1. Problem 6.13

$$
\begin{aligned}
V\frac{dC_R}{dt} &= F(C_{R0}-C_R) - Vk_1C_R + Vk_2C_P - Vk_3C_R \\
V\frac{dC_P}{dt} &= -FC_P + Vk_1C_R - Vk_2C_P \\
\frac{dC_R}{dt} &= -\left(\frac{F}{V}+k_1+k_3\right)C_R + k_2C_P + \frac{F}{V}C_{R0} \\
\frac{dC_P}{dt} &= k_1C_R - \left(\frac{F}{V}+k_2\right)C_P \\
y &= C_P \\
A &= \begin{bmatrix} -\left(\frac{F}{V}+k_1+k_3\right) & k_2 \\ k_1 & -\left(\frac{F}{V}+k_2\right) \end{bmatrix} \\
B &= \begin{bmatrix} \frac{F}{V} \\ 0 \end{bmatrix} \\
C &= \begin{bmatrix} 0 & 1 \end{bmatrix} \\
D &= 0 \\
G(s) &= c\,(sI-A)^{-1}\,b + d \\
sI-A &= \begin{bmatrix} s+\left(\frac{F}{V}+k_1+k_3\right) & -k_2 \\ -k_1 & s+\left(\frac{F}{V}+k_2\right) \end{bmatrix} \\
\mathrm{Adj}(sI-A) &= \begin{bmatrix} s+\left(\frac{F}{V}+k_2\right) & k_2 \\ k_1 & s+\left(\frac{F}{V}+k_1+k_3\right) \end{bmatrix} \\
\det(sI-A) &= \left(s+\left(\frac{F}{V}+k_1+k_3\right)\right)\left(s+\left(\frac{F}{V}+k_2\right)\right) - k_1k_2 \\
\det(sI-A) &= \left(s+\frac{F}{V}+k_1+k_3\right)\left(s+\frac{F}{V}+k_2\right) - k_1k_2 \\
\theta &= \det(sI-A) \\
(sI-A)^{-1} &= \theta^{-1}\begin{bmatrix} s+\left(\frac{F}{V}+k_2\right) & k_2 \\ k_1 & s+\left(\frac{F}{V}+k_1+k_3\right) \end{bmatrix} \\
G(s) &= c\theta^{-1}\begin{bmatrix} s+\left(\frac{F}{V}+k_2\right) & k_2 \\ k_1 & s+\left(\frac{F}{V}+k_1+k_3\right) \end{bmatrix}\begin{bmatrix} \frac{F}{V} \\ 0 \end{bmatrix} \\
G(s) &= c\theta^{-1}\begin{bmatrix} \frac{F}{V}\left(s+\left(\frac{F}{V}+k_2\right)\right) \\ \frac{F}{V}k_1 \end{bmatrix} \\
G(s) &= \theta^{-1}\begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} \frac{F}{V}\left(s+\left(\frac{F}{V}+k_2\right)\right) \\ \frac{F}{V}k_1 \end{bmatrix} \\
G(s) &= \frac{\frac{F}{V}k_1}{\theta} \\
&\boxed{G(s) = \frac{\frac{F}{V}k_1}{\left(s+\frac{F}{V}+k_1+k_3\right)\left(s+\frac{F}{V}+k_2\right) - k_1k_2}}
\end{aligned}
$$

2. Problem 6.7

   (a)

   State space model:

$$A_1\frac{dh_1}{dt} = F_{in} - \frac{h_1 - h_2}{R_1}$$
$$A_2\frac{dh_2}{dt} = \frac{h_1 - h_2}{R_1} - \frac{h_2}{R_2}$$

   At steady state

$$0 = 1 - \frac{1.5 - 0.5}{R_1}$$
$$0 = \frac{1.5 - 0.5}{R_1} - \frac{0.5}{R_2}$$
$$R_1 = 1$$
$$R_2 = 0.5$$
$$\frac{dh}{dt} = \begin{bmatrix} -\frac{1}{A_1R_1} & \frac{1}{A_1R_1} \\ \frac{1}{A_2R_1} & -\left(\frac{1}{A_2R_1} + \frac{1}{A_2R_2}\right) \end{bmatrix} h + \begin{bmatrix} \frac{1}{A_1} \\ 0 \end{bmatrix} F_{in}$$
$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} h$$
$$A = \begin{bmatrix} -1 & 1 \\ 0.5 & -1.5 \end{bmatrix}$$
$$B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$
$$h(0) = \begin{bmatrix} 1.5 \\ 0.5 \end{bmatrix}$$

   Simulate response with lsim

$$D = 0$$

   For h_1 response

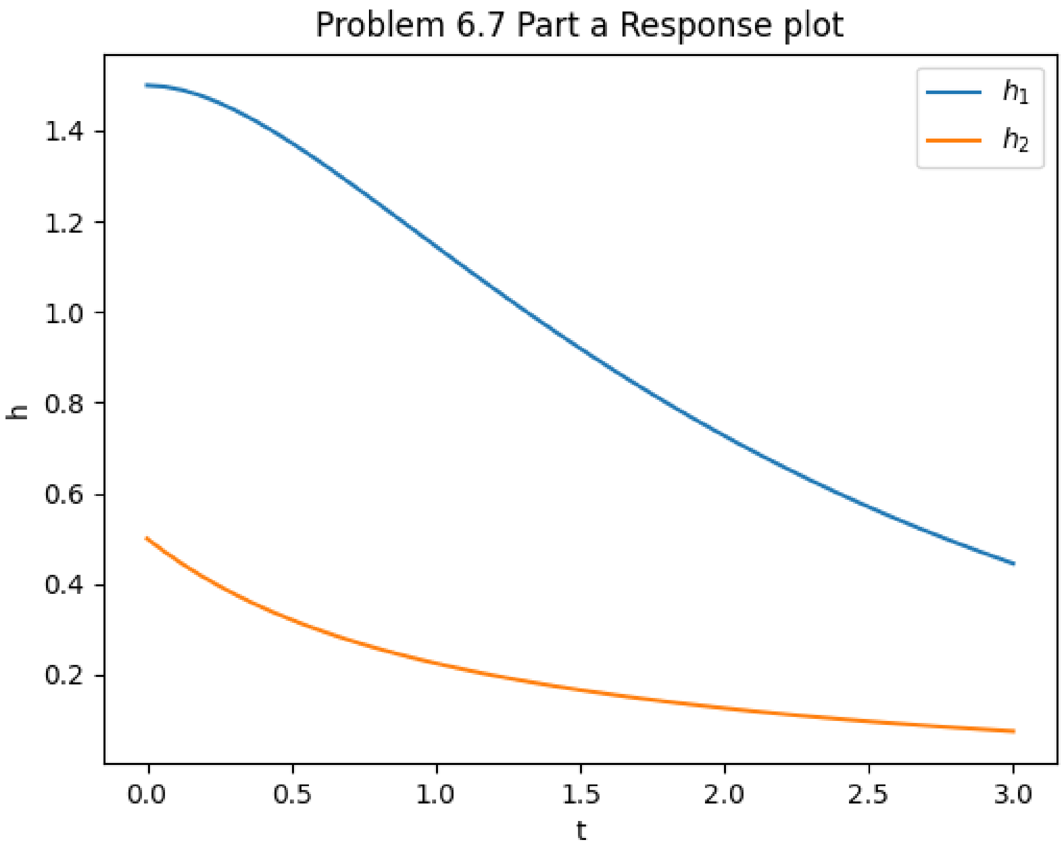
$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

   For h_2 response

$$C = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

   Simulate a response to an unforced input on the system. The initil heights are supplied as the initial condition.

   Simulated response plot:



The plot was generated with the code below:

```
import numpy as np
import matplotlib.pylab as plt
from scipy.signal import lti, step, lsim

h_1_0 = 1.5
h_2_0 = 0.5

A = np.array([
    [-1, 1],
    [.5, -1.5]
])
B = np.array([
    [1],
    [0]
])
C_h1 = np.array([1, 0])
C_h2 = np.array([0, 1])
D = 0

sys_h1 = lti(A, B, C_h1, D)
sys_h2 = lti(A, B, C_h2, D)
```

```python
# Part A
u = 0
t = np.linspace(0, 3, 50)

t_h1_a, h1_a, x1 = lsim(sys_h1, u, t, h_1_0)
t_h2_a, h2_a, x2 = lsim(sys_h2, u, t, h_2_0)

plt.plot(t_h1_a, h1_a, label=r"$h_1$")
plt.plot(t_h2_a, h2_a, label=r"$h_2$")
plt.xlabel(r"t")
plt.ylabel(r"h")
plt.title("Problem 6.7 Part a Response plot")
plt.legend()
plt.show()

# Part B
t_react = np.interp(0.2, np.flip(h2_a), np.flip(t_h2_a))
h_1_react = np.interp(t_react, t_h1_a, h1_a)
print(f"Time when h_2 = 0.2 m: {t_react} min")
print(f"h_1 at that time: {h_1_react}")

# Part C
t_h1_c, h1_c = step(sys_h1, X0=h_1_react)
t_h2_c, h2_c = step(sys_h2, X0=0.2)

plt.plot(t_h1_c, h1_c, label=r"$h_1$")
plt.plot(t_h2_c, h2_c, label=r"$h_2$")
plt.xlabel(r"t")
plt.ylabel(r"h")
plt.title("Problem 6.7 Part c Response plot")
plt.legend()
plt.show()

print(f"Final h_1 = {h1_c[-1]}")
print(f"Final h_2 = {h2_c[-1]}")
```

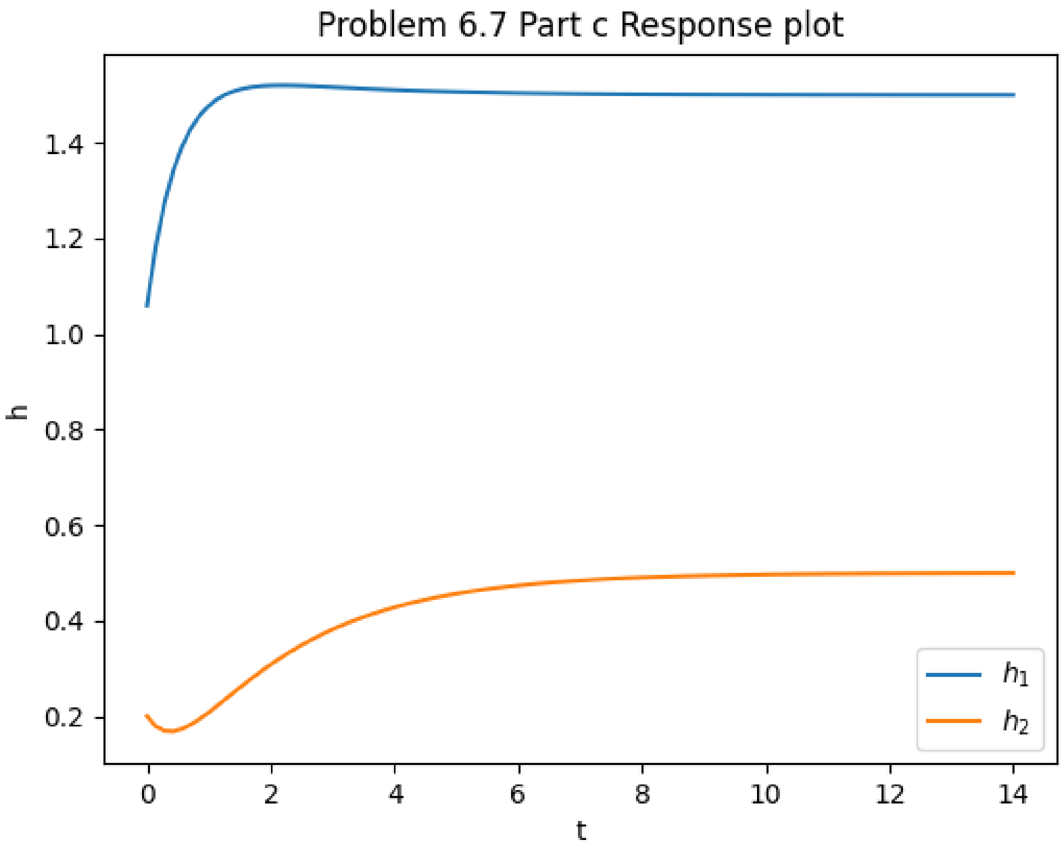
(b) To find the the time that $h_2 = 0.2$ m, the output arrays for the h_2 response were interpolated. With the time value, h_1 at that time was also interpolated from the h_1 response arrays. The code above shows the interpolation and output.

h_2 reaches 0.2 m at $\boxed{1.1845\text{min}}$.

At that time, h_1 is $\boxed{1.0594\text{m}}$.

(c) To find the response of the heights when the flow is restored, another response simulation was made. The flow switching on is equivalent to a step response, and so a step response to the system will be simulated. The code above also contains the code for this simulation.

Response plot:



h_1 ends up at 1.5 m, and h_2 ends up at 0.5 m. Both heights return to their steady state values.

3. Problem 6.8

(a)

$$
\begin{aligned}
\tau\frac{dx_1}{dt} + x_1 &= ku \\
\tau\frac{dx_2}{dt} + x_2 &= kx_1 \\
\frac{dx_1}{dt} &= -\frac{x_1}{\tau} + \frac{k}{\tau}u \\
\frac{dx_2}{dt} &= \frac{k}{\tau}x_1 - \frac{x_2}{\tau} \\
A &= \begin{bmatrix} -\frac{1}{\tau} & 0 \\ \frac{k}{\tau} & -\frac{1}{\tau} \end{bmatrix} \\
B &= \begin{bmatrix} \frac{k}{\tau} \\ 0 \end{bmatrix} \\
sI - A &= \begin{bmatrix} s + \frac{1}{\tau} & 0 \\ -\frac{k}{\tau} & s + \frac{1}{\tau} \end{bmatrix} \\
\mathrm{Adj}(sI - A) &= \begin{bmatrix} s + \frac{1}{\tau} & 0 \\ \frac{k}{\tau} & s + \frac{1}{\tau} \end{bmatrix} \\
\det(sI - A) &= (s + \frac{1}{\tau})^2 \\
(sI - A)^{-1} &= \begin{bmatrix} \frac{1}{s+1/\tau} & 0 \\ \frac{\tau k}{\tau^2 s^2 + 2\tau s + 1} & \frac{1}{s+1/\tau} \end{bmatrix} \\
\mathcal{L}^{-1}\left\{(sI - A)^{-1}\right\} &= \begin{bmatrix} e^{-t/\tau} & 0 \\ \frac{k}{\tau}te^{-t/\tau} & e^{-t/\tau} \end{bmatrix} \\
e^{At} &= \begin{bmatrix} e^{-t/\tau} & 0 \\ \frac{k}{\tau}te^{-t/\tau} & e^{-t/\tau} \end{bmatrix}
\end{aligned}
$$

(b)

$$
\begin{aligned}
x(t) &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} e^{-t/\tau} & 0 \\ \frac{k}{\tau}te^{-t/\tau} & e^{-t/\tau} \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} \\
x(t) &= \begin{bmatrix} e^{-t/\tau}x_1(0) \\ \frac{k}{\tau}te^{-t/\tau}x_1(0) + e^{-t/\tau}x_2(0) \end{bmatrix}
\end{aligned}
$$

(c)

$$A_d = \begin{bmatrix} e^{-T_s/\tau} & 0 \\ \frac{k}{\tau} T_s e^{-T_s/\tau} & e^{-T_s/\tau} \end{bmatrix}$$

$$B_d = \int_0^{T_s} e^{At} B dt$$

$$e^{At} B = \begin{bmatrix} e^{-t/\tau} & 0 \\ \frac{k}{\tau} t e^{-t/\tau} & e^{-t/\tau} \end{bmatrix} \begin{bmatrix} \frac{k}{\tau} \\ 0 \end{bmatrix}$$

$$e^{At} B = \begin{bmatrix} \frac{k}{\tau} e^{-t/\tau} \\ \frac{k^2}{\tau^2} t e^{-t/\tau} \end{bmatrix}$$

$$B_d = \int_0^{T_s} \begin{bmatrix} \frac{k}{\tau} e^{-t/\tau} \\ \frac{k^2}{\tau^2} t e^{-t/\tau} \end{bmatrix} dt$$

$$B_d = \begin{bmatrix} -k e^{-t/\tau} \\ \frac{k^2}{\tau} \left(-\tau t e^{-t/\tau} - \tau^2 e^{-t/\tau} \right) \end{bmatrix}$$

4. MATLAB Problem

(a)

$$\frac{d}{dt}\begin{bmatrix} C_R \\ C_I \\ C_P \end{bmatrix} = \begin{bmatrix} -\left(\frac{F}{V}+k_1\right) & 0 & 0 \\ k_1 & -\left(\frac{F}{V}+k_2\right) & k_3 \\ 0 & k_2 & -\left(\frac{F}{V}+k_3\right) \end{bmatrix} \begin{bmatrix} C_R \\ C_I \\ C_P \end{bmatrix} + \begin{bmatrix} \frac{F}{V} \\ 0 \\ 0 \end{bmatrix} C_{R0}$$

$$k_1 = 0.1$$
$$k_2 = 0.05$$
$$k_3 = 0.02$$
$$F = 10$$
$$V = 100$$
$$\frac{F}{V} = 0.1$$
$$C_{R0,s} = 2$$

$$A = \begin{bmatrix} -(0.1+0.1) & 0 & 0 \\ 0.1 & -(0.1+0.05) & 0.02 \\ 0 & 0.5 & -(0.1+0.02) \end{bmatrix}$$

$$A = \begin{bmatrix} -0.2 & 0 & 0 \\ 0.1 & -0.15 & 0.02 \\ 0 & 0.5 & -0.12 \end{bmatrix}$$

$$Bu = \begin{bmatrix} 0.1 \\ 0 \\ 0 \end{bmatrix} \cdot 2$$

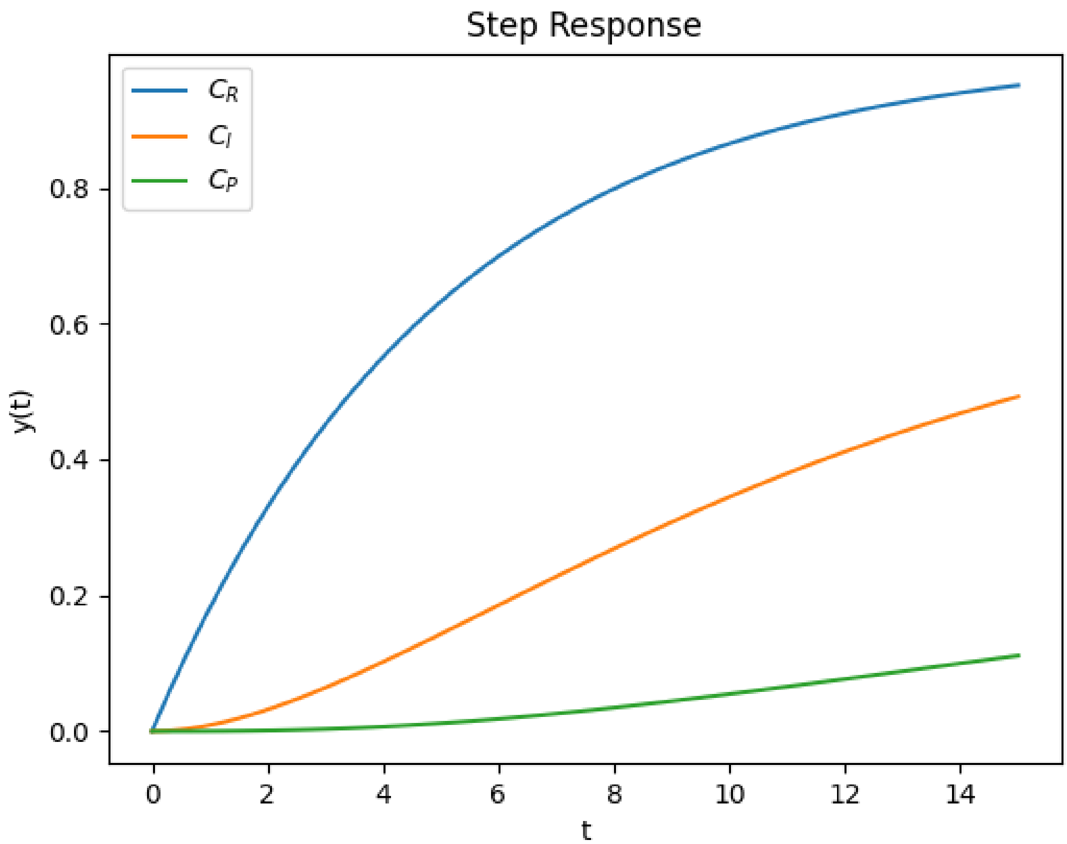
$$Bu = \begin{bmatrix} 0.2 \\ 0 \\ 0 \end{bmatrix}$$

At steady state:

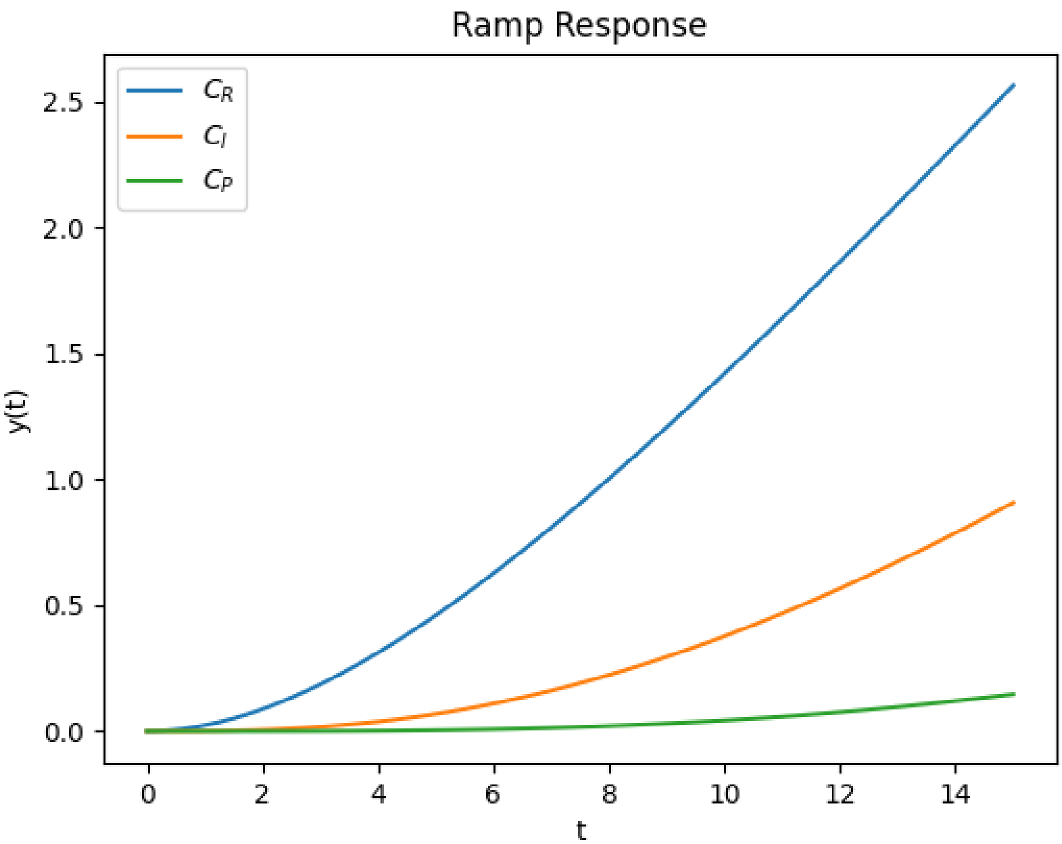
$$0 = Ax + Bu$$
$$x = -A^{-1}Bu$$
$$x = -\begin{bmatrix} -0.2 & 0 & 0 \\ 0.1 & -0.15 & 0.02 \\ 0 & 0.5 & -0.12 \end{bmatrix}^{-1} \begin{bmatrix} 0.2 \\ 0 \\ 0 \end{bmatrix}$$
$$x = \begin{bmatrix} 1 \\ 0.7059 \\ 0.2941 \end{bmatrix}$$
$$C_R = 1$$
$$C_I = 0.7059$$
$$C_P = 0.2941$$

(b) Response plots:

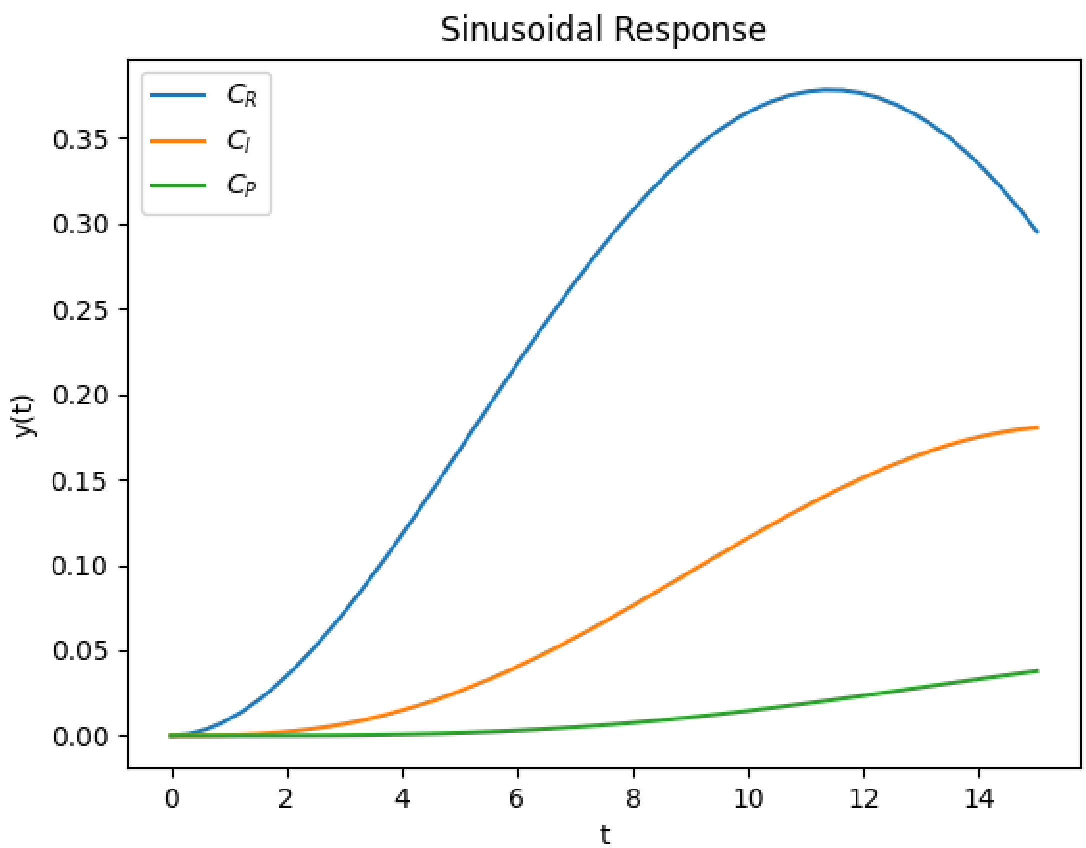
Step response:



Ramp response:



Sinusoidal response:



Code for creating the plots and calculating zero hold disretization:

```
import numpy as np
from scipy import linalg, signal
import matplotlib.pylab as plt
import math

import pprint


def steady_state_composition(A, B) -> None:
    C_ROs = 2

    Bu = B * C_ROs

    steady_state_solution = -linalg.solve(A, Bu)

    print(steady_state_composition)


def step_response(t, M, A, B, c):
    step_response_formula = lambda t: M * (c @ (linalg.expm(A * t) - np.eye(3)) @
    ↪  linalg.inv(A) @ B)
    response = np.zeros(len(t))
```

```
    for i, val in enumerate(t):
        response[i] = step_response_formula(val)[0]

    return response


def ramp_response(t, M, A, B, c):
    ramp_response_formula = lambda t: M * (c @ ((linalg.expm(A * t) - np.eye(3)) @
    ↪  linalg.inv(A) - t * np.eye(3)) @ linalg.inv(A) @ B)
    response = np.zeros(len(t))

    for i, val in enumerate(t):
        response[i] = ramp_response_formula(val)[0]

    return response


def sinusoidal_response(t, M, omega, A, B, c):
    sinusoidal_response_formula = lambda t: c @ (omega * linalg.expm(A * t) - A *
    ↪  math.sin(omega * t) - omega * np.eye(3) * math.cos(omega * t)) @
    ↪  linalg.inv(np.linalg.matrix_power(A, 2) + omega**2 * np.eye(3)) @ B * M
    response = np.zeros(len(t))

    for i, val in enumerate(t):
        response[i] = sinusoidal_response_formula(val)[0]

    return response

def plot_step_response(A, B):
    M = 2

    c_C_R = np.array([1, 0, 0])
    c_C_I = np.array([0, 1, 0])
    c_C_P = np.array([0, 0, 1])

    t_plot = np.linspace(0, 15, 50)

    plt.plot(t_plot, step_response(t_plot, M, A, B, c_C_R))
    plt.plot(t_plot, step_response(t_plot, M, A, B, c_C_I))
    plt.plot(t_plot, step_response(t_plot, M, A, B, c_C_P))
    plt.title("Step Response")
    plt.xlabel("t")
    plt.ylabel("y(t)")
    plt.legend([r"$C_R$", r"$C_I$", r"$C_P$"])
    plt.show()
```

```python
def plot_ramp_response(A, B):
    M = 0.5

    c_C_R = np.array([1, 0, 0])
    c_C_I = np.array([0, 1, 0])
    c_C_P = np.array([0, 0, 1])

    t_plot = np.linspace(0, 15, 50)

    plt.plot(t_plot, ramp_response(t_plot, M, A, B, c_C_R))
    plt.plot(t_plot, ramp_response(t_plot, M, A, B, c_C_I))
    plt.plot(t_plot, ramp_response(t_plot, M, A, B, c_C_P))
    plt.title("Ramp Response")
    plt.xlabel("t")
    plt.ylabel("y(t)")
    plt.legend([r"$C_R$", r"$C_I$", r"$C_P$"])
    plt.show()


def plot_sinusoidal_response(A, B):
    M = 1
    omega = 0.2

    c_C_R = np.array([1, 0, 0])
    c_C_I = np.array([0, 1, 0])
    c_C_P = np.array([0, 0, 1])

    t_plot = np.linspace(0, 15, 50)

    plt.plot(t_plot, sinusoidal_response(t_plot, M, omega, A, B, c_C_R))
    plt.plot(t_plot, sinusoidal_response(t_plot, M, omega, A, B, c_C_I))
    plt.plot(t_plot, sinusoidal_response(t_plot, M, omega, A, B, c_C_P))
    plt.title("Sinusoidal Response")
    plt.xlabel("t")
    plt.ylabel("y(t)")
    plt.legend([r"$C_R$", r"$C_I$", r"$C_P$"])
    plt.show()


def main():
    k_1 = 0.1
    k_2 = 0.05
    k_3 = 0.02
    F = 10
    V = 100

    A = np.array([
        [-(F / V + k_1), 0, 0],
```

```
        [k_1, -(F / V + k_2), k_3],
        [0, k_2, -(F / V + k_3)],
    ])

    B = np.array([
        [F/V],
        [0],
        [0],
    ])

    steady_state_composition(A, B)


    plot_step_response(A, B)


    plot_ramp_response(A, B)


    plot_sinusoidal_response(A, B)

    C = np.array([1, 1, 1])
    D = 0
    sys = signal.lti(A, B, C, D)

    pp = pprint.PrettyPrinter(indent=4)

    pp.pprint(signal.cont2discrete((A, B, C, D), 0.2))

if __name__ == "__main__":
    main()
```

(c) Zero hold discretization was calulated with the code in the previous part.

$$A_d = \begin{bmatrix} 0.960789439 & 0 & 0 \\ 0.0193123181 & 0.970464981 & 3.89347676 \cdot 10^{-5} \\ 9.69160980 \cdot 10^{-5} & 9.73369191 \cdot 10^{-3} & 0.976305197 \end{bmatrix}$$

$$B_d = \begin{bmatrix} 0.0196052804 \\ 1.95395071 \cdot 10^{-4} \\ 6.51198080 \cdot 10^{-7} \end{bmatrix}$$