Dynamical Systems Homework 3

March 19, 2025

Gabe Morris

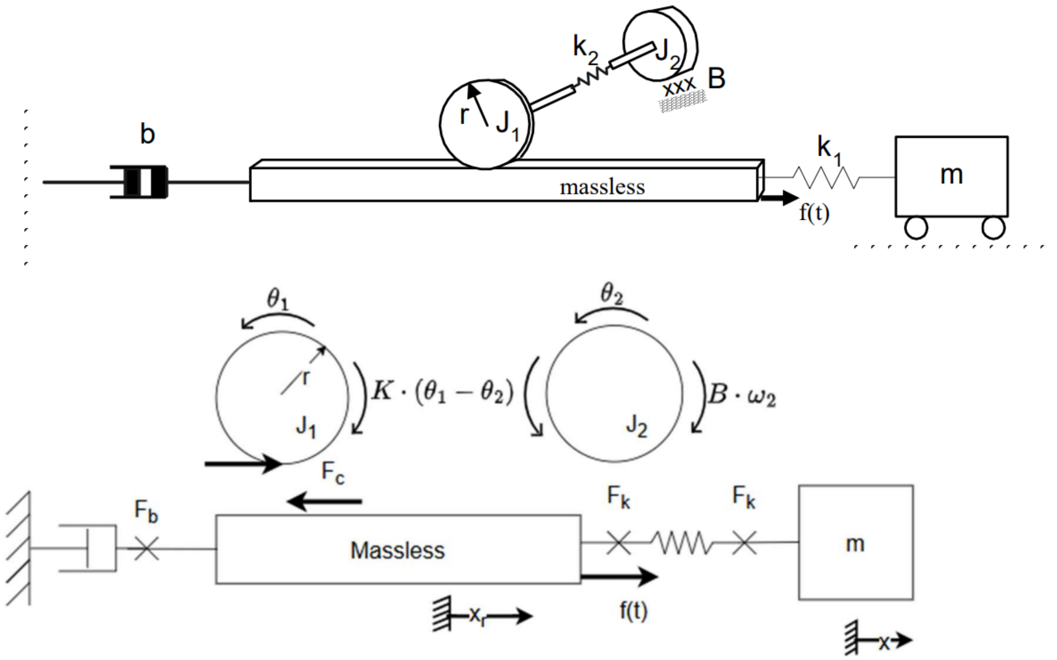
[1]:
```
# toc
import sympy as sp
import matplotlib.pyplot as plt

plt.style.use('../maroon_ipynb.mplstyle')
```

Contents

Problem 1

Given



The constants from the above figure are given as follows:

$$
\begin{gathered}
r = 0.05\,m \\
k_1 = 1000\,N/m \\
k_2 = 500\,N/m \\
b = 100\,N \cdot s/m \\
B = 2.5\,N \cdot m \cdot s \\
J_1 = J_2 = 1\,kg \cdot m^2 \\
m = 1\,kg
\end{gathered}
$$

Find

With the above system,

a. Develop a set of state variable equations. Define the state variable using the general matrix form: $\dot{S} = A \cdot S + B \cdot U$, where A and B are matrices and S is a vector of state variables and U is a vector of inputs. Be careful with labeling since one of the damping coefficients is also labeled B in the figure.

b. Solve the state variable equations using a 4th order Runge-Kutta numerical method for a total simulation time of 20 seconds and assuming that all initial velocities and spring forces are zero and that $f(t) = 10\,N$. Make sure the check that the number of points used in the simulation is adequate.

c. Verify your results using energy conservation. Show that at a given time, t, the total energy that has crossed the boundary through inputs and dampers is equal to the total energy stored in the system through mass/inertias and springs.
d. Use the output matrix form $Y = C \cdot S + D \cdot U$, where Y is the chosen output and C and D are matrices. Use this form to define the contact force between the rack and the pinion using state variable simulation results. Then plot the contact force.
e. Find the input-output equation between the input force and the contact force. One approach of doing this is the use the `ss2tf` function in the `scipy.signal` module.
f. Use the laplace transform approach to solve the input-output equation and compare with results from part d. Comment on how the roots of the characteristic polynomial affect the response.

Solution

Part A

From the above figure, the massless element is denoted as x_1 and the mass denoted by m is x_2.

[2]:
```
# Define symbols
r, k1, k2, b, B, J1, J2, m, F_c, t, s = sp.symbols('r k1 k2 b B J1 J2 m F_c t␣
↪s')
th1, th2 = sp.Function('theta_1')(t), sp.Function('theta_2')(t)
x1, x2 = sp.Function('x1')(t), sp.Function('x2')(t)
f = sp.Function('f')(t)

# Construct equations
eq1 = sp.Eq(0, f - F_c - b*x1.diff() + k1*(x2 - x1))
eq2 = sp.Eq(m*x2.diff(t, 2), k1*(x1 - x2))
eq3 = sp.Eq(J1*th1.diff(t, 2), F_c*r + k2*(th2 - th1))
eq4 = sp.Eq(J2*th2.diff(t, 2), k2*(th1 - th2) - B*th2.diff())
eq5 = sp.Eq(x1.diff(t, 2), r*th1.diff(t, 2))
eqs = [eq1, eq2, eq3, eq4, eq5]
display(*eqs)
```

$$0 = -F_c - b\frac{d}{dt}x_1(t) + k_1\left(-x_1(t) + x_2(t)\right) + f(t)$$

$$m\frac{d^2}{dt^2}x_2(t) = k_1\left(x_1(t) - x_2(t)\right)$$

$$J_1\frac{d^2}{dt^2}\theta_1(t) = F_c r + k_2\left(-\theta_1(t) + \theta_2(t)\right)$$

$$J_2\frac{d^2}{dt^2}\theta_2(t) = -B\frac{d}{dt}\theta_2(t) + k_2\left(\theta_1(t) - \theta_2(t)\right)$$

$$\frac{d^2}{dt^2}x_1(t) = r\frac{d^2}{dt^2}\theta_1(t)$$

To make this a system of first order ODEs, we need to add these state variable relationships:

$$\begin{cases} \dot{x}_1 = x_3 \\ \dot{x}_2 = x_4 \\ \dot{\theta}_1 = \theta_3 \\ \dot{\theta}_2 = \theta_4 \end{cases}$$

The reason why I prefer this method of ensuring each coordinate system is in the state variable model over using the deltas (i.e. Δx and $\Delta\theta$) is that you get a more meaningful solution. I'd rather solve for the coordinate positions instead of the displacement between two coordinate positions. However, I do think that the latter method is more convenient for hand calculations and can be a little faster at setting up the system.

[3]:
```
x3, x4 = sp.Function('x3')(t), sp.Function('x4')(t)
th3, th4 = sp.Function('theta_3')(t), sp.Function('theta_4')(t)

subs = [
    (x1.diff(), x3),
    (x2.diff(), x4),
    (th1.diff(), th3),
    (th2.diff(), th4)
]

state_eqs = [sp.Eq(sub[0], sub[1]) for sub in subs]

eqs = [eq.subs(subs) for eq in eqs] + state_eqs
display(*eqs)
```

$$0 = -F_c - bx_3(t) + k_1\left(-x_1(t) + x_2(t)\right) + f(t)$$

$$m\frac{d}{dt}x_4(t) = k_1\left(x_1(t) - x_2(t)\right)$$

$$J_1\frac{d}{dt}\theta_3(t) = F_c r + k_2\left(-\theta_1(t) + \theta_2(t)\right)$$

$$J_2\frac{d}{dt}\theta_4(t) = -B\theta_4(t) + k_2\left(\theta_1(t) - \theta_2(t)\right)$$

$$\frac{d}{dt}x_3(t) = r\frac{d}{dt}\theta_3(t)$$

$$\frac{d}{dt}x_1(t) = x_3(t)$$

$$\frac{d}{dt}x_2(t) = x_4(t)$$

$$\frac{d}{dt}\theta_1(t) = \theta_3(t)$$

$$\frac{d}{dt}\theta_2(t) = \theta_4(t)$$

[4]:
```
# Solving
# symbols to solve for
diffs = (x1.diff(), x2.diff(), th1.diff(), th2.diff(), x3.diff(), x4.diff(),␣
    ↪th3.diff(), th4.diff(), F_c)
state_sol = sp.solve(eqs, diffs, dict=True)[0]
rhs = []
for key in diffs:
    rhs.append(state_sol[key])
    display(sp.Eq(key, state_sol[key]))
```

$$\frac{d}{dt}x_1(t) = x_3(t)$$

$$\frac{d}{dt}x_2(t) = x_4(t)$$

$$\frac{d}{dt}\theta_1(t) = \theta_3(t)$$

$$\frac{d}{dt}\theta_2(t) = \theta_4(t)$$

$$\frac{d}{dt}x_3(t) = -\frac{br^2x_3(t)}{J_1} - \frac{k_1r^2x_1(t)}{J_1} + \frac{k_1r^2x_2(t)}{J_1} - \frac{k_2r\theta_1(t)}{J_1} + \frac{k_2r\theta_2(t)}{J_1} + \frac{r^2f(t)}{J_1}$$

$$\frac{d}{dt}x_4(t) = \frac{k_1x_1(t)}{m} - \frac{k_1x_2(t)}{m}$$

$$\frac{d}{dt}\theta_3(t) = -\frac{brx_3(t)}{J_1} - \frac{k_1rx_1(t)}{J_1} + \frac{k_1rx_2(t)}{J_1} - \frac{k_2\theta_1(t)}{J_1} + \frac{k_2\theta_2(t)}{J_1} + \frac{rf(t)}{J_1}$$

$$\frac{d}{dt}\theta_4(t) = -\frac{B\theta_4(t)}{J_2} + \frac{k_2\theta_1(t)}{J_2} - \frac{k_2\theta_2(t)}{J_2}$$

$$F_c = -bx_3(t) - k_1x_1(t) + k_1x_2(t) + f(t)$$

This is the state-variable form, but the state-space form is the matrix representation of the above system. The state-space form only uses the ODEs, so we neglect the solution for F_c for now. I stored the solution for F_c to be used later.

[5]:
```
disps = [diff.integrate() for diff in diffs[:-1]]
A, b = sp.linear_eq_to_matrix(rhs[:-1], disps)
B = -b/f
U = sp.Matrix([f])
S_dot = sp.Matrix(diffs[:-1])
ans = sp.Eq(S_dot, sp.Add(sp.MatMul(A, sp.Matrix(disps)), sp.MatMul(B, U)))
ans
```

Answer

$$\begin{bmatrix} \frac{d}{dt}x_1(t) \\ \frac{d}{dt}x_2(t) \\ \frac{d}{dt}\theta_1(t) \\ \frac{d}{dt}\theta_2(t) \\ \frac{d}{dt}x_3(t) \\ \frac{d}{dt}x_4(t) \\ \frac{d}{dt}\theta_3(t) \\ \frac{d}{dt}\theta_4(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{r^2}{J_1} \\ 0 \\ \frac{r}{J_1} \\ 0 \end{bmatrix} \begin{bmatrix} f(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -\frac{k_1 r^2}{J_1} & \frac{k_1 r^2}{J_1} & -\frac{k_2 r}{J_1} & \frac{k_2 r}{J_1} & -\frac{b r^2}{J_1} & 0 & 0 & 0 \\ \frac{k_1}{m} & -\frac{k_1}{m} & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{k_1 r}{J_1} & \frac{k_1 r}{J_1} & -\frac{k_2}{J_1} & \frac{k_2}{J_1} & -\frac{b r}{J_1} & 0 & 0 & 0 \\ 0 & 0 & \frac{k_2}{J_2} & -\frac{k_2}{J_2} & 0 & 0 & 0 & -\frac{B}{J_2} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \theta_1(t) \\ \theta_2(t) \\ x_3(t) \\ x_4(t) \\ \theta_3(t) \\ \theta_4(t) \end{bmatrix}$$