**Part 1 – Technical Causes of the Bereshit Crash**

Based on the livestream footage (https://www.youtube.com/watch?v=HMdUcchBYRA&t=1612s)

The initial phase of the landing appeared to proceed smoothly. The spacecraft transitioned into the braking phase as expected, but from minute 25 onward, a series of technical failures began to unfold, ultimately leading to a crash.

**Minute 25:06 – Braking Phase Begins**

- The spacecraft entered the BRAKING sub-state.

- The system state was LANDING.

- Altitude: 22,629 meters

- Horizontal Speed (HS): 1564.9 m/s

- Vertical Speed (VS): 22.2 m/s

**Minute 25:16 – Point of No Return**

- The spacecraft began its final descent with no abort capability

- Main engine activated successfully

- All systems appeared stable at this point

A screenshot of a computer

AI-generated content may be incorrect.

**Minute 33:02 – First Critical Failure**

- A fault occurred in the IMU2 (Inertial Measurement Unit)

- Reset was issued to IMU2, which temporarily interrupted data from the functioning IMU

- This led to a critical gap in navigation data

- Communication with the NASA ground station was lost

- System data appeared frozen at:

* Altitude: 13,673 meters
* HS: 928.8 m/s
* VS: 24.8 m/s

A computer screen shot of a satellite

AI-generated content may be incorrect.

**Minute 34:28 – Communication Restored**

- Data transmission resumed but telemetry indicated dangerous conditions

- Vertical and horizontal speeds had increased significantly

- Main engine was no longer firing

- Horizontal speed continued to increase, suggesting unintended rotation

**Minute 35:11 – Engine Uncertainty**

- Unclear whether main engine had reignited

- Speeds continued to rise instead of decrease

- Control of the lander was compromised

**Minute 36:34 – System Reset**

- Final recovery attempt via full system reset

**Minute 37:19 – No Signal, Crash Confirmed**

- Communication was permanently lost

- Estimated crash speed: over 3,000 km/h

**Root Cause Analysis**

The crash was caused by a chain reaction initiated by the IMU2 failure. IMUs are critical components that measure a spacecraft's orientation, rotation, and acceleration. The failure sequence can be broken down as follows:

1. **Primary Hardware Failure**: IMU2 experienced a malfunction, the exact nature of which is not publicly disclosed but likely involved sensor drift or complete failure.
2. **Error in Recovery Procedure**: The attempt to reset IMU2 had the unintended consequence of disrupting data flow from the still-functioning IMU1. This created a critical navigation data gap during a phase where precise orientation control was essential.
3. **Control System Response**: Without reliable orientation data, the spacecraft's control system was unable to maintain the correct descent attitude. This likely caused the spacecraft to rotate uncontrollably.
4. **Engine Shutdown**: The flight computer, detecting unreliable orientation data or possibly as a safety measure, may have shut down the main engine. Alternatively, the uncontrolled rotation could have moved the engine's thrust vector in a direction that accelerated rather than decelerated the craft.
5. **Communication Issues**: The orientation problems likely affected the antenna pointing, causing communication disruptions that further complicated recovery efforts.

.

**Part 2 – Simulation and Autonomous Landing Control**

In this part, I developed a Java-based simulation to model the descent and landing of the Bereshit spacecraft on the Moon. The simulation incorporates lunar gravity, fuel consumption, thrust control, and spacecraft dynamics. The objective is to land safely with minimal vertical and horizontal velocity while conserving fuel.

**Physical Modeling and Assumptions**

The simulation starts from an altitude of 13,748 meters, a horizontal speed of 932 m/s, and a vertical speed of 24.8 m/s. The Moon’s gravity is set to 1.622 m/s², and atmospheric effects are neglected. The spacecraft has a main engine that produces 430 N of thrust and eight side thrusters used for angular adjustments, each producing 25 N. The spacecraft’s total mass decreases as fuel burns. Acceleration and velocity updates are based on Newton’s second law.

**Interpolation**

Three separate interpolators are used to define desired profiles for vertical speed, horizontal speed, and angle based on altitude. These interpolators are implemented using piecewise linear interpolation (TrajectoryInterpolator.java). This allows the simulation to adjust target behavior dynamically as the spacecraft descends, creating a smooth and gradual transition from high-speed entry to soft landing.

**PID Control System**

The simulation uses Proportional-Integral-Derivative (PID) controllers to adjust engine thrust and orientation based on the difference between current values and target values (error).

* The vertical speed PID (vsPID) calculates thrust power needed to reduce descent speed.
* The horizontal speed PID (hsPID) corrects horizontal velocity by adjusting the spacecraft’s tilt.
* The angle PID (anglePID) controls the spacecraft orientation to align with the target descent angle.

Each controller is updated at every simulation step, and the outputs are used to compute the desired thrust level (NN) and angular change. The output from these controllers is bounded to ensure realistic limits on acceleration and orientation.

**Simulation Flow**

Every second, the following steps are performed:

1. Compute desired vertical speed, horizontal speed, and angle using interpolators
2. Calculate control signals using PID controllers
3. Adjust NN and angle to reflect control outputs
4. Calculate thrust-based accelerations in vertical and horizontal directions
5. Apply lunar gravity
6. Update velocity, position, fuel, and mass
7. Log results to a CSV file (bereshit\_log.csv)

**Graph Analysis and Output**

Simulation results are exported and analyzed using a Python script (plot\_bereshit\_graphs.py) that generates graphs for:

* Actual vs. desired vertical speed
* Thrust power (NN) over time
* Altitude vs. time
* Horizontal speed vs. time

**Landing Results**

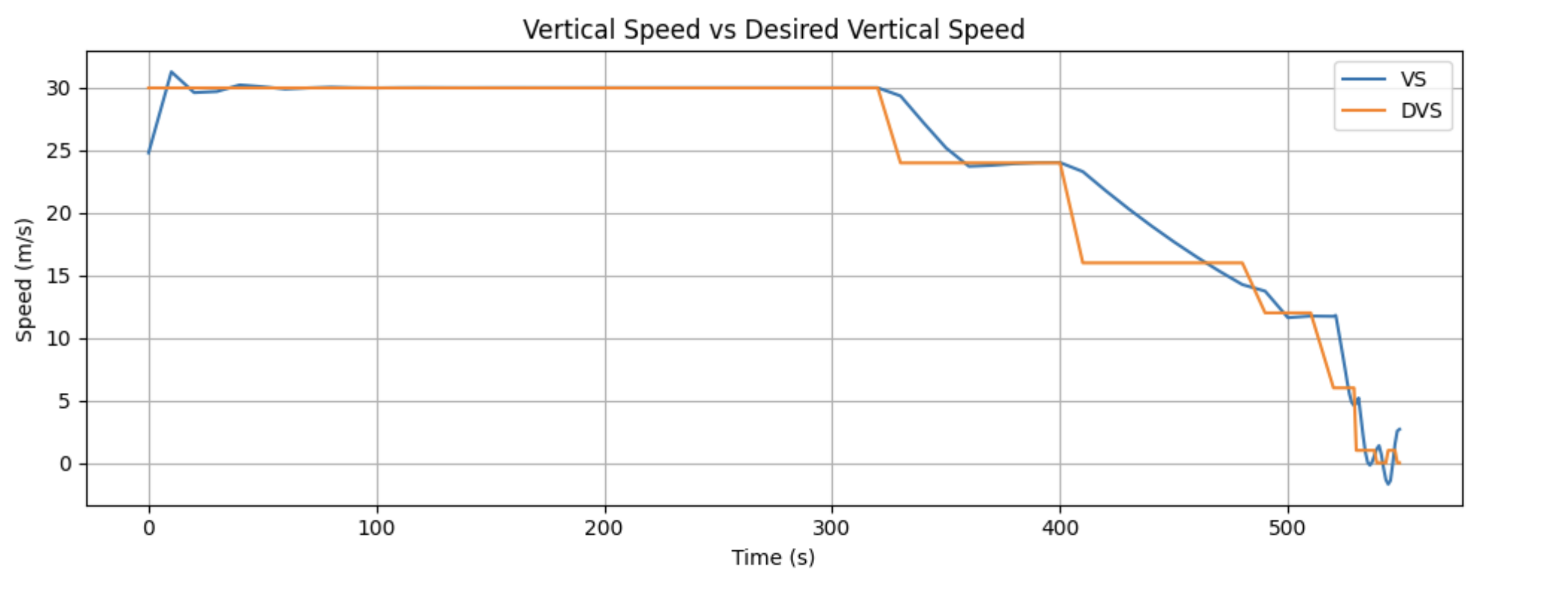
* Final vertical and horizontal speeds were reduced below 2.5 m/s
* Angle reached approximately 0°, indicating vertical orientation
* Fuel usage was minimized, with 50+ liters remaining in successful runs

These results demonstrate a successful autonomous landing under controlled conditions.

**Graphical Results**

The following graphs illustrate the simulation performance:

1. Vertical Speed vs Desired Vertical Speed



1. Normalized Thrust (NN) Over Time

A graph with orange line

AI-generated content may be incorrect.

1. Altitude Over Time

A graph with a green line

AI-generated content may be incorrect.

1. Horizontal Speed

A graph with a line

AI-generated content may be incorrect.