Suspension Example

November 15, 2023

[1]:
```
import sympy as sp
import control as ct
import matplotlib.pyplot as plt
import numpy as np

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

From lecture 17, we are finding the frequency response of the two mass suspension model whose equations are

[2]:
```
m1, m2, k1, k2, c1, s, t = sp.symbols('m1:3 k1:3 c1 s t')
x1, x2, y = sp.Function('x1')(t), sp.Function('x2')(t), sp.Function('y')(t)

m1_, m2_ = 250, 25
k1_, k2_ = 1.0975e4, 1e5
c1_ = 943

eq1 = sp.Eq(m1*x1.diff(t, 2), k1*(x2 - x1) + c1*(x2.diff() - x1.diff()))
eq2 = sp.Eq(m2*x2.diff(t, 2), k2*(y - x2) + k1*(x1 - x2) + c1*(x1.diff() - x2.
 ↪diff()))
display(eq1, eq2)
```

$$m_1 \frac{d^2}{dt^2} x_1(t) = c_1 \left(-\frac{d}{dt} x_1(t) + \frac{d}{dt} x_2(t) \right) + k_1 \left(-x_1(t) + x_2(t) \right)$$

$$m_2 \frac{d^2}{dt^2} x_2(t) = c_1 \left(\frac{d}{dt} x_1(t) - \frac{d}{dt} x_2(t) \right) + k_1 \left(x_1(t) - x_2(t) \right) + k_2 \left(-x_2(t) + y(t) \right)$$

[3]:
```
lp = lambda expr: sp.laplace_transform(expr, t, s)[0]

eq1_s = sp.Eq(lp(eq1.lhs), lp(eq1.rhs.expand()))
eq2_s = sp.Eq(lp(eq2.lhs), lp(eq2.rhs.expand()))

sub_ics = [
    (x1.subs(t, 0), 0),
    (x2.subs(t, 0), 0),
    (x1.diff().subs(t, 0), 0),
    (x2.diff().subs(t, 0), 0)
]
```

```
eq1_s = eq1_s.subs(sub_ics)
eq2_s = eq2_s.subs(sub_ics)

sol_lp = sp.solve([eq1_s, eq2_s], [lp(x1), lp(x2)])
for key, value in sol_lp.items():
    T = (value/lp(y)).simplify()
    display(sp.Eq(key/lp(y), T.collect(s)))
```

$$\frac{\mathcal{L}_t\left[x_1(t)\right](s)}{\mathcal{L}_t\left[y(t)\right](s)} = \frac{k_2\left(c_1 s + k_1\right)}{c_1 k_2 s + k_1 k_2 + m_1 m_2 s^4 + s^3\left(c_1 m_1 + c_1 m_2\right) + s^2\left(k_1 m_1 + k_1 m_2 + k_2 m_1\right)}$$

$$\frac{\mathcal{L}_t\left[x_2(t)\right](s)}{\mathcal{L}_t\left[y(t)\right](s)} = \frac{k_2\left(c_1 s + k_1 + m_1 s^2\right)}{c_1 k_2 s + k_1 k_2 + m_1 m_2 s^4 + s^3\left(c_1 m_1 + c_1 m_2\right) + s^2\left(k_1 m_1 + k_1 m_2 + k_2 m_1\right)}$$

[4]:
```
den = [m1_*m2_, c1_*m1_ + c1_*m2_, k1_*m1_ + k1_*m2_ + k2_*m1_, c1_*k2_,␣
 ↪k1_*k2_]

T1 = ct.tf([c1_*k2_, k2_*k1_], den)
T2 = ct.tf([k2_*m1_, k2_*c1_, k2_*k1_], den)

display(T1, T2)
```

$$\frac{9.43\times10^{7}s + 1.098\times10^{9}}{6250s^4 + 2.593\times10^{5}s^3 + 2.802\times10^{7}s^2 + 9.43\times10^{7}s + 1.098\times10^{9}}$$

$$\frac{2.5\times10^{7}s^2 + 9.43\times10^{7}s + 1.098\times10^{9}}{6250s^4 + 2.593\times10^{5}s^3 + 2.802\times10^{7}s^2 + 9.43\times10^{7}s + 1.098\times10^{9}}$$

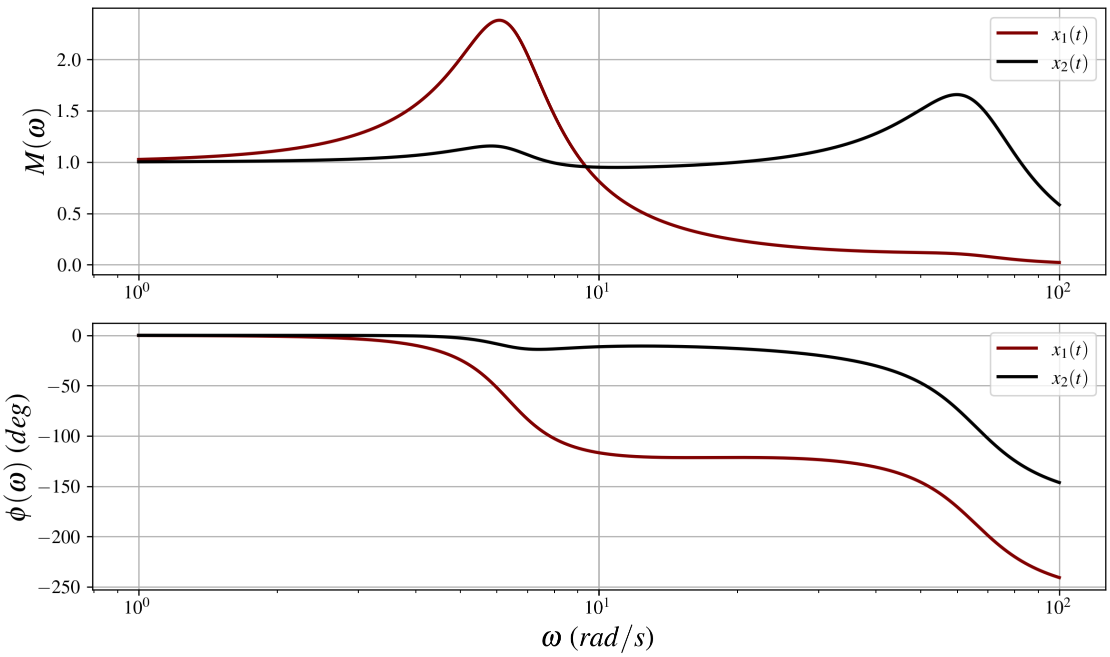
[5]:
```
# Getting the frequency response
omegas = np.linspace(1, 100, 10_000)
mag1, phase1, _ = ct.frequency_response(T1, omegas)
mag2, phase2, _ = ct.frequency_response(T2, omegas)

# The phase output does not go more negative than 180 for the above function,␣
 ↪so we need to fix that here
phase1[phase1 > 0] = phase1[phase1 > 0] - 2*np.pi

fig, (ax1, ax2) = plt.subplots(nrows=2, ncols=1)
ax1.set_xscale('log')
ax2.set_xscale('log')

ax1.plot(omegas, mag1, label='$x_1(t)$')
ax1.plot(omegas, mag2, label='$x_2(t)$')
ax2.plot(omegas, np.rad2deg(phase1), label='$x_1(t)$')
ax2.plot(omegas, np.rad2deg(phase2), label='$x_2(t)$')
```

```
ax1.set_ylabel(r'$M(\omega)$')
ax2.set_ylabel(r'$\phi(\omega)$ ($deg$)')
ax2.set_xlabel(r'$\omega$ ($rad/s$)')
ax1.legend()
ax2.legend()
plt.show()
```



[6]:
```
tire_max = omegas[max(mag2) == mag2][0]
tire_max  # Maximum frequency in rad/s for tire displacement
```

[6]: 59.89108910891089

[7]:
```
chassis_max = omegas[max(mag1) == mag1][0]
chassis_max  # Maximum frequency in rad/s for the chassis displacement
```

[7]: 6.069306930693069

From the above results, we can see that the chassis is not affected by high frequencies. The tires, however, are affected by frequencies around 60 rad/s. This would be a good frequency for rumble strips. Furthermore, the chassis experiences a maximum displacement of 2.38 times the bump amplitude for frequencies around 6 rad/s.