ssresp.vibe A vibration example with two modes

1 In the following example, we explore the a mechanical vibration example, especially with regard to its modes of vibration. Both undamped and (under)damped cases are considered and we discover the effects of damping.

Example ssresp.vibe−1 re: vibration with two modes

2 Consider the system of Fig. vibe.1 in which a velocity source V_S is applied to spring K_1, which connects to mass m_1, which in turn is connected via spring K_2 and damper B to mass m_2 which.[a]

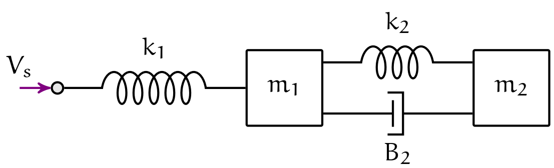

Figure vibe.1: schematic of the two-mass system.

3 The state-space model A-matrix is given as

$$A = \begin{bmatrix} -B/m_1 & -1/m_1 & B/m_1 & 0 \\ K_1 & 0 & -K_1 & 0 \\ B/m_2 & 1/m_2 & -B/m_2 & -1/m_2 \\ 0 & 0 & K_2 & 0 \end{bmatrix} \quad (1)$$

with parameters as follows.[b]

```
m_1 = 0.1;
m_2 = 1.1;
K_1 = 8;
K_2 = 9;
```

4 Two different values for B are given: 0 and 20 N-s/m. We will explore the modes of vibration in each case and plot a corresponding free response.

a. This common situation appears in a slightly modified form in Rowell and Wormley. (Rowell and Wormley, System Dynamics: An Introduction)
b. The programming language used in this example is Matlab.

Without damping

5 Without damping, we expect the system

to be marginally stable and have two pairs of second-order undamped subsystems with their own unique natural frequencies.

```
B = 0;
A = [...
  -B/m_1, -1/m_1, B/m_1, 0; ...
  K_1, 0, -K_1, 0; ...
  B/m_2, 1/m_2, -B/m_2, -1/m_2; ...
  0, 0, K_2, 0 ...
]
```

```
A =

         0  -10.0000         0         0
    8.0000         0   -8.0000         0
         0    0.9091         0   -0.9091
         0         0    9.0000         0
```

6 To explore the modes of vibration, we consider the eigendecomposition of A.

```
[M,L] = eig(A);
```

7 Let's take a closer look at the eigenvalues.

```
disp(diag(L))
```

```
   0.0000 + 9.3818i
   0.0000 - 9.3818i
   0.0000 + 2.7270i
   0.0000 - 2.7270i
```

8 So we have two pairs of purely imaginary eigenvalues. We would say, then, that there are two "modes of vibration," and similarly two second-order systems comprising this fourth-order system. When we consider what the natural frequency and damping ratio is for each pair, we're considering the natural frequencies associated with each "mode of vibration."

9 For a second-order system (see Lec. trans.secondo), the roots of the characteristic equation, which are equal to the eigenvalues corresponding to that

second-order pair, are given in terms of natural frequency ω_n and damping ratio ζ:

(2)

10 So the imaginary part is nonzero only when $\zeta \in [0, 1)$, that is, when the system is underdamped or undamped. In this case,

11 This, taken with the fact that the eigenvalues L have zero real parts, implies either ω_n or ζ is zero. But if ω_n is zero, the eigenvalues L would all be zero, which they are not. Therefore, $\zeta = 0$ for both pairs of eigenvalues.

12 This leaves us with eigenvalues:

$$\pm j\omega_{n_1} \quad \text{and} \quad \pm j\omega_{n_2}. \tag{3}$$

13 So we can easily identify the natural frequencies ω_{n_1} and ω_{n_2} associated with each mode as follows.

```
wn1 = imag(L(1,1));
wn2 = imag(L(3,3));
disp(sprintf('natural frequencies: %g, %g',wn1,wn2))
```

```
natural frequencies: 9.38179, 2.72699
```

Free response

14 Let's compute the free response to some initial conditions. The free state response is given by

15 So we can find this from the state transition matrix Φ, which is known from Lec. ssresp.diag to be ________.

16 First, we construct Φ' symbolically.

```
syms t
Phi_p = @(t) diag(diag(exp(L*t)));
vpa(Phi_p(t),3)
```

```
ans =

[ exp(t*9.38i),            0,           0,
↪  0]
[            0, exp(-t*9.38i),           0,
↪  0]
[            0,            0, exp(t*2.73i),
↪  0]
[            0,            0,           0,
↪  exp(-t*2.73i)]
```

17 Now we can apply our transformation.

```
M_inv = M^-1; % compute just once, not on every call
Phi = @(t) M*Phi_p(t)*M_inv; % messy to print
```

18 So our symbolic solution is to multiply the initial conditions by this matrix.

```
x_0 = [1;0;0;0]; % initial conditions
x = Phi(t)*x_0; % free response, symbolically
vpa(x,3)
```

```
ans =

        0.45*exp(-t*9.38i) + 0.45*exp(t*9.38i) +
        ↪  0.0498*exp(-t*2.73i) +
        ↪  0.0498*exp(t*2.73i)
  exp(-t*9.38i)*0.422i - exp(t*9.38i)*0.422i +
  ↪  exp(-t*2.73i)*0.0136i - exp(t*2.73i)*0.0136i
  - 0.0451*exp(-t*9.38i) - 0.0451*exp(t*9.38i) +
  ↪  0.0451*exp(-t*2.73i) + 0.0451*exp(t*2.73i)
 - exp(-t*9.38i)*0.0433i + exp(t*9.38i)*0.0433i +
 ↪  exp(-t*2.73i)*0.149i - exp(t*2.73i)*0.149i
```

Plotting a free response

19 Let's make the symbolic solution into something we can evaluate numerically and plot.

```
x_fun = matlabFunction(x);
```

20 Now let's set up our time array for the plot.

```
t_a = linspace(0,5,300);
```

21 Plot the state responses through time. The output is shown in Fig. vibe.2.

```
figure
plot(t_a,real(x_fun(t_a)))
xlabel('time (s)')
ylabel('state free response')
legend('x_1','x_2','x_3','x_4')
```

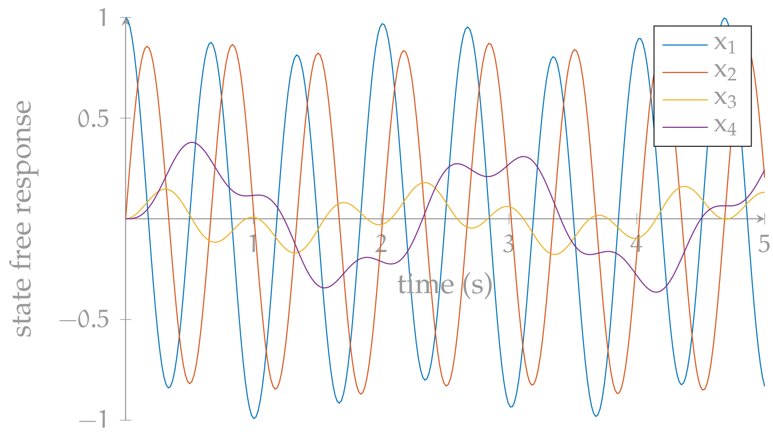

Figure vibe.2: state free response with zero damping. Note the visible vibrations modes, which are clearly mixing in x_3 and x_4.

With a little damping

22 Now consider the case when the damping coefficent B is nonzero. Let's recompute A and the eigendecomposition.

```
B = 20;
A = [...
  -B/m_1, -1/m_1, B/m_1, 0; ...
  K_1, 0, -K_1, 0; ...
  B/m_2, 1/m_2, -B/m_2, -1/m_2; ...
  0, 0, K_2, 0 ...
]
```

```
A =

 -200.0000  -10.0000  200.0000         0
    8.0000         0   -8.0000         0
   18.1818    0.9091  -18.1818   -0.9091
         0         0    9.0000         0
```

```
[M,L] = eig(A);
```

23 Let's take a closer look at the eigenvalues.

```
disp(diag(L));
```

```
   1.0e+02 *

  -2.1778 + 0.0000i
  -0.0000 + 0.0274i
  -0.0000 - 0.0274i
  -0.0040 + 0.0000i
```

24 We can see that one of the second-order systems is now "overdamped" or, equivalently, has split into two first-order systems. The other is now underdamped (but barely damped). Let's compute the natural frequency of the remaining vibratory mode.

```
wn1 = abs(L(2,2))
```

```
wn1 =

    2.7384
```

25 So the effect of damping was to eliminate the ≈ 10 rad/s mode and leave us with a slightly modified version of the ≈ 2.7 rad/s mode.

Free response now

```
syms t
Phi_p = @(t) diag(diag(exp(L*t)));
vpa(Phi_p(t),3)
```

```
ans =

[ exp(-218.0*t),                          0,
↪   0,              0]
[             0, exp(t*(- 0.00154 + 2.74i)),
↪   0,              0]
[             0,                          0,
↪   exp(t*(- 0.00154 - 2.74i)),             0]
[             0,                          0,
↪   0, exp(-0.401*t)]
```

26 Now we can apply our transformation.

```
M_inv = M^-1; % compute just once, not on every call
Phi = @(t) M*Phi_p(t)*M_inv; % messy to print
```

27 So our symbolic solution is to multiply the initial conditions by this matrix.

```
x_0 = [1;0;0;0]; % initial conditions
x = Phi(t)*x_0; % free response, symbolically
```

Plotting a free response

28 Let's make the symbolic solution into something we can evaluate numerically and plot.

```
x_fun = matlabFunction(x);
```

29 Now let's set up our time array for the plot.

```
t_a = linspace(0,5,300);
```

30 Plot the state responses through time. The output is shown in Fig. vibe.3.

```
figure
plot(t_a,real(x_fun(t_a)))
xlabel('time (s)')
ylabel('state free response')
legend('x_1','x_2','x_3','x_4')
```

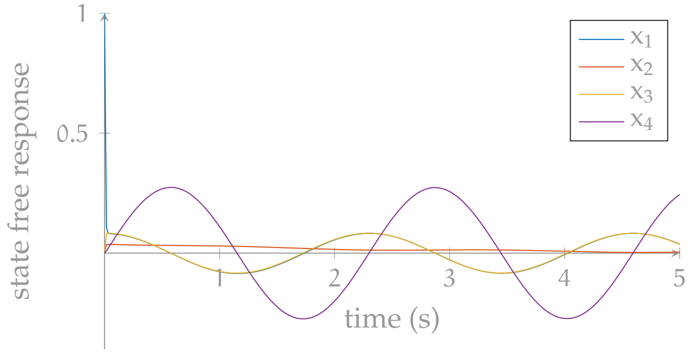

Figure vibe.3: state free response with some damping. Note that there's only one remaining vibration mode because the other mode is now overdamped. The remaining mode is very lightly damped and does not decay appreciably in five seconds.