test

May 22, 2023

San Francisco State University: School of Engineering

ENGR 463: Final Exam Project (Spring 2023)

By Elon Goliger Mallimson

Professor: Dr. Douglas Couldron

Date: 5/18/2021

Part A: Variable Speed, Constant Load Test

Setup variables and dependencies as well as given data

[ ]:
```python
#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_exhaust = Qdot_exhaust.to(W)
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           372.723  \
1       14.788141  226.194671        0.538580           496.964
2       15.264479  226.194671        0.521774           621.205
3       15.146207  226.194671        0.525848           878.238
4       14.034514  226.194671        0.567501          1161.930
5       14.136460  226.194671        0.563408          1408.400

   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

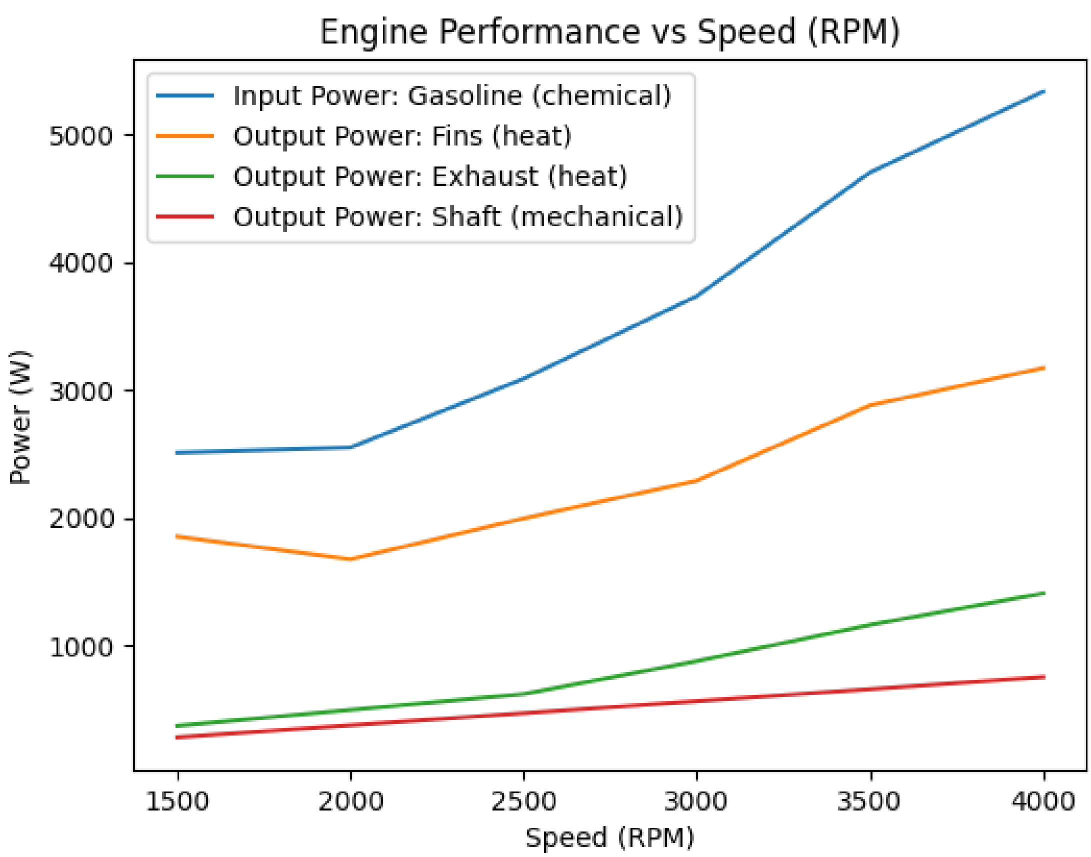
[ ]:

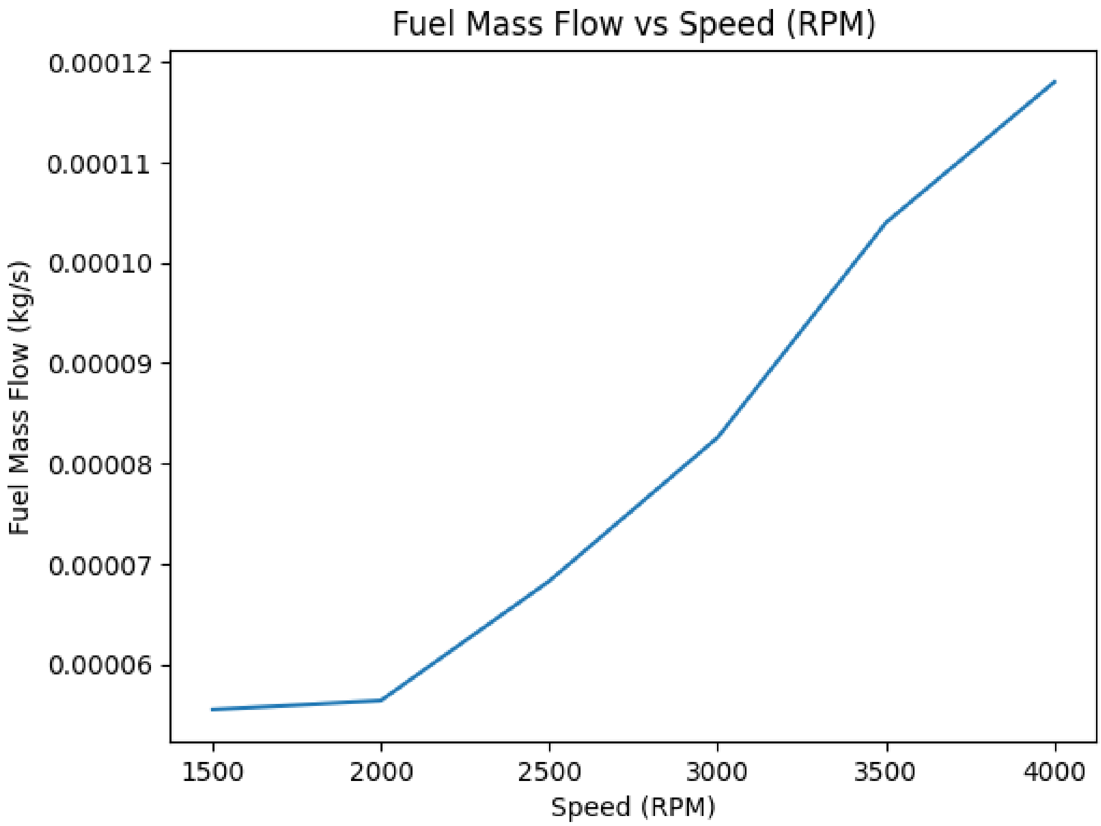
```
# Plot Qdot_shaft, Qdot_in, Qdot_exhaust, Qdot_fins vs RPM
plt.figure()
plt.plot(df['Speed (RPM)'], df['Power Input (W)'], label="Input Power: Gasoline␣
 ↪(chemical)")
plt.plot(df['Speed (RPM)'], df['Fins Heat (W)'], label="Output Power: Fins␣
 ↪(heat)")
plt.plot(df['Speed (RPM)'], df['Exhaust Heat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

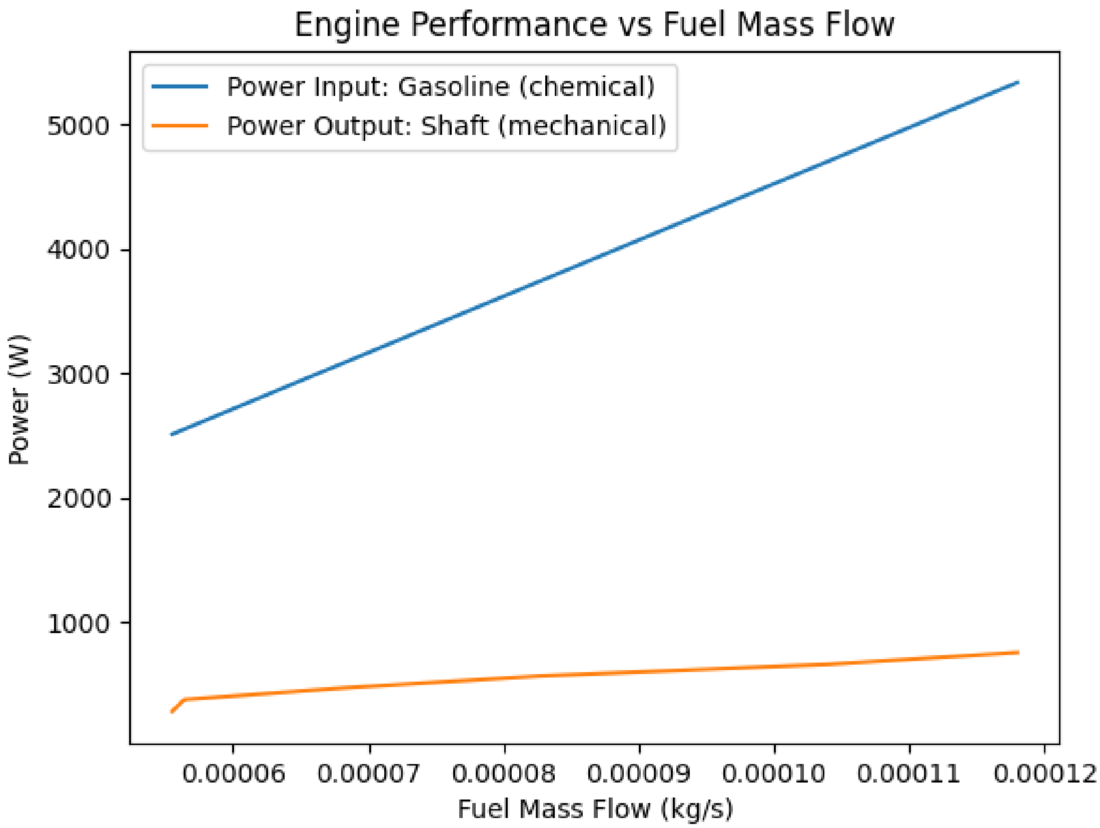
```
plt.show()

# 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()
```




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) #␣
↪torque
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_exhaust = Qdot_exhaust.to(W)
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 = {
    'Percent Load (%)': percent_load,
    '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
```

[ ]:
```
   Percent Load (%)  Speed (RPM) Torque (N * m) Power Shaft (W)
0              0.75         1500           1.35      212.057504  \
1              1.00         1500            1.8      282.743339
2              1.25         1500           2.25      353.429174
3              1.50         1500            2.7      424.115008
4              1.75         1500           3.15      494.800843
5              2.00         1500            3.6      565.486678

   Power Input (W) Efficiency (%)   MEP (kPa) BSFC (g/(W*h))
0         2228.812        9.514374 169.646003        0.837113  \
1         2507.244       11.277057 226.194671        0.706266
2         3820.304        9.251336 282.743339        0.860914
3         3820.304       11.101604 339.292007        0.717428
4         4222.584       11.717963 395.840674        0.679692
5         5333.600       10.602345 452.389342        0.751211

   Exhaust Heat (W) Fins Heat (W)  Air Mass Flow (kg/s)
0           354.615   1662.139496                0.0015  \
1           372.723   1851.777661                0.0015
2           377.250   3089.624826                0.0015
3           434.592   2961.596992                0.0015
4           473.826   3253.957157                0.0015
5           520.605   4247.508322                0.0015

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

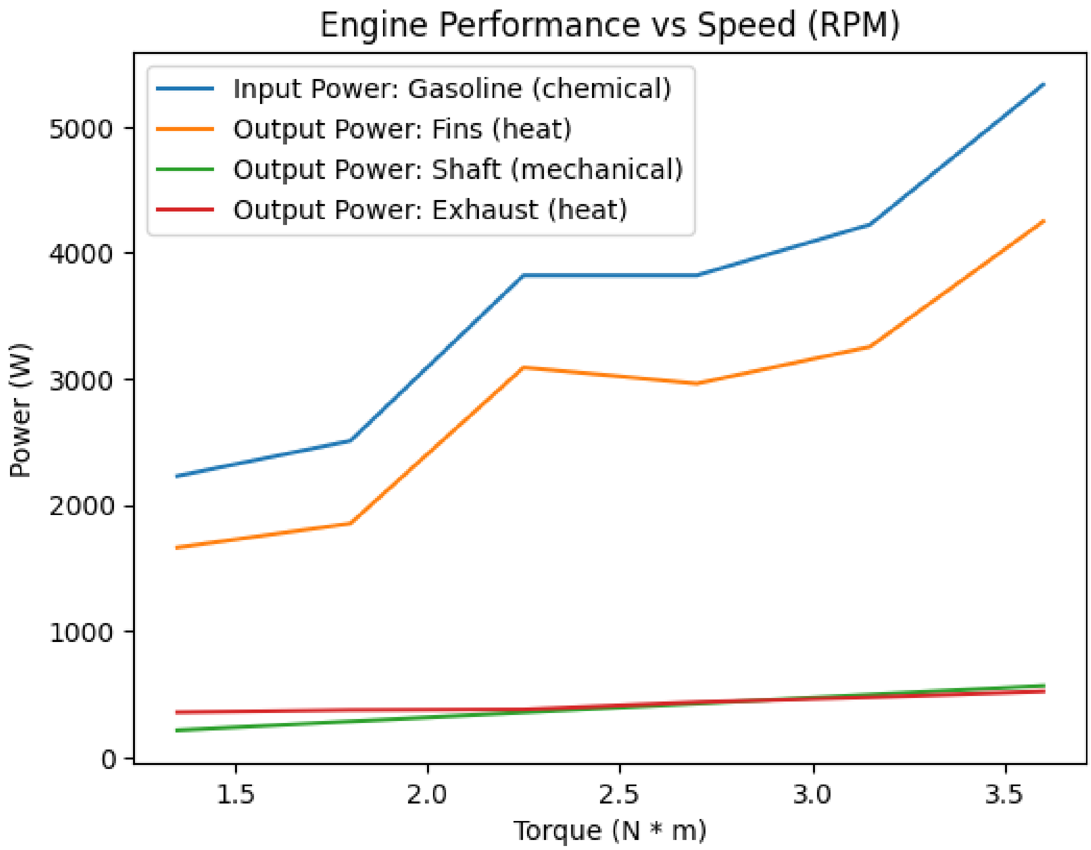
Graphs

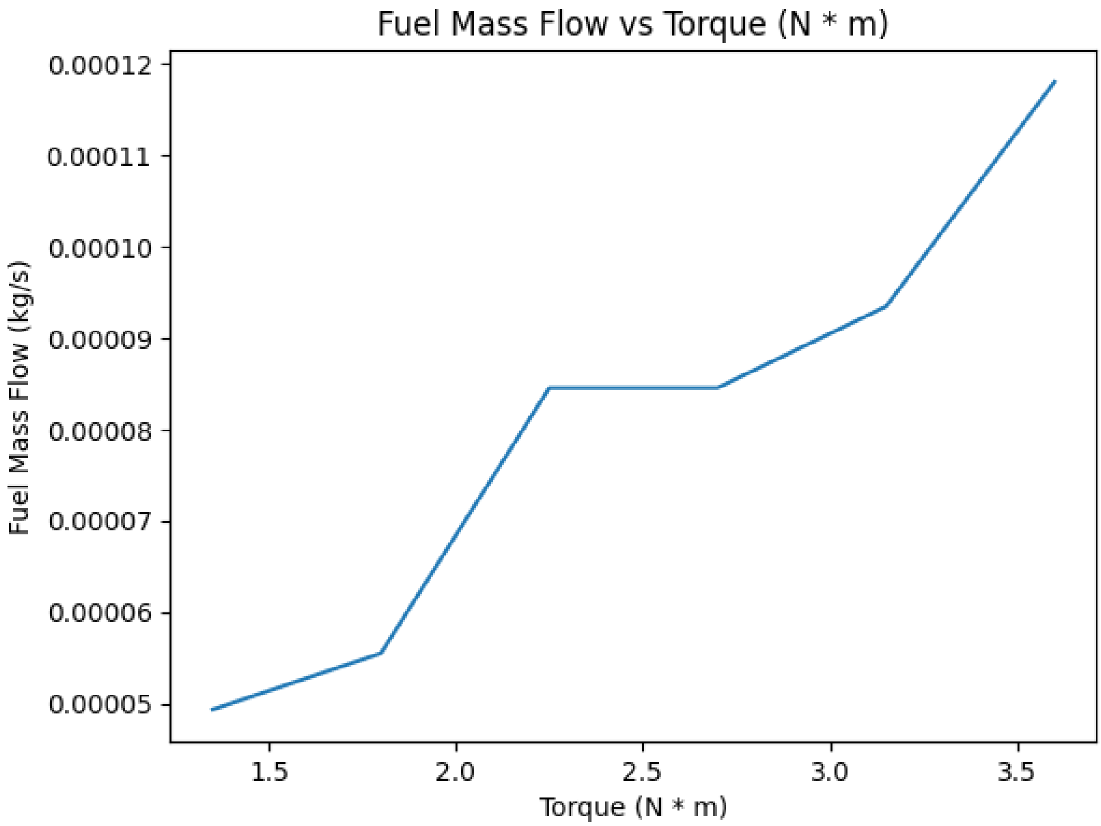
[ ]:
```
# Plot Qdot_shaft, Qdot_in, Qdot_exhaust, Qdot_fins vs Torque
plt.figure()
```

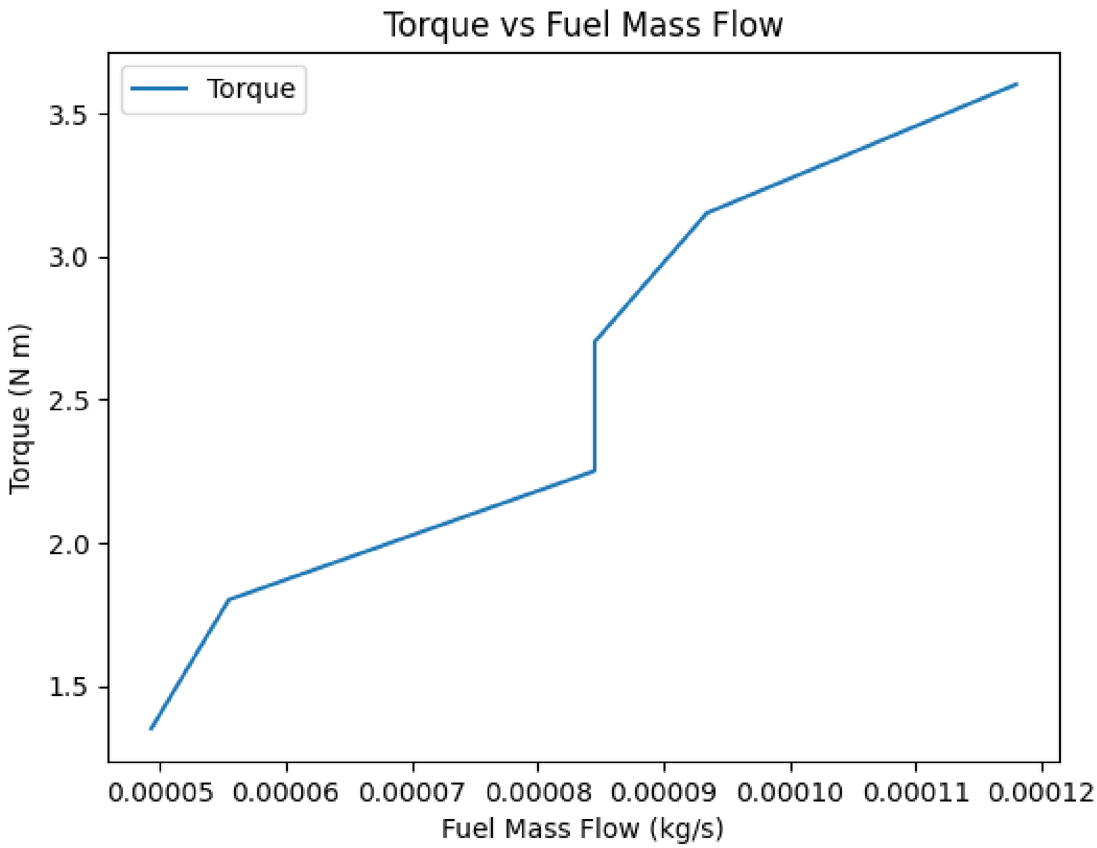
```
plt.plot(df['Torque (N * m)'], df['Power Input (W)'], label="Input Power:␣
↪Gasoline (chemical)")
plt.plot(df['Torque (N * m)'], df['Fins Heat (W)'], label="Output Power: Fins␣
↪(heat)")
plt.plot(df['Torque (N * m)'], df['Power Shaft (W)'], label="Output Power:␣
↪Shaft (mechanical)")
plt.plot(df['Torque (N * m)'], df['Exhaust Heat (W)'], label="Output Power:␣
↪Exhaust (heat)")
plt.xlabel('Torque (N * m)')
plt.ylabel('Power (W)')
plt.legend()
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()
```


Engine Performance vs Speed (RPM)
Input Power: Gasoline (chemical)
Output Power: Fins (heat)
Output Power: Shaft (mechanical)
Output Power: Exhaust (heat)
Power (W)
Torque (N * m)
0
1000
2000
3000
4000
5000
1.5
2.0
2.5
3.0
3.5


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)



Conclusion Questions

1) Comment on the efficiency of engine for both loading cases (Table A vs. Table B). How could we improve the efficiency of the engine? Do these results make sense?

Efficiency is higher for higher RPM's and lower loads. This makes sense because the engine is running faster and therefore has more power to spare. We could improve the efficiency of the engine by increasing the compression ratio, which would increase the power output of the engine. These results make sense because the engine is more efficient at higher RPM's and lower loads.

2) What are some of the things we would need to be conscious of in our experiment to obtain reliable results?

We would have to be conscious that the engine is warmed up before we start the experiment. We would also have to be aware of how the engine is performing, making sure that it does not overheat and cause errant results. Engine should not run too lean or too rich. We would want to make sure the oil and gas is fresh to make sure we don't lose more efficiency then we would expect. It is also imperative that there are no leaks in the system.

3) Why may it be important to allow an engine to warm up before placing it under test (in terms of reliable results)?

Engines are more efficient when they are warm. This is because the oil is more viscous when it is cold, and therefore causes more friction. More friction will decrease efficiency.