EXPERIMENT – VII

Anirban Mukherjee

16BIT0200

Sprung Mass Displacement in a Quarter Car (One Wheel) Model

Aim:

- To write MATLAB code for system of second order differential equations of the form X"+ AX = 0 using Diagonalization.
- Determining the sprung mass displacement in a car suspension for one wheel. Solving a coupled system of ordinary differential equations derived from the mathematical model.

Problem Statements:

Using the transformation Y = P X, the given system of differential equations can be reduced to uncoupled system as X" + AX = 0 , where D is the diagonal matrix with eigen values of A as diagonal elements and Q is the modal matrix corresponding to A.

Matlab Commands:

- eig: *eig(A)* returns a vector of the eigenvalues of matrix A
- solve: *solve(eq, x)* returns the set of all complex solutions of an equation or inequality eq with respect to x.

Description of Physical Experiment:

Sprung Mass Displacement in a Quarter Car (One Wheel) Model

•The vehicle suspension system differ depending on the manufacturer which ensures a wide range of models. Whichever solution is adopted to design, a suspension system has the primary role to ensuring the safety function.

• It is known that road unevenness produce oscillations of the vehicle wheels which will transmitted to their axles. It becomes clear that the role of the suspension system witch connect the axles to the car body is to reduce as much vibrations and shocks occurring in the operation. This causes, the necessity to using a suspension of a better quality.

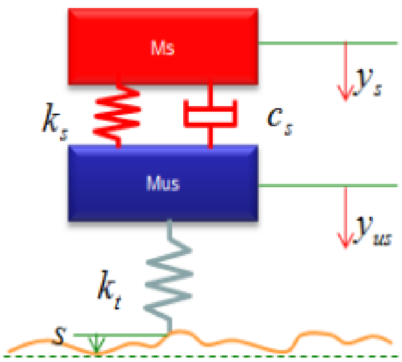
• A quality suspension must achieve a good behavior of the vehicle and a degree of comfort depending on the interaction with uneven road surface. When the vehicle is requested by uneven road profile, it should not be too large oscillations, and if this occurs, they must be removed as quickly. The design of a vehicle suspension is an issue that requires a series of calculations based on the purpose.

• Suspension systems are classified in the well-known terms of passive, semi-active, active and various in between systems. Typical features are the required energy and the characteristic frequency of the actuator. Passive systems are the most common. So far, several models have been developed, such as quarter car, half car or full car suspension.

The system shown in Figure.1 is an quarter car system where

- M_s - is the sprung mass
- M_{us} - is the unsprung mass
- k_s - is the stiffness coefficient of the suspension
- k_t - is the vertical stiffness of the tire
- c_s - is the damping coefficient of the suspension
- b_2 - is the damping coefficient of the tire
- y_s - the vertical displacement of sprung mass
- y_{us} - is the vertical displacement of unsprung mass
- s - is the road excitation

<u>Sprung Mass equation in a Quarter car (one Wheel)</u>



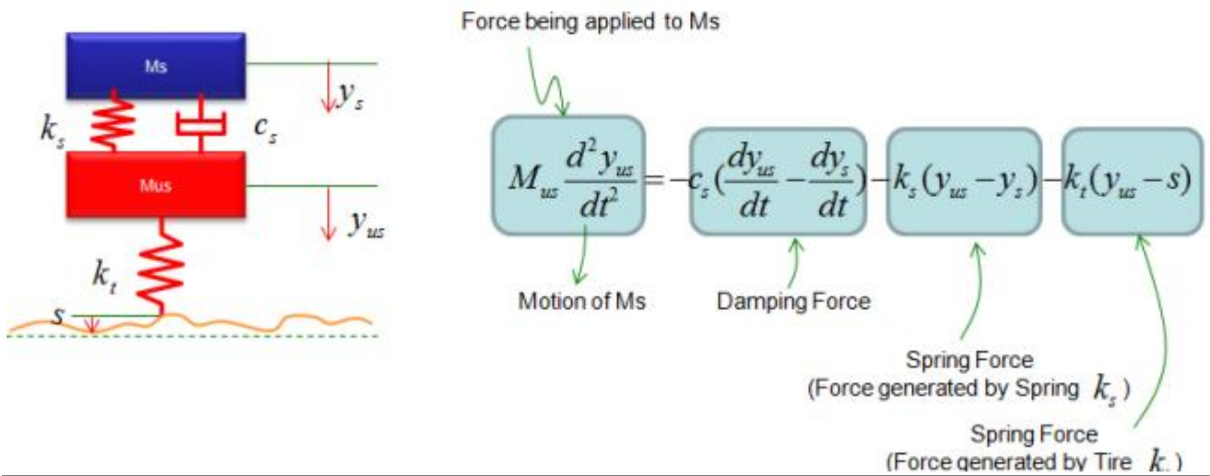
Force being applied to Ms

$$\underbrace{M_s \frac{d^2 y_s}{dt^2}}_{\text{Motion of Ms}} = \underbrace{-c_s\left(\frac{dy_s}{dt} - \frac{dy_{us}}{dt}\right)}_{\text{Damping Force}} \underbrace{-k_s(y_s - y_{us})}_{\text{Spring Force}}$$

Spring Force
(Force generated by Spring k_s)

$$M_s \frac{d^2 y_s}{dt^2} + c_s \left(\frac{dy_s}{dt} - \frac{dy_{us}}{dt} \right) + k_s (y_s - y_{us}) = 0 \quad (1)$$

Unsprung Mass equation in a Quarter car (one Wheel)



$$M_{us} \frac{d^2 y_s}{dt^2} + c_s \left(\frac{dy_{us}}{dt} - \frac{dy_s}{dt} \right) + k_s (y_{us} - y_s) + k_t y_{us} = k_t s \quad (2)$$

Connection to Mathematics:

The differential equation of the sprung and un sprung masses of the quarter-car model:

$$M_s \frac{d^2 y_s}{dt^2} + c_s \left(\frac{dy_s}{dt} - \frac{dy_{us}}{dt} \right) + k_s (y_s - y_{us}) = 0$$

$$M_{us} \frac{d^2 y_s}{dt^2} + c_s \left(\frac{dy_{us}}{dt} - \frac{dy_s}{dt} \right) + k_s (y_{us} - y_s) + k_t y_{us} = k_t s$$

Matrix form of the equations

$$\begin{bmatrix} M_s & 0 \\ 0 & M_{us} \end{bmatrix} \begin{bmatrix} \dfrac{d^2 y_s}{dt^2} \\ \dfrac{d^2 y_{us}}{dt^2} \end{bmatrix} + \begin{bmatrix} c_s & -c_s \\ -c_s & c_s \end{bmatrix} \begin{bmatrix} \dfrac{dy_s}{dt} \\ \dfrac{dy_{us}}{dt} \end{bmatrix} + \begin{bmatrix} k_s & -k_s \\ -k_s & k_s + k_t \end{bmatrix} \begin{bmatrix} y_s \\ y_{us} \end{bmatrix} = \begin{bmatrix} 0 \\ k_t s \end{bmatrix}$$

$$\mathbf{M}\ddot{\mathbf{y}} + \mathbf{c}\dot{\mathbf{y}} + \mathbf{k}\mathbf{y} = \mathbf{f}$$

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \frac{d^2}{dt^2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} (k_1 + k_2) & -k_2 \\ -k_2 & (k_2 + k_3) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\mathbf{M} \frac{d^2\mathbf{x}}{dt^2} + \mathbf{K}\mathbf{x} = 0$$

Code:

Question:

Solve the following system of equations

y1" = −5y1 −2y1

y2" = −2y1 −2y2

```
clc
clear all
syms x1(t) x2(t)
A = input('Enter the coefficient matrix A: '); lambda = eig(A);
fprintf('eigen values of A are %f, %f\n\n',lambda);
for i=1:length(lambda) temp = null(A-lambda(i)*eye(size(A)),'r');
P(:,i) = temp./min(temp);
end
disp('The Modal Matrix is: ');
disp(P);
D = inv(P)*A*P;
X= [x1;x2];
Sol1 = dsolve(diff(x1,2) + D(1)*x1 == 0);
Sol2 = dsolve(diff(x2,2) + D(4)*x2 == 0);
disp('The solution of the system diff(X,2)+DX=0 is: ');
disp(Sol1);
disp(Sol2);
disp('The Solution of the given system is: ');
Y= P*[Sol1; Sol2]
```

MATLAB output:

```
Enter the coefficient matrix A: [5 2; 2 2]

eigen values of A are 1.000000, 6.000000


The Modal Matrix is:

     1     2

    -2     1


The solution of the system diff(X,2)+DX=0 is:

C2*cos(t) + C3*sin(t)


C5*cos(6^(1/2)*t) + C6*sin(6^(1/2)*t)


The Solution of the given system is:


Y =


 2*C5*cos(6^(1/2)*t) + 2*C6*sin(6^(1/2)*t) + C2*cos(t) + C3*sin(t)
```

```
C5*cos(6^(1/2)*t) + C6*sin(6^(1/2)*t) - 2*C2*cos(t) - 2*C3*sin(t)
```

Exercise question:

Enter the coefficient matrix A: [-5 2; 2 -2]

eigen values of A are -6.000000, -1.000000

The Modal Matrix is:

```
    1.0000    1.0000
   -0.5000    2.0000
```

The solution of the system diff(X,2)+DX=0 is:

C2*exp(6^(1/2)*t) + C3*exp(-6^(1/2)*t)

C6*exp(t) + C5*exp(-t)

The Solution of the given system is:

Y =

```
        C2*exp(6^(1/2)*t) + C3*exp(-6^(1/2)*t) + C6*exp(t) + C5*exp(-t)
 2*C6*exp(t) - (C3*exp(-6^(1/2)*t))/2 - (C2*exp(6^(1/2)*t))/2 + 2*C5*exp(-t)
```

<u>My Work:</u>

Using system of differential equations with matrix method, we can apply many machine learning concepts.

We would for example design a gaming system, for the non playing characters to react to the main character; we would use systems of differential equations. We can train the system to react to the player’s behaviour according to his behaviour on the game.

We can also use it for making an interactive operating system. If we want to open a certain software, we input a command, which is converted to differential equation then to matrix form, that will generate an output prompting the system to open that software.

Take a simulation of a chat assistant for a bit more complex example,

Say for saying a greeting, for doing that we generate a matrix A converted from its differential equation form.

If we abuse or violate rules of the chat bot, B matrix is generated.

When the matrix A is input, the system would give an output that would make the non assistant be friendly to the user and comply to its requirements, if B is input, the user would be reported and blocked.

Let matrix A be $\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$

When Y =

(910872158600853*C5*cos((2947644224044330^(1/2)*t)/33554432))/562949953421312 + (910872158600853*C6*sin((2947644224044330^(1/2)*t)/33554432))/562949953421312 + C2*cos((430055496483542^(1/2)*t)/33554432) + C3*sin((430055496483542^(1/2)*t)/33554432)

C5*cos((2947644224044330^(1/2)*t)/33554432) + C6*sin((2947644224044330^(1/2)*t)/33554432) - (910872158600853*C2*cos((430055496483542^(1/2)*t)/33554432))/562949953421312 - (910872158600853*C3*sin((430055496483542^(1/2)*t)/33554432))/562949953421312

A friendly greeting is generated back as say

Display(‘Good Morning to you too!’)

If say matrix B be $\begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$ for any violation

When

Y =

C2*cos((785004844525955^(1/2)*t)/33554432) + C3*sin((785004844525955^(1/2)*t)/33554432) + (3718594782844541*C5*cos((1211123672421791^(1/2)*t)/16777216))/1125899906842624 + (3718594782844541*C6*sin((1211123672421791^(1/2)*t)/16777216))/1125899906842624

C5*cos((1211123672421791^(1/2)*t)/16777216) - (3718594782844541*C3*sin((785004844525955^(1/2)*t)/33554432))/1125899906842624 - (3718594782844541*C2*cos((785004844525955^(1/2)*t)/33554432))/1125899906842624 + C6*sin((1211123672421791^(1/2)*t)/16777216)

The user is reported and blocked.