System Dynamics Homework 3

September 27, 2023

Gabe Morris

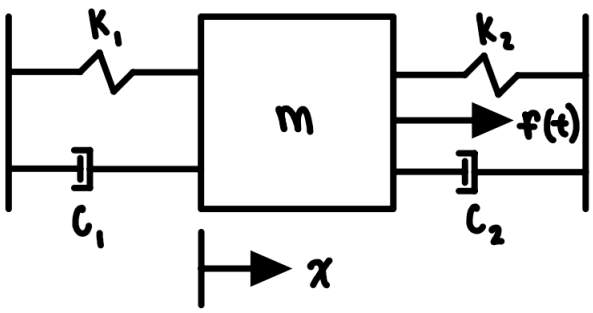
[1]:
```
import control as ct
import matplotlib.pyplot as plt
import numpy as np
import sympy as sp
import pandas as pd

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

Contents

ME 3613 Homework 3 Gabe Morris
gnm54

1 Problem 1

1.1 Given



$$m = 10\ kg$$

$$k_1 = 85\ N/m,\ k_2 = 30\ N/m$$

$$c_1 = 4\ N \cdot s/m,\ c_2 = 3\ N \cdot s/m$$

The input force $f(t)$ is in the `data.xlsx` file.

1.2 Find

Using the control package `tf()` function to get the forced response, find the following:

a. The equation of motion for the system.
b. The transfer function $X(s)/F(s)$.
c. The forced response for the no noise data. Plot the input force and the response on separate axes.
d. Repeat part c for the high frequency noise data.
e. Repeat part c for the low frequency noise data.

1.3 Solution

1.3.1 Part A

[2]:
```
t = sp.Symbol('t')
x = sp.Function('x')(t)
f = sp.Function('f')(t)

eq = sp.Eq(10*x.diff(t, 2), f - 85*x - 30*x - 4*x.diff() - 3*x.diff())
eq
```

[2]:
$$10\frac{d^2}{dt^2}x(t) = f(t) - 115x(t) - 7\frac{d}{dt}x(t)$$

1.3.2 Part B

[3]:
```
s = sp.Symbol('s')
eq_s = sp.Eq(
    sp.laplace_transform(eq.lhs, t, s)[0],
    sp.laplace_transform(eq.rhs, t, s)[0]
)
eq_s
```

[3]:
$$10s^2\mathcal{L}_t\left[x(t)\right](s) - 10sx(0) - 10\left.\frac{d}{dt}x(t)\right|_{t=0} = -7s\mathcal{L}_t\left[x(t)\right](s) + \mathcal{L}_t\left[f(t)\right](s) - 115\mathcal{L}_t\left[x(t)\right](s) + 7x(0)$$

[4]:
```
eq_s = eq_s.subs([
    (x.subs(t, 0), 0),
    (x.diff().subs(t, 0), 0)
])
eq_s
```

[4]:
$$10s^2\mathcal{L}_t\left[x(t)\right](s) = -7s\mathcal{L}_t\left[x(t)\right](s) + \mathcal{L}_t\left[f(t)\right](s) - 115\mathcal{L}_t\left[x(t)\right](s)$$

[5]:
```
X = sp.solve(eq_s, sp.laplace_transform(x, t, s)[0])[0]
X
```

[5]:
$$\frac{\mathcal{L}_t\left[f(t)\right](s)}{10s^2 + 7s + 115}$$

[6]:
```
# Transfer function
sys1 = ct.tf(1, [10, 7, 115])
sys1
```

[6]:
$$\frac{1}{10s^2 + 7s + 115}$$

1.3.3 Part C

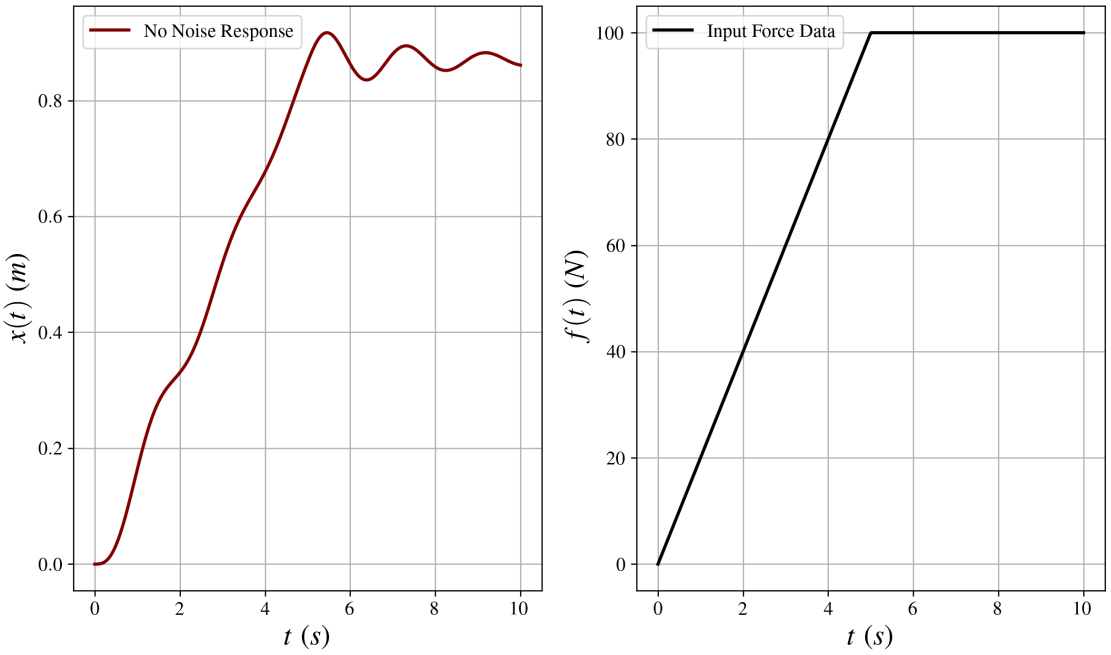
[7]:
```
data = pd.read_excel('data.xlsx', sheet_name='Input Force Data')
data
```

[7]:

	Time (s)	High (N)	Low (N)	No Noise (N)
0	0.00000	-4.849845	1.076581	0.000000
1	0.02004	-1.574263	1.076581	0.400802
2	0.04008	-2.771360	1.076581	0.801603
3	0.06012	2.071162	1.076581	1.202405
4	0.08016	3.660872	1.076581	1.603206
..	…	…	…	…
495	9.91984	95.248337	95.648604	100.000000
496	9.93988	101.726636	95.648604	100.000000

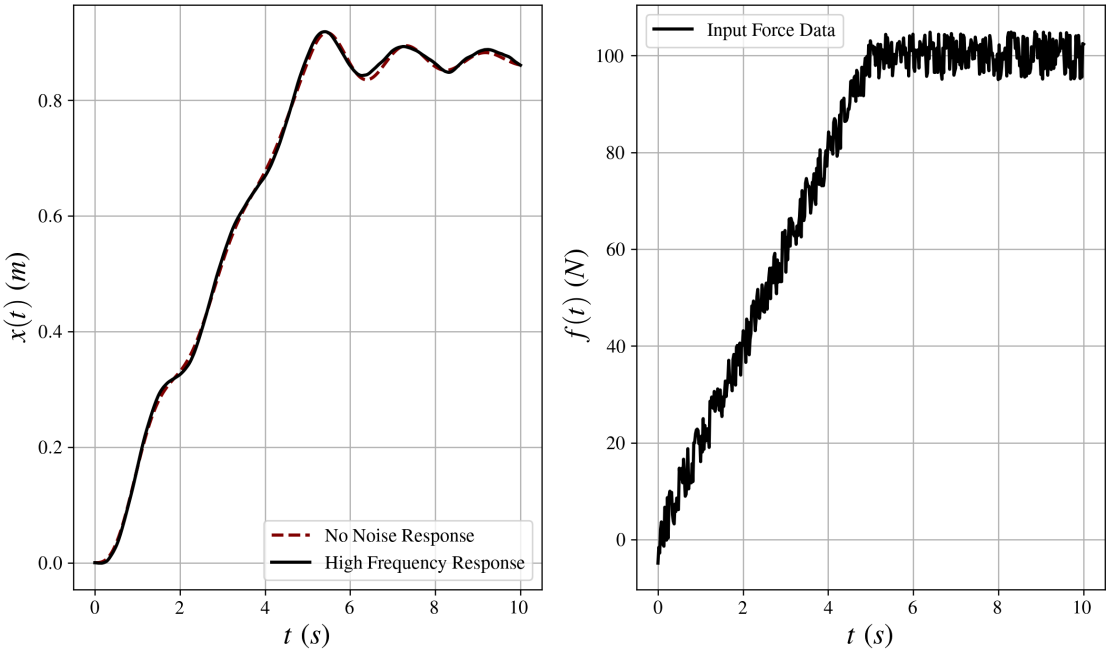
```
497    9.95992    95.527882  95.648604    100.000000
498    9.97996   102.458740  95.648604    100.000000
499   10.00000   102.326431  95.648604    100.000000

[500 rows x 4 columns]
```

[8]:
```
t, f_high, f_low, f_none = np.array(data['Time (s)']), np.array(data['High␣
 ↪(N)']), np.array(data['Low (N)']), np.array(data['No Noise (N)'])
_, x_none = ct.forced_response(sys1, T=t, U=f_none)

fig, ax = plt.subplots(nrows=1, ncols=2)
ax[0].set_xlabel('$t$ ($s$)')
ax[1].set_xlabel('$t$ ($s$)')
ax[0].set_ylabel('$x(t)$ ($m$)')
ax[1].set_ylabel('$f(t)$ ($N$)')

ax[1].plot(t, f_none, label='Input Force Data', color='black')
ax[0].plot(t, x_none, label='No Noise Response')

ax[0].legend()
ax[1].legend()
plt.show()
```



1.3.4 Part D

[9]:
```
_, x_high = ct.forced_response(sys1, T=t, U=f_high)

fig, ax = plt.subplots(nrows=1, ncols=2)
ax[0].set_xlabel('$t$ ($s$)')
ax[1].set_xlabel('$t$ ($s$)')
ax[0].set_ylabel('$x(t)$ ($m$)')
ax[1].set_ylabel('$f(t)$ ($N$)')

ax[1].plot(t, f_high, label='Input Force Data', color='black')
ax[0].plot(t, x_none, label='No Noise Response', ls='--')
ax[0].plot(t, x_high, label='High Frequency Response')

ax[0].legend()
ax[1].legend()
plt.show()
```



1.3.5 Part E

[10]:
```
_, x_low = ct.forced_response(sys1, T=t, U=f_low)

fig, ax = plt.subplots(nrows=1, ncols=2)
ax[0].set_xlabel('$t$ ($s$)')
ax[1].set_xlabel('$t$ ($s$)')
ax[0].set_ylabel('$x(t)$ ($m$)')
```

```
ax[1].set_ylabel('$f(t)$ ($N$)')

ax[1].plot(t, f_low, label='Input Force Data', color='black')
ax[0].plot(t, x_none, label='No Noise Response', ls='--')
ax[0].plot(t, x_low, label='Low Frequency Response')

ax[0].legend()
ax[1].legend()
plt.show()
```