Parametric Rocket: Phase 3

Python analysis of the flight data

What I did

In Phase 3 I used Python and pandas to analyze the three OpenRocket flight exports for the F42, G80-7, and G125-9. I wrote one function to load and clean an OpenRocket CSV, loaded all three files into a dictionary, built a summary table of each motor's max altitude, max velocity, max acceleration, and time to apogee, and plotted altitude, velocity, and acceleration against time with all three motors overlaid on each chart.

Loading and cleaning the data

OpenRocket's CSV exports start with comment lines that begin with #, so I could not load them straight into pandas. My function opens the file, loops through each line, skips the comments and blank lines, and splits each remaining line on commas. It then builds a DataFrame with named columns for time, altitude, velocity, and acceleration, and converts everything to floats.

```python
def load_openRocket_csv(filepath):
    with open(filepath) as file:
        lines = file.readlines()

    data_rows = []
    for line in lines:
        line = line.strip()
        if line.startswith('#') or line == '':
            continue
        parts = line.split(',')
        data_rows.append(parts)

    columns = ['time_s', 'altitude_m', 'velocity_ms', 'acceleration_ms2']
    df = pd.DataFrame(data_rows, columns=columns)

    df = df.astype(float)

    return df
```

Figure 1. The function that loads and cleans an OpenRocket flight CSV.

Summary table

For each motor I pulled the max altitude, max velocity, and max acceleration, and found the time to apogee by taking the time at the max altitude. These match the values in OpenRocket's own simulation table from Phase 2, which tells me the files loaded correctly.

```
[Running] python -u "c:\Users\Joshu\OneDrive\Desktop\OpenRocket Sims\Rocket 1.1\rocket_analysis.py"
 Motor  Max Altitude (m)  Max Velocity (m/s)  Max Acceleration (m/s^2)  Time to Apogee (s)
   F42             507.2               185.0                     316.2                8.07
 G80-7             809.1               302.5                     380.1                9.12
G125-9             866.6               342.3                     510.3                9.63

[Done] exited with code=0 in 34.402 seconds
```

Figure 2. Summary table printed by my script.

Flight charts

Altitude. Only the F42 comes near the target range. It peaks at 507.2 m, just above the 500 m line, while the G80-7 and G125-9 overshoot to 809.1 m and 866.6 m. After apogee, each descent is a nearly straight line, so the rocket falls at a steady speed under the parachute.

Velocity. All three motors peak about 1 second after launch, then slow down until the curve is near zero at apogee (8.07, 9.12, and 9.63 s). When the parachute opens, the curve lifts slightly and then levels off at a few m/s. The small bump on the F42 curve near 9.5 s is the parachute opening, which shows the late ejection delay I found in Phase 2.

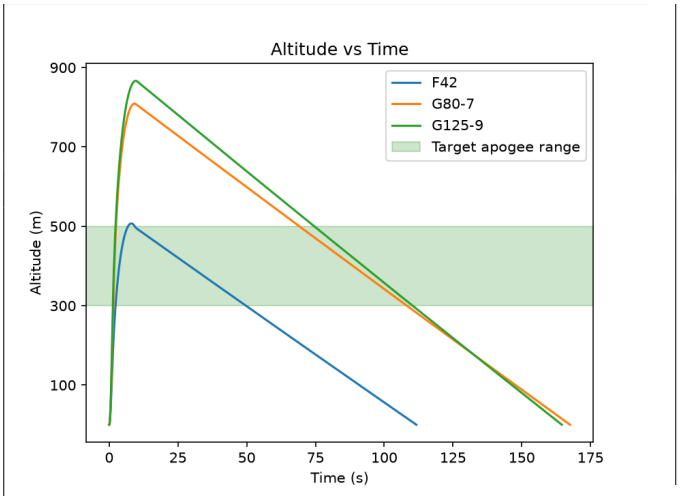


Figure 3. Altitude vs. time, with the 300-500 m target range shaded.

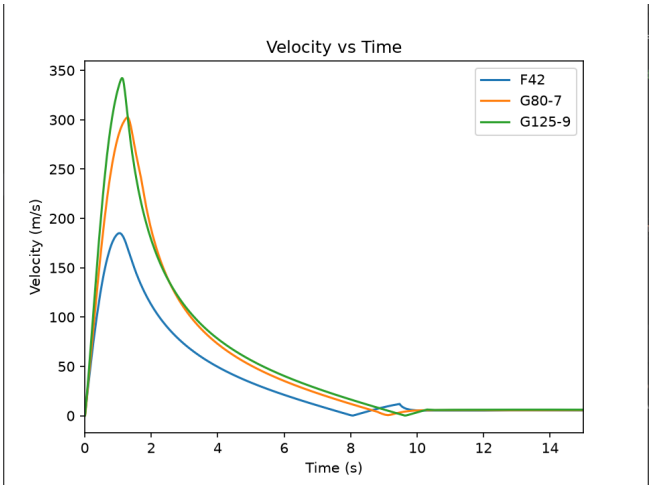


Figure 4. Velocity vs. time (first 15 seconds).

Acceleration. The G125-9 has the sharpest launch spike at 510.3 m/s^2 (about 52 g), compared with 316.2 m/s^2 for the F42. All three curves drop quickly once the motor stops pushing. The small spike near 9.5 s on the F42 curve is the same parachute event.

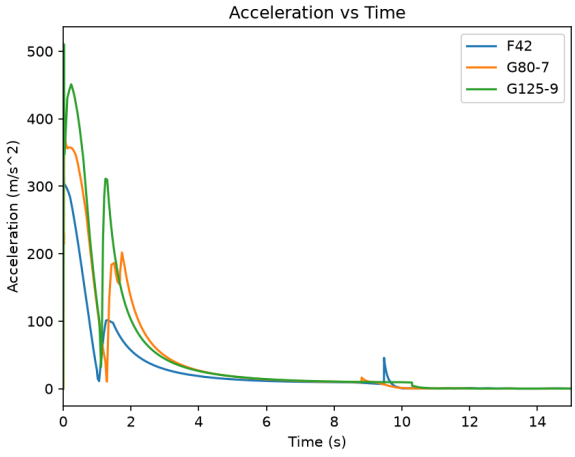


Figure 5. Acceleration vs. time (first 15 seconds).

What I learned

- **Real data needs cleaning first.** The comment lines in the OpenRocket exports would have broken a plain load. Handling them in one function meant all three motors went through the same steps.
- **A value and its time come from two calls.** `.max()` gives the largest value, `.idxmax()` gives the row where it happens, and `.loc[]` pulls the time from that row. That is how I got time to apogee.
- **Check your analysis against a known result.** My max altitude, velocity, and acceleration match OpenRocket's table. The one small difference is the G125-9 time to apogee, 9.63 s here and 9.66 s in OpenRocket.
- **Charts show what a table can't.** The summary table says the F42 is closest to the target, but the plots show the shape of each flight, including the parachute opening as a small bump in velocity and acceleration.
- **Core tools were enough.** I used loops, conditionals, string handling, and basic pandas and matplotlib, which are the PD101 skills.