Multiple Inputs

September 27, 2023

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

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

This is an extension to Example 2.5.1 in the book. This is how you get the response to a system that has multiple inputs. You add the forced response from each input function. For this example, the first input is $f(t) = t^2$, and the second input is $g(t) = e^{-5t}$.

[2]:
```
t = sp.Symbol('t')
x = sp.Function('x')(t)
f = t**2
g = sp.exp(-5*t)

eq = sp.Eq(5*x.diff(t, 2) + 30*x.diff() + 40*x, 6*f - 20*g)
eq
```

[2]:

$$40x(t) + 30\frac{d}{dt}x(t) + 5\frac{d^2}{dt^2}x(t) = 6t^2 - 20e^{-5t}$$

[3]:
```
sol = sp.dsolve(eq, ics={
    x.subs(t, 0): 0,
    x.diff().subs(t, 0): 0
})
sol
```

[3]:

$$x(t) = \frac{3t^2}{20} - \frac{9t}{40} + \frac{21}{160} - \frac{49e^{-2t}}{60} + \frac{323e^{-4t}}{160} - \frac{4e^{-5t}}{3}$$

[4]:
```
x_lamb = sp.lambdify(t, sol.rhs, modules='numpy')
t_ = np.linspace(0, 10, 500)

den = [5, 30, 40]  # all transfer functions have the same denominator␣
  ↪(characteristic equation)
X_F = ct.tf(6, den)
X_G = ct.tf(-20, den)
display(X_F, X_G)
```

$$\frac{6}{5s^2 + 30s + 40}$$

$$\frac{-20}{5s^2 + 30s + 40}$$

[5]:
```
_, f_response = ct.forced_response(X_F, T=t_, U=t_**2)
_, g_response = ct.forced_response(X_G, T=t_, U=np.exp(-5*t_))
response = f_response + g_response

plt.plot(t_, response, label='Control Package Result')
plt.plot(t_, x_lamb(t_), label='Analytical Result', ls='--', color='darkgrey')
plt.legend()
plt.show()
```