201933914 김상우

메트랩 ODE45 과제 제출

코드

Main file

%{

filename : main\_solver\_4

date 2021.06.05 22:05

athorizer : sangwoo michael kim

language : matlab R2019b

객체지향 프로그래밍 기법 중 Class기능을 이용하여서

전체 코드를 기능별로 분할 모듈화 한 후

Quarter\_Car\_Class로 응집시켜 동작 코드를 정리하였음.

%}

clc

clear

fprintf("calculating...\n");

Quarter\_Car\_solver\_object = Quarter\_Car\_Class\_try\_4();

QCS\_obj = Quarter\_Car\_solver\_object;

QCS\_obj.plot\_all();

fprintf("done!\n");

Class file

%{

filename : Quarter\_Car\_Class\_try\_4

date 2021.06.05 22:23

athorizer : sangwoo michael kim

language : matlab R2019b

%}

classdef Quarter\_Car\_Class\_try\_4

properties

m1 = 49.8;

m2 = 466.5;

k1 = 135000;

k2 = 5700;

c1 = 1400;

c2 = 0;%it is going to be input by propertie set method

velocity\_kmh = 0

velocity\_ms = 0

length\_L = 5.2

height\_Y = 0.1

t\_range = [0 15]; % 시뮬레이션 시간 범위

initial = [0 0 0 0]; % 변수 초기값

result\_t = 0;

result\_x = 0;

end

methods

function dx = Quarter\_Car(obj, t, x)

% Y = half sign signal Part ? if 문 포함 %

[y, dy] = obj.input\_displacement(t);

dx = zeros(4, 1);

dx(1) = x(2);

dx(3) = x(4);

dx(2) =(obj.k1\*y+obj.c1\*dy-(obj.k1+obj.k2).\*x(1)-(obj.c1+obj.c2).\*x(2)+obj.k2.\*x(3)+obj.c2.\*x(4))/obj.m1;

dx(4) =(-obj.c2.\*x(4)+obj.c2.\*x(2)-obj.k2.\*x(3)+obj.k2.\*x(1))/obj.m2;

end

function [y, dy] = input\_displacement(obj, t)

T = obj.length\_L/obj.velocity\_ms;

w = pi\*obj.velocity\_ms/obj.length\_L;

y = 0;

dy = 0;

if(0 <= t)

if(t <= T)

y = obj.height\_Y\*sin(w\*t);

dy = w\*obj.height\_Y\*cos(w\*t);

end

end

end

function obj = solve\_ode(obj)

[t, x] = ode45(@obj.Quarter\_Car, obj.t\_range, obj.initial); % ODE45 function

obj.result\_t = t;

obj.result\_x = x(:,3);%바퀴가 아닌 차량만의 변위 만을 확인하고자 함

%위 같은 처리를 하지 않았을 때 x가 벡터의 집합으로 처리되어 최대 최소 값을 구하면 각각원 소별로 4개의 최대 최소값이 발생하였다.

end

function [maximun, minimum] = maxmin\_value(obj)

maximun = max(obj.result\_x);

minimum = min(obj.result\_x);

end

function plot\_result(obj)

plot(obj.result\_t, obj.result\_x);

end

% function set\_properties\_2(obj, c2\_input, velocity)

% c2 = c2\_input;

% velocity\_kmh = velocity;

% velocity\_ms = velocity\*(1000/3600);

% end

function obj = set\_properties(obj, c2\_input, velocity)

obj.c2 = c2\_input;

obj.velocity\_kmh = velocity;

obj.velocity\_ms = velocity\*(1000/3600);

end

function get\_result\_and\_plot(obj)

obj = obj.solve\_ode();

obj.plot\_result();

end

function plot\_unit(obj)

obj = obj.set\_properties(1000, 5);

hold on

obj.get\_result\_and\_plot();

obj = obj.set\_properties(1000, 10);

obj.get\_result\_and\_plot();

obj = obj.set\_properties(1000, 15);

obj.get\_result\_and\_plot();

obj = obj.set\_properties(1000, 20);

obj.get\_result\_and\_plot();

obj = obj.set\_properties(1000, 25);

obj.get\_result\_and\_plot();

hold off

end

function plot\_all(obj)

c = [1000, 5000, 10000, 15000];

v = [5, 10, 15, 20, 25];

maximum = zeros(size(4, 5));

minimum = zeros(size(4, 5));

tiledlayout(3,2)

a = size(c);

b = size(v);

for i = 1:a(2)%매트랩은 배열의 첫째항의 인덱스가 1이다.

nexttile

hold on

for j = 1:b(2)

obj = obj.set\_properties(c(i), v(j));

obj = obj.solve\_ode();

[maximum(i, j), minimum(i, j)] = obj.maxmin\_value();%solve\_ode() 함수를 수정하므로써 대입 시 차원 불일치 문제를 수정함.

obj.plot\_result();

end

hold off

xlabel("time(s)")

ylabel("displacement(m)")

switch i

case 1

lgd = legend("v = 5", "v = 10", "v = 15", "v = 20", "v = 25");

title("responce1");

title(lgd, "c = 1000");

case 2

lgd = legend("v = 5", "v = 10", "v = 15", "v = 20", "v = 25");

title("responce2");

title(lgd, "c = 5000");

case 3

lgd = legend("v = 5", "v = 10", "v = 15", "v = 20", "v = 25");

title("responce3");

title(lgd, "c = 10000");

case 4

lgd = legend("v = 5", "v = 10", "v = 15", "v = 20", "v = 25");

title("responce4");

title(lgd, "c = 15000");

end

end

%최대 최소 변위 출력

nexttile

hold on

for i = 1:a(2)

plot(v, maximum(i, :))

end

hold off

legend("c2 = 1000", "c2 = 5000", "c2 = 10000", "c2 = 15000");

xlabel("velocity(km/h)")

ylabel("displacement(m)")

title("maximum displacement")

nexttile

hold on

for i = 1:a(2)

plot(v, minimum(i, :))

end

hold off

legend("c2 = 1000", "c2 = 5000", "c2 = 10000", "c2 = 15000");

xlabel("velocity(km/h)")

ylabel("displacement(m)")

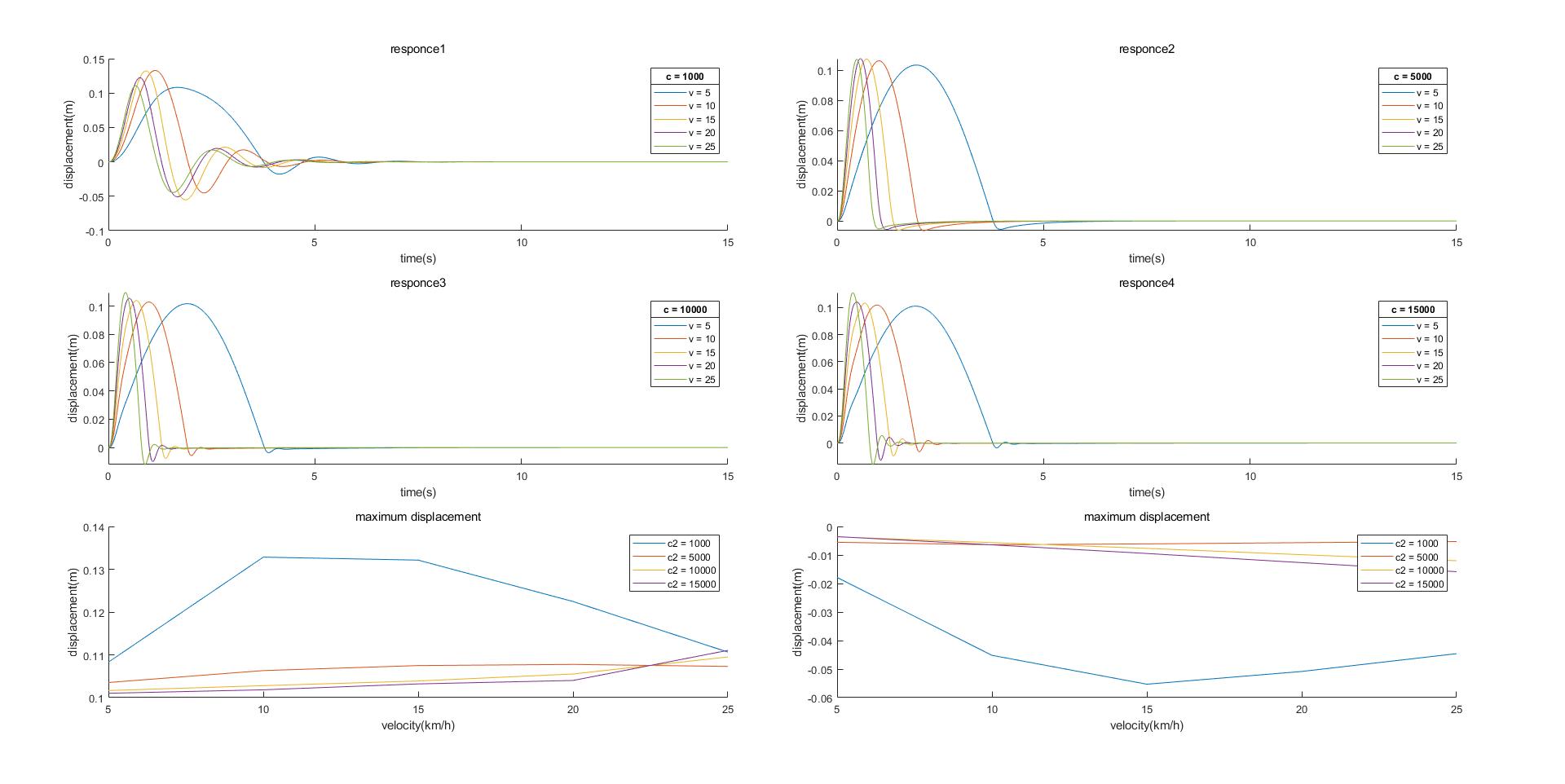
title("maximum displacement")

end

end

end

분석



C 가 작을 때에는 단순 m-k 시스템처럼 입력 주파수(주행 속도)에 따른 최대 변위 반응 변화가 두드러진 반면, C가 5000Nm/s 이상인 경우부터는 주행속도 변화가 무색해질 정도로 C = 1000Nm/s 인 경우에 비해 진폭 감소가 두드러졌다.

최대 변위 기준으로 C = 1000Nm/s인 경우에는 속도에 따라서 10 Km/h 일 때와 15Km/h 일 때 정적 변위 0.1m 에 대하여 동적변위가 각각 0.1328m, 0.1321m 로 증폭계수가 M = 1.328, M = 1.321 로 30%p 가량 증폭되는 것을 확인하였다. 반면에 C = 5000Nm/s 이상인 경우의 최대변위를 확인하면 최고 0.111m로 댐핑이 Nm/s 이상만 되어도 즉폭계수가 1.11 이하가 되는 것을 확인하였다.

단, 댐핑이 아주 큰 경우 속력이 증가할수록 최대 변위 값이 증가하는 양상을 보였다. V = 25 km/h 일때 C = 1000 인 경우 최대 변위가 0.1106m, C = 15000인 경우 0.111m로 댐핑이 큰 경우의 변위의 증폭계수가 오히려 더 커져서, 댐핑이 더 클수록 최대변위가 더 커졌다. C와 M = X/ 의 관계가 역전되는 양상이 발견되었다. 따라서 25km/h이하의 환경에서 설계되는 사례의 경우 이번 실험자료와 결과를 인용하여도 좋지만, 25km/h 이상의 고속환경에서 차량이 설계될 경우 기존 자료를 그대로 이용하지 말고, 25km/h 이상의 속도에서의 추가실험을 통해 고속환경에서의 반응 역전 현상을 더욱 면밀히 조사할 필요가 있다.