**Landing Simulation Report**

In part A of the assignment we explained the cause of the crash of the "Beresheet" spacecraft on the moon. Briefly, communication with the IMU controller was cut off, The spacecraft reversed, The spacecraft failed to fix its angle, The horizontal and vertical speeds increased until it was too late for a soft landing.

We came to this claim by looking at the video about the simulation data and fiddling with our simulation. From the video: After being told that the IMU controller was back to work, the horizontal and vertical speeds increased at a rapid rate while the engines were working normally.

When the spacecraft is at the right position, the goal of the engines functionality is supposed to reduce the horizontal and vertical velocities (last 30 kilometer) and not increase it as happened in reality.

In our exercise we try and catch the controller data until we reached an optimal landing position. We have found that PID controller are one of the most important availability of any autonomous robot. Incorrect PID controller calculation lead to very quickly crashing, on the one hand , and on the other getting to the ground without fuel (sometimes with a negative amount of fuel)

SpaceIL's original design (by the video <https://www.youtube.com/watch?v=VYd5vRjsfQE>): The spacecraft rotates around the moon about 30 km from the ground (rotation speed depends on the gravity of the moon and the spacecraft engines). The spacecraft is at an angle of 58.3 (with the engines facing in the direction of its motion) and begins to slow its speed by operating the engines. As the speed drops, the spacecraft approaches the ground, to the landing area. About 2 km away, the spacecraft changes its angle so that its engines are downward (angle 0). The last km uses all engines, including the main one, to slow down the speed to a soft landing.

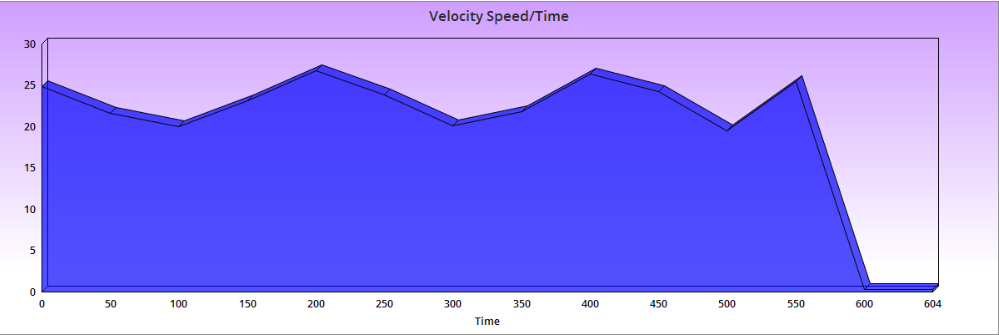
In the project, we tried to stick to this design. Spacecraft modeling was done by creating the “Beresheet\_Spacecraft” class, which has the velocity and horizontal speed data, distance from the landing point, angle, ground height, time, acceleration, fuel, weight, power (for engines), An collection of 8 navigation engines under different names (locations) and a PID controller. The controller main goal is to activate the secondly engine’s power when the spacecraft needs to change its angle from 58.3 to 0

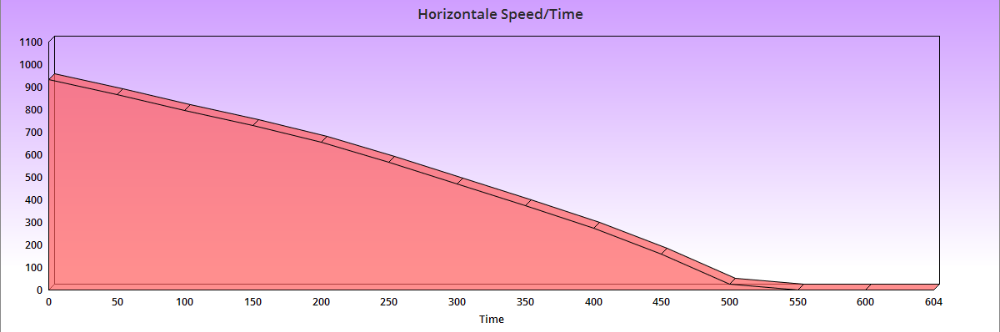
How does the simulation work? When the project is run, a window opens with a picture of space, stars, moon's bottom below, a spaceship at the far end of the upper left and a small green circle on the moon from the bottom right. Behind the scenes, a loop is activated that updates the spacecraft's data, including its location on the window. First, we gave all engines the same power, so the spacecraft lowers its speed (horizontal and vertical) until it reaches 2 km from the ground. Vertical velocity is updated by a simple calculation of the angular sine of double acceleration radians Horizontal velocity is updated by the same calculation only with the angle cosine in radians. The distance is calculated by formulating a distance between the spacecraft point and the predicted landing point. It is important to note that we normalized the data from KM to pixels. The fuel and the weight of the spacecraft are updated by the power given to the engines, the amount of engines and the time.

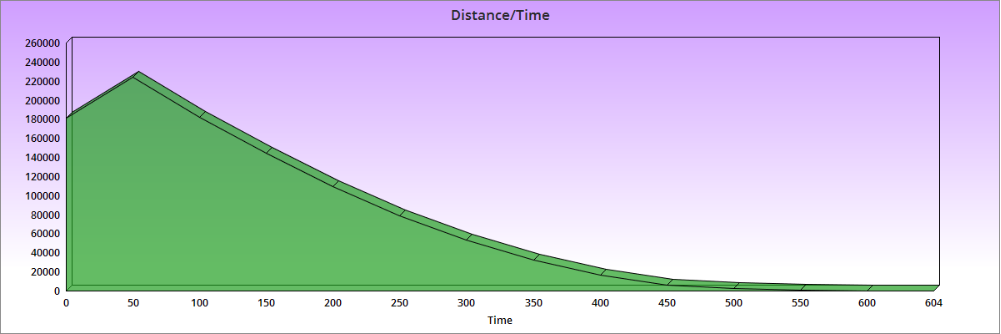
After reaching about 2 km from the ground, the left engines gave more reliable power to the spacecraft to change its angle until it reached 0 (because it is a two-dimensional simulation and we only have left, right, up, down). Here we send the power of the motors to the PID controller, so that it maintains around half the power of each engine. 125 meters away from the target and with an angle of 0, we gave maximum power to all engines so that the spacecraft would slow down to a soft landing. 5 meters from the ground, speeds were reduced enough so we reduced the power of the engines. We also saved some fuel.

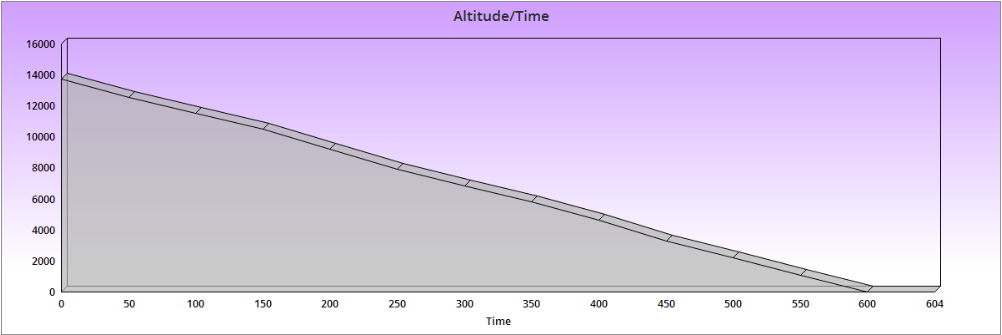
At the end of the simulation, the spacecraft reaches the landing area (green circle) with speeds, distance and altitude close to 0 and a quantity of fuel greater than 0.

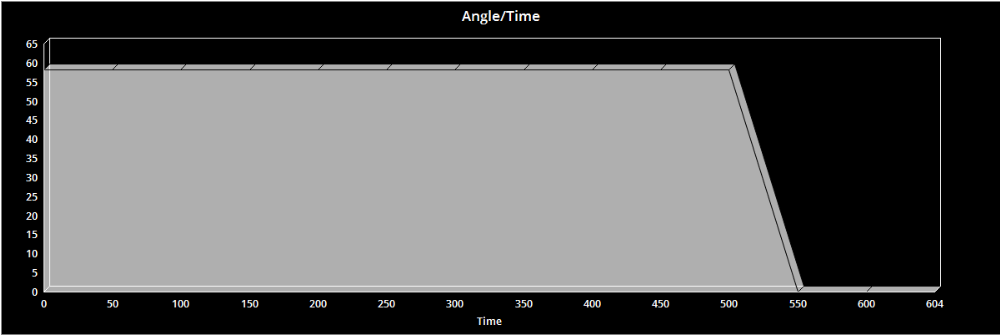
Our simulation data:

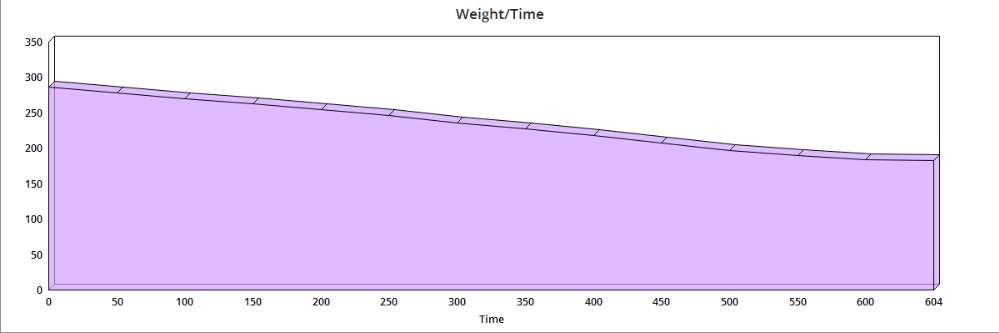


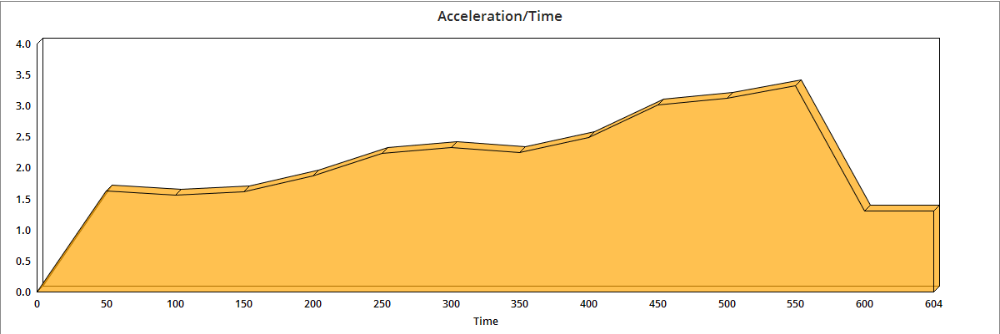


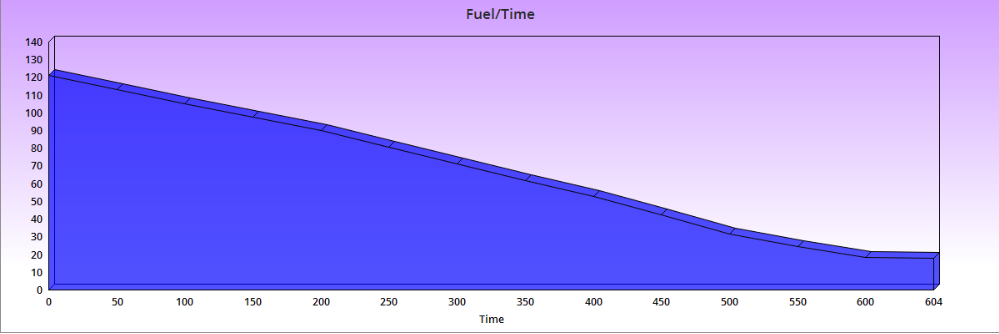












**Authors**: Snir Shaharabani & Lioz Elmalem