Ex0 Report:

GITHUB: https://github.com/LidorMalichi/Ex0.git

To write this report, we primarily relied on the live broadcast of the Bereshit landing.
Please note that our focus is on the braking stage, which begins at the 25:00-minute mark.

In the crash report, we focus on three main parameters:

- **Altitude (meters):** Indicates the spacecraft's height above the lunar surface.
- **Horizontal Velocity (m/s):** Measures the spacecraft's speed across the surface.
- **Vertical Velocity (m/s):** Measures the spacecraft's speed toward or away from the surface.

1. At 25:03 Bereshit shift from Orientation sub-state to Breaking.
2. 25:18 the aircraft supress the point of no return. The spacecraft data as follows
    - Altitude: 25248
    - Horizontal velocity: 1697.7
    - Vertical velocity: 34.9
    - Fuel Mass (KG): 215.06

3. Until 33:03, the spacecraft proceeded according to plan, where:
    - Altitude: 13.674
    - Distance: 178.288KM
    - Horizontal speed: 928.8
    - Vertical speed: 24.8
    - Battery Charge: 71.3
    - Tilt angle: 58

4. The failure began with the IMU 2 (Inertial Measurement Unit) gyroscope malfunctioning. Shortly after, an attempt was made to restart IMU 2, which may have led to the failure of the second IMU as well.
5. At 33:48, comminution with the NASA ground unit was lost, as the data being transmitted was inaccurate.

6. Communication is back at 34:23 as the updated data is as follows:
    - Altitude: 10997
    - Horizontal speed: 880.2
    - Vertical speed: 47.9
    - Fuel: 105.06

7. Velocity is increased by every second, at 35:00 can be observed :
    - Altitude: 8814
    - Horizontal velocity: 900.0
    - Vertical velocity: 74.2

    Instead of gradually reducing both horizontal and vertical velocity, the spacecraft continued to accelerate before landing. The vertical velocity, indicated in red, clearly shows this increase.

8. The spacecraft continued to lose altitude while its speed increased. The crew struggled to determine whether the main engine was active, as the rapid descent and high velocity suggested a lack of thrust

9. At 35:44, an attempt was made to restart the spacecraft.

10. At 36:40, a status update was received with the following data:

    - Altitude: 547
    - Horizontal velocity: 948.0
    - Vertical velocity: 131.1

These values indicate that the spacecraft was on a collision course with the Moon and was expected to crash within seconds, likely resulting in total disintegration upon impact.

<u>THE SIMULATION</u>

In this simulation, we developed a dynamic landing control strategy using two dedicated **PID (Proportional-Integral-Derivative) controllers** to guide the Bereshit lander safely and efficiently to the lunar surface.

1. Vertical Speed PID Controller (Thrust Control - NN)

The first PID controller regulates the lander's **thrust (NN)**, which directly influences the **vertical speed (VS)**. This controller continuously adjusts the thrust level to match a target vertical speed that varies according to the current altitude. The error input to the PID is the difference between the current and target vertical speed, allowing the controller to throttle up or down in response. This tightly coupled feedback ensures the descent remains smooth while minimizing fuel consumption.

2. Angle Control PID Controller (Attitude Adjustment)

The second PID controller is responsible for managing the **lander's angle**, which in turn affects the balance between vertical and horizontal motion. This controller becomes active when the altitude drops below **2000 meters**—prior to that, the lander maintains its initial descent angle to conserve control efforts and reduce unnecessary side thrust. As the lander approaches the lunar surface, the PID parameters are **dynamically tuned** to become more responsive, ensuring a gentle and vertically aligned touchdown.

3. Horizontal Speed PID Controller (Thrust Control - NN)

The third PID controller is responsible for the **lander's thrust (NN)**, During the breaking phase. This controller is responsible for continuously adjusting the thrust level to match a target horizontal speed that varies according to the current altitude. It works solely on the breaking phase.

4. Adaptive Targeting via Linear Interpolation

To enhance the flexibility and realism of the simulation, we employed **Linear Interpolation** to determine both:

- The **target vertical speed**, based on current altitude.
- The **desired angle**, also based on altitude.

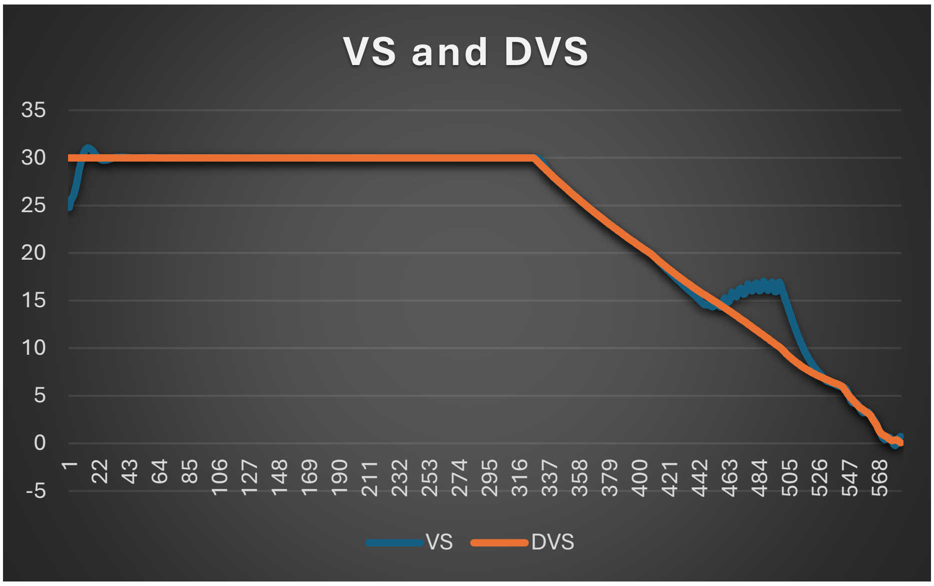
This interpolation technique ensures the system can adapt smoothly to changes in descent profile without hardcoded thresholds. It also makes the simulation more robust and extensible for future refinements.

- The **target horizontal speed,** based on altitude.

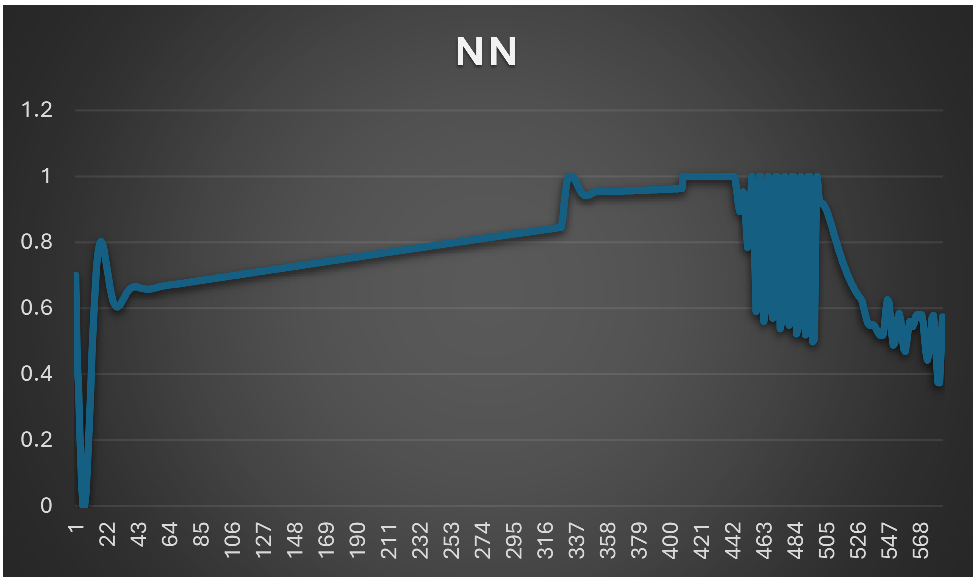
Simulation Results:●

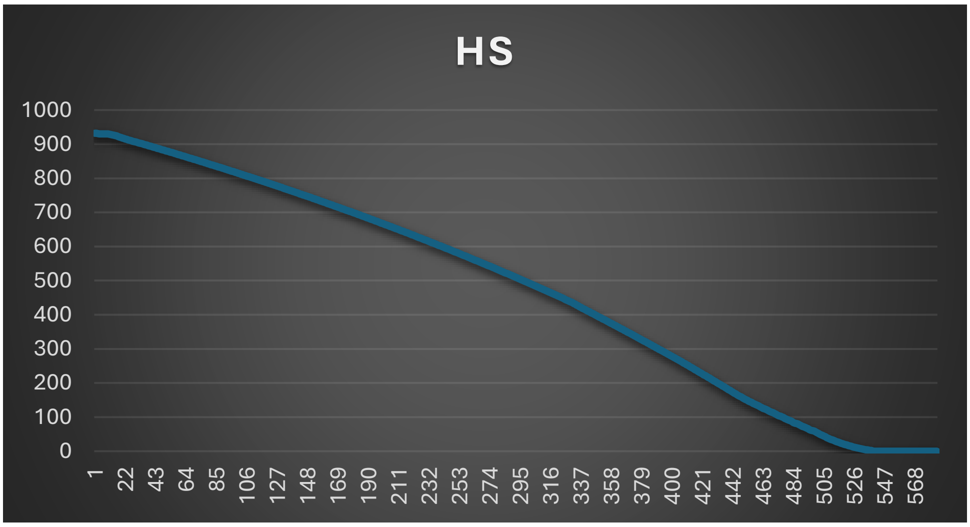
The full results are saved in BereshitSimulation.csv file.

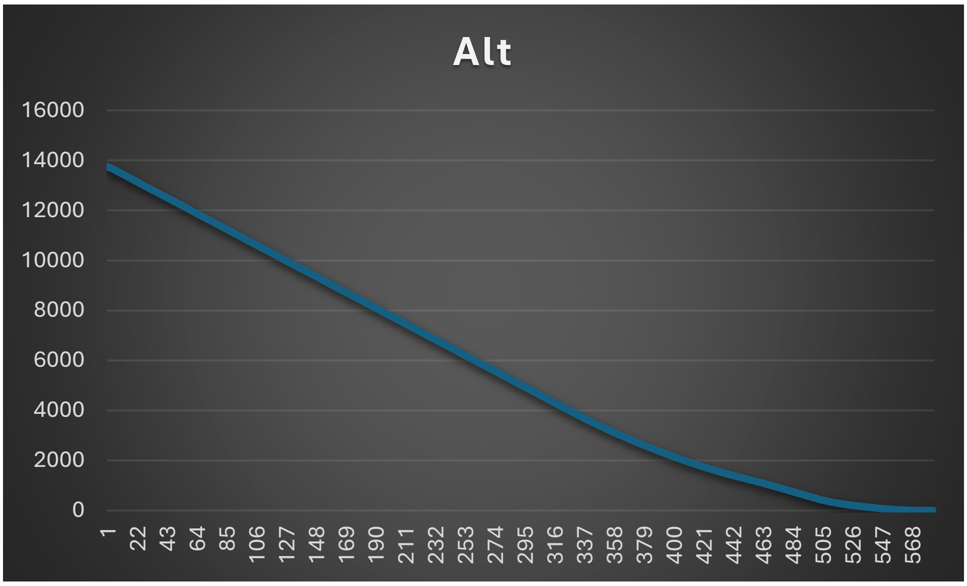
Added Graphical representation to main parameters:



The little hill at the vs chart Is caused by the breaking phase were we use more thrust to •slow down – this consumes more fuel indeed, but it also provides more robustness when descending with a much higher horizontal speed. Our assumption was that in that phase the most crucial thing is to break keep that as the focus.






Alt
16000
14000
12000
10000
8000
6000
4000
2000
0
1
22
43
64
85
106
127
148
169
190
211
232
253
274
295
316
337
358
379
400
421
442
463
484
505
526
547
568