

1. Problem 4.8

(a)

$$Y_1(s) = \frac{k_1}{\tau_1 s + 1} U(s)$$

$$Y_2(s) = \frac{-k_2}{\tau_2 s + 1} U(s)$$

$$Y(s) = \left( \frac{k_1}{\tau_1 s + 1} - \frac{k_2}{\tau_2 s + 1} \right) U(s)$$

$$U(s) = \frac{M}{s}$$

$$\mathcal{L}^{-1}\{Y(s)\} = \mathcal{L}^{-1}\left\{\left(\frac{k_1}{\tau_1 s + 1} - \frac{k_2}{\tau_2 s + 1}\right) \frac{M}{s}\right\}$$

$$y(t) = k_1 M (1 - e^{-t/\tau_1}) - k_2 M (1 - e^{-t/\tau_2})$$

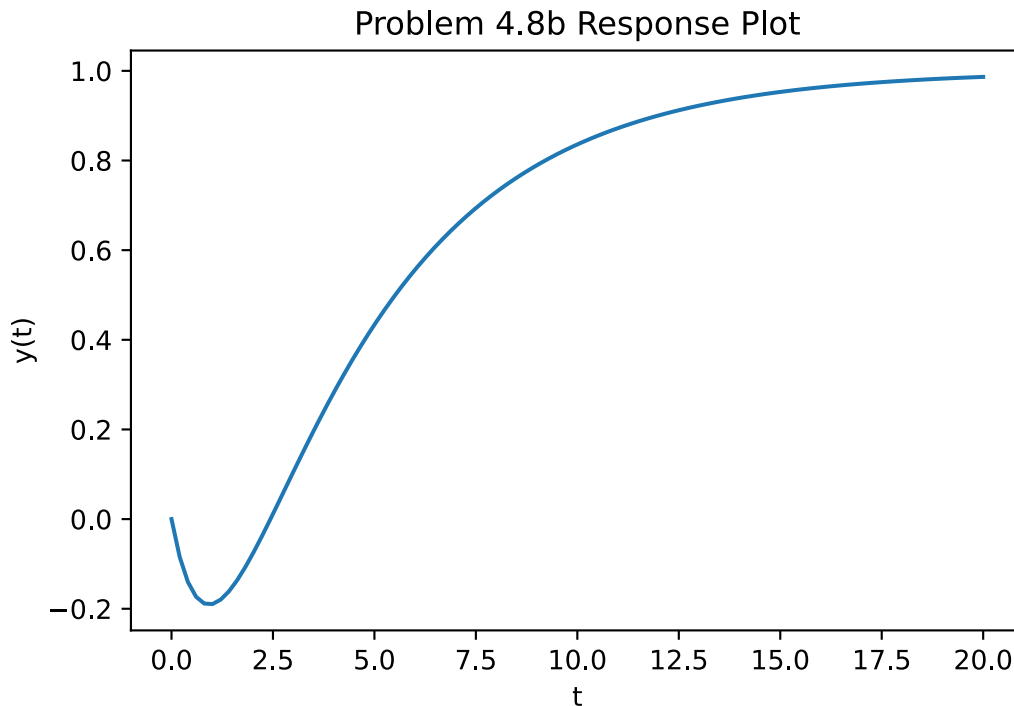
$$\frac{dy}{dt} = \frac{k_1 M}{\tau_1} e^{-t/\tau_1} - \frac{k_2 M}{\tau_2} e^{-t/\tau_2}$$

$$\frac{dy}{dt}(0) = \frac{k_1 M}{\tau_1} - \frac{k_2 M}{\tau_2}$$

As  $t \rightarrow \infty$

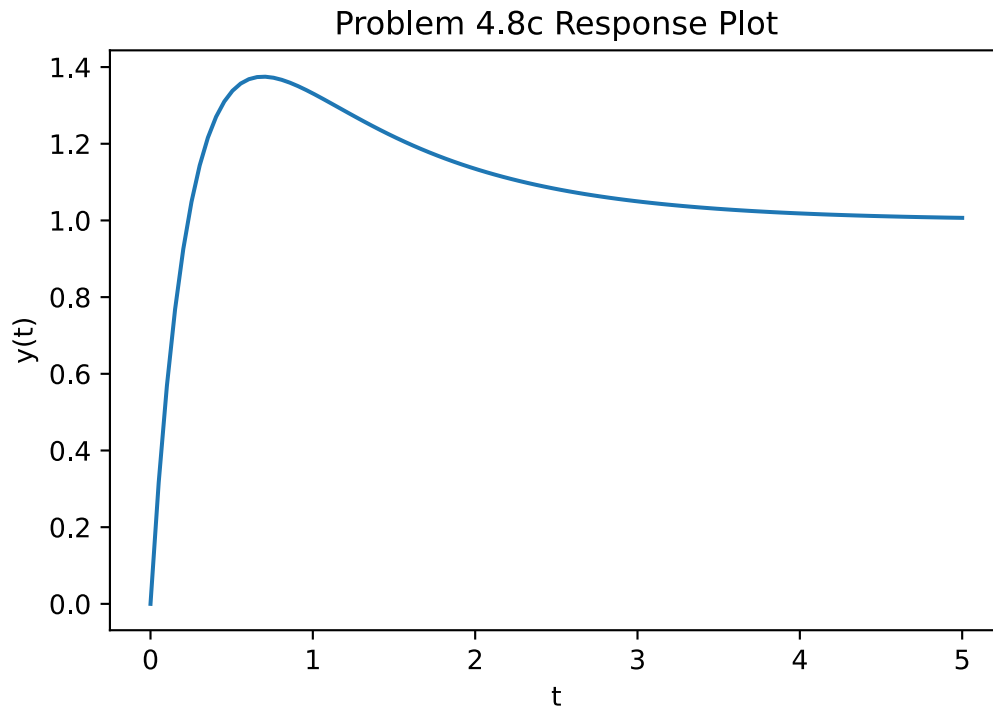
$$y(t) = \frac{k_1 M}{\tau_1} - \frac{k_2 M}{\tau_2}$$

(b) Plot:



Initially, the second block's output is larger than the first block's output. As a result, the second block's output pulls the combined output negative. The first output eventually outpaces the second output, and the output reaches a new higher steady state.

(c) Plot:



A similar situation to that of part b occurs in part c. Except, the first output is initially larger than the second output. The combined output reaches a peak before the second output matches the first output and pulls the combined output to a new steady state, lower than the peak but higher than the initial.

Plotting code:

```
import numpy as np
import matplotlib.pyplot as plt

k_1 = 2
tau_1 = 4
k_2 = 1
tau_2 = 1
M = 1

func = lambda t: k_1 * M * (1 - np.exp(-t / tau_1)) - k_2 * M * (1 - np.exp(-t / tau_2))

t_ran = np.linspace(0, 20, 100)
plt.plot(t_ran, func(t_ran))
plt.xlabel(r"t")
plt.ylabel(r"y(t)")
plt.title("Problem 4.8b Response Plot")

k_1 = 2
tau_1 = 1/4
k_2 = 1
tau_2 = 1
```

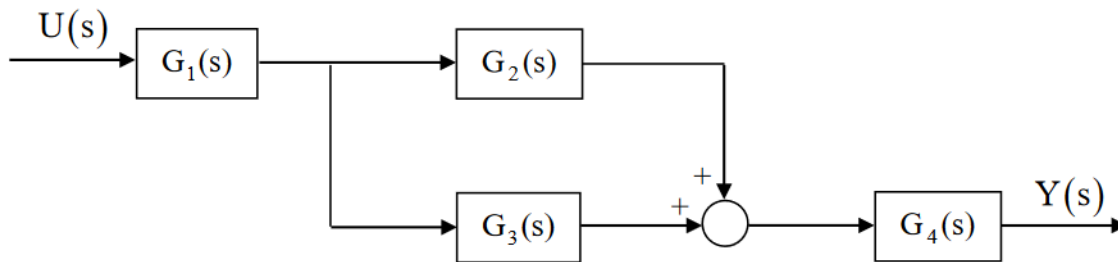
```
M = 1

func = lambda t: k_1 * M * (1 - np.exp(-t / tau_1)) - k_2 * M * (1 - np.exp(-t / tau_2))

t_ran = np.linspace(0, 5, 100)
plt.plot(t_ran, func(t_ran))
plt.xlabel(r"t")
plt.ylabel(r"y(t)")
plt.title("Problem 4.8c Response Plot")
```

## 2. Problem 2

Calculate the overall transfer function of the system represented by the following block diagram:



$$Y_1(s) = G_1(s)U(s)$$

$$Y_2(s) = (G_2(s) + G_3(s))U_2(s)$$

$$Y_2(s) = (G_2(s) + G_3(s)) G_1(s)U(s)$$

$$Y(s) = G_4(s)U_3(s)$$

$$Y(s) = G_4(s) (G_2(s) + G_3(s)) G_1(s)U(s)$$

$$\boxed{Y(s) = (G_2(s) + G_3(s)) G_1(s)G_4(s)U(s)}$$

3. Problem 5.8

$$kM \approx 38$$

$$A \approx 47 - 38 = 9$$

$$\frac{A}{kM} = \exp\left(-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}\right)$$

$$\frac{9}{38} = \exp\left(-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}\right)$$

$$\boxed{\zeta = 0.4168}$$

$$T \approx 20$$

$$T = \frac{2\pi\tau}{\sqrt{1-\zeta^2}}$$

$$20 = \frac{2\pi\tau}{\sqrt{1-0.4168^2}}$$

$$\boxed{\tau = 2.893}$$

$$M = 0.5 \cdot 76 = 38$$

$$\boxed{k = 1}$$

4. Problem 4.2

(a) State space:

$$V \frac{dC_R}{dt} = FC_{in,R} - FC_R - Vk_1 C_R$$

$$V \frac{dC_P}{dt} = Vk_1 C_R - FC_P - Vk_2 C_P$$

(b)

$$\mathcal{L} \left\{ V \frac{dC_R}{dt} \right\} = \mathcal{L} \{ FC_{in,R} - FC_R - Vk_1 C_R \}$$

$$VC_R(s)s = FC_{in,R}(s) - FC_R(s) - Vk_1 C_R(s)$$

$$C_R(s) = \frac{F}{Vs + F + Vk_1} C_{in,R}(s)$$

$$\boxed{G_1(s) = \frac{F}{Vs + F + Vk_1}}$$

$$\mathcal{L} \left\{ V \frac{dC_P}{dt} \right\} = \mathcal{L} \{ Vk_1 C_R - FC_P - Vk_2 C_P \}$$

$$VC_P(s)s = Vk_1 C_R(s) - FC_P(s) - Vk_2 C_P(s)$$

$$C_P(s)(Vs + F + Vk_2) = Vk_1 C_R(s)$$

$$C_P(s)(Vs + F + Vk_2) = \frac{FVk_1}{Vs + F + Vk_1} C_{in,R}(s)$$

$$C_P(s) = \frac{FVk_1}{(Vs + F + Vk_1)(Vs + F + Vk_2)} C_{in,R}(s)$$

$$\boxed{G_2(s) = \frac{FVk_1}{(Vs + F + Vk_1)(Vs + F + Vk_2)}}$$

(c) Partial fraction decomposition:

$$\begin{aligned}
\frac{1}{(Vs + F + Vk_1)(Vs + F + Vk_2)} &= \frac{A}{Vs + F + Vk_1} + \frac{B}{Vs + F + Vk_2} \\
1 &= A(Vs + F + Vk_2) + B(Vs + F + Vk_1) \\
1 &= s(AV + BV) + (AF + AVk_2 + BF + BVk_1) \\
AV + BV &= 0 \\
B &= -A \\
AF + AVk_2 + BF + BVk_1 &= 1 \\
AF + AVk_2 - AF - AVk_1 &= 1 \\
AV(k_2 - k_1) &= 1 \\
A &= \frac{1}{V(k_2 - k_1)} \\
B &= -\frac{1}{V(k_2 - k_1)} \\
G(s) &= \frac{FVk_1}{V(k_2 - k_1)} \left( \frac{1}{Vs + F + Vk_1} - \frac{1}{Vs + F + Vk_2} \right) \\
G(s) &= \frac{Fk_1}{V(k_2 - k_1)} \left( \frac{1}{s + \frac{F}{V} + k_1} - \frac{1}{s + \frac{F}{V} + k_2} \right)
\end{aligned}$$

Impulse input:

$$\begin{aligned}
C_{in,R} &= \frac{n_R}{V} \delta(t) \\
C_{in,R}(s) &= \frac{n_R}{V} \\
\mathcal{L}^{-1} \{C_P(s)\} &= \mathcal{L}^{-1} \left\{ \frac{Fk_1}{V(k_2 - k_1)} \left( \frac{1}{s + \frac{F}{V} + k_1} - \frac{1}{s + \frac{F}{V} + k_2} \right) \frac{n_R}{V} \right\} \\
\boxed{C_P(t) &= \frac{Fk_1 n_R}{V^2(k_2 - k_1)} \left( e^{-(\frac{F}{V} + k_1)t} - e^{-(\frac{F}{V} + k_2)t} \right)} \\
C_R(s) &= \frac{F}{Vs + F + Vk_1} \frac{n_R}{V} \\
\mathcal{L}^{-1} \{C_R(s)\} &= \mathcal{L}^{-1} \left\{ \frac{Fn_R}{V^2} \frac{1}{s + \frac{F}{V} + k_1} \right\} \\
\boxed{C_R(t) &= \frac{Fn_R}{V^2} e^{-(\frac{F}{V} + k_1)t}}
\end{aligned}$$

(d) Reversible

State Space:

$$\begin{aligned}
V \frac{dC_R}{dt} &= FC_{in,R} - FC_R - Vk_1 C_R + Vk_{-1} C_P \\
V \frac{dC_P}{dt} &= Vk_1 C_R - FC_P - Vk_2 C_P - Vk_{-1} C_P
\end{aligned}$$

Transfer Function:

$$\begin{aligned}
\mathcal{L}\left\{V\frac{dC_R}{dt}\right\} &= \mathcal{L}\{FC_{in,R} - FC_R - Vk_1C_R + Vk_{-1}C_P\} \\
VsC_R(s) &= FC_{in,R}(s) - C_R(s)(F + Vk_1) + Vk_{-1}C_P(s) \\
C_R(s)(Vs + F + Vk_1) &= FC_{in,R}(s) + Vk_{-1}C_P(s) \\
C_R(s) &= \frac{F}{Vs + F + Vk_1}C_{in,R}(s) + \frac{Vk_{-1}}{Vs + F + Vk_1}C_P(s) \\
\mathcal{L}\left\{V\frac{dC_P}{dt}\right\} &= \mathcal{L}\{Vk_1C_R - FC_P - Vk_2C_P - Vk_{-1}C_P\} \\
VsC_P(s) &= Vk_1C_R(s) - FC_P(s) - Vk_2C_P(s) - Vk_{-1}C_P(s) \\
C_P(s)(Vs + F + Vk_2 + Vk_{-1}) &= Vk_1C_R(s) \\
C_P(s)(Vs + F + Vk_2 + Vk_{-1}) &= \frac{FVk_1}{Vs + F + Vk_1}C_{in,R}(s) + \frac{V^2k_{-1}k_1}{Vs + F + Vk_1}C_P(s) \\
\frac{FVk_1}{Vs + F + Vk_1}C_{in,R}(s) &= C_P(s)\left((Vs + F + Vk_2 + Vk_{-1}) - \frac{V^2k_{-1}k_1}{Vs + F + Vk_1}\right) \\
(FVk_1)C_{in,R}(s) &= C_P(s)\left((Vs + F + Vk_2 + Vk_{-1})(Vs + F + Vk_1) - V^2k_{-1}k_1\right) \\
C_P(s) &= \frac{FVk_1}{(Vs + F + Vk_2 + Vk_{-1})(Vs + F + Vk_1) - V^2k_{-1}k_1}C_{in,R}(s) \\
G(s) &= \frac{FVk_1}{(Vs + F + Vk_2 + Vk_{-1})(Vs + F + Vk_1) - V^2k_{-1}k_1}
\end{aligned}$$



(Matlab)

$$V \frac{dC_R}{dt} = FC_{in,R} - FC_R - V k_1 C_R + V k_{-1} C_P$$

$$V \frac{dC_P}{dt} = V k_1 C_R - FC_P - V k_2 C_P - V k_{-1} C_P$$

$$\frac{dC_R}{dt} = \frac{F}{V} C_{in,R} - \frac{F}{V} C_R - k_1 C_R + k_{-1} C_P$$

$$\frac{dC_P}{dt} = k_1 C_R - \frac{F}{V} C_P - k_2 C_P - k_{-1} C_P$$

$$\frac{dx_1}{dt} = a_{11} x_1 + a_{12} x_2 + b_1 u$$

$$\frac{dx_2}{dt} = a_{21} x_1 + a_{22} x_2 + b_2 u$$

$$y = c_1 x_1 + c_2 x_2 + du$$

$$\frac{dC_R}{dt} = - \left( \frac{F}{V} + k_1 \right) C_R + k_{-1} C_P + \frac{F}{V} C_{in,R}$$

$$\frac{dC_P}{dt} = - \left( \frac{F}{V} + k_2 + k_{-1} \right) C_P + k_1 C_R$$

$$C_P = C_P$$

$$\frac{d}{dt} \begin{bmatrix} C_R \\ C_P \end{bmatrix} = \begin{bmatrix} - \left( \frac{F}{V} + k_1 \right) & k_{-1} \\ k_1 & - \left( \frac{F}{V} + k_2 + k_{-1} \right) \end{bmatrix} \begin{bmatrix} C_R \\ C_P \end{bmatrix} + \begin{bmatrix} \frac{F}{V} \\ 0 \end{bmatrix} C_{in,R}$$

$$C_P = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} C_R \\ C_P \end{bmatrix} + 0 \cdot du$$

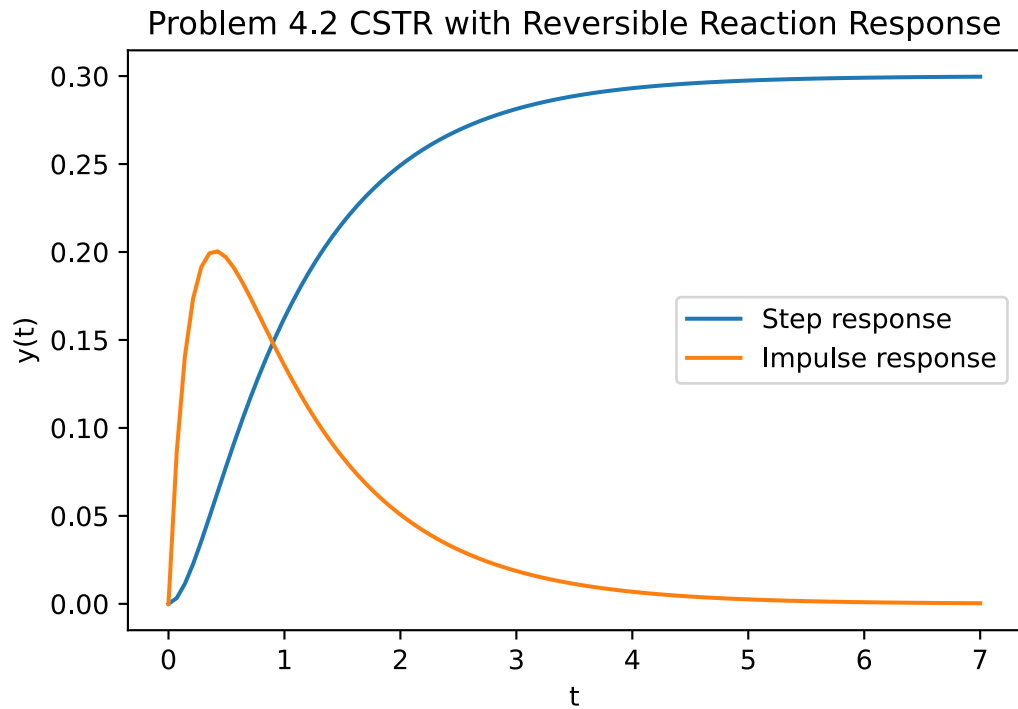
$$\frac{F}{V} = 0.5$$

$$k_1 = 3$$

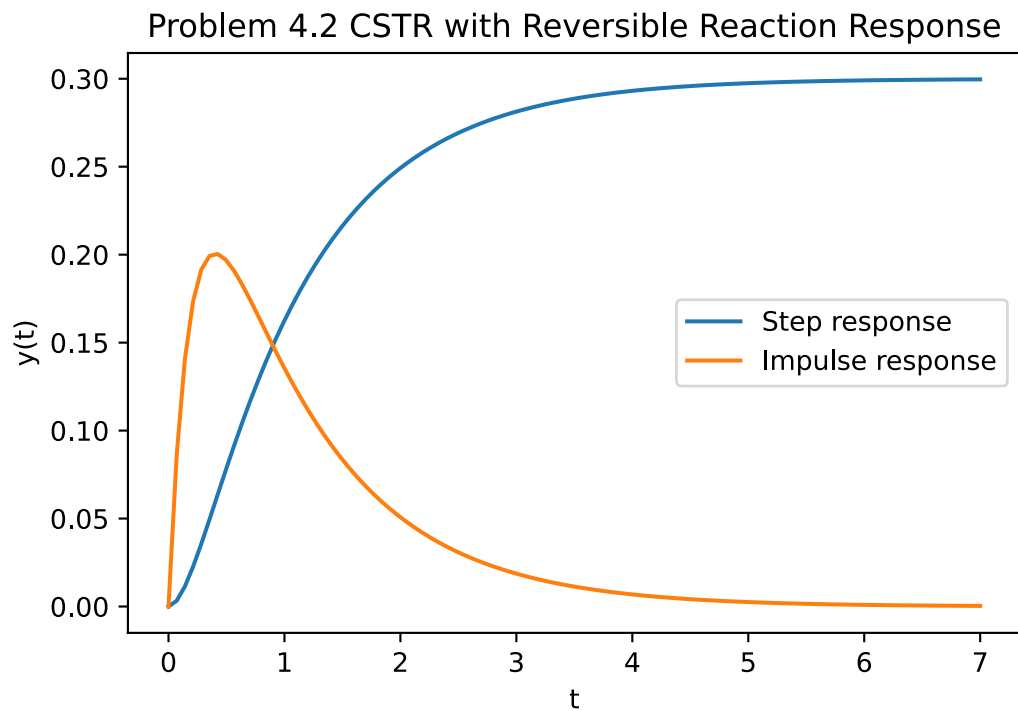
$$k_{-1} = 1.25$$

$$k_2 = 0.75$$

Response plot:



Response plot from manual transfer function:



Transfer function output:

TransferFunctionContinuous( array([1.5]), array([1., 6., 5.]), dt: None )

$$G(s) = \frac{1.5}{s^2 + 6s + 5}$$

Code for solving the system

```

import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as signal

# constants
F_V = 0.5
k_1 = 3
k_b = 1.25
k_2 = 0.75

# matrices
A = np.array([
    [-(F_V + k_1), k_b],
    [k_1, -(F_V + k_2 + k_b)]
])
B = np.array([
    [F_V],
    [0]
])
C = np.array([0, 1])
D = 0

# define state space system
sys = signal.StateSpace(A, B, C, D)

# compute step/impulse response
t_step, y_step = sys.step()

t_impulse, y_impulse = sys.impulse()

# plot
plt.plot(t_step, y_step, label="Step response")
plt.plot(t_impulse, y_impulse, label="Impulse response")
plt.xlabel(r"$t$")
plt.ylabel(r"$y(t)$")
plt.title("Problem 4.2 CSTR with Reversible Reaction Response")
plt.legend()

# compute transfer function
transfer_func = sys.to_tf()

print(transfer_func)

# manually create system from transfer function
sys_manual = signal.lti(transfer_func.num, transfer_func.den)

# compute step/impulse response
t_step, y_step = sys_manual.step()

```

```
t_impulse, y_impulse = sys_manual.impulse()

plt.plot(t_step, y_step, label="Step response")
plt.plot(t_impulse, y_impulse, label="Impulse response")
plt.xlabel(r" $t$ ")
plt.ylabel(r" $y(t)$ ")
plt.title("Problem 4.2 CSTR with Reversible Reaction Response")
plt.legend()
```