main

May 11, 2023

0.0.1 San Francisco State University: School of Engineering

0.0.2 ENGR 463: Final Exam Project (Spring 2023)

0.0.3 By Elon Goliger Mallimson

0.0.4 Professor: Dr. Douglas Couldron

0.0.5 Date: 5/10/2023

Part A: Variable Speed, Constant Load Test

Setup variables and dependencies as well as given data

[ ]:
```
#dependencies
from pint import UnitRegistry
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

#unit setup
ureg = UnitRegistry()
m = ureg.meter
g = ureg.gram
kg = ureg.kilogram
s = ureg.second
min = ureg.minute
hr = ureg.hour
rpm = ureg.revolution / min
turns = ureg.turn
kPa = ureg.kilopascal
W = ureg.watt
N = ureg.newton
K = ureg.kelvin

#variable array setup
speed = ureg.Quantity(np.array([1500, 2000, 2500, 3000, 3500, 4000]), rpm)
torque = ureg.Quantity(np.array([1.8, 1.8, 1.8, 1.8, 1.8, 1.8]), N * m)
Qdot_shaft = ureg.Quantity(np.array([None, None, None, None, None, None]), W) #
 ↪Qdot_shaft
```

```
Qdot_in = ureg.Quantity(np.array([None, None, None, None, None, None]), W) #␣
↪heat input
efficiency = np.array([None, None, None, None, None, None]) # n_th percent
MEP = ureg.Quantity(np.array([None, None, None, None, None, None]), kPa) # mean␣
↪effective pressure
bsfc = ureg.Quantity(np.array([None, None, None, None, None, None]), g / (W *␣
↪hr)) # brake specific fuel consumption
Qdot_exhaust = ureg.Quantity(np.array([None, None, None, None, None, None]), W)␣
↪# exhaust heat
Qdot_fins = ureg.Quantity(np.array([None, None, None, None, None, None]), W) #␣
↪fins heat
mdot_air = ureg.Quantity(np.array([None, None, None, None, None, None]), kg/s)␣
↪# mass flow rate of air
mdot_fuel = ureg.Quantity(np.array([5.55E-05, 5.64E-05, 6.83E-05, 8.26E-05, 0.
↪000104, 0.000118]), kg / s) # mass flow rate of fuel
deltaT = ureg.Quantity(np.array([247, 247, 247, 291, 330, 350]), K) #␣
↪temperature difference

#given value
D = 10**-4 * m**3 # displacement of the engine: given 100cc
#constants
rho_air = 1.2 * ureg.kilogram / ureg.meter**3
Cp_air = 1.006 * ureg.kilojoule / (ureg.kilogram * ureg.kelvin)
LHV_gas = 45.2 * ureg.kilojoule / ureg.gram
```

Calculations

[ ]:
```
mdot_air = (rho_air * D * speed) / (2*turns) # The 2 is present in the␣
↪denominator because engine will only draw air every second revolution
mdot_air = mdot_air.to(kg / s)
Qdot_exhaust = mdot_air * Cp_air * deltaT
Qdot_in = mdot_fuel * LHV_gas
Qdot_in = Qdot_in.to(W)
Qdot_shaft = speed*torque
Qdot_shaft = Qdot_shaft.to(W)
Qdot_fins = Qdot_in - Qdot_exhaust - Qdot_shaft
efficiency = (Qdot_shaft / Qdot_in)*100 # n_th percent
bsfc = mdot_fuel / Qdot_shaft
bsfc = bsfc.to(g / (W * hr))
MEP = (4 * np.pi * torque )/ D
MEP = MEP.to(kPa)
```

Table A

[ ]:
```
# Create a dictionary with the column names and data
data = {
    'Speed (RPM)': speed.magnitude,
    'Torque (N * m)': torque.magnitude,
```

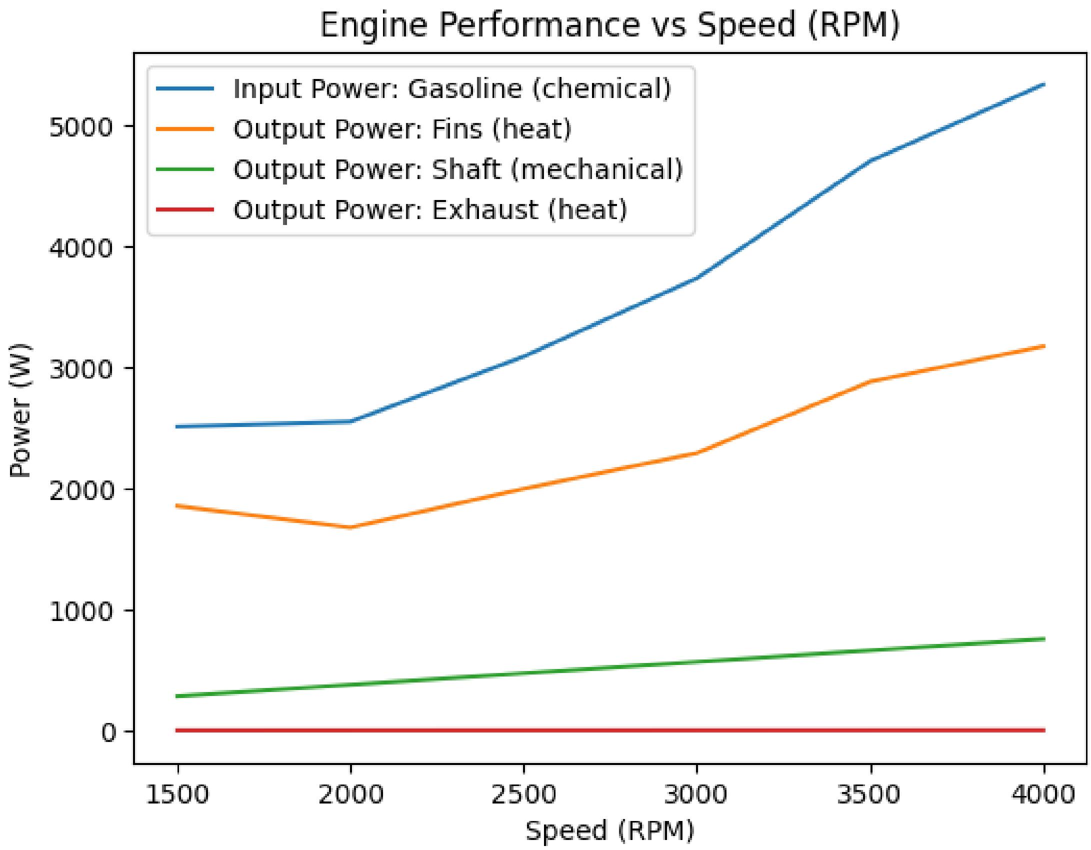
```
    'Power Shaft (W)': Qdot_shaft.magnitude,
    'Power Input (W)': Qdot_in.magnitude,
    'Efficiency (%)': efficiency.magnitude,
    'MEP (kPa)': MEP.magnitude,
    'BSFC (g/(W*h))': bsfc.magnitude,
    'Exhaust Heat (W)': Qdot_exhaust.magnitude,
    'Fins Heat (W)': Qdot_fins.magnitude,
    'Air Mass Flow (kg/s)': mdot_air.magnitude,
    'Fuel Mass Flow (kg/s)': mdot_fuel.magnitude,
    'Delta T (delta_K)': deltaT.magnitude
}

# Create a DataFrame from the dictionary and add a caption
df = pd.DataFrame(data)
df
```

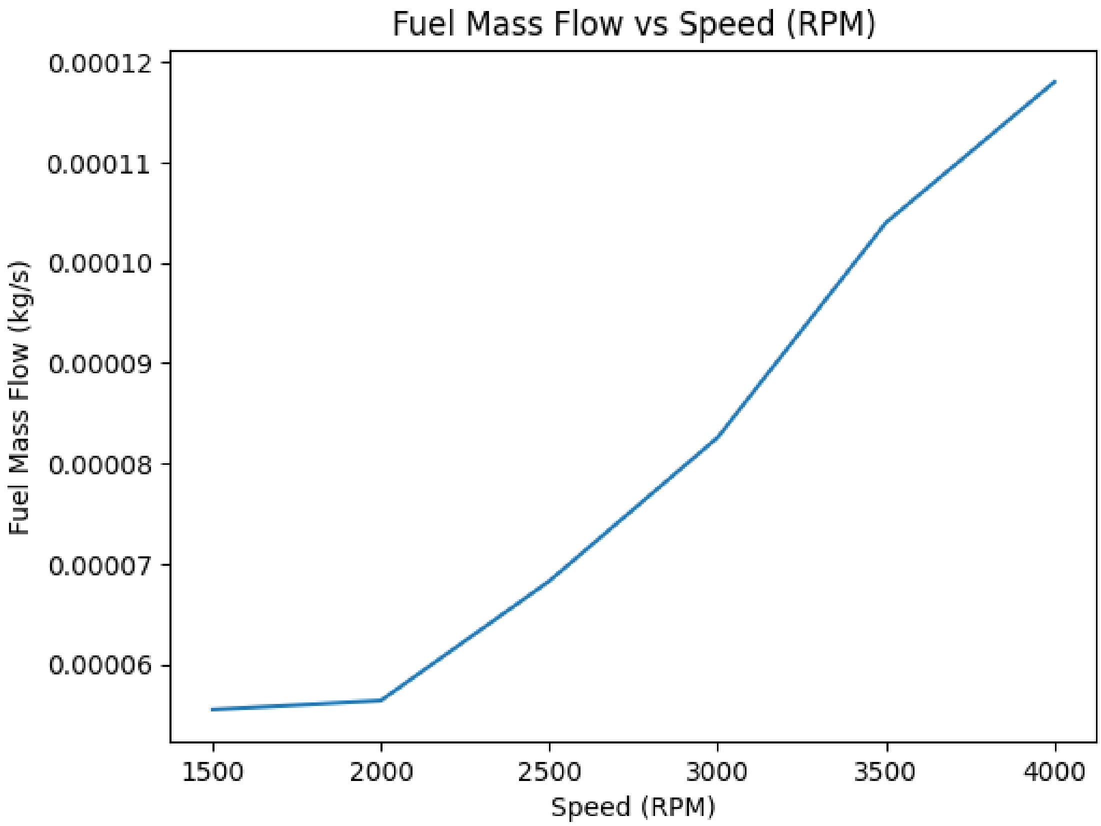
[ ]:

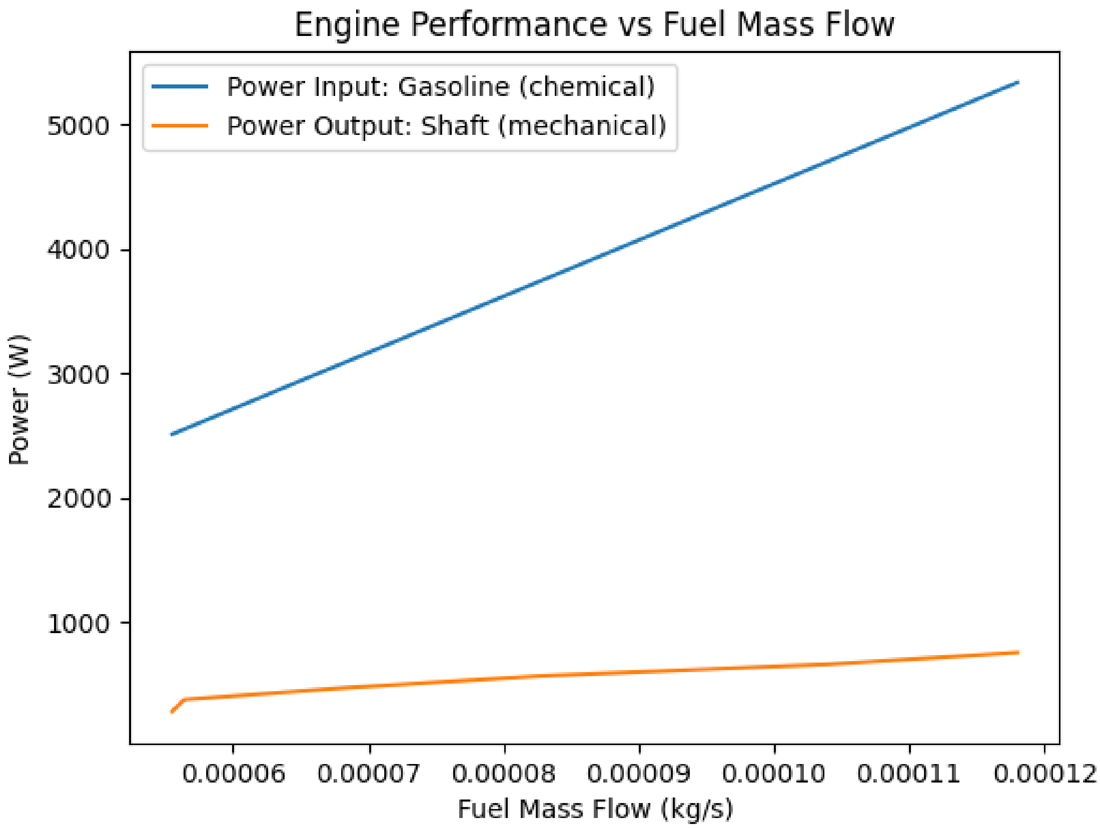
```
   Speed (RPM)  Torque (N * m)  Power Shaft (W)  Power Input (W)
0         1500             1.8       282.743339          2508.60  \
1         2000             1.8       376.991118          2549.28
2         2500             1.8       471.238898          3087.16
3         3000             1.8       565.486678          3733.52
4         3500             1.8       659.734457          4700.80
5         4000             1.8       753.982237          5333.60

   Efficiency (%)   MEP (kPa)  BSFC (g/(W*h))  Exhaust Heat (W)
0       11.270961  226.194671        0.706648          0.372723  \
1       14.788141  226.194671        0.538580          0.496964
2       15.264479  226.194671        0.521774          0.621205
3       15.146207  226.194671        0.525848          0.878238
4       14.034514  226.194671        0.567501          1.161930
5       14.136460  226.194671        0.563408          1.408400

   Fins Heat (W)  Air Mass Flow (kg/s)  Fuel Mass Flow (kg/s)
0    1853.133661                0.0015               0.000056  \
1    1675.324882                0.0020               0.000056
2    1994.716102                0.0025               0.000068
3    2289.795322                0.0030               0.000083
4    2879.135543                0.0035               0.000104
5    3171.217763                0.0040               0.000118

   Delta T (delta_K)
0                247
1                247
2                247
3                291
4                330
5                350
```

Graphs

```
[ ]: # Plot Qdot_shaft, Qdot_in, Qdot_exhaust, Qdot_fins vs RPM
plt.figure()
plt.plot(df['Speed (RPM)'], df['Power Input (W)'], label="Input Power: Gasoline␣
 ↪(chemical)")
plt.plot(df['Speed (RPM)'], df['Fins Heat (W)'], label="Output Power: Fins␣
 ↪(heat)")
plt.plot(df['Speed (RPM)'], df['Power Shaft (W)'], label="Output Power: Shaft␣
 ↪(mechanical)")
plt.plot(df['Speed (RPM)'], df['Exhaust Heat (W)'], label="Output Power:␣
 ↪Exhaust (heat)")
plt.xlabel('Speed (RPM)')
plt.ylabel('Power (W)')
plt.legend()
plt.title('Engine Performance vs Speed (RPM)')
plt.show()

# Plot fuel flow rate vs rpm
plt.figure()
plt.plot(df['Speed (RPM)'], df['Fuel Mass Flow (kg/s)'])
plt.xlabel('Speed (RPM)')
plt.ylabel('Fuel Mass Flow (kg/s)')
plt.title('Fuel Mass Flow vs Speed (RPM)')
plt.show()

# Plot Qdot_shaft, Qdot_in vs fuel flow rate
plt.figure()
plt.plot(df['Fuel Mass Flow (kg/s)'], df['Power Input (W)'], label="Power Input:
 ↪ Gasoline (chemical)")
plt.plot(df['Fuel Mass Flow (kg/s)'], df['Power Shaft (W)'], label="Power␣
 ↪Output: Shaft (mechanical)")
plt.xlabel('Fuel Mass Flow (kg/s)')
plt.ylabel('Power (W)')
plt.legend()
plt.title('Engine Performance vs Fuel Mass Flow')
plt.show()
```


Engine Performance vs Speed (RPM)
Input Power: Gasoline (chemical)
Output Power: Fins (heat)
Output Power: Shaft (mechanical)
Output Power: Exhaust (heat)
Power (W)
Speed (RPM)
0
1000
2000
3000
4000
5000
1500
2000
2500
3000
3500
4000


Fuel Mass Flow vs Speed (RPM)
Fuel Mass Flow (kg/s)
0.00012
0.00011
0.00010
0.00009
0.00008
0.00007
0.00006
1500
2000
2500
3000
3500
4000
Speed (RPM)



Part B: Constant Speed (1500 RPM), Variable Load Test

Setup variables and dependencies as well as given data

[ ]:
```
#variable array setup
percent_load = np.array([0.75, 1, 1.25, 1.5, 1.75, 2]) # percent
speed = ureg.Quantity(np.array([1500, 1500, 1500, 1500, 1500, 1500]), rpm) # rpm
torque = ureg.Quantity(np.array([1.35, 1.8, None, None, None, None]), N * m)
Qdot_shaft = ureg.Quantity(np.array([None, None, None, None, None, None]), W) #
 ↪Qdot_shaft
Qdot_in = ureg.Quantity(np.array([None, None, None, None, None, None]), W) #
 ↪heat input
efficiency = np.array([None, None, None, None, None, None]) # n_th percent
MEP = ureg.Quantity(np.array([169.64, 226.19, None, None, None, None]), kPa) #
 ↪mean effective pressure
bsfc = ureg.Quantity(np.array([None, None, None, None, None, None]), g / (W *
 ↪hr)) # brake specific fuel consumption
Qdot_exhaust = ureg.Quantity(np.array([None, None, None, None, None, None]), W)
 ↪# exhaust heat
```

```
Qdot_fins = ureg.Quantity(np.array([None, None, None, None, None, None]), W) #␣
↪fins heat
mdot_air = ureg.Quantity(np.array([None, None, None, None, None, None]), kg/s)␣
↪# mass flow rate of air
mdot_fuel = ureg.Quantity(np.array([4.931E-05, 5.547E-05, 8.452E-05, 8.452E-05,␣
↪9.342E-05, 0.000118]), kg / s) # mass flow rate of fuel
deltaT = ureg.Quantity(np.array([235, 247, 250, 288, 314, 345]), K) #␣
↪temperature difference
```

Calculations

[ ]:
```
# Calculate slope of load-torque line
slope = (torque[1].magnitude - torque[0].magnitude) / (percent_load[1] -␣
↪percent_load[0])

# Calculate torque for remaining percent loads
for i in range(2, len(percent_load)):
    torque[i] = (slope * (percent_load[i] - percent_load[1]) + torque[1].
↪magnitude) * N * m

mdot_air = (rho_air * D * speed) / (2*turns) # The 2 is present in the␣
↪denominator because engine will only draw air every second revolution
mdot_air = mdot_air.to(kg / s)
Qdot_exhaust = mdot_air * Cp_air * deltaT
Qdot_in = mdot_fuel * LHV_gas
Qdot_in = Qdot_in.to(W)
Qdot_shaft = speed*torque
Qdot_shaft = Qdot_shaft.to(W)
Qdot_fins = Qdot_in - Qdot_exhaust - Qdot_shaft
efficiency = (Qdot_shaft / Qdot_in)*100 # n_th percent
bsfc = mdot_fuel / Qdot_shaft
bsfc = bsfc.to(g / (W * hr))
MEP = (4 * np.pi * torque )/ D
MEP = MEP.to(kPa)
```

Table B

[ ]:
```
# Create a dictionary with the column names and data
data = {
    'Speed (RPM)': speed.magnitude,
    'Torque (N * m)': torque.magnitude,
    'Power Shaft (W)': Qdot_shaft.magnitude,
    'Power Input (W)': Qdot_in.magnitude,
    'Efficiency (%)': efficiency.magnitude,
    'MEP (kPa)': MEP.magnitude,
    'BSFC (g/(W*h))': bsfc.magnitude,
    'Exhaust Heat (W)': Qdot_exhaust.magnitude,
    'Fins Heat (W)': Qdot_fins.magnitude,
```

```python
    'Air Mass Flow (kg/s)': mdot_air.magnitude,
    'Fuel Mass Flow (kg/s)': mdot_fuel.magnitude,
    'Delta T (delta_K)': deltaT.magnitude
}

# Create a DataFrame from the dictionary and add a caption
df = pd.DataFrame(data)
df
```

[ ]:

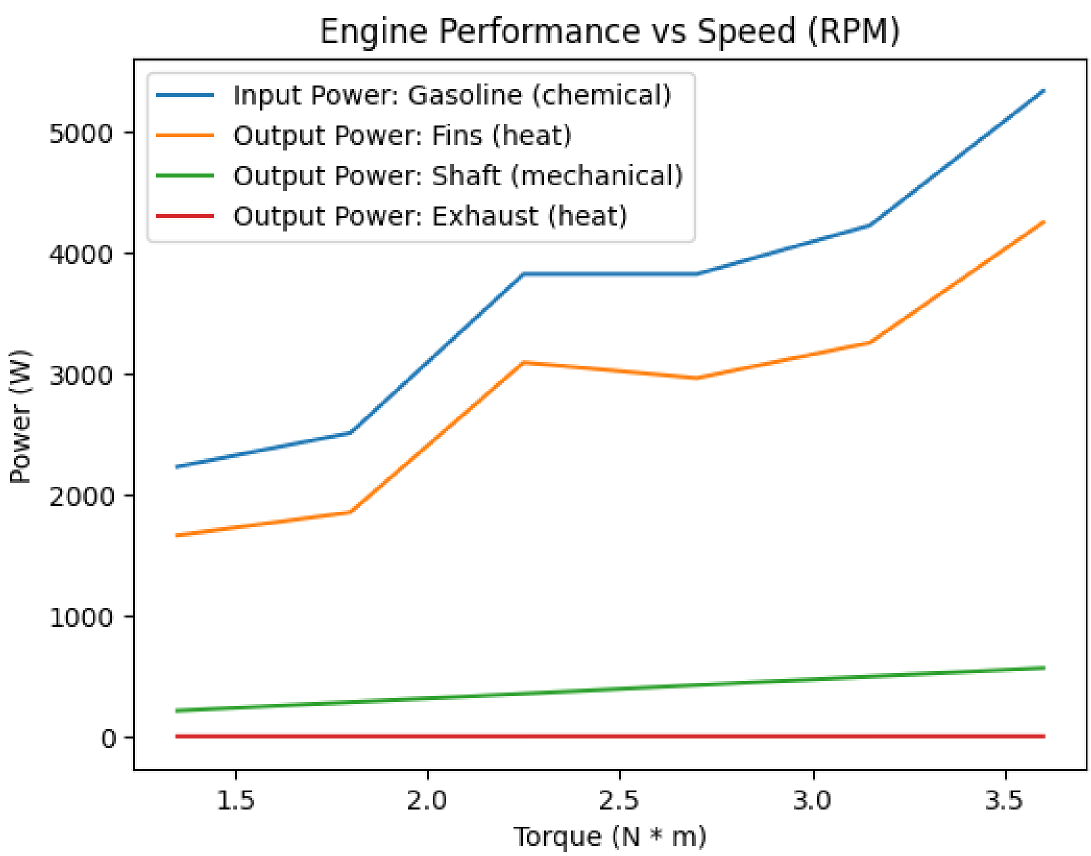
```
   Speed (RPM) Torque (N * m) Power Shaft (W)  Power Input (W) Efficiency (%)
0         1500           1.35      212.057504         2228.812       9.514374  \
1         1500            1.8      282.743339         2507.244      11.277057
2         1500           2.25      353.429174         3820.304       9.251336
3         1500            2.7      424.115008         3820.304      11.101604
4         1500           3.15      494.800843         4222.584      11.717963
5         1500            3.6      565.486678         5333.600      10.602345

    MEP (kPa) BSFC (g/(W*h))  Exhaust Heat (W) Fins Heat (W)
0  169.646003       0.837113          0.354615   1662.139496  \
1  226.194671       0.706266          0.372723   1851.777661
2  282.743339       0.860914          0.377250   3089.624826
3  339.292007       0.717428          0.434592   2961.596992
4  395.840674       0.679692          0.473826   3253.957157
5  452.389342       0.751211          0.520605   4247.508322

   Air Mass Flow (kg/s)  Fuel Mass Flow (kg/s)  Delta T (delta_K)
0                0.0015               0.000049                235
1                0.0015               0.000055                247
2                0.0015               0.000085                250
3                0.0015               0.000085                288
4                0.0015               0.000093                314
5                0.0015               0.000118                345
```

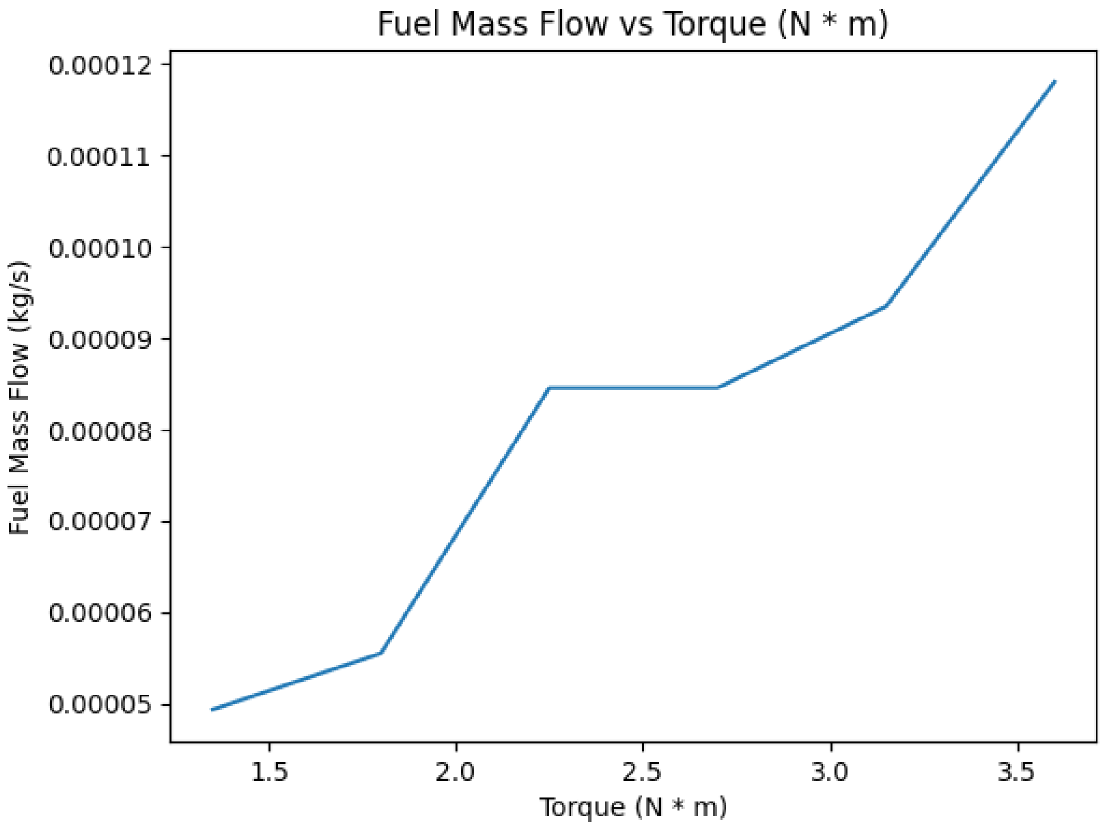
Graphs

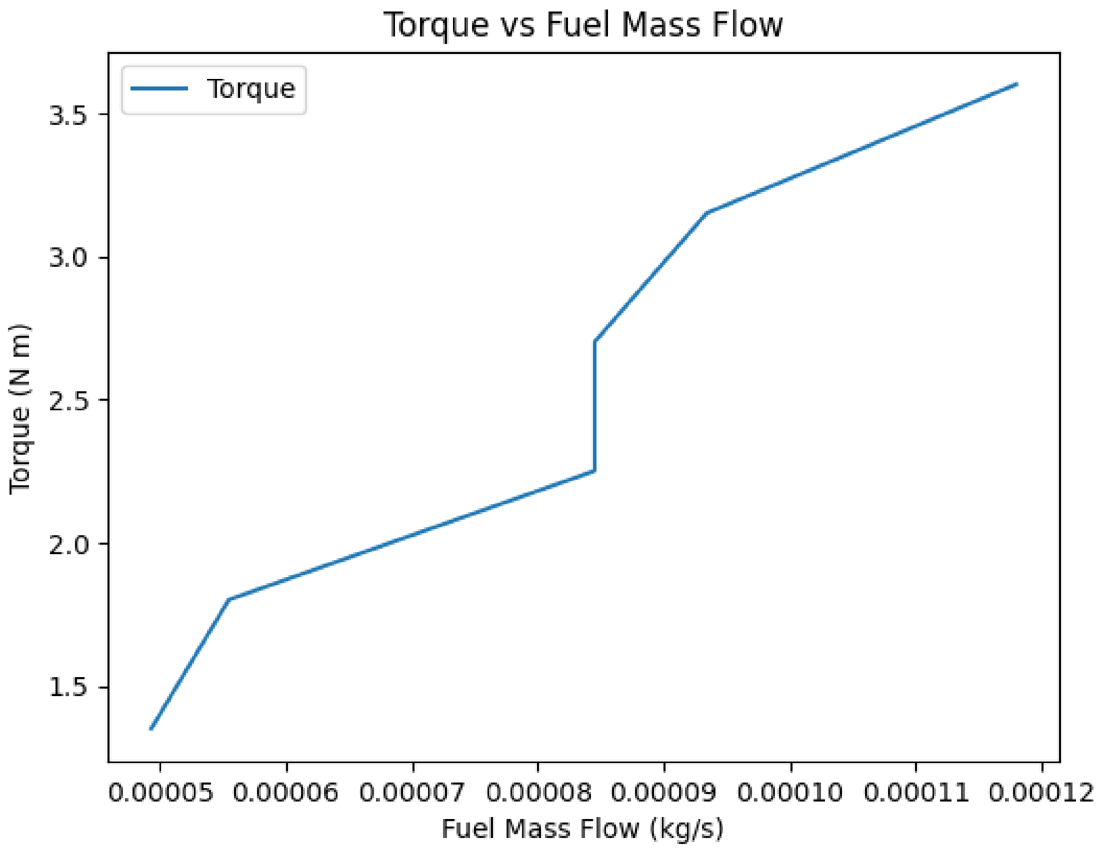
[ ]:

```python
# Plot Qdot_shaft, Qdot_in, Qdot_exhaust, Qdot_fins vs Torque
plt.figure()
plt.plot(df['Torque (N * m)'], df['Power Input (W)'], label="Input Power: Gasoline (chemical)")
plt.plot(df['Torque (N * m)'], df['Fins Heat (W)'], label="Output Power: Fins (heat)")
plt.plot(df['Torque (N * m)'], df['Power Shaft (W)'], label="Output Power: Shaft (mechanical)")
plt.plot(df['Torque (N * m)'], df['Exhaust Heat (W)'], label="Output Power: Exhaust (heat)")
plt.xlabel('Torque (N * m)')
plt.ylabel('Power (W)')
plt.legend()
```

```python
plt.title('Engine Performance vs Speed (RPM)')
plt.show()

# Plot fuel flow rate vs rpm
plt.figure()
plt.plot(df['Torque (N * m)'], df['Fuel Mass Flow (kg/s)'])
plt.xlabel('Torque (N * m)')
plt.ylabel('Fuel Mass Flow (kg/s)')
plt.title('Fuel Mass Flow vs Torque (N * m)')
plt.show()

# Plot Qdot_shaft, Qdot_in vs fuel flow rate
plt.figure()
plt.plot(df['Fuel Mass Flow (kg/s)'], df['Torque (N * m)'], label="Torque")
plt.xlabel('Fuel Mass Flow (kg/s)')
plt.ylabel('Torque (N m)')
plt.legend()
plt.title('Torque vs Fuel Mass Flow')
plt.show()
```




Fuel Mass Flow vs Torque (N * m)
Fuel Mass Flow (kg/s)
0.00012
0.00011
0.00010
0.00009
0.00008
0.00007
0.00006
0.00005
1.5
2.0
2.5
3.0
3.5
Torque (N * m)


Torque vs Fuel Mass Flow
Torque
3.5
3.0
2.5
2.0
1.5
Torque (N m)
0.00005
0.00006
0.00007
0.00008
0.00009
0.00010
0.00011
0.00012
Fuel Mass Flow (kg/s)