ASSIGNMENT 4: CO-CURRENT FLOW

Ishan Singh 200457

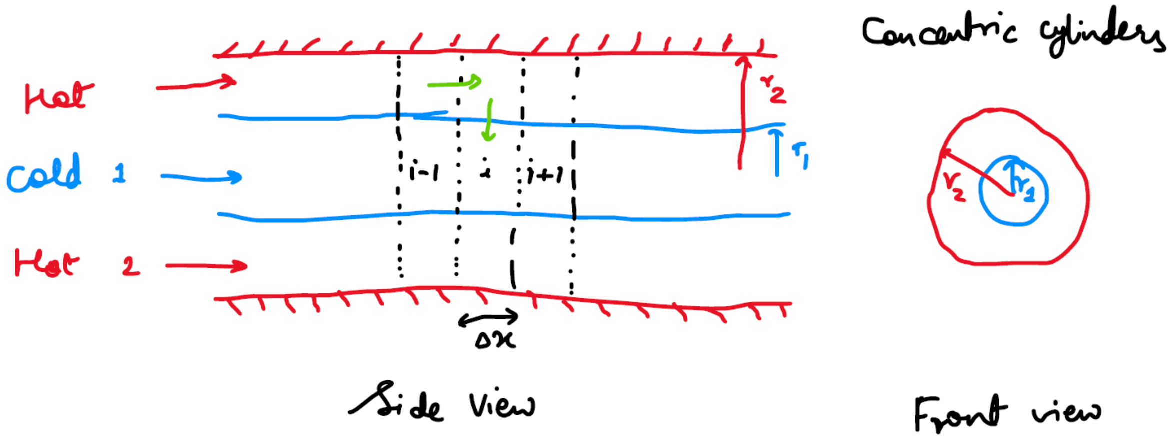
Solve, and obtain the transient response of Temperature with time for the concentric cylinder double pipe heat exchanger, as shown above.

Details:

1) Length of pipe = L = 60 m
2) Inner radius = r1 = 0.1 m
3) Outer radius = r2 = 0.15 m
4) Number of internal points = n = 100 (Can increase this for better accuracy)
5) For fluid 1 (Water here):
   1) m1 = Mass flow rate = 3 kg/s
   2) Cp1 = Heat capacity of fluid (water) = 4180 J/kg.K
   3) rho1 = Density of fluid (water) = 1000 kg/m^3
6) For fluid 2 (Water here again):
   1) m2 = Mass flow rate = 5 kg/s
   2) Cp2 = Heat capacity of fluid (water) = 4180 J/kg.K
   3) rho2 = Density of fluid (water) = 1000 kg/m^3
7) Initial temperature of fluid throughout the pipe = T0 = 300K
8) Inlet temperature of fluid 1 = T1i = 400 K
9) Inlet temperature of fluid 2 = T2i = 800 K
10) Overall heat transfer coefficient = U = 340 W/m^2

Simulate for t_final = 1000 seconds, with a time step (Δt) of 1 sec for each step.

Co-current flow



$A_{c_1} = \pi r_1^2$, $A_{c_2} = \pi(r_2^2 - r_1^2)$

Energy balance :

```
In [12]:
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation
from matplotlib.animation import FuncAnimation
```

```
In [14]:
l=int(input("Etnter the length of the pipe"))
r1=float(input("Inner Radius"))
r2=float(input("Outer Radius"))
n=int(input("Number of internal points"))
pi=3.14
a_1=pi*(r1**2)
a_2=pi*(r2**2-r1**2)
m_flow1=int(input("Mass flow rate 1"))
cp1=int(input("Heat capacity of fluid 1"))
den_1=int(input("Density of fluid 1"))
m_flow2=int(input("Mass flow rate 2"))
cp2=int(input("Heat capacity of fluid 2"))
den_2=int(input("Density of fluid 2"))
```

```
Etnter the length of the pipe60
Inner Radius0.1
Outer Radius0.15
Number of internal points100
Mass flow rate 13
Heat capacity of fluid 14180
Density of fluid 11000
Mass flow rate 25
Heat capacity of fluid 24180
Density of fluid 21000
```

In [15]:

```
To=int(input("Initial temperature of fluid throughout the pipe"))
t1_inlet=int(input("Inlet temperature of fluid 1"))
t2_inlet=int(input("Inlet temperature of fluid 2"))
u=int(input("Overall heat transfer coefficient"))
```

```
Initial temperature of fluid throughout the pipe300
Inlet temperature of fluid 1400
Inlet temperature of fluid 2800
Overall heat transfer coefficient340
```

In [16]:

```
del_x=l/n
t_final=1000 #tf
del_t=1
```

In [17]:

```
x=np.linspace(del_x/2,l-del_x/2,n)
t_1=np.ones(n)*To
t_2=np.ones(n)*To
delT1_dt=np.zeros(n)
delT2_dt=np.zeros(n)
t_in=np.zeros((t_final,n))
t_out=np.zeros((t_final,n))
```

In [20]:

```
t=np.arange(0,t_final,del_t)
```

Accumulation = In − Out + Generation

For inner cylinder:

$$\rho_1 C_{p,1} A_{c1} \Delta x \frac{dT_1}{dt} = \dot{m}_1 C_{p1}\left(T(i-1) - T(i)\right) + U \cdot 2\pi r_1 \Delta x \left(T_2(i) - T_1(i)\right)$$

$$\Rightarrow \boxed{\frac{dT_1}{dt} = \frac{\dot{m}_1 C_{p1}\left(T_1(i-1) - T_1(i)\right) + U \cdot 2\pi r_1 \Delta x \left(T_2(i) - T_1(i)\right)}{\rho_1 C_{p,1} A_{c1} \Delta x}} \quad -(1)$$

For outer cylinder:

$$\rho_2 C_{p,2} A_{c2} \Delta x \frac{dT_2}{dt} = \dot{m}_2 C_{p2}\left(T_2(i-1) - T_2(i)\right) - U \cdot 2\pi r_1 \Delta x \left(T_2(i) - T_1(i)\right)$$

$$\Rightarrow \boxed{\frac{dT_2}{dt} = \frac{\dot{m}_2 C_{p2}\left(T_2(i-1) - T_2(i)\right) - U \cdot 2\pi r_1 \cdot \Delta x \left(T_2(i) - T_1(i)\right)}{\rho_2 C_{p2} A_{c2} \Delta x}} \quad -(2)$$

In [25]:
```
for j in range(1,len(t)):
    plt.clf()
    delT1_dt[1:n]=(m_flow1*cp1*(t_1[0:n-1]-t_1[1:n])+u*2*pi*r1*del_x*(t_2[1:n]-t_1[1:n]))/(den_1*cp1*del_x*
    delT1_dt[0]=(m_flow1*cp1*(t1_inlet-t_1[0])+u*2*pi*r1*del_x*(t_2[0]-t_1[0]))/(den_1*cp1*del_x*a_1)
    delT2_dt[1:n]=(m_flow2*cp2*(t_2[0:n-1]-t_2[1:n])-u*2*pi*r1*del_x*(t_2[1:n]-t_1[1:n]))/(den_2*cp2*del_x*
    delT2_dt[0]=(m_flow2*cp2*(t2_inlet-t_2[0])-u*2*pi*r1*del_x*(t_2[0]-t_1[0]))/(den_2*cp2*del_x*a_2)
    t_1=t_1+delT1_dt*del_t
    t_2=t_2+delT2_dt*del_t
    t_in[j,:]=t_1
    t_out[j,:]=t_2
```

In [27]:

```
def plotmap(Tin,Tout):
    plt.clf()
    plt.plot(x,Tin,label="Inner Temp: T1")
    plt.plot(x,Tout,label="Outer Temp: T2")
    plt.xlabel('Distance(m)')
    plt.ylabel('Temp.(Kelvin)')
    plt.axis([0,l,0,900])
    return plt
```

In [31]:

```
def animate(j):
    plotmap(t_in[j,:],t_out[j,:])
anim = animation.FuncAnimation(plt.figure(), animate, interval=del_t, frames=t_final, repeat=False)
anim.save("Temperature.gif")
plt.draw()
plt.show()
```