Air Standard cycle Simulator using Python

Aim: Plotting P-V diagram and efficiency calculation of the Otto Cycle.

OBJECTIVES OF THE PROJECT

The main objective of the project is to calculate all unknowns of each state by help of governing equations and given values. After that plotting P-V diagram and calculating efficiency of the cycle.

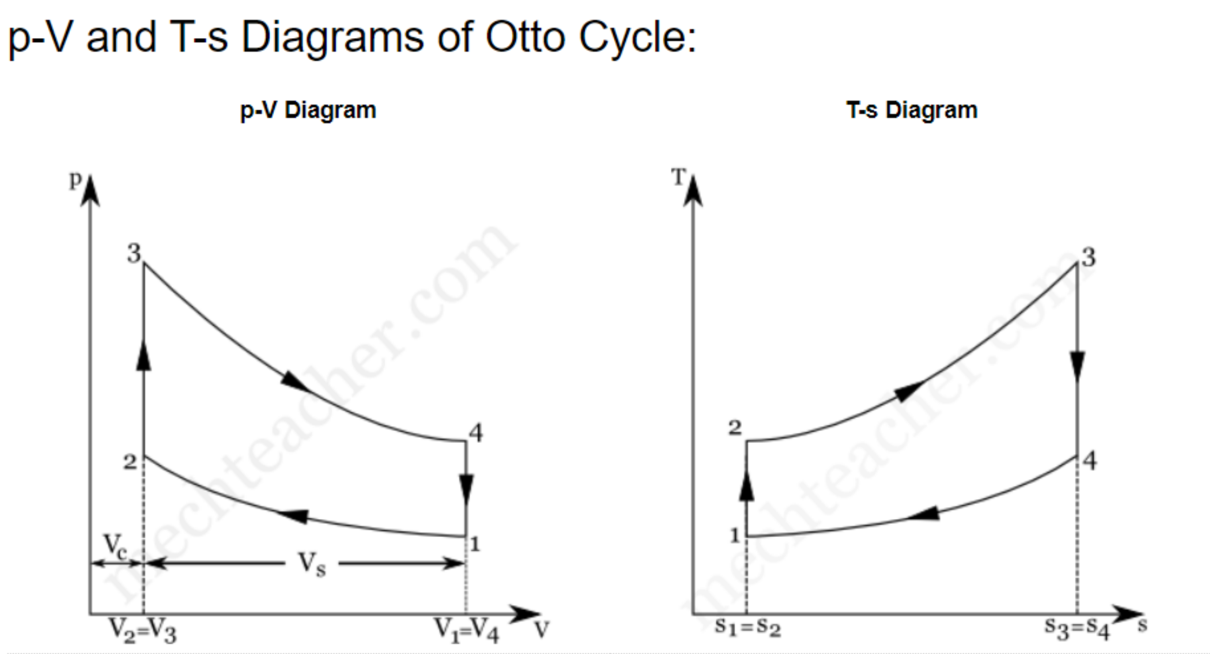
GOVERNING EQUATIONS:

Otto cycle is a gas power cycle that is used in a spark-ignition Internal combustion engine (modern petrol engines). It is an air standard cycle which approximates the processes in petrol or diesel engines.

An Otto cycle consists of four processes:

1. Two isentropic (reversible adiabatic) processes
2. Two isochoric (constant volume) processes

These processes can be easily understood if we understand p-V (Pressure-Volume) and T-s (Temperature-Entropy) diagrams of the Otto cycle.



In the above diagrams,

- P = Pressure
- V= volume
- T = Temperature
- S = Entropy
- Vc= Clearance volume
- V = Stroke volume

Processes in Otto Cycle:

Process 1-2: Isentropic compression

In this process, the piston moves from the bottom dead centre (BDC) to top dead centre (TDC) position. Air undergoes reversible adiabatic (isentropic) compression. We know that compression is a process in which volume decreases and pressure increases. Hence, in this process, the volume of air decreases from V_1 to V_2 and pressure increases from p_1 to p_2. Temperature increases from T_1 to T_2. As this an isentropic process, entropy remains constant (i.e., $s_1=s_2$).

Process 2-3: Constant Volume Heat Addition

Process 2-3 is isochoric (constant volume) heat addition process. Here, piston remains at the top dead centre for a moment. Heat is added at constant volume ($V_2 = V_3$) from an external heat source. Temperature increases from T_2 to T_3, pressure increases from p_2 to p_3 and entropy increases from s_2 to s_3.

In this process, Heat Supplied = $mC_v(T_3-T_2)$

where, m → Mass, C_v→ Specific heat at constant volume

Process 3-4: Isentropic expansion

In this process, air undergoes isentropic (reversible adiabatic) expansion. The piston is pushed from the top dead centre (TDC) to bottom dead centre (BDC) position. Here, pressure decreases from p_3 to p_4, volume rises from V_3 to V_4, the temperature falls from T_3 to T_4 and entropy remains constant (s_3 to s_4).

Process 4-1: Constant Volume Heat Rejection

The piston rests at BDC for a moment and heat is rejected at constant volume (V_4 to V_1). In this process, the pressure falls from p_4 to p_1, temperature decreases from T_4 to T_1 and entropy falls from s_4 to s_1.

In process 4-1

Heat Rejected = $mC_v(T_4-T_1)$

Thermal efficiency (air-standard efficiency) of otto cycle,

$$\eta_{\text{th}} = \frac{\text{Heat Supplied - Heat Rejected}}{\text{Heat Supplied}}$$

$$\eta_{\text{th}} = 1 - \frac{\left(T_4 - T_1\right)}{\left(T_3 - T_2\right)} = \eta_{\text{Otto}}$$

1 to 2 and from 3 to 4 are isentropic, therefore

$$T_4V_1^{\gamma-1} = T_3V_2^{\gamma-1}, \qquad T_1V_1^{\gamma-1} = T_2V_2^{\gamma-1}$$

Volume swept by the piston, Vs = (π*(bore)2*stroke)/4

Clearance volume, Vc = Vs/(cr-1)

Volume swept as a function of theta:

A = (0.5)*(cr-1)

B = R+1-cosd(theta)

C = ((R^2)-((sind(theta))^2))^0.5

V=(1+(A*(B-C)))*Vc

GIVEN:

bore=0.8 meters

stroke=0.2 meters

connecting rod length = 0.15 meters

compression ratio (cr) = 10

p1=101325

Ɣ=1.4

T1=450

T3=3050

Simulation Code

```
# Otto cycle simulator

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

#Engine geometrical parameters
bore=0.8
stroke=0.2
con_rod=0.15
cr=10

def engine_kinematics(bore,stroke,con_rod,cr,theta1,theta2):

    v_stroke=(math.pi*pow(bore,2)*stroke)/4
    v_clear=v_stroke/(cr-1)
    a=stroke/2;
    R=con_rod/a

    theta=np.linspace(theta1,theta2,180)
    V=[]

    for t in theta:

        A=(0.5)*(cr-1)
        B=R+1-math.cos(math.radians(t))
        C=pow((R**2)-(math.sin(math.radians(t))**2),0.5)
        V.append((1+(A*(B-C)))*v_clear)

    return V

#known state variables
p1=101325
#gamma
y=1.4
t1=450
t3=3050

#calculation of volumes
v_stroke=(math.pi*pow(bore,2)*stroke)/4
v_clear=v_stroke/(cr-1)
```

```
v_clear=v_stroke/(cr-1)

#state 1
v1=v_stroke+v_clear

#state 2
v2=v_clear
p2=pow(v1/v2,y)*p1
t2=pow(v1/v2,y-1)*t1

#state 3
v3=v2
p3=p2*(t3/t2)

#state 4
v4=v1
p4=pow(v3/v4,y)*p3
t4=pow(v3/v4,y-1)*t3

#During Compression stroke
v_comp=engine_kinematics(bore,stroke,con_rod,cr,180,0)
c1=p1*pow(v1,y)
p_comp=[]

for v in v_comp:
    p_comp.append(c1/(pow(v,y)))


#During Expansion stroke
v_exp=engine_kinematics(bore,stroke,con_rod,cr,0,180)
c2=p3*pow(v3,y)
p_exp=[]

for ve in v_exp:
    p_exp.append(c2/(pow(ve,y)))


#Calculation efficiency
n=1-((t4-t1)/(t3-t2))
print('Efficiency of engine = ', n);

```

```

#plotting state point
plt.plot(v1,p1,'*')
plt.plot(v2,p2,'*')
plt.plot(v3,p3,'*')
plt.plot(v4,p4,'*')
plt.plot(v_comp,p_comp, "-b",linewidth=3,label="Compression Stroke")
plt.plot([v2,v3],[p2,p3], "-r",linewidth=3,label="Heat Addition")
plt.plot(v_exp,p_exp, "-g",linewidth=3,label="Expansion Stroke")
plt.plot([v1,v4],[p1,p4], "-y",linewidth=3,label="Heat Rejection")

plt.xlabel('Volume (m^3)')
plt.ylabel('Pressure (pa)')
plt.legend()
plt.title('OTTO CYCLE')
plt.grid()
plt.show()

```

```
# Otto cycle simulator
import math
import numpy as np
import matplotlib.pyplot as plt
#Engine geometrical parameters
bore=0.8
stroke=0.2
con_rod=0.15
cr=10
def engine_kinematics(bore,stroke,con_rod,cr,theta1,theta2):
        v_stroke=(math.pi*pow(bore,2)*stroke)/4
        v_clear=v_stroke/(cr-1)
        a=stroke/2;
        R=con_rod/a
        theta=np.linspace(theta1,theta2,180)
        V=[]
        for t in theta:
                A=(0.5)*(cr-1)
                B=R+1-math.cos(math.radians(t))
                C=pow((R**2)-(math.sin(math.radians(t))**2),0.5)
                V.append((1+(A*(B-C)))*v_clear)
        return V
#known state variables
p1=101325
#gamma
ɣ=1.4
t1=450
t3=3050
#calculation of volumes
v_stroke=(math.pi*pow(bore,2)*stroke)/4
v_clear=v_stroke/(cr-1)

#state 1
v1=v_stroke+v_clear

#state 2
v2=v_clear
p2=pow(v1/v2,y)*p1
t2=pow(v1/v2,y-1)*t1

#state 3
v3=v2
p3=p2*(t3/t2)

#state 4
v4=v1
p4=pow(v3/v4,y)*p3
t4=pow(v3/v4,y-1)*t3

#During Compression stroke
v_comp=engine_kinematics(bore,stroke,con_rod,cr,180,0)
c1=p1*pow(v1,y)
p_comp=[]

for v in v_comp:
        p_comp.append(c1/(pow(v,y)))


#During Expansion stroke
v_exp=engine_kinematics(bore,stroke,con_rod,cr,0,180)
c2=p3*pow(v3,y)
p_exp=[]

for ve in v_exp:
```

```
        p_exp.append(c2/(pow(ve,y)))


#Calculation efficiency
n=1-((t4-t1)/(t3-t2))
print('Efficiency of engine = ', n);

#plotting state point
plt.plot(v1,p1,'*')
plt.plot(v2,p2,'*')
plt.plot(v3,p3,'*')
plt.plot(v4,p4,'*')
plt.plot(v_comp,p_comp, "-b",linewidth=3,label="Compression Stroke")
plt.plot([v2,v3],[p2,p3], "-r",linewidth=3,label="Heat Addition")
plt.plot(v_exp,p_exp, "-g",linewidth=3,label="Expansion Stroke")
plt.plot([v1,v4],[p1,p4], "-y",linewidth=3,label="Heat Rejection")

plt.xlabel('Volume (m^3)')
plt.ylabel('Pressure (pa)')
plt.legend()
plt.title('OTTO CYCLE')
plt.grid()
plt.show()
```

STEPS:

1. All known engine geometry parameters and state variable were defined.

2. The formula for volume swept and volume clearance was given.

3. With the help of Isentropic formulate for process 1-2 and 3-4, all state variable was calculated.

4. By this, we got all four-point of the p-v diagram but for tracing isentropic volume we need to define how the volume changes in the engine piston.

5. For this, we define a function, engine_kinematics which calculates the volume swept for isentropic processes for each value of angle covered by the connecting rod.

6. For this function, bore, stroke, length of connecting rod (cr), and swept angle were given as arguments.

7. After this P-V diagram plot was completed by the help of plotting command.

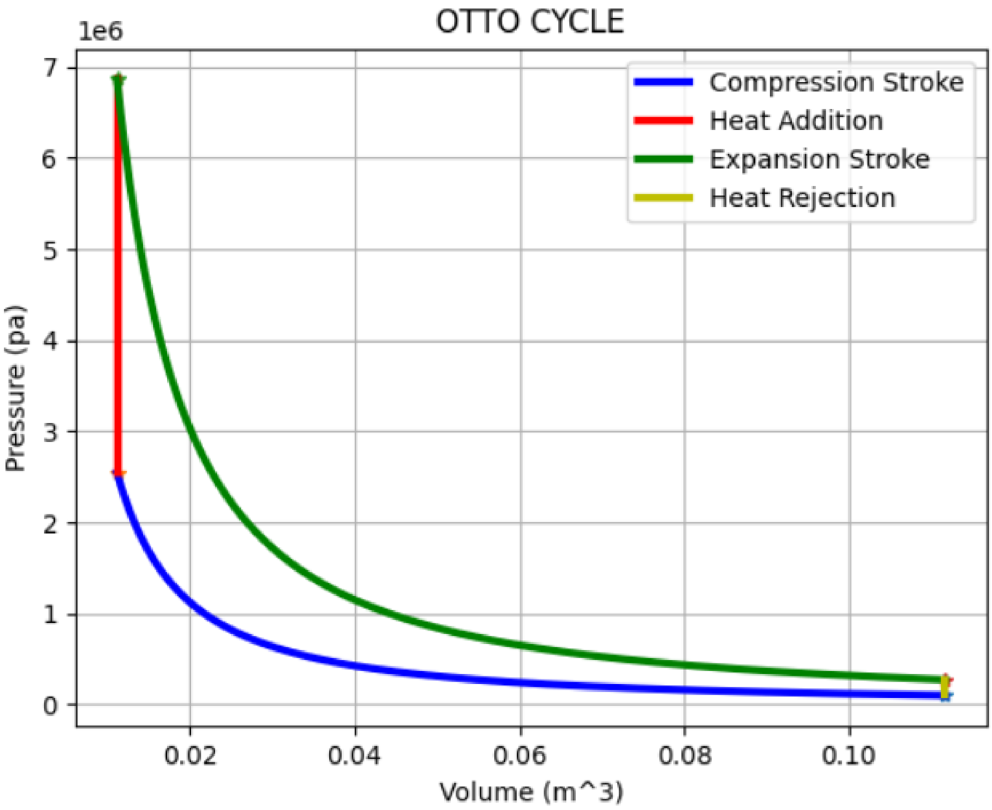
8. The efficiency formula was also defined to get the efficiency of the cycle.

OUTPUT:

The program outputs the efficiency of the Otto cycle and plots P-V diagram. In the P-V diagram we can see that for heat addition and heat removal process, the volume is constant and for the isentropic processes, the graph is following an isentropic path.

Efficiency of Engine = 0.6018928294465027

```
Efficiency of engine =  0.6018928294465027
```



CONCLUSION:

Using PYTHON programming, the efficiency of the Otto cycle was calculated and the P-V diagram is plotted. This is very helpful in the case where there is any change in engine geometry, we can easily update that in the code and quickly see the changes in the P-V diagram.