MAE 5070 **Exercise Set IV** April 1, 2011

D. A. Caughey

Consider the equation of motion for a spring-mass-damper system, characterized as

$$m\frac{\mathrm{d}^2y}{\mathrm{d}t^2} + c\frac{\mathrm{d}y}{\mathrm{d}t} + ky = 0 \tag{1}$$

which can be written as

$$\frac{\mathrm{d}^2y}{\mathrm{d}t^2} + 2\zeta\omega_n\frac{\mathrm{d}y}{\mathrm{d}t} + \omega_n^2 y = 0 \tag{2}$$

where $\omega_n = \sqrt{k/m}$ is the *undamped natural frequency* and $\zeta = c/(2\sqrt{mk})$ is the *damping ratio*.

As seen in class (and the course notes), this equation can be written in state space form as

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \tag{3}$$

where $x_1 = y$ corresponds to position and $x_2 = \dot{y}$ corresponds to velocity in the original formulation.

1. Write the characteristic equation for the above system and solve it for the case when $\omega_n = 6.0$ and $\zeta = 2/3$. After finding the eigenvalues by hand, use the MATLAB function **eig** to verify your values, and also to compute the corresponding eigenvectors of the plant matrix. Use the eigenvalues you have computed to determine the period of oscillation of the solution, and verify that it is equal to

$$T = \frac{2\pi}{\omega_n\sqrt{1-\zeta^2}} = 1.405 \tag{4}$$

What are the components of the eigenvectors $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$?

2. For the same *system* – i.e., same values of ω_n and ζ, determine the coefficients a_i in the modal expansion of the solution as a linear combination of the eigenvectors $\mathbf{x}^{(i)}$

$$\mathbf{x} = \sum_{i=1}^{2} a_i\mathbf{x}^{(i)}e^{\lambda_i t} \tag{5}$$

for the solution having initial conditions

$$y(0) = 1.0$$
$$\dot{y}(0) = 0.0$$

Note: you can determine the coefficients by using the modal matrix $\mathbf{Q}$ you determined using the MATLAB function **eig** in Problem 1, then solving the equations

$$\mathbf{Q}\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} y(0) \\ \dot{y}(0) \end{pmatrix} \tag{6}$$

Plot the components x_1 and x_2 of the state vector over the time interval $0 \le t \le 5.0$.

3. Solve the same problem as in **2** using the MATLAB function **ss** to define a state-space system data element and use the MATLAB function **initial** to study the system response to the specified initial perturbation. Compare the result with the solution you found for question **2**.

4. For the same system, analyze the response for the initial perturbation for which $y(0) = 0$ and $\dot{y}(0) = 1.0$. What is the maximum displacement in this case, and what is the time at which the maximum displacement occurs?

5. Finally, consider the system for which $\omega_n = 2.0$ and $\zeta = 1.0$. Determine the system response for the following initial perturbations using the methods of *both* Problems **2** and **3**. If you encounter differences, explain their origin and suggest how the method of Problem **2** must be modified to produce the correct answer.

a. Initial condition: $y(0) = 1$, $\dot{y}(0) = -2$; and

b. Initial condition: $y(0) = 1$, $\dot{y}(0) = 0$.

These notes are intended to identify the MATLAB functions that you will need to solve the problems in Exercise Set IV, and to summarize their properties. Of course, more complete information is available using the MATLAB help command.

1 Solution of Linear Systems of Equations

The linear system of equations

$$\mathbf{A}\mathbf{x} = \mathbf{y} \tag{7}$$

can be solved using the MATLAB function **inv**. If $\mathbf{A}$ is a non-singular $n \times n$ matrix and $\mathbf{y}$ is a column vector containing n elements, the statement

```
x = inv(A)*y;
```

puts the solution into the n-vector $\mathbf{x}$. The following MATLAB statement, using the *right division* operator,

```
x = A\y;
```

is equivalent to multiplication by the inverse (and is the preferred form as it is slightly more stable and efficient – although this usually really matters only for rather large or poorly conditioned matrices). Note that in Eq. (7) both $\mathbf{x}$ and $\mathbf{y}$ are *column* vectors; e.g., in MATLAB $\mathbf{y} = \mathbf{x}(0)$ would be defined as

```
y = [y_0 ; ydot_0];
```

or, equivalently, as

```
y = [y_0 ydot_0]';
```

where the ' indicates the transpose of the *row* vector [y_0 ydot_0].

The MATLAB function **eig**, called as

```
[modal, dlam] = eig(A);
```

determines the eigenvalues and eigenvectors of the matrix $\mathbf{A}$. If $\mathbf{A}$ is an $n \times n$ matrix, both *modal* and *dlam* are also $n \times n$ matrices. The matrix *dlam* is a diagonal matrix whose entries are the eigenvalues of $\mathbf{A}$. The n-th column of *modal* is the eigenvector corresponding to the eigenvalue in the n-th row (or column) of *dlam*.

2 Response of Linear Systems

The MATLAB function **ss** can be used to create the data object corresponding to a state-space model. For the *linear time-invariant* model defined by

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\delta \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\delta \end{aligned} \tag{8}$$

the MATLAB command

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

defines *sys* as the corresponding data object, assuming the matrices **A**, **B**, **C**, and **D** have been defined. In Eqs. (8) **x** is the *state vector*, δ is the *control input* vector, and **y** is the *output* vector. The matrix **A** is the *plant matrix*, and **B** is the *control matrix*. The matrices **C** and **D** are used to define the output vector in terms of the system state and control inputs. This output vector is primarily used when studying control systems with *feedback*. At this point, we are interested only in open-loop systems, so we will set $\mathbf{D} = 0$ and define $\mathbf{C} = \mathbf{I}$ to be the identity matrix (so the system output will be identical to the state vector, i.e., $\mathbf{y} = \mathbf{x}$).

Once the state-space system data object *sys* has been created, the MATLAB command

initial(sys, x_0, t_final);

computes the system response for the initial condition contained in the vector *x_0* and displays the trajectories of the elements of the output vector up to the time *t_final*. Alternatively, specifying

[y, t, x] = initial(sys, x_0, t_final);

returns the output response in the vector **y** and the state variable response in the vector **x**, at the times in the vector t. The variables **x** and **y** are actually *matrices* with each row containing the n elements of the system state variable (or output vector) at the corresponding time t; i.e., these matrices have *length(t)* rows and n columns.