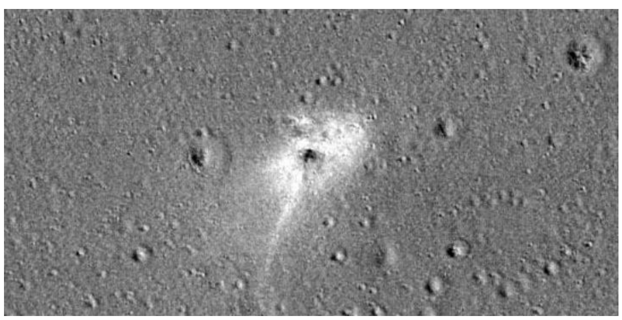
21.05.2020

Autonomous Robots- Task 2:

Beresheet: Soft Lunar Landing Simulation

Yinon Sinay
Yarden Ben- Amitai

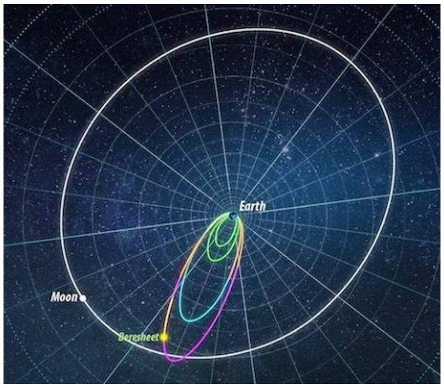
The moon-lander Beresheet is a small, partly crowdsourced, private-sector owned spacecraft, operated by Israel Aerospace Industries and equipped with a magnetometer, digital time capsule, and laser retroreflector, all meant to land on the moon. Due to various reasons (listed in the additional text "Report: Beresheet Crash-Landing"), Beresheet could not complete its mission and instead, crashed into the surface of the moon. In this task, we attempt to remodel and simulate the lunar-lander Beresheet and its course of travel, using the Eclipse platform, various sources of information regarding physics in changing gravity, reviews of the mission's malfunctions etc.



Beresheet crash site

The following paragraphs contain a layout of our process of work and its products, beginning with basic information and data of the relevant astronomical bodies involved, followed by a detailed description of the simulated spacecraft, with all its relevant components and mechanism, then moving to a dissected view of our designed route, the process of landing, possible future improvements that can be applied to our model and lastly an overall summary and conclusions.

The moon is a natural body orbiting the earth with an average orbital distance of 384,402 km. Its angular speed of its orbit about the earth is $2.7 * 10^{-6}\ (r/s)$, and its gravity is $1.62\ m/s^2$.
When simulating the landing of our craft we took notice of the moon's gravity and used it to influence the data. The equations we used can be found at "Class Physics", which contains the essential calculations of mass, acceleration, force applied by gravity and Newton's laws.

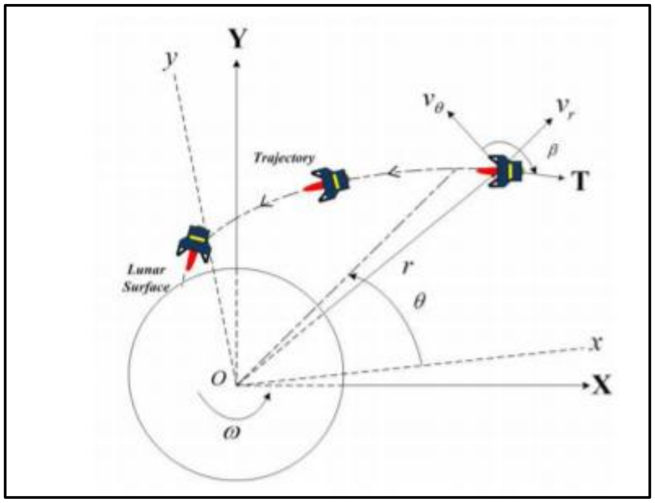


The modeling of the spacecraft is based on the information provided by Aerospace with some simplifications. The weight of fuel, weight of spacecraft, engine force for main and secondary engines are as published [1], however, Beresheet carried 8 secondary engines used for directing its course sideways, each one has the force of 25N, but our model uses all 8 of them as thrusters. Fuel consumption is roughly 0.15 kg per second for the main engine and 0.009 kg per second for each secondary engine. Since our simulation is starting from an altitude of 30 km above the moon's surface, the craft's fuel tank contains 216 kg of fuel [2], whereas a full tank is 420 kg.

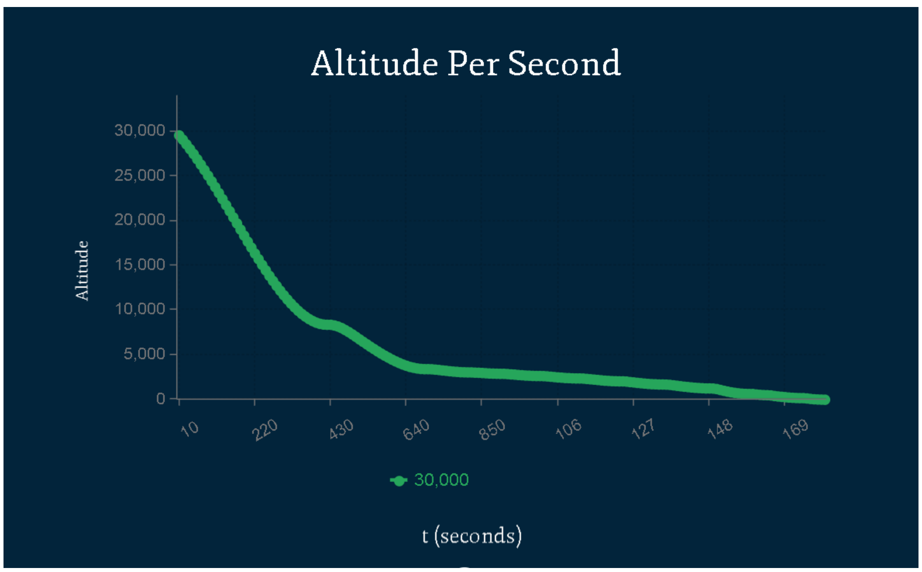
As mentioned above, our simulation's starting point is from an altitude of 30,000 m above the moon's surface, where the craft is at an angle of 35∘, and traveling at approximately 1,700 m/s. The known variables at the beginning of the simulation, are:

$mass_0^{spacecraft}$	$mass_0^{fuel}$	g_{moon}	alt_0	$Vertical\ Velocity$	$Horizontal\ Velocity$
381 kg	216 kg	$1.622\ m/s^2$	30,000 m	1701.6 m/s	43 m/s

We separated the landing simulation into 3 phases, each one is manipulating its engine power differently while responding to the changes in altitude, angle, vertical velocity and horizontal velocity, with the goal of reducing both velocities to 0 before reaching 0 altitude, while maintaining vertical landing.



T(thrust), v_r(vertical velocity), v_θ(horizontal velocity)



Even though our simulation is capable of soft lunar landing, it does not optimise it's variables to achieve minimal descension time with minimal fuel consumption. A possible improvement to our program would be to implement Optimal Descent Trajectory. Optimisation is the act of achieving the best possible result under given circumstances. In trajectory optimisation problems we are usually trying to minimise the fuel consumption. This effort can be expressed as a function of certain parameter variables. Hence, optimisation is the process of finding the conditions that give the maximum or the minimum value of a function[3].

$$J(u) := \int_0^{t_f} c\,(t, x(t), u(t))\,dt$$

By choosing the value of $u(t)$, the state trajectory $x(t)$ can be controlled. In case of optimal descent trajectory, $u(t)$ is a control function of a lunar module engine thrust. We are trying to find a thrust to minimise the "cost" function J. Function c is the cost function and t_f is the final time. Unfortunately, java is not the best language to compute optimisation functions, whereas MatLab would have been the logical choice.

In conclusion, even though our model was capable of soft lunar landing, it still took approximately 30 minutes to touchdown, whereas the original calculations for Beresheet are 17 minutes. We suggested an applicable way to improve our results with regard to modern solutions.

[1]https://davidson.weizmann.ac.il/online/sciencepanorama/%D7%92%D7%99%D7%9C%D7%95%D7%99%D7%99%D7%9D-%D7%97%D7%93%D7%A9%D7%99%D7%9D-%D7%A2%D7%9C-%D7%91%D7%A8%D7%90%D7%A9%D7%99%D7%AA

[2] https://www.youtube.com/watch?v=ll4nm8EkvjM&t=1500s

[3]https://rc.library.uta.edu/uta-ir/bitstream/handle/10106/23955/Ocampo_uta_2502M_12322.pdf;sequence=1

Link to Github Project:

https://github.com/Yinonss/Autonomous-robots-Ex2.git