

1. Problem 3.8

$$\tau \frac{dy}{dt} + y(t) = ku(t)$$

$$u(t) = t\mathcal{H}(t) - t\mathcal{H}(t-2) + (-t+4)\mathcal{H}(t-2) - (-t+4)\mathcal{H}(t-4)$$

$$u(t) = t\mathcal{H}(t) - 2(t-2)\mathcal{H}(t-2) + (t-4)\mathcal{H}(t-4)$$

$$0.5 \cdot \frac{dy}{dt} + y(t) = t\mathcal{H}(t) - 2(t-2)\mathcal{H}(t-2) + (t-4)\mathcal{H}(t-4)$$

$$\mathcal{L} \left\{ 0.5 \cdot \frac{dy}{dt} + y(t) \right\} = \mathcal{L} \{ t\mathcal{H}(t) - 2(t-2)\mathcal{H}(t-2) + (t-4)\mathcal{H}(t-4) \}$$

$$0.5sY(s) + Y(s) = \frac{1}{s^2} - \frac{2e^{-2s}}{s^2} + \frac{e^{-4s}}{s^2}$$

$$Y(s) \left(\frac{s}{2} + 1 \right) = \frac{1}{s^2} - \frac{2e^{-2s}}{s^2} + \frac{e^{-4s}}{s^2}$$

$$\frac{1}{\left(\frac{s}{2} + 1 \right) s^2} = \frac{A}{s} + \frac{B}{s^2} + \frac{C}{\frac{s}{2} + 1}$$

$$1 = A \left(\frac{s^2}{2} + s \right) + B \left(\frac{s}{2} + 1 \right) + Cs^2$$

$$\frac{A}{2} + C = 0$$

$$A + \frac{B}{2} = 0$$

$$B = 1$$

$$A = -\frac{1}{2}$$

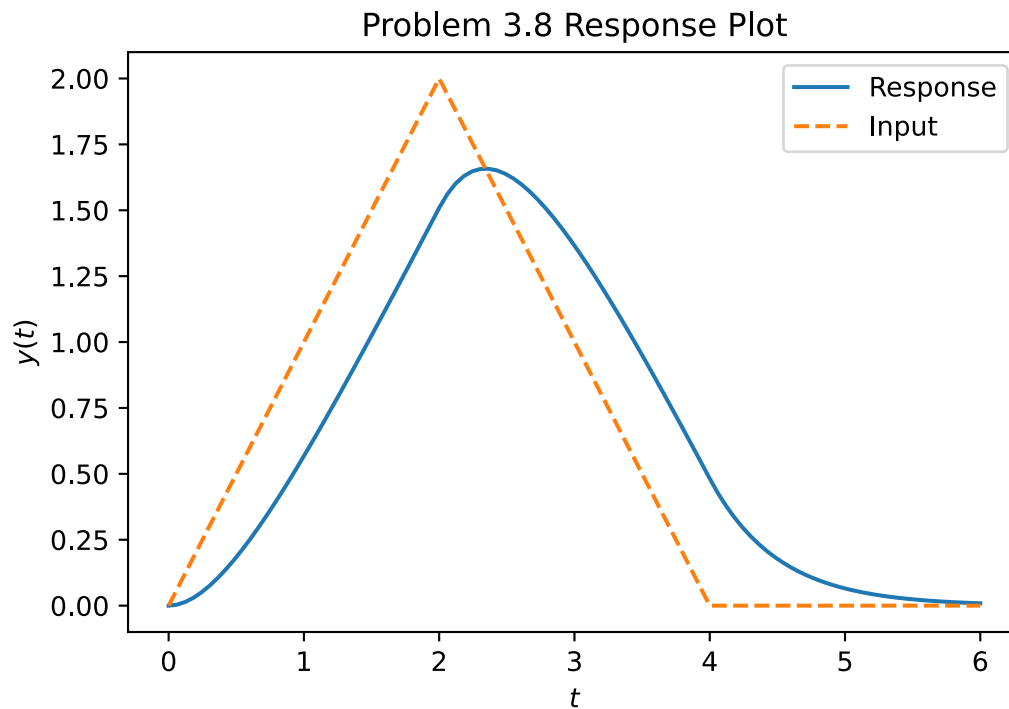
$$C = -\frac{1}{4}$$

$$\begin{aligned} Y(s) &= \frac{1}{2} \left(\frac{2}{s^2} - \frac{1}{s+2} - \frac{1}{s} \right) \\ &\quad - \frac{1}{2} \left(\frac{2}{s^2} - \frac{1}{s+2} - \frac{1}{s} \right) (2e^{-2s}) \\ &\quad + \frac{1}{2} \left(\frac{2}{s^2} - \frac{1}{s+2} - \frac{1}{s} \right) (e^{-4s}) \end{aligned}$$

Take the inverse Laplace Transform of the above equation.

$$y(t) = \frac{1}{2} \left[(2t - 9 + e^{8-2t}) \mathcal{H}(t-4) - 2(2t - 5 + e^{4-2t}) \mathcal{H}(t-2) + (2t - 1 + e^{-2t}) \mathcal{H}(t) \right]$$

Solution plot:



Code to produce the output plot:

```
import numpy as np
import matplotlib.pyplot as plt
from sympy.integrals import inverse_laplace_transform, laplace_transform
from sympy.abc import t, s
from sympy.functions import exp, Heaviside
from sympy import lambdify
# input a sympy expression
u_expr = t*Heaviside(t) - 2*(t-2)*Heaviside(t-2) + (t-4)*Heaviside(t-4)
# convert sympy expression u(t) into lambda func
u = lambdify(t, u_expr, 'numpy')
# lambdify won't work on Heaviside() with sympy version <1.11
# inverse laplace transform Y(s)
y_expr = inverse_laplace_transform((1 - 2*exp(-2*s) + exp(-4*s)) / s**2 / (s/2 + 1), s, t)
# convert sympy expression y(t) into lambda func
y = lambdify(t, y_expr, 'numpy')
t_range = np.linspace(0, 6, 100)
# plotting
plt.plot(t_range, y(t_range))
plt.plot(t_range, u(t_range), '--')
plt.xlabel(r'$t$')
plt.ylabel(r'$y(t)$')
plt.title(r'Problem 3.8 Response Plot')
plt.legend(['Response', 'Input'])
```

2. Problem 3.9

Response:

$$y(t) = \frac{kM}{\epsilon} \left(1 - e^{\frac{-t}{\tau}}\right)$$

Assume the sensor is perfectly calibrated:

$$k = 1$$

$$\epsilon = 1\text{min}$$

At $t = 1$ min:

$$T \approx 39^\circ\text{C}$$

$$T_0 = 20$$

$$\frac{M}{\epsilon} = 50 - 20 = 30$$

$$T = 19$$

$$19 = 30 \cdot \left(1 - e^{\frac{-1}{\tau}}\right)$$

$$\boxed{\tau = 0.997\text{min}}$$

3. Problem 3.10

Input:

$$T_L(t) = 20 + \sin(2t)$$

$$\omega = 2$$

$$M = 1$$

$$\text{output amplitude} = \frac{kM}{\sqrt{1 + (\omega\tau)^2}}$$

$$\text{output amplitude} = 0.1 \text{ from plot}$$

Assume the sensor is perfectly calibrated:

$$k = 1$$

$$0.1 = \frac{1}{\sqrt{1 + (2\tau)^2}}$$

Solve for τ :

$$\boxed{\tau = 4.975}$$

4. Problem 3.12

$$\tau \frac{dT}{dt} + T(t) = ku(t)$$

$$\tau = 10\text{s} = \frac{10\text{s}}{60\text{s}/\text{min}} = \frac{1}{6}\text{min}$$

Assume the sensor is perfectly calibrated:

$$k = 1$$

$$u(t) = 20 + 5e^{-t}$$

Make $T_0 = 20^\circ\text{C}$:

$$T(0) = 0, 0 \text{ initial condition}$$

$$u(t) = 5e^{-t}$$

$$\frac{1}{6} \frac{dT}{dt} + T(t) = 5e^{-t}$$

$$\mathcal{L} \left\{ \frac{1}{6} \frac{dT}{dt} + T(t) \right\} = \mathcal{L} \{ 5e^{-t} \}$$

$$\frac{sT(s)}{6} + T(s) = \frac{5}{s+1}$$

$$T(s) = \frac{5}{(s+1)(\frac{s}{6}+1)}$$

$$\frac{5}{(s+1)(\frac{s}{6}+1)} = \frac{A}{s+1} + \frac{B}{\frac{s}{6}+1}$$

$$5 = A \left(\frac{s}{6} + 1 \right) + B(s+1)$$

$$\frac{A}{6} + B = 0$$

$$A + B = 5$$

$$A = 6$$

$$B = -1$$

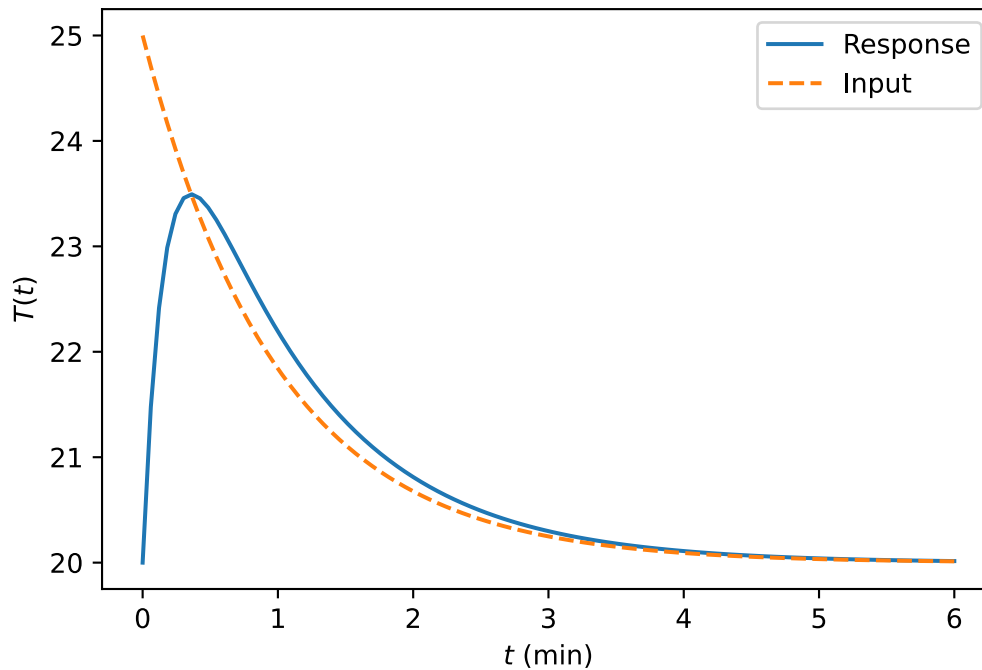
$$T(s) = \frac{6}{s+1} + \frac{-1}{\frac{s}{6}+1}$$

$$\mathcal{L}^{-1} \{ T(s) \} = \mathcal{L}^{-1} \left\{ \frac{6}{s+1} - \frac{6}{s+6} \right\}$$

$$T(t) = 6(e^{-t} - e^{-6t})$$

Solution plot:

Problem 3.12 Response Plot



```
import numpy as np
import matplotlib.pyplot as plt
from sympy.integrals import inverse_laplace_transform, laplace_transform
from sympy.abc import t, s
from sympy.functions import exp, Heaviside
from sympy import lambdify
# input function
u = lambda t: 20 + 5 * np.exp(-t)
# inverse laplace transform Y(s)
y_expr = inverse_laplace_transform(5 / (s + 1) / (s/6 + 1), s, t)
# convert sympy expression y(t) into lambda func
y = lambdify(t, y_expr, 'numpy')
# lambdify won't work on Heaviside() with sympy version <1.11
t_range_min = np.linspace(0, 6, 100)
# plotting
plt.plot(t_range_min, y(t_range_min) + 20) # +20 to add initial condition back
plt.plot(t_range_min, u(t_range_min), '--')
plt.xlabel(r'$t$ (min)')
plt.ylabel(r'$T(t)$')
plt.title(r'Problem 3.12 Response Plot')
plt.legend(['Response', 'Input'])
```