File - C:\Users\geoff\Desktop\FlyOx Concept\Python\API\perfIntroAPI.py

```
 1
 2 # coding: utf-8
 3 from math import sqrt
 4
 5 class atmosphere:
 6     def __init__ (self,altitude,rhoSL):
 7 #             self.initialWeight = initialWeight
 8 #             self.finalWeight = finalWeight
 9         self.altitude = altitude
10         self.rhoSL = rhoSL
11         #             self.cruiseSpeed = cruiseSpeed
12 #             self.S = S
13
14     def pressure(self):
15         altitudePresssure = 2116*(1-(0.0000068756*self.altitude))**5.2561
16         return altitudePresssure
17
18     def pressureRatio(self):
19         pressureRatio = (1-(0.0000068756*self.altitude))**5.2561
20         return pressureRatio
21
22     def density(self):
23         ### fetch this function in other script
24         altitudeDensity = self.rhoSL*(1-0.0000068756*self.altitude)**4.2561
25         return altitudeDensity
26
27     def densityRatio(self):
28         ### fetch this function in other script
29         densityRatio = (1-0.0000068756*self.altitude)**4.2561
30         return densityRatio
31
32     def temperature(self):
33         temperature = 518.7*(1-0.0000068756*self.altitude)
34         return temperature
35
36     def temperatureRatio(self):
37         temperatureRatio = (1-0.0000068756*self.altitude)
38         return temperatureRatio
39
40
41 # In[97]:
42
43 class speeds:
44
45
46     def __init__(self,altitude,altitude_temperature,altitude_pressure,cruiseSpeed,gas_constant,gamma,
   altitude_density,
47                  altitude_pressureRatio,rhoSL):
48         self.altitude = altitude
49         self.temperature = altitude_temperature
50         self.pressure = altitude_pressure/altitude_pressureRatio ######AAAAAAAAAAAAAAAAAA confirm i
   am guessing
51         self.altitude_pressure = altitude_pressure
52         self.cruiseSpeed = cruiseSpeed
53         self.gas_constant = gas_constant
54         self.gamma = gamma
55         self.altitude_density = altitude_density
56         self.altitude_pressureRatio = altitude_pressureRatio
57         self.rhoSL = rhoSL
58
59
60     def soundSpeed(self):
61         soundSpeed = sqrt(self.gamma*self.temperature*self.gas_constant)
62         return soundSpeed
63
64     def dynamicPressure(self):
65         dynamicPressure = self.altitude_pressure*((1 + 0.2*(self.cruiseSpeed/self.soundSpeed())**2)**3
   .5 - 1)
66         return dynamicPressure
67
68     def EAS(self):
69         EAS = self.cruiseSpeed * sqrt(self.altitude_density/self.rhoSL)
70         return EAS
71
72     def CAS(self):
73         CAS = self.EAS()*sqrt(self.altitude_pressureRatio**-1)*sqrt((((self.dynamicPressure()/(self.
   pressure/self.altitude_pressureRatio))+1)**0.2857 - 1)/( (((self.dynamicPressure()/(self.pressure))+1)
```

```
**0.2857 - 1)))
        return CAS


# In[ ]:


class flightEnvelope:

    def __init__(self,finalMTOW,S,maxSpeed,stallSpeed,negCLmin,rhoSL):
        self.finalMTOW = finalMTOW
        self.S = S
        self.maxSpeed = maxSpeed
        self.stallSpeed = stallSpeed         ########## make this EAS
        self.negCLmin = negCLmin
        self.rhoSL = rhoSL

    def loadFactor(self):
        n = 2.1 + ( 24000/(self.finalMTOW + 10000))
        return n

    def negloadFactor(self):
        negloadFactor = -0.4*flightEnvelope.loadFactor(self)
        return negloadFactor

    def minCruiseSpeed(self):
        vcmin = 33*sqrt(self.finalMTOW/self.S)
        return vcmin   #This is in KEAS by default

    def maxCruiseSpeed(self):
        vcmax = 0.9 * self.maxSpeed   ############## convert to KTAS if not
        return vcmax

    def diveSpeed(self):
        vD = 1.40 * flightEnvelope.minCruiseSpeed(self)
        return vD  ### dive speed should be greater than this value always

    def maneuveringSpeed(self):
        vA = self.stallSpeed *sqrt(flightEnvelope.loadFactor(self))
        return vA

    def negmaneuveringSpeed(self):
        VG = sqrt( (2*abs(flightEnvelope.negloadFactor(self))*self.finalMTOW)/(self.rhoSL*self.S*abs(
    self.negCLmin)))
        return VG/1.688

    def maxGustSpeed(self):
        pass
```