ME 563
Mechanical Vibrations
Lecture #11

Single Degree of Freedom

Free Response (Matlab)

Free Response (Matlab)

When using Matlab to analyze the free response of a system, we can either:

- Derive the analytical solution and program this solution into Matlab along with the parameters of the model.
- Allow Matlab to numerically integrate the equation of motion to obtain the response.

However, Matlab requires that we define the system before it will solve for the response solution. There are two options:

- State variable model
- Transfer function model

State Variable Modeling

Consider the disc on an incline with no forcing function:

$$\rightarrow \quad M_{eff}\ddot{x}_d + C\dot{x}_d + Kx_d = 0$$

This equation is of second order; however, integrators are generally written to solve first order equations. To convert this equation of motion to a first order equation, the system states are defined as follows:

$$\rightarrow \quad q_1 = x_d, q_2 = \dot{x}_d$$

Then the equation of motion is rewritten as follows:

$$\rightarrow \quad M_{eff}\ddot{x}_d = -Kx_d - C\dot{x}_d \Rightarrow M_{eff}\dot{q}_2 = -Kq_1 - Cq_2$$

First-Order Equations

This equation is one of the two state variable equations we need. If we start with one second order equation, then we must have two first order equations to define the system.

$$\rightarrow \quad M_{eff}\dot{q}_2 = -Kq_1 - Cq_2$$

The second equation is a kinematic identity:

$$\rightarrow \quad \dot{q}_1 = q_2$$

Then the two equations can be expressed in matrix form as:

$$\rightarrow \quad \begin{Bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ -\dfrac{K}{M_{eff}} & -\dfrac{C}{M_{eff}} \end{bmatrix} \begin{Bmatrix} q_1 \\ q_2 \end{Bmatrix}$$

State Variable Formulation

We must also define the input force (as zeros in this case) using the following formulation:

$$\begin{Bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{Bmatrix}_{2\times1} = \begin{bmatrix} 0 & 1 \\ -\dfrac{K}{M_{eff}} & -\dfrac{C}{M_{eff}} \end{bmatrix}_{2\times2} \begin{Bmatrix} q_1 \\ q_2 \end{Bmatrix}_{2\times1} + \left[0\right]_{2\times1} f(t)$$

This equation is written in general as follows:

$$\{\dot{q}\}_{2\times1} = [A]_{2\times2}\{q\}_{2\times1} + [B]_{2\times1} f(t)$$

Input

State matrix

Input matrix

State vector

State Variable Formulation

We must also define the outputs of interest. Sometimes, the outputs are just the state variables (displacements, etc.), but in other cases the outputs are functions of the states (forces).

We use the following formulation to define the outputs, *y*:

$$\rightarrow \begin{Bmatrix} y_1 \\ y_2 \end{Bmatrix}_{2\times1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}_{2\times2} \begin{Bmatrix} q_1 \\ q_2 \end{Bmatrix}_{2\times1} + \begin{bmatrix} 0 \end{bmatrix}_{2\times1} f(t)$$

Or: $$\rightarrow \{y\}_{2\times1} = [C]_{2\times2}\{q\}_{2\times1} + [D]_{2\times1} f(t)$$ **Input**

Direct feed-through matrix

Output matrix **State vector**

Other Output Examples

If we are interested in the forces acting on the mass, what is the appropriate output equation in state variable form?

$$\begin{Bmatrix} y_1 \\ y_2 \end{Bmatrix}_{2\times 1} = \begin{bmatrix} -K & 0 \\ 0 & -C \end{bmatrix}_{2\times 2} \begin{Bmatrix} q_1 \\ q_2 \end{Bmatrix}_{2\times 1} + [0]_{2\times 1} f(t)$$

What about if we are interested in the acceleration as when we measure the acceleration experimentally?

$$y = \begin{bmatrix} -\frac{K}{M_{eff}} & -\frac{C}{M_{eff}} \end{bmatrix}_{1\times 2} \begin{Bmatrix} q_1 \\ q_2 \end{Bmatrix}_{2\times 1} + [0] f(t)$$

Free Response in MATLAB

Once the equations of motion are written in state variable form, then we can define the system in MATLAB:

```
sys = ss(A,B,C,D);
```

Then we can simulate the response using

```
initial(sys,[x0 xd0]);
```

where `x0` and `xd0` are the initial conditions on displacement and velocity.

We can also including time vectors in the argument or in the output from the function for plotting purposes.

Free Response in MATLAB

MATLAB also allows us to define the system using the transfer function:

```
sys = tf(NUM,DEN);
```

where the NUM and DEN vectors define the polynomials of the transfer function between the input and output of the system. In the single degree of freedom case,

```
sys = tf([1],[Meff C M]);
```

Then we call the `initial` function as described previously.