According to Ref. [1], fin flutter speed is governed by an equation where geometric dimension of fins, atmospheric pressure, speed of sound and shear modulus of fin material appear. Since geometric dimensions and shear modulus of the fins are not varying during the whole flight, remaining parameters, namely atmospheric pressure and speed of sound is carefully examined so that fin flutter speed can be revealed for the whole flight up to apogee.

Atmospheric pressure and temperature for each altitude are modelled using the Ref. [2] provided by NASA. Depending on the temperature, speed of sound is determined by equation , where the only variable is temperature. Flutter speed is, then accordingly, calculated with respect to altitude as shown in Figure 1. The same figure monitors the expected speed of the rocket with respect to altitude to compare them. Obviously, the fins flutter speed is significantly higher than expected rocket speed, which indicates that fins are not concerned to be exposed to flutter.

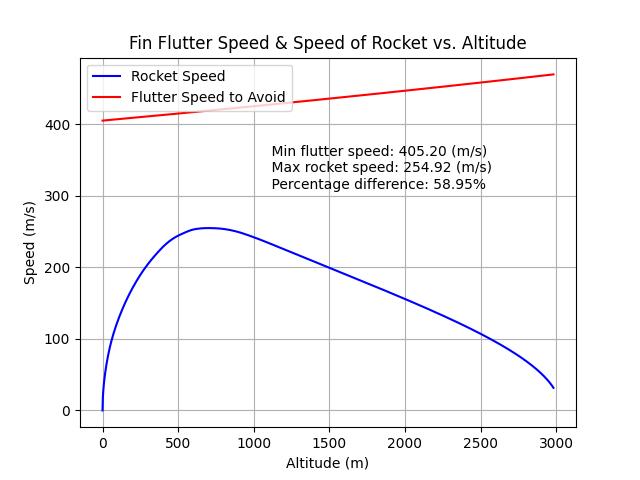


Figure . Fin Flutter Speed and Speed of Rocket vs. Altitude

Figure 1 also shows that minimum fin flutter speed, which is , is higher than the maximum expected rocket velocity, which is . These numerical values agree with the requirement 8.2.2 in Design, Test, & Evaluation Guide (Revision: 2025 V1.1.4).

The Python script (Ref. [3]) used for calculations can be referred to for further calculation details.

References

[1] <https://apogeerockets.com/education/downloads/Newsletter291.pdf>

[2] <https://www.grc.nasa.gov/WWW/K-12/airplane/atmosmet.html>

[3]:

## Refer to below link for further info related to calculations:

## https://apogeerockets.com/education/downloads/Newsletter291.pdf

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

## Please make sure that you use metric units!

## At the and we will convert the required ones into imperial units to use the formula as it is.

## JUST CHANGE PARAMETERS: G, t, cr, ct, b and max altitude. other parameters and formulas are

## inherently determined by the nature of the problem.

# Shear modulus of fin material (in Pa)

# Reference: https://www.matweb.com/search/datasheet.aspx?matguid=39e40851fc164b6c9bda29d798bf3726&ckck=1

G = 5 \* 10\*\*9

# Geometric dimensions of the fin (thickness, root&tip chord, semi span)(in m)

t = 5 \* 10\*\*-3

cr = 400 \* 10\*\*-3

ct = 80 \* 10\*\*-3

b = 145 \* 10\*\*-3

# Import expected speed of the rocket UP TO APOGEE

data = pd.read\_csv('Altitude\_vs\_velocity\_rev3.csv')

x = data['Altitude (ft)'].astype(float)

y = data['Vertical velocity (m/s)'].astype(float)

# Convert altitude to meters

x = x \* 0.3048

# Define altitude range from zero to apogee (in m)

altitudeAtApogee = int(x.iloc[-1])

altitude = np.linspace(0, altitudeAtApogee, altitudeAtApogee)

# Linearly interpolate to increase the number of points

rocketSpeed = np.interp(altitude, x, y)

# Wing area (in m^2), aspect ratio (no unit) and taper ratio (no unit)

S = 0.5\*(cr+ct)\*b

AR = b\*\*2/S

l = ct/cr

# Temperature (degree celcius NOT KELVIN) variation w.r.t altitude

# Reference: https://www.grc.nasa.gov/WWW/K-12/airplane/atmosmet.html

temperature = 15.04 - 0.00649\*altitude

# Pressure (KPa NOT Pa)

# Reference: https://www.grc.nasa.gov/WWW/K-12/airplane/atmosmet.html

pressure = ((temperature+273.1)/288.08)\*\*5.256 \* 101.29

# Convert it to Pa

pressure = pressure \* 10\*\*3

# Speed of sound = sqrt(k\*Rair\*T), T(Kelvin), Rair = 287.05 J/kgK

a = (1.4 \* 287.05 \* (temperature+273.15))\*\*0.5

# Now, we HAVE TO convert the units according to reference document.

# Just G and P will be enough since the others are just ratio.

# Since the a remains in m, the flutter speed result will be in m as well.

G = G \* 0.0001450377

pressure = pressure \* 0.00014503773800722

# Finally, calculate the fin flutter speed flutterSpeed(m/s)

# Since I need element\*wise multiplication, I will calculate the sqrt term seperately.

sqrtTerm = ((G\*2\*(AR+2)\*(t/cr)\*\*3)/(1.337\*AR\*\*3\*pressure\*(l+1)))\*\*0.5

flutterSpeed = np.multiply(a, sqrtTerm)

text = f'''

    Min flutter speed: {min(flutterSpeed):.2f} (m/s)

    Max rocket speed: {max(rocketSpeed):.2f} (m/s)

    Percentage difference: {(min(flutterSpeed)-max(rocketSpeed))\*100/max(rocketSpeed):.2f}%'''

# Print the min flutter speed, max rocket speed and percentage difference.

print(text)

# Plot both speed in one plot.

plt.plot(altitude, rocketSpeed, 'b-', label="Rocket Speed")

plt.plot(altitude, flutterSpeed, 'r-', label="Flutter Speed to Avoid")

plt.text(1000, 310, text, fontsize = 10)

plt.title('Fin Flutter Speed & Speed of Rocket vs. Altitude')

plt.xlabel('Altitude (m)')

plt.ylabel('Speed (m/s)')

plt.legend(loc="upper left")

plt.grid()

plt.savefig('speedComparison.jpg')

plt.show()