기계진동학 [실습2] 2023. 4. 5.

1자유도계 자유응답

< 복 습 >

1. MATLAB 이란?

2. 관련 자료

3. MATLAB의 시작

시작 → 프로그램 → StudentMATLAB

4. (DOS용) 명령어 모음

```
EDU? cd [enter]  % 현 폴더(디렉토리) 확인
EDU? cd  . . [enter] % 상위 폴더로 이동
EDU? cd  vtoolbox [enter]  % 하위 폴더로 이동
EDU? dir(또는 ls)  % 현 폴더내의 file 확인
```

5. 변수의 형태와 규칙

- 변수 이름은 대,소문자를 구분함
- 변수 이름은 문자로 시작, 숫자나 밑줄 가능
- MATLAB 의 고유 변수들은 사용할 수 없음

6. Plot

plot(가로축 변수, 세로축 변수)

label 추가

7. Scalar 연산

- 덧셈, 뺄셈, 곱셈은 일반 계산기와 동일
- 나눗셈은 두 종류(/, ₩) 구분, 지수는 ^를 사용
- 변수를 정의한 후에 변수들만으로 연산 가능

8. Matrix 연산

- matrix 형성

  행 구분은 (ENTER) 또는 ‘;’

  열 구분은 (SPACE) 또는 ‘,’
- 행렬간의 덧셈과 뺄셈은 크기가 동일해야 함
- 행렬간의 곱셈은 두 가지로 구분

  A * B , A .* B

< m-file 프로그램 이해 >

1. 함수

MATLAB에서의 함수는 m-file로 저장되며
다른 언어의 서브루틴이나 함수와 동일

(1) 오직 하나의 변수만 되돌려주는 함수

m-file Window

```
function y=demof3_(x)
y=(2*x.^3+ 7*x.^2+ 3*x-1)./(x.^2-3*x+ 5*exp(-x));
```

Save as -> demof3_.m으로 저장
(vtoolbox 폴더에 저장)

Command Window

```
y=demof3_(3)
y=demof3_([3, 1; 0, -1])
```

(2) 여러 개의 변수들을 돌려주는 함수

m-file Window

```
function [mean, stdv] = mean_st3_(x)
        n = length(x);
        mean = sum(x)/n;
        stdv = sqrt(sum((x – mean).^2) / n);
```

Save as -> mean_st3_.m으로 저장

Command Window

```
x=[1  5  3  4  6  5  8  9  2  4];
[m, s]=mean_st3_(x)
```

2. plot 명령

데이터 (xi, yi), i=1,2,3,....,n 을 2 차원 평면상에 그래프로 나타낸다.

m-file Window

```
function ex_graph3_(n)
        delx = 10/(n-1);
    for k = 1 : n
        x(k) = (k-1) * delx;
        y(k) = sin(x(k)) * exp(-0.4*x(k));
    end
    figure(1); clf
    plot(x,y)
    xlabel('x');ylabel('y')
```

Save as -> ex_graph3_.m

Command Window

```
ex_graph3_(100)
```

3. fprintf 명령 (생략)

출력 명령

fprintf('%f %d %e', x1, x2, x3)

지정자	설명	지정자	설명
%f	부동 소수점	%d	부호가 있는 10진수
%e	소문자 e를 사용한 지수	%s	문자열로 출력
%g	%e나 %f중에서 길이가 짧은 쪽을 택해서 그 서식으로 출력, 필요 없는 소수점이나 0은 제거		

Command Window

```
a = 'It is beautiful Wednesday.';
b = 20230405.31234550000;
fprintf('%s ₩t %f ₩t %e ₩t %g ₩n', a, b, b, b);
 fprintf('%.4f ₩t %.3f ₩t %.2f ₩n', b, b, b);
```

4. for 문

(1) Loop 연산

m-file Window

```
function ex_graph3_(n)
figure(1);clf
figure(1)
 for x = 1 : 0.1 : n
    y = sin(x) * exp(-0.5*x);
    plot(x, y)
    hold on
 end
```

Save as -> ex_graph3_.m

command Window

```
ex_graph3_(10)
```

(2) Vector 연산

m-file Window

```
function ex_graph4_(n)
figure(2);clf
        x = 1 : 0.1 : n;
        y = sin(x) .* exp(-0.5*x);
        figure(2)
        plot(x, y)
        hold on
```

Save as -> ex_graph4_.m

command Window

```
ex_graph4_(10)
```

< 1자유도계 자유진동 >

1. m-file 폴더(vtoolbox)로 이동

```
EDU? cd [enter]  % 현 폴더 확인
EDU? cd  .. [enter]  %상위 폴더로 이동
EDU? cd vtoolbox [enter]  %하위 폴더로 이동
EDU? dir(또는 ls)  % 현 폴더내의 file 확인
```

2. 변수 대입법

물성치 3개와 초기조건 2개 (m, c, k, x0, v0, tf) 또는 물성치 2개와 초기조건 2개 (zeta, wn, x0, v0, tf)를 입력한다. tf는 그래프의 시간 구간(final time)

```
EDU? figure(1) [enter]   % fig1 창을 띄움
EDU? vtb1_1(1, 0.9, 10, 5, 0, 20) [enter]
EDU? figure(2) [enter]  % fig2 창을 띄움
EDU? vtb1_1(0.1432, 3.16228, 5, 0, 20) [enter]
EDU? close all [enter]   % 모든 fig 창을 닫음
```

EDU? type vtb1_1.m

3. 임계감쇠운동

[1.3.3절, 그림1.12] z=1, wn=2 rad/s, x0=0.4 mm, tf=3.5 s로 고정하고, 초기속도v0를 변화시킨다.

```
EDU? figure(1) [enter]
EDU? vtb1_1(1, 2, 0.4, 1, 3.5) [enter]
EDU? hold on [enter] % 그래프를 중복시킴
EDU? vtb1_1(1, 2, 0.4, 0, 3.5) [enter]
EDU? vtb1_1(1, 2, 0.4, -1, 3.5) [enter]
EDU? grid on [enter] % 보조치수선을 보여줌
EDU? grid off [enter] % 보조치수선을 숨김
```

4. 과도감쇠운동

[1.3.2절 그림 1.11] z=1.5, wn=2 rad/s, tf=6 s로 고정하고, 초기속도와 초기변위를 변화시킨다.

```
EDU? figure(2) [enter]
EDU? vtb1_1(1.5, 2, 0.3, 0, 6) [enter]
EDU? hold on [enter]
EDU? vtb1_1(1.5, 2, 0, 1, 6) [enter]
EDU? vtb1_1(1.5, 2, -0.3, 0, 6) [enter]
EDU? grid on [enter]
```

5. 부족감쇠운동

[1.3.1절, 그림 1.10]

'2. 변수대입법'에서 그린 그래프.

지금까지 그린 감쇠계의 그래프를 모두 비교해 본다.

```
EDU? figure(3) [enter]
EDU? vtb1_1(0.1, 2, 10, 1, 30) [enter]
EDU? grid on [enter]
EDU? close all [enter]
```

6. 초기조건의 변화에 따른 자유응답의 변화 (부족감쇠계에서)

x(0)이 변할 경우

```
EDU? hold on [enter]
EDU? vtb1_1(1, 0.9, 10, 1, 0, 20) [enter]
EDU? vtb1_1(1, 0.9, 10, -1, 0, 20) [enter]
EDU? vtb1_1(1, 0.9, 10, 10, 0, 20) [enter]
EDU? vtb1_1(1, 0.9, 10, -10, 0, 20) [enter]
EDU? close [enter]
```

v(0)이 변할 경우

```
EDU? hold on [enter]
EDU? vtb1_1(1, 0.9, 10, 1, 1, 20) [enter]
EDU? vtb1_1(1, 0.9, 10, 1, -1, 20) [enter]
EDU? vtb1_1(1, 0.9, 10, 1, 10, 20) [enter]
EDU? vtb1_1(1, 0.9, 10, 1, -10, 20) [enter]
EDU? close [enter]
```

7. 연습

(연습문제 1.53)

고유진동수 wn = 2 rad/s 이고, 초기조건이 x0 = 1 mm, v0 = 0 mm/s이며, 감쇠비 z = 0.01, 0.1, 0.2, 0.4, 0.6, 0.8 인 감쇠계에 대해 응답 x(t)를 그려라.

(연습문제 1.54)

wn = 2 rad/s, z = 0.1, v0 = 0 mm/s이고, 초기변위 x0 = 1, 5, 10, 100 mm 인 부족감쇠계의 응답 x(t)를 그려라.

vtb1_1 설명

```
EDU? type vtb1_1.m
function VTB1_1(m,c,k,x0,v0,tf)
%VTB1_1 Free response of a single degree of freedom system.
% VTB1_1(m,c,k,x0,v0,tf) plots the free response of a single degree of freedom system.
%The arguments x0 and v0 represent the initial conditions and tf represents the total time of the response.
%The system is described by mass m, damping c, and stiffness k.
% VTB1_1(zeta,w,x0,v0,tf) plots the free response of a single degree of freedom system.
%The arguments x0 and v0 represent the initial conditions and tf represents the total time of the response.
% The system is in non-dimensional form, where zeta is the damping ratio, and w is the natural frequency in rad/s.
% This loop determines which type of input format you are using. =>M-file의 전반적인 설명
if nargin==5
   z=m;w=c;tf=v0;v0=x0;x0=k;m=1;c=2*z*w;k=w^2;
End  => 입력 변수 개수 구분
w=sqrt(k/m);    => 고유 진동수 계산
z=c/2/w/m;%(1.30)  => 감쇠비 계산
wd=w*sqrt(1-z^2);%(1.37)  => 감쇠 고유진동수 계산
fprintf('The natural frequency is %.3g rad/s.₩n',w);
fprintf('The damping ratio is %.3g.₩n',z);
fprintf('The damped natural frequency is %.3g.₩n',wd);
t=0:tf/1000:tf;  => 시간 축을 0에서부터 tf까지 tf/1000 간격으로 지정
if z < 1  => 부족 감쇠인 경우 x(t) 계산 및 출력
      A=sqrt(((v0+z*w*x0)^2+(x0*wd)^2)/wd^2);%(1.38)  => amplitude 계산
      phi=atan2(x0*wd,v0+z*w*x0);%(1.38)  => phase 계산
      x=A*exp(-z*w*t).*sin(wd*t+phi);%(1.36)  => x(t) 계산
      fprintf('A= %.3g₩n',A);
      fprintf('phi= %.3g₩n',phi);
   elseif z==1    => 임계 감쇠인 경우
      a1=x0;%(1.46)
      a2=v0+w*x0;%(1.46)
      fprintf('a1= %.3g₩n',a1);
      fprintf('a2= %.3g₩n',a2);
      x=(a1+a2*t).*exp(-w*t);%(1.45)
   else    => 과도 감쇠인 경우
      …….
plot(t,x)  => 위의 계산 결과를 그래프로 출력
xlabel('Time') => x축 지정
ylabel('Displacement')    => y축 지정
title('Displacement versus Time')  => graph title 지정
text(0,0,'Press Return To Continue','sc')
```