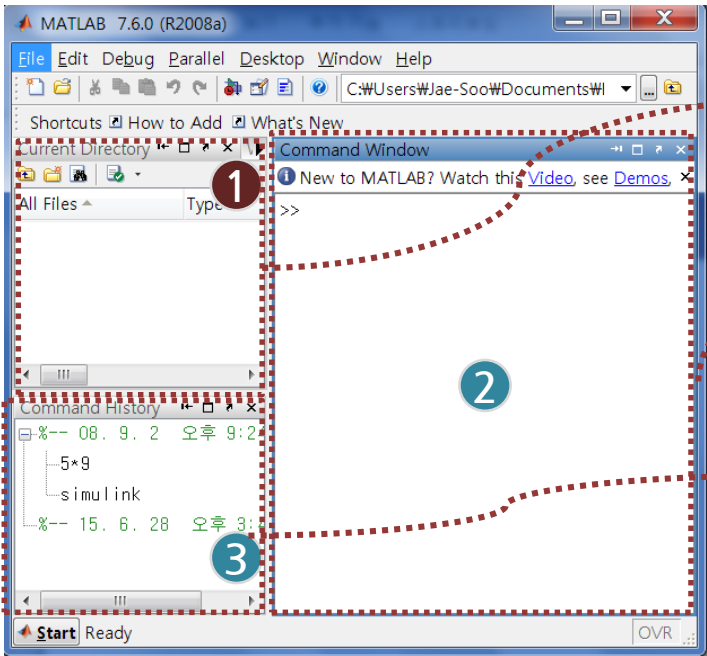
- 명령 수행 결과의 일부



4) SIMULINK

- 여러 가지 모듈을 이용하여 시뮬레이션을 수행

5) Matlab 프로그램 실행



- Current Directory  
현재 작업하고 있는 디렉토리의 파일들을 보여주는 창
- Command Window  
명령어창
- Command History  
명령어창에서 명령어를 실행한 과거 히스토리를 보여주는 창



## Matlab 소개

### 3. 간단한 공학용 계산기 기능

- Matlab은 기본적인 공학 계산이 가능함
- Matlab은 변수명에 대한 규칙을 가지고 있음
- 변수명은 대소문자를 구분함

예

Cost, cost, CoSt, COST는 모두 다른 Matlab 변수들임



행렬 및 벡터 생성

1. 행렬 입력 규칙

1) Matlab에서의 형 선언

- 다른 프로그래밍 언어들과 달리 차원의 선언이나 형 선언이 필요 없음  
→ 컴퓨터가 사용 가능한 크기까지 자동적으로 저장 공간을 할당함
- 빈칸 · 쉼표: 각각의 행렬 원소들은 빈칸 또는 쉼표를 사용하여 분리
- 대괄호 ( [ ] ): 전체 원소들은 대괄호 ( [ ] )로 감쌘
- 세미콜론 ( ; ): 원소의 끝에 세미콜론 ( ; )을 붙이면 한 행의 종료를 의미함

예제 04-01

a = [ 1 2 3 ] a 라는 변수에 1행 3열의 3개의 행렬원소를 입력하면?

[예제풀이]

```
>> a = [1, 2, 3]; 또는
>> a = [1 2 3];
```

예제 04-02

b라는 2행 2열의 변수에 4개의 원소를 입력하면?

[예제풀이]

$$b = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

```
>> b = [1, 2 ; 3, 4]; 또는
>> b = [ 1,2
          3, 4];
```



## 행렬 및 벡터 생성

## 1. 행렬 입력 규칙

## 2) Matlab에서 사용되는 원소 지정

- Matlab에서 사용되는 임의의 표현들 (수치, 함수, 수식, 문자 등)은 모두 행렬의 원소로 사용될 수 있음

```
>> x = [-2.5exp(2.5)2*3/4]
```

```
x =
```

```
-2.5000 12.1825 1.5000
```

```
>> y = ['abc'; 'x"y"z']
```

```
y =
```

```
abc
```

```
xyz
```





## 행렬 및 벡터 생성

## 1. 행렬 입력 규칙

## 3) Matlab에서 행렬 원소의 추가 및 변경

- 행렬의 원소 중에 어느 한 원소만을 추가하거나 바꾸고 싶을 때는 색인을 지정하고 원하는 값을 입력하면 됨

```
Command Window
>> x = [-2.5 12.1825 1.5000 0]

x =

    -2.5000    12.1825     1.5000         0

>> x(5)=abs(x(1));
>> x

x =

    -2.5000    12.1825     1.5000         0     2.5000

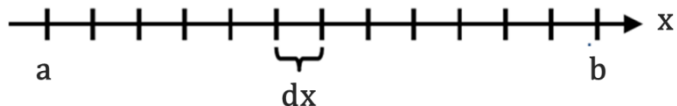
>> |
```



행렬 및 벡터 생성

2. 벡터의 합을 통한 방법

- a와 b 사이에서 dx 간격으로 n개의 분할된 벡터 값 생성



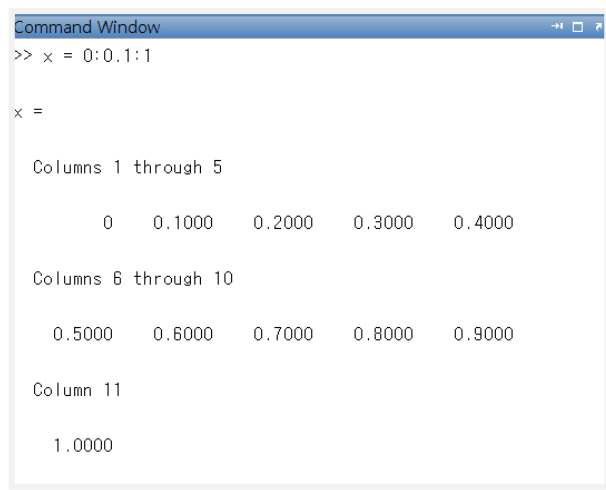
```
>> x = a : dx : b ;  
>> x = linspace(a, b, n) ;
```

} row vectors

예제 04-03

```
x = [ 0 0.1 0.2 L 0.9 1 ]  
x = 0 : 0.1 : 1
```

[예제풀이]





행렬 및 벡터 생성

2. 벡터의 합을 통한 방법

예제 04-04

$$x = linspace(0, 1, 10)$$

[예제풀이]

Command Window

```
>> x=linspace(0,1,10)

x =

Columns 1 through 5

    0    0.1111    0.2222    0.3333    0.4444

Columns 6 through 10

    0.5556    0.6667    0.7778    0.8889    1.0000
```



기본적인 행렬 연산

1. 행렬 트랜스포즈와 행렬 연결 연산

1) 행렬 트랜스포즈(Transpose) 연산

- $A'$  : 행렬의 행과 열을 바꿈

```
Command Window

A =

     1     2     3
     4     5     6
     7     8     9

>> A'

ans =

     1     4     7
     2     5     8
     3     6     9
```

2) 행렬 연결(Concatenation) 연산

- $C = [A, B]$  ,  $D = [A;B]$  : 행렬을 여러 개 연결하여 새로운 행렬을 생성

```
Command Window

>> A
A =
     1     2     3
     4     5     6
     7     8     9

>> B
B =
     1     0     0
     0     1     0
     0     0     1

>> C = [A,B]
C =
     1     2     3     1     0     0
     4     5     6     0     1     0
     7     8     9     0     0     1

>> D = [A;B]
D =
     1     2     3
     4     5     6
     7     8     9
     1     0     0
     0     1     0
     0     0     1
```



기본적인 행렬 연산

2. 행렬의 덧셈과 뺄셈

- 숫자의 덧셈 및 뺄셈과 마찬가지로 '+'와 '-'기호 사용
- 단 행렬 연산의 대상이 되는 두 행렬의 차원이 같아야 함
- 행렬 또는 벡터의 덧셈과 뺄셈: 각 행렬의 같은 위치, 행렬 상의 색인이 같은 원소끼리 이루어짐
- 1×1 행렬인 스칼라의 경우: 어떤 한 행렬이나 벡터와도 연산이 가능하며 행렬이나 벡터의 모든 원소들에 스칼라를 더하거나 빼면 됨

예제 04-05

```
□ A=[1 2 3;4 5 6];
□ B=[2 4 6;1 3 5];
□ C=A+B
```

[예제풀이]

```
C =
    3     6     9
    5     8    11
```

예제 04-06

```
□ C-5
```

[예제풀이]

```
ans =
   -2     1     4
     0     3     6
```

기초적인 행렬 연산

3. 행렬의 곱셈과 나눗셈

1) 행렬의 곱셈 연산

- \* 연산: 기본적인 두 행렬의 곱셈 경우에만 정의됨
- .\* 연산: 배열 연산의 곱셈으로 두 행렬의 차원이 같을 때 각 행렬의 같은 위치에 있는 원소들끼리 곱셈을 하여 새로운 행렬을 구성함

\* 연산

X=A \* B

```
A =
     1     2
    -3     2

>> B
     1     2
     3     4

>> X=A*B

X =
     7     10
     3      2
```

.\* 연산

X=A .\* B

```
A =
     1     2
    -3     2

>> B
     1     2
     3     4

>> X=A .* B

X =
     1     4
    -9     8
```

2) 행렬의 나눗셈 연산

- 스칼라와는 달리 일반적으로 좌측 나누기와 우측 나누기의 결과가 일치하지 않음
- 좌측 나누기:  $X=AWB \rightarrow A * X=B$ 의 해 ( $X = \text{inv}(A) * B$ )
- 우측 나누기:  $X=A/B \rightarrow X * B=A$ 의 해 ( $X = A * \text{inv}(B)$ )

Command Window

좌측 나누기

```
>> A=[1 2; -3 2]

A =
     1     2
    -3     2

>> B = [ 1 2 ; 3 4]

B =
     1     2
     3     4

>> X=AWB
```

X =

```
-0.5000    -0.5000
 0.7500     1.2500
```

X=A\B -----> X=AWB

A \* X=B의 해(X = inv(A)\*B)

Command Window

우측 나누기

```
>> A=[1 2; -3 2]

A =
     1     2
    -3     2

>> B = [ 1 2 ; 3 4]

B =
     1     2
     3     4

>> X=A/B
```

X =

```
1  0
9 -4
```

X=A/B -----> X=A/B

X \* B=A의 해 (X = A\*inv(B))



기본적인 행렬 연산

4. 기타 행렬의 연산

1) 관계연산

- Matlab에서는 차원이 같은 두 행렬에 대하여 적용할 수 있는 6가지의 관계연산자가 있음

연산자	의 미
<	미만(less than)
<=	이하(less than and equal)
>	초과(greater than)
>=	이상(greater than and equal)
==	같음(equal)
~=	같지 않음(not equal)

- 두 행렬의 대응 원소들을 비교 → 그 결과를 0과 1로 구성된 행렬의 형태로 나타냄 ( 0: 거짓, 1: 참 )

2) 논리연산

연산자	의 미
&	그리고(and)
	또는(or)
~	부정(not)

- $C = A \& B$ : 행렬 A와 행렬 B의 대응 원소들이 둘 다 0이 아닐 경우에는 1을 돌려주고 하나라도 0이면 0을 돌려줌
- $C = A | B$ : 행렬 A와 행렬 B의 대응 원소들 중 하나라도 0이 아닐 경우에는 1을. 두 원소들 중에서 어느 하나라도 0이 아닐 경우에는 1을 돌려줌
- $B = \sim A$ : A의 원소가 0이 아닐 경우에는 0을 돌려줌. 1이 아닐 경우에는 1을 돌려줌

## 핵심정리

### Matlab 소개

- Matlab: MATrix LABoratory의 약자로 기본적으로 행렬을 사용하여 다양한 응용분야에서 활용되고 있는 컴퓨터 개발 S/W
- Matlab의 구성 요소: Matlab 명령어를 수행하는 명령창, 과거 명령어 히스토리를 보여주는 명령어 히스토리 창, 현재 작업하고 있는 디렉토리의 파일들을 보여주는 디렉토리 창으로 구성

### 행렬 및 벡터 생성

- 행렬 입력 규칙: 각각의 행렬 원소들은 빈칸 또는 쉼표를 사용하여 분리해서 행렬 값을 입력하고, 전체 원소는 대괄호( [ ] )로 감싸며, 원소의 끝에 세미콜론(:)을 붙이면 한 행의 종료를 의미함
- $x = a:dx:b$ ; 명령어: a와 b 사이에서 dx 간격으로 분할 된 행렬벡터를 생성하는 명령어

### 기본적인 행렬 연산

- 행렬 연결 연산: 행렬을 여러 개 연결하여 새로운 행렬을 생성하는 연산
- 행렬 트랜스포즈 연산: 행렬의 행과 열을 바꾸는 연산
- 행렬 또는 벡터의 덧셈과 뺄셈: 각 행렬의 같은 위치, 즉 행렬 상의 색인이 같은 원소끼리 이루어짐
- '\*' 연산: 기본적인 두 행렬의 곱셈
- '.'\* 연산: 두 행렬의 차원이 같을 때 각 행렬의 같은 위치에 있는 원소들끼리 곱셈을 하여 새로운 행렬을 구성
- 좌측 나눗셈:  $X = A \backslash B$ 는  $A * X = B$ 의 해를 의미
- 우측 나눗셈:  $X = A / B$ 는  $X * B =$  의 해를 의미