Performance of an Automobile

Ishaan Kharbanda

UE179039

11/23/2020

Car Model : Jaguar F- Type P340

Engine Specifications

In [553]:

```
P_max = 340               # Maximum Engine power (hp)
T_max = 450               # Maximum Engine torque (Nm)

N_Pmax = 6500             # Engine Speed at Maximum Power (rpm)
Nmax_Tmax = 3500          # Engine Speed at Maximum Torque (rpm)

max_speed = 260           # Maximum speed (kph)
```

Transmission

In [648]:

```
# Gear Ratios

ig1 = 4.71          # First gear ratio
ig2 = 3.14          # Second gear ratio
ig3 = 2.11          # Third gear ratio
ig4 = 1.67          # Fourth gear ratio
ig5 = 1.29          # Fifth gear ratio
ig6 = 1.00          # Sixth gear ratio
ig7 = 0.84          # Sevennth gear ratio
ig8 = 0.67          # Eighth gear ratio

i_f = 3.31           # Final Drive

gear_Ratios = [ig1,ig2,ig3,ig4,ig5,ig6,ig7,ig8]
print(" Gear ratios are :\n",gear_Ratios)
```

```
 Gear ratios are :
 [4.71, 3.14, 2.11, 1.67, 1.29, 1.0, 0.84, 0.67]
```

Exterior Dimensions

In [507]:

```
W_v = 1741          # Vehivle mass
W_d = 80            # Driver Mass
f_m = 1.05          # Vehocle mass factor

print("***** Exterior Dimensions **********")
m_v = round(f_m*W_v + W_d , 2)

print('-------------------------------------')
print('Total Vehicle Mass    :  ',m_v,'kg')
print("-------------------------------------")

f_a = 2.42

print("Frontal area          :   {} m^2".format(f_a))
print("-------------------------------------")

wr_s = 228.6
wr_d = 0.98 * wr_s

L = 176.5
B_w = 75.7
B_h = 51.6
Wb = 103.2
Gc = 4.0


print("Static wheel radius  :  ",wr_s,"mm")
print("-------------------------------------")

print("Dynamic wheel radius :  ",wr_d,"mm")
print("-------------------------------------")

print("Length               :  ", L , 'inches')
print("-------------------------------------")

print("Body Width           :  ",B_w,'inches')
print("-------------------------------------")

print("Body Height          :  ",B_h,"inches")
print("-------------------------------------")

print("WheelBase            :  ", Wb,'inches')
print("-------------------------------------")

print("Ground Clearence     :  ",Gc,'inches')
print("-------------------------------------")
```

```
***** Exterior Dimensions **********
-------------------------------------
Total Vehicle Mass   :   1908.05 kg
-------------------------------------
Frontal area         :   2.42 m^2
-------------------------------------
Static wheel radius  :   228.6 mm
-------------------------------------
Dynamic wheel radius :   224.028 mm
-------------------------------------
Length               :   176.5 inches
-------------------------------------
Body Width           :   75.7 inches
-------------------------------------
Body Height          :   51.6 inches
-------------------------------------
WheelBase            :   103.2 inches
-------------------------------------
Ground Clearence     :   4.0 inches
-------------------------------------
```

Performance coefficients

In [508]:

```
c_d = 0.36          # Drag Coefficient
c_r = 0.011         # Rolling Resistance Coefficient
g = 9.81            # Gravitational Acceleration
rho = 1.202         # Air Density(kg/m3)
n_t = 0.85          # Driveline Efficiency
theta = 0           # Gradient
```

RPM Range array

In [661]:

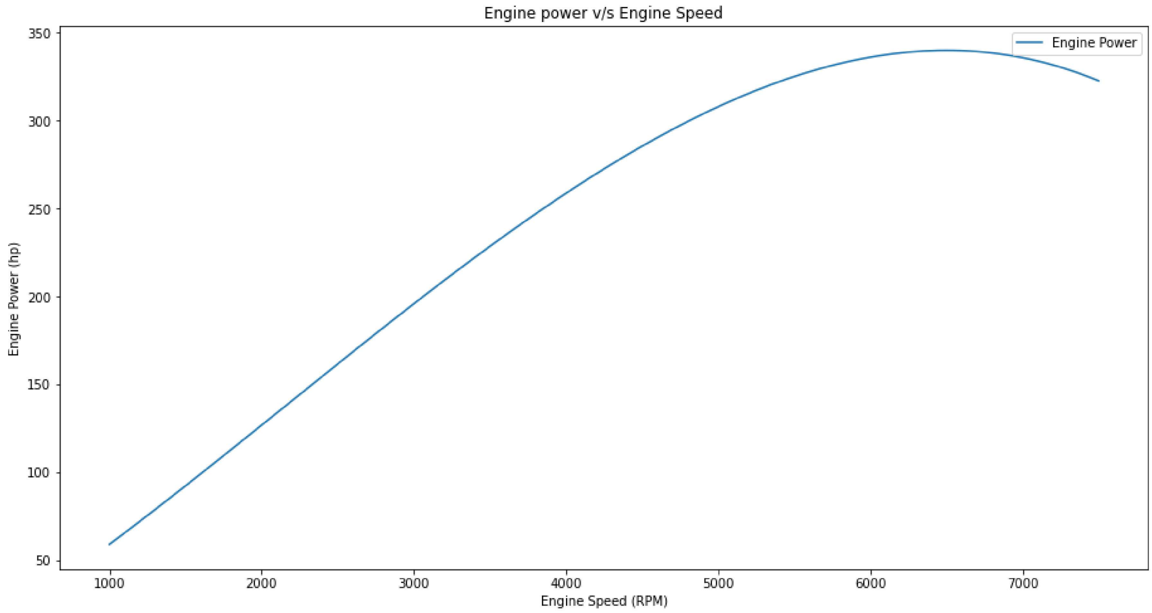
```
import numpy as np

N_min = 1000    # Minimum RPM
N_max = 7500    # Maximum RPM

N_e = np.array(list(range(N_min,N_max+1,50)))
print(N_e)
```

```
[1000 1050 1100 1150 1200 1250 1300 1350 1400 1450 1500 1550 1600 1650
 1700 1750 1800 1850 1900 1950 2000 2050 2100 2150 2200 2250 2300 2350
 2400 2450 2500 2550 2600 2650 2700 2750 2800 2850 2900 2950 3000 3050
 3100 3150 3200 3250 3300 3350 3400 3450 3500 3550 3600 3650 3700 3750
 3800 3850 3900 3950 4000 4050 4100 4150 4200 4250 4300 4350 4400 4450
 4500 4550 4600 4650 4700 4750 4800 4850 4900 4950 5000 5050 5100 5150
 5200 5250 5300 5350 5400 5450 5500 5550 5600 5650 5700 5750 5800 5850
 5900 5950 6000 6050 6100 6150 6200 6250 6300 6350 6400 6450 6500 6550
 6600 6650 6700 6750 6800 6850 6900 6950 7000 7050 7100 7150 7200 7250
 7300 7350 7400 7450 7500]
```

Engine Power vs Engine Speed

In [662]:

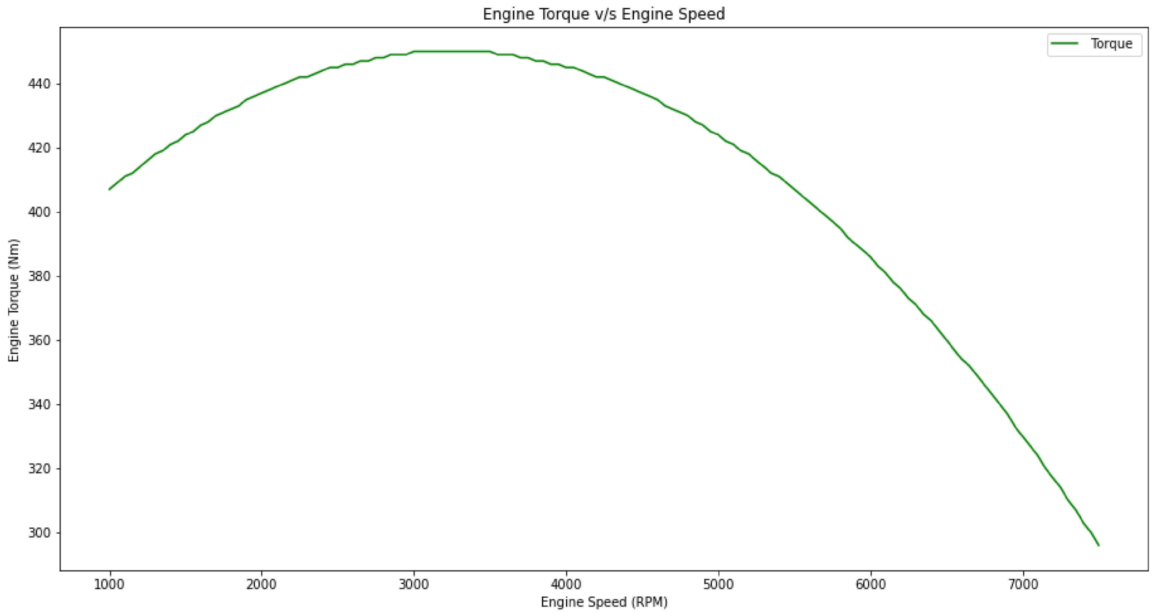
```
import matplotlib.pyplot as plt

N_ratio = N_e / N_Pmax

beta = N_ratio + N_ratio**2 - N_ratio**3

P_e = P_max * beta

plt.axes([0 , 0 , 2.00, 1.5])
Power_plot = plt.plot(N_e,P_e,label = 'Engine Power')
plt.legend()
plt.title("Engine power v/s Engine Speed")
plt.xlabel("Engine Speed (RPM)")
plt.ylabel("Engine Power (hp)")
plt.show(Power_plot)

#for i in range(len(N_e)):
#    print('For RPM : {} rpm , Engine power is : {} hp\n'.format(N_e[i] , P_engine[i]))
```



Engine Torque v/s Engine Speed

In [663]:

```
T_e =  np.round((P_e * 0.98 * 716 * 9.81) / N_e , 0)


plt.axes([0 , 0 , 2.0, 1.5])


Torque_plot = plt.plot(N_e, T_e,'g', label = ' Torque ')
plt.legend()
plt.title("Engine Torque v/s Engine Speed")
plt.xlabel("Engine Speed (RPM)")
plt.ylabel("Engine Torque (Nm)")

plt.show(Torque_plot)
print(T_e.shape)
```




```
(131,)
```

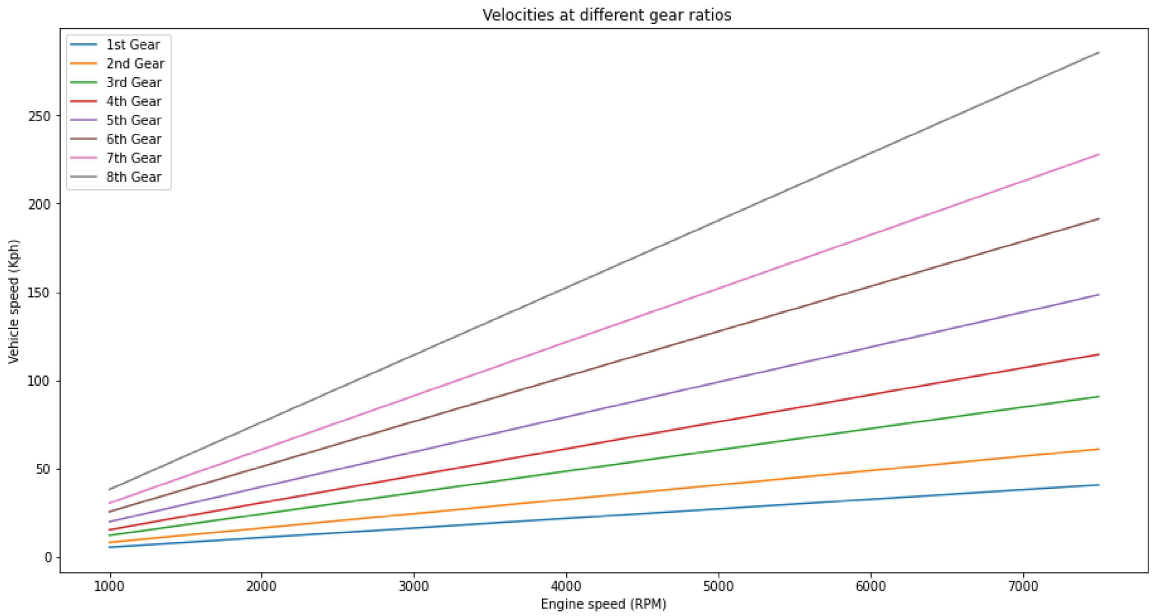
Velocites at different RPMs and Different Gear ratios

In [664]:

```
# First Gear      Gear ratio : 4.71
V1 = np.round(0.377 * wr_d *(N_e /( ig1 * i_f ) ) / 10**3 ,2)
print('First Gear      Gear ratio : 4.71')
#print(V1)
print('**********************************')

# Second Gear      Gear ratio : 3.14
V2 = np.round(0.377 * wr_d *(N_e /( ig2 * i_f ) ) / 10**3 ,2)
print('Second Gear     Gear ratio : 3.14')
#print(V2)
print('**********************************')

# Third Gear      Gear ratio : 2.11
V3 = np.round(0.377 * wr_d *(N_e /( ig3 * i_f ) ) / 10**3 ,2)
print('Third Gear      Gear ratio : 2.11')
#print(V3)
print('**********************************')

# Fourth Gear      Gear ratio : 1.67
V4 = np.round(0.377 * wr_d *(N_e /( ig4 * i_f ) ) / 10**3 ,2)
print('Fourth Gear     Gear ratio : 1.67')
#print(V4)
print('**********************************')

# Fifth Gear      Gear ratio : 1.29
V5 = np.round(0.377 * wr_d *(N_e /( ig5 * i_f ) ) / 10**3 ,2)
print('Fifth Gear      Gear ratio : 1.29')
#print(V5)
print('**********************************')

# Sixth Gear      Gear ratio : 1.00
V6 = np.round(0.377 * wr_d *(N_e /( ig6 * i_f ) ) / 10**3 ,2)
print('Sixth Gear      Gear ratio : 1.00')
#print(V6)
print('**********************************')

# Seventh Gear      Gear ratio : 0.84
V7 = np.round(0.377 * wr_d *(N_e /( ig7 * i_f ) ) / 10**3 ,2)
print('Seventh Gear    Gear ratio : 0.84')
#print(V7)
print('**********************************')

# Eigth Gear      Gear ratio : 0.67
V8 = np.round(0.377 * wr_d *(N_e /( ig8 * i_f ) ) / 10**3 ,2)
print('Eigth Gear      Gear ratio : 0.67')
#print(V8)
print('**********************************')



plt.axes([0 , 0 , 2.0, 1.5])
V1_plot = plt.plot(N_e,V1,label = "1st Gear")
plt.legend()
#plt.legend()

V2_plot = plt.plot(N_e,V2,label = "2nd Gear")
plt.legend()

V3_plot = plt.plot(N_e,V3,label = "3rd Gear")
plt.legend()
```

```
V4_plot = plt.plot(N_e,V4,label = "4th Gear")
plt.legend()

V5_plot = plt.plot(N_e,V5,label = "5th Gear")
plt.legend()

V6_plot = plt.plot(N_e,V6,label = "6th Gear")
plt.legend()

V7_plot = plt.plot(N_e,V7,label = "7th Gear")
plt.legend()

V8_plot = plt.plot(N_e,V8,label = "8th Gear")
plt.legend()
plt.title("  Velocities at different gear ratios ")
plt.xlabel("Engine speed (RPM)")
plt.ylabel("Vehicle speed (Kph)")

#plt.axis([0, 250, 0, 250])

plt.show(Velocity_plot)
#print(len(TE_1))
```

```
First Gear      Gear ratio : 4.71
**********************************
Second Gear     Gear ratio : 3.14
**********************************
Third Gear      Gear ratio : 2.11
**********************************
Fourth Gear     Gear ratio : 1.67
**********************************
Fifth Gear      Gear ratio : 1.29
**********************************
Sixth Gear      Gear ratio : 1.00
**********************************
Seventh Gear    Gear ratio : 0.84
**********************************
Eigth Gear      Gear ratio : 0.67
**********************************
```

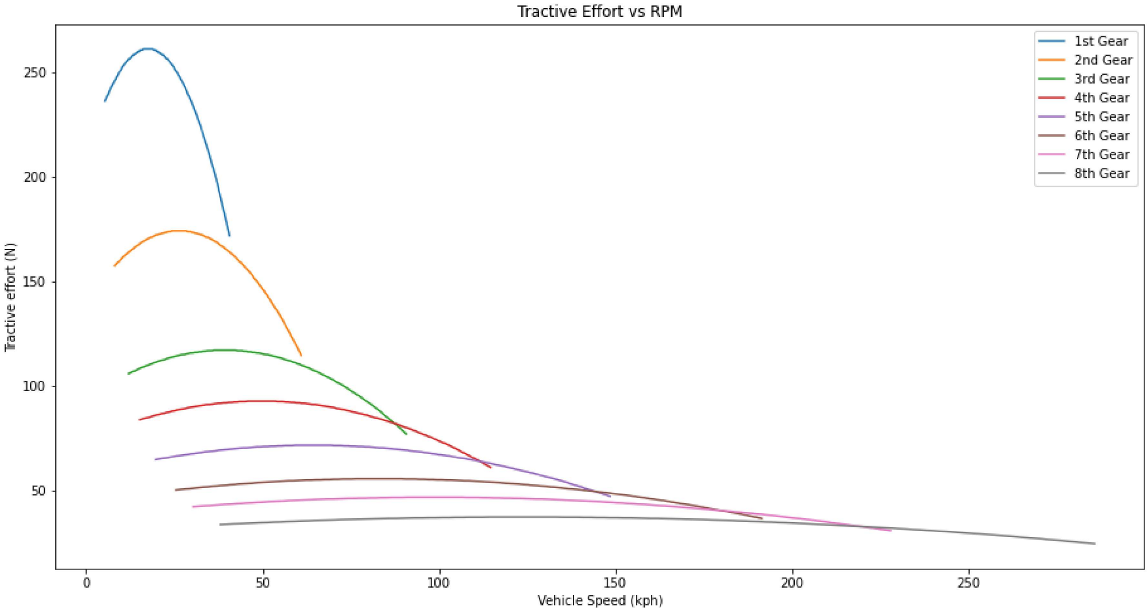


Tractive Effort Plot at different speeds and different Gear Ratios

In [665]:

```
TE_1 = np.round((T_e * ig1 * i_f * n_t * 9.80665 ) / wr_d , 2 )
#print(TE_1)

TE_2 = np.round((T_e * ig2 * i_f * n_t * 9.80665 ) / wr_d , 2 )
#print(TE_2)

TE_3 = np.round((T_e * ig3 * i_f * n_t * 9.80665 ) / wr_d , 2 )
#print(TE_3)

TE_4 = np.round((T_e * ig4 * i_f * n_t * 9.80665 ) / wr_d , 2 )
#print(TE_4)

TE_5 = np.round((T_e * ig5 * i_f * n_t * 9.80665 ) / wr_d , 2 )
#print(TE_5)

TE_6 = np.round((T_e * ig6 * i_f * n_t * 9.80665 ) / wr_d , 2 )
#print(TE_6)

TE_7 = np.round((T_e * ig7 * i_f * n_t * 9.80665 ) / wr_d , 2 )
#print(TE_7)

TE_8 = np.round((T_e * ig8 * i_f * n_t * 9.80665 ) / wr_d , 2 )
#print(TE_8)

#plt.axes([0 , 0 , 1.50, 1.5])
#TE_plot = plt.plot(V1,TE_1,V2,TE_2,V3,TE_3,V4,TE_4,V5,TE_5,V6,TE_6,V7,TE_7,V8,TE_8)


plt.axes([0 , 0 , 2.0, 1.5])
TE1_plot = plt.plot(V1,TE_1,label = "1st Gear")
plt.legend()
#plt.legend()

TE2_plot = plt.plot(V2,TE_2,label = "2nd Gear")
plt.legend()

TE3_plot = plt.plot(V3,TE_3,label = "3rd Gear")
plt.legend()

TE4_plot = plt.plot(V4,TE_4,label = "4th Gear")
plt.legend()

TE5_plot = plt.plot(V5,TE_5,label = "5th Gear")
plt.legend()

TE6_plot = plt.plot(V6,TE_6,label = "6th Gear")
plt.legend()

TE7_plot = plt.plot(V7,TE_7,label = "7th Gear")
plt.legend()

TE8_plot = plt.plot(V8,TE_8,label = "8th Gear")
plt.legend()

#plt.legend()
plt.title("Tractive Effort vs RPM")
plt.xlabel("Vehicle Speed (kph)")
plt.ylabel("Tractive effort (N)")
```

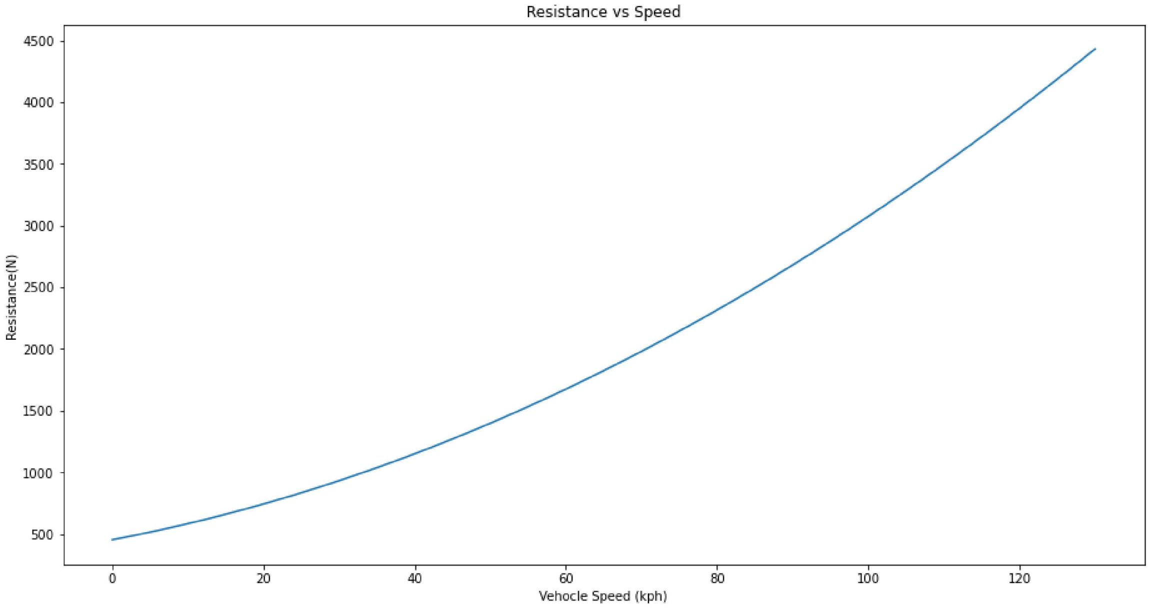
```
#plt.axis([0, 250, 0, 250])

plt.show(TE_plot)
#print(len(TE_1))
```



Resistance

In [666]:

```
R_a = ( 0.5*rho*f_a*c_d*(V8/3.6)**2 )
R_r= (0.015+0.00016*V8)*m_v*9.8
#print(R_r)

R = R_a + R_r
#print(R)
plt.axes([0 , 0 , 2.0, 1.5])
plt.title(" Resistance vs Speed ")
plt.xlabel("Vehocle Speed (kph)")
plt.ylabel("Resistance(N)")
plt.plot(R)
P = R*V8/3600
#print(P)
```



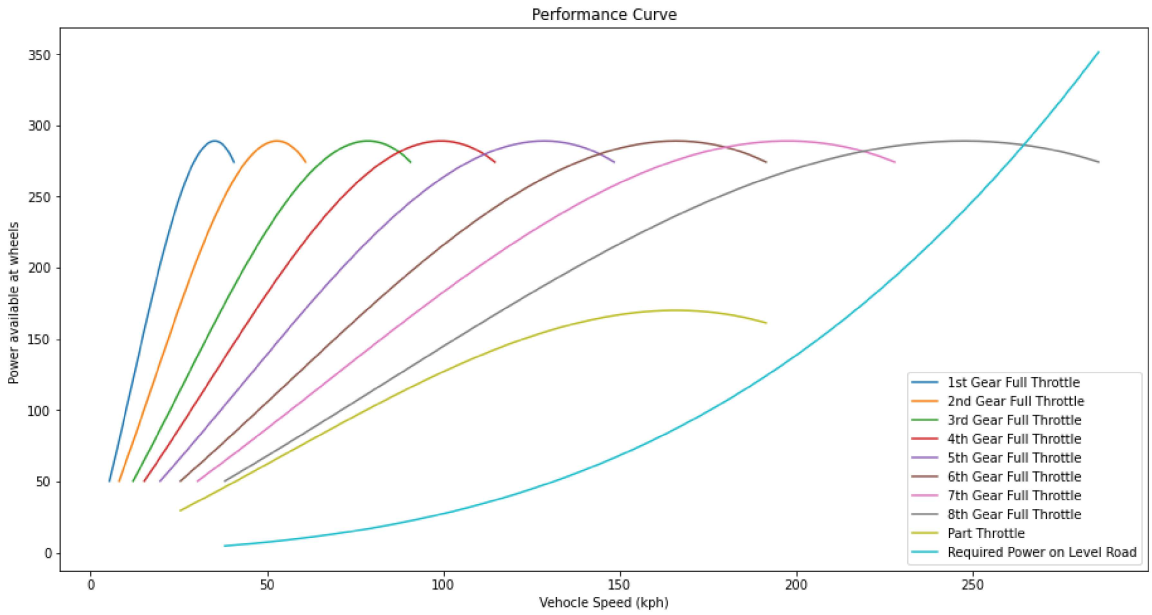
Performance Curve

In [667]:

```
a = P_e * n_t


plt.axes([0 , 0 , 2.0, 1.5])
Pw_1_plot = plt.plot(V1,a,label = "1st Gear Full Throttle")
plt.legend()
#plt.legend()

Pw_2_plot = plt.plot(V2,a,label = "2nd Gear Full Throttle")
plt.legend()

Pw_3_plot = plt.plot(V3,a,label = "3rd Gear Full Throttle")
plt.legend()

Pw_4_plot = plt.plot(V4,a,label = "4th Gear Full Throttle")
plt.legend()

Pw_5_plot = plt.plot(V5,a,label = "5th Gear Full Throttle")
plt.legend()

Pw_6_plot = plt.plot(V6,a,label = "6th Gear Full Throttle")
plt.legend()

Pw_7_plot = plt.plot(V7,a,label = "7th Gear Full Throttle")
plt.legend()

Pw_8_plot = plt.plot(V8,a,label = "8th Gear Full Throttle")
plt.legend()

part_throttle = P_e*0.5
Pw_6_part_plot = plt.plot(V6,part_throttle,label = "Part Throttle")
plt.legend()

plt.title(" Performance Curve ")
plt.xlabel("Vehocle Speed (kph)")
plt.ylabel("Power available at wheels ")

P_req_plot = plt.plot(V8,P,label = "Required Power on Level Road")
plt.legend()
```

Out[667]:

```
<matplotlib.legend.Legend at 0x2b88636abb0>
```



Gear ratio for maximum acceleration

Assuming values for moment of Inertias

In [668]:

```
I_w = 62.00
I_e = 6.2

tractive_resistance = TE_1[10]
velocity = V1[10]
print("The effective moment of inertia of the four road wheels and the rear axle togeth
er is :  ",I_w)
print("Moment of Inertia of Engine and Flywheel : ",I_e)
print("Tractive resistance at speed {} kph is {} N".format(velocity,tractive_resistance
))

Resistance = round(tractive_resistance + m_v*9.81/4,2)
print("Resistance R :",Resistance,'N')
radius = wr_d / 10**3

a = 450 * n_t

b = Resistance * (radius / 10**3)

c = (I_w /( radius *  9.81) ) + (( m_v*9.81 * radius )/9.81)

d = (I_e * n_t ) / (9.81 * radius)

alpha = (b/a)**2 + c/d

Gear_ratio = round((b/a) + (alpha**0.5),2)
print('\nGear Ratio for maximum acceleration is : ',Gear_ratio,':1')

max_acc = round((a*Gear_ratio - b) /( c + d*( Gear_ratio**2 )),2)
print("Maximum Acceleration is : ",max_acc,'m/s^2')
```

```
The effective moment of inertia of the four road wheels and the rear axle
together is :   62.0
Moment of Inertia of Engine and Flywheel :  6.2
Tractive resistance at speed 8.13 kph is 245.95 N
Resistance R : 4925.44 N

Gear Ratio for maximum acceleration is :  13.79 :1
Maximum Acceleration is :  5.78 m/s^2
```

Custom Gear ratios

In [669]:

```
print("The gear ratio are decided in the basis of gp having a ratio of ",1.25)
print("Since 6th Gear is direct drive so gear ratio at 6th gear is 1.00  ")

print("\nComputing other gear ratios on the basis of the above statement .....\n")

New_gear_ratios = [3.05 , 2.44 , 1.95 , 1.56 , 1.25 , 1.00 ,0.8 , 0.64]


for i in range(len(New_gear_ratios)):
    print('Gear',i+1,' : ',New_gear_ratios[i])


ig1_n = New_gear_ratios[0]
ig2_n = New_gear_ratios[1]
ig3_n = New_gear_ratios[2]
ig4_n = New_gear_ratios[3]
ig5_n = New_gear_ratios[4]
ig6_n = New_gear_ratios[5]
ig7_n = New_gear_ratios[6]
ig8_n = New_gear_ratios[7]
```

```
The gear ratio are decided in the basis of gp having a ratio of  1.25
Since 6th Gear is direct drive so gear ratio at 6th gear is 1.00

Computing other gear ratios on the basis of the above statement .....

Gear 1  :  3.05
Gear 2  :  2.44
Gear 3  :  1.95
Gear 4  :  1.56
Gear 5  :  1.25
Gear 6  :  1.0
Gear 7  :  0.8
Gear 8  :  0.64
```

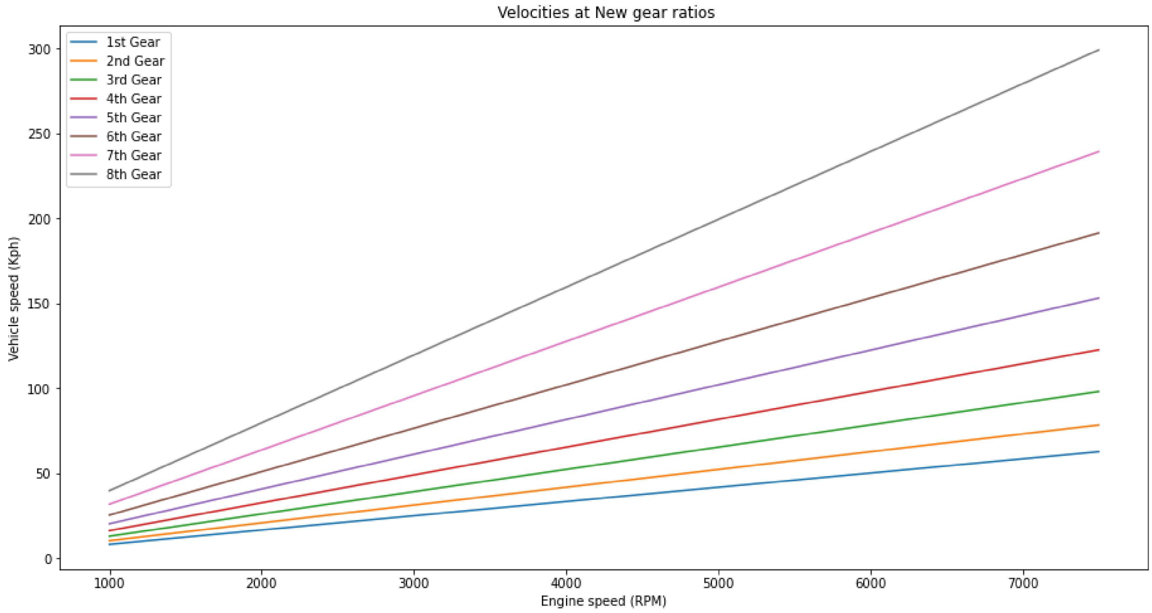
New Velocity Profile

In [670]:

```
# First Gear      Gear ratio : 3.05
V1_n = np.round(0.377 * wr_d *(N_e /( ig1_n * i_f ) ) / 10**3 ,2)
print('First Gear      Gear ratio : 3.05')
#print(V1)
print('**********************************')

# Second Gear      Gear ratio : 2.44
V2_n = np.round(0.377 * wr_d *(N_e /( ig2_n * i_f ) ) / 10**3 ,2)
print('Second Gear     Gear ratio : 2.44')
#print(V2)
print('**********************************')

# Third Gear      Gear ratio : 1.95
V3_n = np.round(0.377 * wr_d *(N_e /( ig3_n * i_f ) ) / 10**3 ,2)
print('Third Gear      Gear ratio : 1.95')
#print(V3)
print('**********************************')

# Fourth Gear      Gear ratio : 1.56
V4_n = np.round(0.377 * wr_d *(N_e /( ig4_n * i_f ) ) / 10**3 ,2)
print('Fourth Gear     Gear ratio : 1.56')
#print(V4)
print('**********************************')

# Fifth Gear      Gear ratio : 1.25
V5_n = np.round(0.377 * wr_d *(N_e /( ig5_n * i_f ) ) / 10**3 ,2)
print('Fifth Gear      Gear ratio : 1.25')
#print(V5)
print('**********************************')

# Sixth Gear      Gear ratio : 1.00
V6_n = np.round(0.377 * wr_d *(N_e /( ig6_n * i_f ) ) / 10**3 ,2)
print('Sixth Gear      Gear ratio : 1.00')
#print(V6)
print('**********************************')

# Seventh Gear      Gear ratio : 0.80
V7_n = np.round(0.377 * wr_d *(N_e /( ig7_n * i_f ) ) / 10**3 ,2)
print('Seventh Gear    Gear ratio : 0.80')
#print(V7)
print('**********************************')

# Eigth Gear      Gear ratio : 0.64
V8_n = np.round(0.377 * wr_d *(N_e /( ig8_n * i_f ) ) / 10**3 ,2)
print('Eigth Gear      Gear ratio : 0.64')
#print(V8)
print('**********************************')



plt.axes([0 , 0 , 2.0, 1.5])
V1_n_plot = plt.plot(N_e,V1_n,label = "1st Gear")
plt.legend()
#plt.legend()

V2_n_plot = plt.plot(N_e,V2_n,label = "2nd Gear")
plt.legend()

V3_n_plot = plt.plot(N_e,V3_n,label = "3rd Gear")
plt.legend()
```

```
V4_n_plot = plt.plot(N_e,V4_n,label = "4th Gear")
plt.legend()

V5_n_plot = plt.plot(N_e,V5_n,label = "5th Gear")
plt.legend()

V6_n_plot = plt.plot(N_e,V6_n,label = "6th Gear")
plt.legend()

V7_n_plot = plt.plot(N_e,V7_n,label = "7th Gear")
plt.legend()

V8_n_plot = plt.plot(N_e,V8_n,label = "8th Gear")
plt.legend()
plt.title("  Velocities at New gear ratios ")


plt.xlabel("Engine speed (RPM)")
plt.ylabel("Vehicle speed (Kph)")

#plt.axis([0, 250, 0, 250])


plt.show(Velocity_plot)
#print(len(TE_1))
```

```
First Gear       Gear ratio : 3.05
**********************************
Second Gear      Gear ratio : 2.44
**********************************
Third Gear       Gear ratio : 1.95
**********************************
Fourth Gear      Gear ratio : 1.56
**********************************
Fifth Gear       Gear ratio : 1.25
**********************************
Sixth Gear       Gear ratio : 1.00
**********************************
Seventh Gear     Gear ratio : 0.80
**********************************
Eigth Gear       Gear ratio : 0.64
**********************************
```

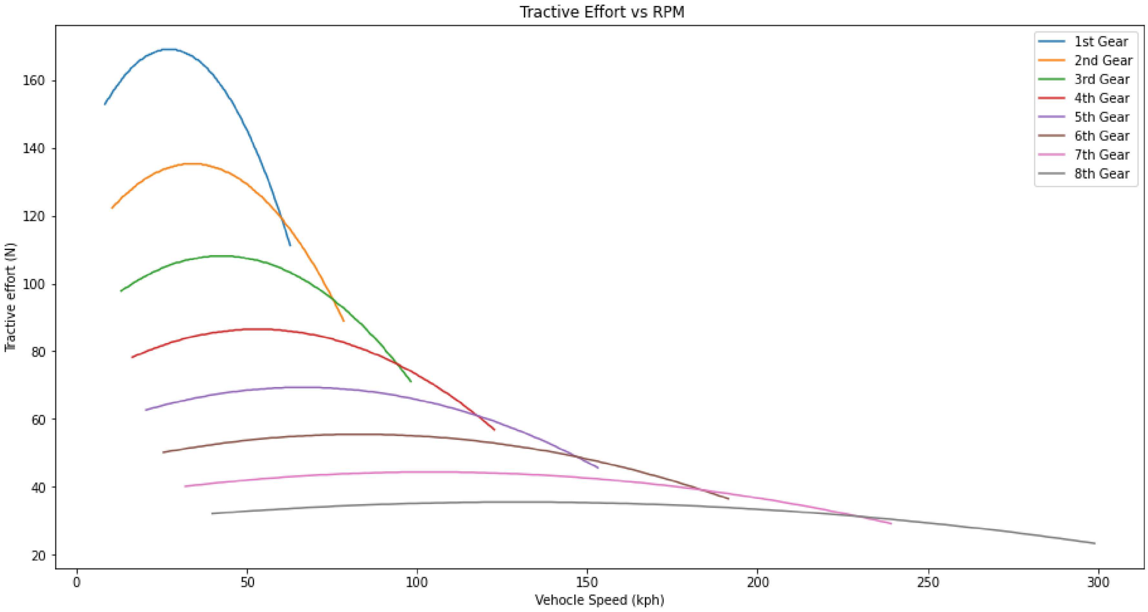


New Tractive Effort

In [671]:

```
TE_1_n = np.round((T_e * ig1_n * i_f * n_t * 9.80665 ) / wr_d , 2 )
#print(TE_1)

TE_2_n = np.round((T_e * ig2_n * i_f * n_t * 9.80665 ) / wr_d , 2 )
#print(TE_2)

TE_3_n = np.round((T_e * ig3_n * i_f * n_t * 9.80665 ) / wr_d , 2 )
#print(TE_3)

TE_4_n = np.round((T_e * ig4_n * i_f * n_t * 9.80665 ) / wr_d , 2 )
#print(TE_4)

TE_5_n = np.round((T_e * ig5_n * i_f * n_t * 9.80665 ) / wr_d , 2 )
#print(TE_5)

TE_6_n = np.round((T_e * ig6_n * i_f * n_t * 9.80665 ) / wr_d , 2 )
#print(TE_6)

TE_7_n = np.round((T_e * ig7_n * i_f * n_t * 9.80665 ) / wr_d , 2 )
#print(TE_7)

TE_8_n = np.round((T_e * ig8_n * i_f * n_t * 9.80665 ) / wr_d , 2 )
#print(TE_8)

#plt.axes([0 , 0 , 1.50, 1.5])
#TE_plot = plt.plot(V1,TE_1,V2,TE_2,V3,TE_3,V4,TE_4,V5,TE_5,V6,TE_6,V7,TE_7,V8,TE_8)


plt.axes([0 , 0 , 2.0, 1.5])
TE1_n_plot = plt.plot(V1_n,TE_1_n,label = "1st Gear")
plt.legend()
#plt.legend()

TE2_n_plot = plt.plot(V2_n,TE_2_n,label = "2nd Gear")
plt.legend()

TE3_n_plot = plt.plot(V3_n,TE_3_n,label = "3rd Gear")
plt.legend()

TE4_n_plot = plt.plot(V4_n,TE_4_n,label = "4th Gear")
plt.legend()

TE5_n_plot = plt.plot(V5_n,TE_5_n,label = "5th Gear")
plt.legend()

TE6_n_plot = plt.plot(V6_n,TE_6_n,label = "6th Gear")
plt.legend()

TE7_n_plot = plt.plot(V7_n,TE_7_n,label = "7th Gear")
plt.legend()

TE8_n_plot = plt.plot(V8_n,TE_8_n,label = "8th Gear")
plt.legend()

#plt.legend()
plt.title(" Tractive Effort vs RPM")
plt.xlabel("Vehocle Speed (kph)")
plt.ylabel("Tractive effort (N)")
```

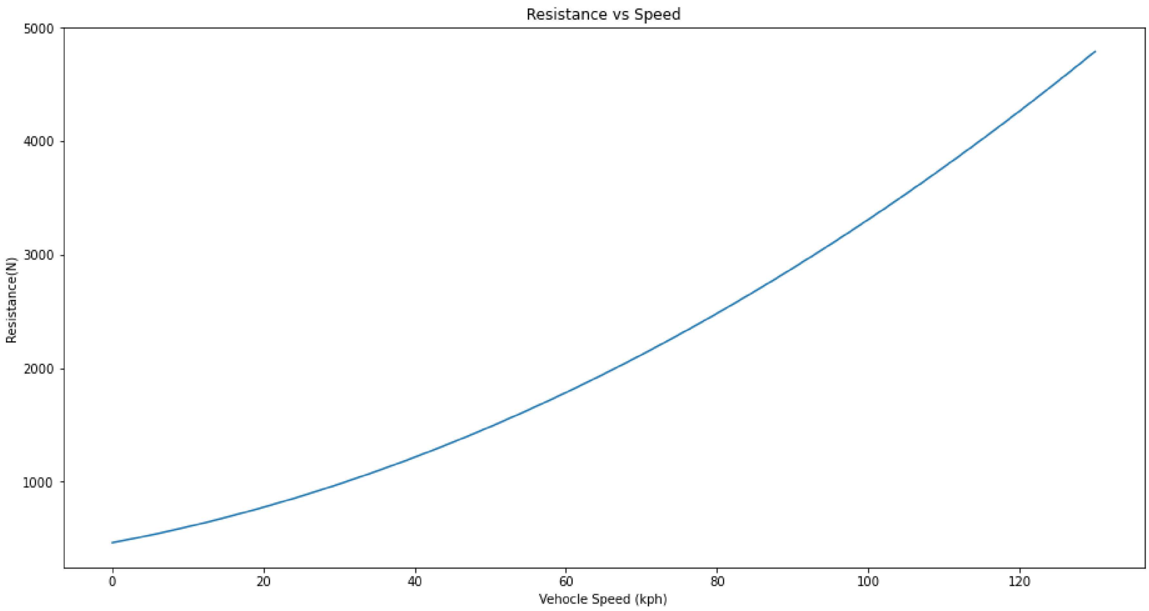
```
#plt.axis([0, 250, 0, 250])

plt.show(TE_plot)
#print(len(TE_1))
```



New Resistance

In [672]:

```
R_a_n = ( 0.5*rho*f_a*c_d*(V8_n/3.6)**2 )
R_r_n= (0.015+0.00016*V8_n)*m_v*9.8
#print(R_r)

R_n = R_a_n + R_r_n
#print(R)
plt.axes([0 , 0 , 2.0, 1.5])
plt.title(" Resistance vs Speed ")
plt.xlabel("Vehocle Speed (kph)")
plt.ylabel("Resistance(N)")
plt.plot(R_n)
P_n = R_n*V8_n/3600
#print(P)
```



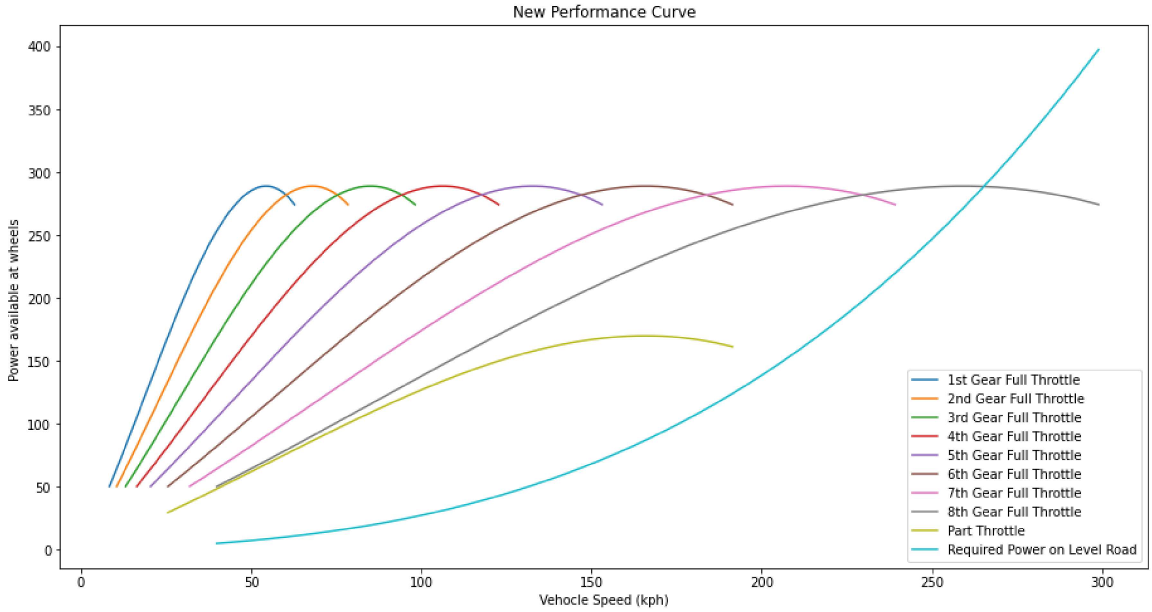
New Performance Curves

In [673]:

```
a = P_e * n_t


plt.axes([0 , 0 , 2.0, 1.5])
Pw_1_plot = plt.plot(V1_n,a,label = "1st Gear Full Throttle")
plt.legend()
#plt.legend()

Pw_2_plot = plt.plot(V2_n,a,label = "2nd Gear Full Throttle")
plt.legend()

Pw_3_plot = plt.plot(V3_n,a,label = "3rd Gear Full Throttle")
plt.legend()

Pw_4_plot = plt.plot(V4_n,a,label = "4th Gear Full Throttle")
plt.legend()

Pw_5_plot = plt.plot(V5_n,a,label = "5th Gear Full Throttle")
plt.legend()

Pw_6_plot = plt.plot(V6_n,a,label = "6th Gear Full Throttle")
plt.legend()

Pw_7_plot = plt.plot(V7_n,a,label = "7th Gear Full Throttle")
plt.legend()

Pw_8_plot = plt.plot(V8_n,a,label = "8th Gear Full Throttle")
plt.legend()

part_throttle = P_e*0.5
Pw_6_part_plot = plt.plot(V6_n,part_throttle,label = "Part Throttle")
plt.legend()

#POWER_REQ = R * V8 / 270

P_req_plot = plt.plot(V8_n,P_n,label = "Required Power on Level Road")
plt.legend()


plt.title(" New Performance Curve ")
plt.xlabel("Vehocle Speed (kph)")
plt.ylabel("Power available at wheels ")
```

Out[673]:

```
Text(0, 0.5, 'Power available at wheels ')
```



Comparison Between Manufacturer's and Custom Gear ratios

In [674]:

```
TE_original_max = np.max(TE_1)
TE_new_max = np.max(TE_1_n)

print("Manufacturer's Tractive effort                    : ",TE_original_max,'N')
print("New Tractive effort based on Custom Gear ratios : ",TE_new_max,'N')
```

```
Manufacturer's Tractive effort                    :  261.03 N
New Tractive effort based on Custom Gear ratios :  169.04 N
```