

BRNO UNIVERSITY OF TECHNOLOGY VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

VYSORE OCENI TECHNICKE V BRNE

FACULTY OF INFORMATION TECHNOLOGY FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

LUNAR LANDING SIMULATION

SIMULACE PŘISTÁNÍ NA MĚSÍCI

BACHELOR'S THESIS BAKALÁŘSKÁ PRÁCE

AUTHOR AUTOR PRÁCE **JAKUB FILO**

SUPERVISOR VEDOUCÍ PRÁCE doc. Ing. PETER CHUDÝ, Ph.D., MBA

BRNO 2019

Department of Computer Graphics and Multimedia (DCGM)

Bachelor's Thesis Specification



Student: Filo Jakub

Programme: Information Technology

Title: Lunar Landing Simulation

Category: Modelling and Simulation

Assignment:

- 1. Investigate the history of Lunar Landing in the Apollo project.
- 2. Research the descent trajectory physics.
- 3. Perform a computation of optimal descent trajectory.
- 4. Design and implement visualization environment, for which you create or download from relevant sources basic 3D models and 3D engine for descent maneuver interpretation.
- 5. Evaluate achieved results and discuss potential further improvements.

Recommended literature:

• According to supervisor's recommendations.

Requirements for the first semester:

• Items 1, 2, 3 and partially item 4.

Detailed formal requirements can be found at http://www.fit.vutbr.cz/info/szz/Supervisor:Chudý Peter, doc. Ing., Ph.D. MBAHead of Department:Černocký Jan, doc. Dr. Ing.Beginning of work:November 1, 2018Submission deadline:May 15, 2019Approval date:May 10, 2019

Abstract

The goal of this bachelor thesis is to compute and visualize lunar module's optimal descent trajectory. An application for the visualization of the descent trajectory and lunar module's position from dataset was developed with the use of a 3D engine Godot. The measured coordinates were acquired through digitization of graphs from the NASA published Apollo 11 flight plan. Subsequently the optimal trajectory was computed and compared with the measured historical data from Apollo 11 Lunar landing. Using the designed application it is possible to interactively visualize the differences between optimal descent trajectories.

Abstrakt

Cílem této bakalářské práce je vypočítat a vizualizovat optimální sestupovou trajektorii lunárního modulu. Pro tento účel byla s pomocí 3D enginu Godot vytvořena aplikace, která vizualizuje sestupovou trajektorii a polohu lunárního modulu z datové sady. Naměřené souřadnice byly získány digitalizací grafů z letového plánu mise Apollo 11, který publikovala NASA. Následně byla vypočtena optimální trajektorie a tato porovnána s naměřenými historickými daty z přistání na Měsíci mise Apollo 11. S vytvořenou aplikací je možné interaktivně vizualizovat rozdíly mezi optimálními sestupovými trajektoriemi.

Keywords

optimal trajectory, optimal control, flight trajectory visualization, Lunar landing, Apollo 11, Godot Engine, Bocop

Klíčová slova

optimální trajektórie, optimální řízení, vizualizace trajektorie letu, přistání na Měsíci, Apollo 11, Godot Engine, Bocop

Reference

FILO, Jakub. *Lunar Landing Simulation*. Brno, 2019. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor doc. Ing. Peter Chudý, Ph.D., MBA

Rozšířený abstrakt

Cílem této bakalářské práce je provést výpočet optimální trajektorie přistání lunárního modulu na Měsíci a vizualizovat sestupový manévr v 3D prostředí.

Téma simulace a optimalizace sestupových trajektorií je v současné době aktuální -NASA plánuje návrat lidské posádky na povrch Měsíce před rokem 2020 [26]. V praxi se optimální trajektorie simuluje především z důvodu minimalizace spotřeby paliva vesmírné lodě, což je důležitý faktor ovlivňující cenu vesmírní mise.

V první části práce byla nastudována historie mise Apollo 11, z níž se v práci vychází. Z dostupných zdrojů byly získány technické parametry jednotlivých částí lunárního modulu a fyzikální parametry astronomických objektů.

Pro výpočet optimální trajektorie bylo nutné definovat fyzikální model sestupu lunárního modulu. Pohyb modulu byl zúžen z prostorového pohybu do abstrakce se třemi stupněmi volnosti, protože se v něm odehrává podstatná část přistávacího manévru. Taktéž to umožňuje zjednodušení výpočtů. Dále byly na základě Newtonovy klasické mechaniky odvozené potřebné pohybové rovnice.

Optimální trajektorie byla modelována jako problém optimálního řízení. To znamená, že trajektorie může být měněna v čase pomocí řídícího parametru. V našem připadě je tento parametr tah motoru lunárního modulu. Jako užitková funkce byla vybrána finální hmotnost lunárního modulu, kterou se snažíme maximalizovat. Problém byl implementován v softwarovém nástroji Bocop [34]. Výsledky optimalizace splnily omezující požadavky a zbývajíci hmotnost paliva při hladkém přistání zůstala výrazně nad limitem.

Druhá část práce spočívala v implementaci aplikace pro vizualizaci přistávacího manévru. Pro tento účel byl použit 3D engine Godot [13]. Ve výsledné aplikaci si uživatel po zvolení vstupního souboru může přehrát animaci sestupové trajektorie lunárního modulu, pozastavit ji, nebo přetočit dozadu. Zobrazují se v reálnem čase parametry letu jako je výška, vzdálenost od místa přistání a úhly sklonu lunárního modulu. Je též možné měnit úhel kamery, případně se podívat na přistání z pohledu kokpitu.

Vstupem aplikace je datová sada, která obsahuje souřadnice polohy a úhel sklonu lunárního modulu. Datová sada byla vytvořena digitalizací grafů z plánů mise Apollo 11. Souřadnice jsou v aplikaci modifikovány tak, aby zohlědňovaly zakřivení povrchu Měsíce. Kromě datové sady z reálné mise byl vytvořen dataset z výstupních dat optimalizačního problému. Obě trajektorie byly na závěr porovnány.

Lunar Landing Simulation

Declaration

Hereby I declare that this bachelor's thesis was prepared as an original author's work under the supervision of doc. Ing. Peter Chudý, Ph.D., MBA. All the relevant information sources, which were used during preparation of this thesis, are properly cited and included in the list of references.

> Jakub Filo May 27, 2019

Acknowledgements

I would like to thank my supervisor doc. Ing. Peter Chudý, Ph.D., MBA for all the help, guidance and an enormous patience during my work on this bachelor's thesis.

Contents

1	Introduction	7
2	History of Lunar Landing in project Apollo2.1Project Apollo2.2Apollo 11 spacecraft2.3Apollo 11 mission profile	8 8 9 13
3	Descent trajectory physics3.1 Position of Lunar Module in coordinate system3.2 Physics of Lunar landing problem	16 16 19
4	Optimal descent trajectory computation4.1Optimization problem theory4.2Implementation of optimal descent trajectory	23 23 26
5	Design and implementation of visualization environment for descent maneuver interpretation5.1Application implementation5.2Creation of the input dataset	33 33 37
6	Evaluation of achieved results	40
7	Conclusion	42
Bi	bliography	43
Α	A Astronomical objects and spacecrafts in Apollo 11 mission	
в	3 Content of the included CD	

List of Figures

2.1	Apollo spacecraft launch configuration [33]
2.2	Command Module technical drawing [20] 10
2.3	Service Module technical drawing [23] 10
2.4	LM's yaw, pitch, roll axes 12
2.5	Render of Lunar Module 3D model 12
2.6	Staging of Saturn V [24] 13
2.7	$Translunar injection [24] \dots \dots$
2.8	Transposition maneuver $[24]$
2.9	Lunar orbit insertion $[24]$
2.10	Separation of the LM from the CSM [24]
2.11	Landing on the Moon [24]
2.12	Rendezvous and docking [24] 15
2.13	Transearth insertion $[24]$
2.14	Recovery of the Command Module [24]
3.1	Polar coordinate system
3.2	Cartesian coordinate system
3.3	Lunar Module position in polar coordinate system
3.4	Inertial and rotating frames
4.1	Global and local minimum
4.2	One step of midpoint method [8]
4.3	Bocop pipeline [5]
4.4	Main window of Bocop
4.5	Optimal descent trajectory altitude in time
4.6	Optimal descent trajectory central angle in time
4.7	Optimal descent trajectory radial velocity in time
4.8	Optimal descent trajectory tangential velocity in time
4.9	Optimal descent trajectory, lunar module pitch angle in time
4.10	Optimal descent trajectory, lunar module mass in time
4.11	Optimal descent trajectory, lunar module throttle command in time 32
5.1	Screenshot of the application menu
5.2	Screenshot of application main view
5.3	Bézier curve representation
5.4	Visual output from WebPlotDigitizer application containing one part of the
	Apollo 11 descent trajectory graph
5.5	Digitized of Apollo 11 descent trajectory, altitude vs. distance from the
	landing site

5.6 5.7	Optimal descent trajectory altitude vs. distance from the landing site Optimal trajectory pitch vs. distance from the landing site	39 30
0.1	Optimial trajectory pitch vs. distance from the failung site	00
6.1	Comparison of Apollo 11 and optimal descent trajectories	41
A.1	The Earth [21] The Manual [27]	47
A.2 A.3	The model of the Saturn V rocket[15]	47 48

List of Tables

2.1	Lunar Module's technical specifications	11
2.2	Lunar Module's descent propulsion system specifications	11
2.3	Lunar Module's ascent propulsion system specifications	11
4.1	Constants used in the optimal descent trajectory problem	29
4.2	Boundaries for state and control variables	29
A.1	The Earth's physical characteristics [38]	47
A.2	The Moon's physical characteristics [39]	47
A.3	Saturn V technical specifications [22]	48

Nomenclature

- α Inertial angular acceleration
- \ddot{r} Object acceleration in two-body problem
- $\Delta \theta \,$ Angular increment
- $\Delta d\,$ Travelled distance increment
- \dot{r} Object velocity in two-body problem
- \hat{i} Object velocity in rotating frame
- \hat{j} Object acceleration in rotating frame
- μ Gravitational parameter
- ω Inertial angular velocity
- ψ^r Lunar Module engine thrust direction angle
- θ Lunar Module central angle
- \vec{a} Object acceleration vector
- \vec{F} Resultant of the forces acting on the object
- \vec{p} Object momentum
- \vec{r} Relative position vector in two-body problem
- \vec{v} Object velocity vector
- B(t) Bézier curve
- f(x) Objective function
- G Universal Gravitational Constant
- $g_i(x)$ Inequality constraints
- H Hessian
- i Rotating frame i axis
- I_x Inertial frame x axis

- I_y Inertial frame y axis
- j Rotating frame j axis

J(u) Cost function

- K Bézier curve knot point
- k Lunar Module engine throttle command
- $l_i(x)$ Equality constraints
- m Lunar Module mass
- P Bézier curve control point
- r Lunar Module radial coordinate
- R_{eq} Moon equatorial radius
- $T_{max}\,$ Lunar Module engine maximal thrust
- u(t) Control parameter
- u_T Lunar Module engine thrust vector
- v_r Radial velocity
- v_{θ} Tangential velocity
- $V_{ex}\,$ Lunar Module engine specific impulse
- x(t) State trajectory

Chapter 1

Introduction

Landing a man on the surface of the Moon is without a doubt one of the greatest achievements of human civilization. In July we will be celebrating 50th anniversary of the first Moon Landing. Nowadays Moon is also an important target for the future space missions. NASA would like to land astronauts on its surface in the late 2020s [26].

A common part of planning a space mission is to create a simulator where whole range of boundary conditions can be tested. The mission is less error-prone with analyzing simulator data where various parameters of flight can be optimized. One of the most important parameters is the trajectory design. The trajectory must meet selected mission objectives while considering spacecraft limitations. The guidance algorithms are implemented based on the designed trajectory. They are created with the objective of minimizing the fuel consumption, because the fuel weight is a major part of spacecraft mass and thus mission cost.

The first part of this document is primarily focused on establishing theoretical basis needed for the computation of an optimal descent trajectory, while second part discuses an implementation details of visualization application. The last part focuses on the evaluation of achieved results. In the second chapter reader can learn about the history of Lunar landing in Apollo 11 mission. The technical parameters of the lunar module and other spacecrafts are provided here. The third chapter is dedicated to the explanation of the descent trajectory physics along with the equations of motion necessary for its computation. Moving to the fourth chapter all the previous knowledge is used for the computation of optimal descent trajectory for Lunar Module. The trajectory has been computed using software for solving optimal control problems Bocop. The math needed for solving optimization problems is introduced in this chapter. Next, the application for visualization of descent trajectories was developed with use of 3D engine Godot. The implementation details are discussed in Chapter 5. The evaluation of created trajectories and discussion of possible improvements form the content of Chapter 6.

Chapter 2

History of Lunar Landing in project Apollo

The aim of this chapter is to get the reader acquainted with the history of project Apollo and to provide a knowledge base about the vehicles and spacecraft used in the Apollo 11 mission. The technical specifications of the Lunar Module, which are then used in the later chapters are provided, along with description of actual flight.

2.1 Project Apollo

The Apollo space program originated during the Eisenhower administration in 1960, as a successor to project Mercury, which put the first Americans in space. It was carried out by the National Aeronautics and Space Administration (NASA). The Apollo's initial plan was to build a spacecraft which would carry three astronauts on Earth orbit. Previous spacecraft could carry only one. In the space race enthusiasm this goal was quickly expanded and possible missions of Apollo included carrying crews to a space station, flights around the Moon and even manned lunar landings. The NASA manager Abe Silverstein named the Apollo program after the Greek god of light, music and the sun, because he felt "Apollo riding his chariot across the Sun was appropriate to the grand scale of the proposed program" [17].

On May 25, 1961, president Kennedy proposed a vision of landing men on the Moon before end of the decade. Thus the race for the Moon began. This courageous goal required the largest commitment of resources ever made by any nation in peacetime. At its peak, the Apollo program employed 400 000 people and over 20 000 industrial companies and universities participated [1]. The cost of the program was enormous - \$25 billion, which translates to today's \$112 billion when counting with inflation [11].

The prime mission objective of Apollo 11 was to: "Perform a manned lunar landing and return". The famous crew of three - *Neil A. Armstrong (Commander), Michael Collins (Command Module Pilot)* and *Edwin "Buzz" E. Aldrin Jr. (Lunar Module Pilot)* begun their mission on July 16, 1969 from NASA's Kennedy Space Center Launch Complex 39-A. A description of this outstanding journey follows. Apollo 11 Press Kit [24] and Apollo 11 Mission Report [25] were used as a frame of reference.

2.2 Apollo 11 spacecraft

The Apollo 11 had a three part spacecraft. It consisted of the **Command Module** *Columbia*, **Service Module** and **Lunar Module** *Eagle*. Figure 2.1 shows the launch configuration of the spacecraft along with the carrier rocket Saturn V (technical specifications are provided in Table A.3). The technical drawing was retrieved from the NASA archive [33].



Figure 2.1: Apollo spacecraft launch configuration [33]

Command Module

The Command Module (CM) provided the living space for the astronauts. It was the only part to return to Earth. The conical design was chosen to protect the astronauts from the heat during reentering the Earth atmosphere. A special material was developed to burn away during reentry and dissipate the extremely high temperatures caused by friction. The CM was divided into 3 compartments. The part in the nose of the cone held parachutes and docking mechanism. Second part was situated around the base and contained propellant tanks, reaction control engines and electrical wiring/plumbing. There were 12 reaction control thrusters, with thrust of 420 N each. The crew compartment provided three couches for astronauts, controls, displays, navigational and other systems. In the center above the

couches there was a large access hatch. It provided a re-boarding capability at the end of the mission. Figure 2.2 shows detailed description of individual systems of the Command Module. Figure 2.2 was retrieved from [20].



Figure 2.2: Command Module technical drawing [20]

Service Module

The aluminum panels composed the exterior of a cylindrical Service Module (SM). At the back of the SM was mounted a re-startable 9.1 kN engine and a cone shaped engine nozzle. The attitude control was provided by four 450 N reaction control thrusters spaced 90 degrees around the SM. Propellant tanks and electrical power system were situated in six radially divided sections around the central cylinder. Figure 2.3 shows technical drawing of the Service Module [23].



Figure 2.3: Service Module technical drawing [23]

Lunar Module

The Lunar Module (LM) was a two-stage vehicle composed of an descent and ascent stage. At descent to the lunar surface both stages operated as one compact unit. On the return trip back to the Lunar orbit, the ascent stage acted as a single spacecraft for rendezvous and docking with the Command and Service Module (CSM).

The descent stage was an octagonal prism with four landing legs fitted with round footpads which held the vehicle 1.5 m above surface. One of the legs had a small ladder used by the astronauts for descending down to the Lunar surface. The descent stage contained a landing rocket, propellant tanks and storage space for the lunar experiments equipment. It also served as launching platform for the ascent stage and was left behind on the Moon.

The ascent stage of an irregular shape was mounted on top of the LM's descent stage. It housed the astronauts in a small 6.65 m^3 compartment, which operated as a base for the lunar operations. On top, there was a docking hatch for connecting to the CSM. The LM's ascent stage also contained all necessary telecommunication equipment. On the bottom, there was a fixed, constant thrust engine. There were no seats for the astronauts - they were standing. Another important component was the Apollo's guidance computer which operated on a frequency of 2.048 MHz and provided guidance, navigation and control of the spacecraft [37].

Table 2.1 shows the Lunar Module's technical specifications retrieved from [18]. Tables 2.2 and 2.3 describe the descent and ascent stage propulsion system. Technical specifications of the propulsion systems were taken from [19].

Height (legs extended)	$6.985 \ m$
Width (diagonally across extended landing gear)	$9.449 \ m$
Weight (with crew and propellant)	$15 \ 103 \ kg$
Weight (dry)	$4479 \ kg$
Propellant weight (ascent stage)	$2376 \ kg$
Propellant weight (descent stage)	$8248 \ kg$

Table 2.1: Lunar Module's technical specifications

Table 2.2: Lunar Module's descent propulsion system specifications

Manufacturer	TRW
Propellant	$N_2O_4/\text{Aerozine }50$
Maximum thrust (vacuum)	$45.04 \ kN$
Specific impulse	$3050 \ ms^{-1}$
Throttle	between 10% and 60% of full thrust

Table 2.3: Lunar Module's ascent propulsion system specifications

Manufacturer	Bell Aircraft / Rocketdyne
Propellant	N_2O_4 /Aerozine 50
Constant thrust (vacuum)	$16 \ kN$

LM attitude description

The Lunar Module is free to rotate in three dimensions. As displayed in Figure 2.4, yaw is rotation angle about Z axis, pitch angle is rotation about Y axis and roll angle is rotation about X axis, all according to right hand rule. Axes have their origin at the center of gravity of the lunar module [41].



Figure 2.4: LM's yaw, pitch, roll axes

Figure 2.5 shows a 3D model of the Lunar Module, which was used in this work. It was retrieved from [32].



Figure 2.5: Render of Lunar Module 3D model

2.3 Apollo 11 mission profile

Launch and translunar injection

The Saturn V launch vehicle set the astronauts to the three day voyage to the Moon. First, it entered the Earth's orbit at an altitude of 183.2 km. One and half orbits later the third stage of the rocket restarted and autopilot pushed Apollo 11 into the translunar trajectory (translunar injection). Figure 2.6 shows the separation of the first Saturn V stage. The translunar injection is displayed on Figure 2.7 retrieved from [24].



Figure 2.6: Staging of Saturn V [24]



Figure 2.7: Translunar injection [24]

Translunar flight and lunar orbit insertion

After a complete checkout of vehicle readiness the command/service module separated from the Saturn V rocket, turned around and docked with lunar module. Docking maneuvers were performed in a speed of approximately $11.2 \times 10^3 m s^{-1}$. The separated stage of the Saturn V rocket was directed into a "slingshot" trajectory to miss the Moon and go into the solar orbit. Apollo spacecraft with lunar module docked on top of it and continued to its destination - Moon.

Four trajectory correction maneuvers were made during translunar flight. Primary reason for these maneuvers was to establish the horizon altitude for optical marks in the navigation computer. The digital autopilot was used to keep the spacecraft in positive roll rate of $0.3 \ deg.s^{-1}$ to stabilize its thermals due to the solar heating. It allowed the crew to sleep without fear of encountering unacceptable thermal conditions.

After the service module propulsion maneuver, the spacecraft was inserted into a 111.12 km by 314.84 km elliptical lunar orbit which was adjusted to 100 km by 122.23 km after two revolutions. The ellipticity of this orbit was supposed to disappear slowly, because of lunar gravitational field irregularities and put the spacecraft into circular orbit. However the ellipticity decay was less than excepted and lunar rendezvous maneuver solution differed from preflight estimates. Figure 2.8 shows the transposition maneuver and Figure 2.9 the Lunar orbit insertion taken from [24].



Figure 2.8: Transposition maneuver [24]



Figure 2.9: Lunar orbit insertion [24]

Descent preparations and undocking

Planned routines before powered descent went smoothly and were done approximately 30-40 minutes early. These included suiting, transfer of Commander (Armstrong) and LM Pilot (Aldrin) to the lunar module and power-up of computer to perform calculations for aligning the command and lunar modules before the signal lost on the lunar far side. After activation of all systems and pressurization of lunar module cabin, command and service modules maneuvered both spacecrafts to undocking attitude and final check before undocking was accomplished. Lunar module then undocked from the CSM and begun powered descent phase. Figure 2.10 displays separation of the Lunar Module from the Command and Service Module taken from [24].

Powered descent and landing

Powered descent started at altitude of $15.34 \ km$ about $481.52 \ km$ from landing site - Sea of tranquility. We can divide it into three major phases: breaking phase, approach or visibility phase and final landing phase. Three separated computer programs, one for each phase were used. They were designed to execute desired trajectory and satisfy various constraints such as position, velocity and acceleration. There was also the manual landing phase program - to enable pilot a manual control of lunar module.

The breaking phase was initiated in a facedown attitude as it enabled crew to make time marks on selected landmarks. At altitude of about 13.99 km the faceup maneuver was executed after passing Maskelyne W crater. Acquisition of radar followed. The computer alarm occurred due to large difference between computed and radar altitude, but differences converged within 30 seconds.

Lunar module descended to altitude of 2.17 km and switched to final approach phase program. At 1.52 km, the Commander switched to attitude-hold to check manual control in anticipation of the final descent. After pitch-over maneuver (pitch angle down to 0 degrees) systems indicated that the approach path was leading into large crater. To avoid it, Armstrong switched to manual controls and increased the pitch angle to extend range. This dangerous maneuver caused approximately 335.28 meters down range from the initial aim point, but the Eagle has landed. Figure 2.11 shows the Lunar Landing illustration taken from [24].



Figure 2.10: Separation of the LM from the CSM [24]



Figure 2.11: Landing on the Moon [24]

Lunar ascent and rendezvous

After touchdown the astronauts first prepared the lunar module for ascent. Armstrong was the first man to step onto the lunar surface, followed by Aldrin about 40 minutes later. During their almost 3 hours stay they conducted several experiments - measured meteoroid impacts and installed reflector which mirrored laser beams back on Earth.

The descent part of the Lunar Module served as launching pad for the ascent part. Rendezvous radar helped with tracking the command module. The ascent phase started with 10 second long vertical rise. Pitch-over maneuver provided the correct attitude to place the spacecraft in 18.52 km by 83.34 km orbit to establish initial conditions for rendezvous. Lunar module then performed four other maneuvers which brought it to command module for docking. Figure 2.12 shows rendezvous and docking maneuvers [24].

Transearth injection and recovery

After 12 hour rest, the transearth maneuver was initiated on time and spacecraft followed the same roll procedure as in translunar flight. Figure 2.13 shows the transearth insertion [24]. The course was adjusted several times, but the journey back home was not problematic. Apollo 11 entered the Earth's atmosphere at 195 hours five minutes after launch at speed of approximately $11 \times 10^3 m s^{-1}$. The touchdown point was about 1926 km from Honolulu, Hawaii. Recovery after the touchdown is displayed in Figure 2.14.





Figure 2.12: Rendezvous and docking [24]





Figure 2.14: Recovery of the Command Module [24]

Chapter 3

Descent trajectory physics

This chapter describes the physics necessary to compute an optimal descent trajectory of a Lunar Module. Coordinate systems are explained along with motion equations and rotational kinematics. Apollo 11 Lunar Module (LM) will be used as reference vehicle, because it allows us to compare computed and real descent trajectories.

The Moon has no atmosphere and is assumed to be spherical. As mentioned in MIT thesis [10], its rotation can be neglected, because the extra fuel expenditure would be at maximum 0.3% of the total fuel usage.

3.1 Position of Lunar Module in coordinate system

The first step to define a descent trajectory physics for a lunar lander is to place the vehicle into coordinate system. This allows us to uniquely determine its position. For the trajectory computation we can reduce the Lunar Module into a point mass. Two major coordinate systems are used in this thesis - Polar and Cartesian. For the computation of the optimal descent trajectory the polar coordinate system has been chosen, because it provides mathematical simplifications in motion equations. In 3D application for the trajectory visualization the Cartesian system is used. Since the motion of the Lunar Module is mainly within one plane, we can work in 2D coordinate system during the optimal trajectory computation.

Polar coordinate system

In Polar coordinate system, a point M is represented by an ordered pair of numbers (r, θ) , where r is distance of point M from coordinate origin 0 and θ is an angle between positive part of x axis and join of coordinate origin 0 and point M, $r \in [0, \infty]$ and $\theta \in [0, 2\pi]$. The polar coordinates can be converted to Cartesian coordinates [x, y] by using *sin* and *cos* [16]:

$$x = r.\cos\theta \tag{3.1}$$

$$y = r.\sin\theta \tag{3.2}$$



Figure 3.1: Polar coordinate system

Cartesian coordinate system

The Cartesian coordinate system is defined by two mutually perpendicular lines. They are called coordinate axes. Each of the coordinate axes has an orientation and a unit length. The orientation of the axis defines its positive and negative half with the coordinate origin (0,0) in the point of the axes intersection.

The point M is represented by an ordered pair of numbers (x, y). We determine the values of the x and y coordinate by drawing a line through the M which is perpendicular to the x and y axis respectively. The position of the intersection with the axis is then interpreted as a value of the coordinate.



Figure 3.2: Cartesian coordinate system

Position the of Lunar Module in polar coordinate system

The position of LM is constrained within inertial frame¹, which is defined by axes I_x and I_y . They originate in point F which symbolizes the center of Moon's gravity. Moon equatorial radius R_{eq} is defined in Table A.2. The LM's position is then defined by polar coordinates r and θ . Next an engine thrust vector u_T is defined along with the thrust direction angle ψ^r . ψ^r is defined from the radius vector r to the engine thrust vector u_T [10].



Figure 3.3: Lunar Module position in polar coordinate system

Additionally to the fixed inertial frame, the rotating frame $F = (LM, \hat{i}, \hat{j})$ is defined. It will be used for derivation of velocity and acceleration components for the Equations of Motion (EOM). Frame axis j is perpendicular to the frame axis i, which is parallel to the Lunar Module radial coordinate r.

 $^{^{1}}$ Inertial frame is a frame of reference in which a body with zero net force acting upon it is not accelerating - it is at rest or is moving at constant speed in a straight line.



Figure 3.4: Inertial and rotating frames

3.2 Physics of Lunar landing problem

The physics is modeled with use of Newtonian classical mechanics. In order to create the equations of motion, the Newton's laws will be needed. Three Newton's laws of motion are [6]:

First Law

Every object continues in its state of rest or of uniform motion in a straight line unless it is compelled to change that state by forces impressed upon it.

The first law requires the identification of inertial system (see Figure 3.3) where the motion is defined.

Second Law

The rate of change of momentum is proportional to the force impressed and is in the same direction as that force. It can be expressed as:

$$\vec{F} = \frac{d\vec{p}}{dt} \tag{3.3}$$

where \vec{F} is the resultant of the forces acting on the object and $\vec{p} = m\vec{v}$ is object's momentum. For an object which mass *m* changes over time, the equation of motion is defined as:

$$\vec{F} = m\vec{a} + \dot{m}\vec{v} \tag{3.4}$$

where \vec{a} is object's acceleration.

Third Law

To every action there is always opposed an equal reaction.

Law of Universal Gravitation

Newton also formulated the law of Universal Gravitation by stating, that two bodies, the masses of which are M and m, respectively, attract one another along the line joining them with a force proportional to the product of their masses and inversely proportional to the square of the distance between them [6]:

$$F = G \frac{Mm}{r^2} \tag{3.5}$$

where G is the Universal Gravitational Constant and it equals to $6.673 \times 10^{-11} m^3 kg^{-1}s^{-2}$.

Equations of motion

The Moon landing is a two-body problem. Two bodies are represented by the Moon and the Lunar Module respectively. These bodies interact with each other within the rules of Newton's laws. The equations of motion can be derived from those interactions.

In system of two bodies with masses m_1 and m_2 where $m_1 > m_2$ we can define the position of m_2 relative to m_1 as [6]:

$$\vec{r} = \vec{R_2} - \vec{R_1} \tag{3.6}$$

where $\vec{R_1}$ and $\vec{R_2}$ describe the position of masses m_1 and m_2 respectively.

When derived, one can obtain object velocity and its acceleration:

$$\dot{\vec{r}} = \dot{\vec{R}_2} - \dot{\vec{R_1}}$$
 (3.7)

$$\ddot{\vec{r}} = \vec{\vec{R}_2} - \vec{\vec{R}_1} \tag{3.8}$$

With use of Newton's Law of Universal Gravitation, the force acting on each mass is used to describe its motion:

$$m_1 \ddot{\vec{R}_1} = G \frac{m_1 m_2}{r^3} \vec{r}$$
(3.9)

$$m_2 \ddot{\vec{R_2}} = -G \frac{m_1 m_2}{r^3} \vec{r}$$
(3.10)

After subtracting the second equation from first, the relative motion can be defined as:

$$\ddot{\vec{r}} = -G\frac{m_1 + m_2}{r^3}\vec{r}$$
(3.11)

Since mass of the Moon is larger than mass of lunar module by several orders of magnitude $(m_1 = M \text{ and } m_2 = m, \text{ with } M \gg m)$, the equation of relative motion can be simplified to [6]:

$$\ddot{\vec{r}} = -\frac{\mu^3}{r}\vec{r} \tag{3.12}$$

where $\mu = GM$ is gravitational parameter (see Table A.2).

In rotating frame (Figure 3.4), we can define the velocity and acceleration components:

$$\hat{i} = \cos\theta \hat{I}_x + \sin\theta \hat{I}_y \tag{3.13}$$

$$\hat{j} = -\sin\theta \hat{I}_x + \cos\theta \hat{I}_y \tag{3.14}$$

When derivation in time is performed we get:

$$\hat{i} = \dot{\theta}\hat{j} \tag{3.15}$$

$$\dot{\hat{j}} = -\dot{\theta}\hat{i} \tag{3.16}$$

The velocity vector now can be expressed as:

$$\vec{v} = v_r \hat{i} + v_\theta \hat{j} \tag{3.17}$$

where v_r is radial velocity and v_{θ} tangential velocity. After derivation of the position vector, the radial and tangential velocity can be defined as [6]:

$$v_r = \dot{r} \tag{3.18}$$

$$v_{\theta} = r\dot{\theta} \tag{3.19}$$

According to [6], when there are other forces acting on the spacecraft - such as thrust of the engine, the vector equation of motion can be expressed as:

$$\ddot{\vec{r}} = -\frac{\mu}{r^3}\vec{r} + \vec{a_T} \tag{3.20}$$

where $\vec{a_T}$ is acceleration of the engine. This vector can be split in radial and tangential components:

$$a_r = \ddot{r} - r\dot{\theta}^2 = -\frac{\mu}{r^2} + a_T \cos\psi^r$$
(3.21)

$$a_{\theta} = r\ddot{\theta} + 2\dot{r}\dot{\theta} = a_T \sin\psi^r \tag{3.22}$$

where ψ^r is an angle of engine thrust defined in Chapter 3.1.

The acceleration vector can be expressed as [10]:

$$\vec{a_T} = \frac{T_{max}k}{m} \tag{3.23}$$

where T_{max} is engine's maximal thrust defined in Table 2.2 and $k \in [0,1]$ is the throttle command, which is used to regulate the engine power.

Because:

$$v_r = \dot{r} \Rightarrow \dot{v_r} = \ddot{r} \tag{3.24}$$

$$v_{\theta} = r\dot{\theta} \Rightarrow \dot{v}_{\theta} = \dot{r}\dot{\theta} + r\ddot{\theta}$$
(3.25)

the equations of motion can be derived as [6]:

$$\dot{r} = v_r \tag{3.26}$$

$$\dot{\theta} = \frac{v_{\theta}}{r} \tag{3.27}$$

$$\dot{v_r} = \frac{v_{\theta}^2}{r} - \frac{\mu}{r^2} + \frac{T_{max}k}{m}\cos\psi^r$$
 (3.28)

$$\dot{v_{\theta}} = -\frac{v_r v_{\theta}}{r} + \frac{T_{max}k}{m} \sin \psi^r \tag{3.29}$$

$$\dot{\psi}^r = \omega - \dot{\theta} \tag{3.30}$$

$$\dot{\omega} = \alpha \tag{3.31}$$

where ω is the inertial angular velocity and α an inertial angular acceleration, which can be regulated.

The flow of the lunar module's mass in time can be derived as [10]:

$$\dot{m} = -\frac{T_{max}k}{V_{ex}} \tag{3.32}$$

where V_{ex} is the lunar module engine specific impulse defined in Table 2.2.

With equations of the motion derived, we can proceed to the computation of the optimal descent trajectory, which is content of the next chapter.

Chapter 4

Optimal descent trajectory computation

The content of this chapter is focused on computation of optimal descent trajectory. First the mathematical theory behind optimization problems is explained. Following is an introduction to the used optimization software. Finally, the definition and implementation of discussed problem is described and the results are presented.

4.1 Optimization problem theory

Optimization is the act of achieving the best possible result under given circumstances. The best can vary. For example when we are using optimization in finance we may want to **maximize** the profit. On the other hand, in trajectory optimization problems we are usually trying to **minimize** the fuel consumption. This effort can be expressed as a function of certain parameter variables. Hence, optimization is the process of finding the conditions that give the maximum or the minimum value of a function.

If point x^* corresponds to the minimum value of a function f(x), the same point corresponds to the maximum value of the function -f(x). We can thus take the optimization as minimization [3].

General definition of optimization problem

According to [3], an optimization problem can be stated as follows: Find

$$x = (x_1, x_2, \dots, x_n) \tag{4.1}$$

which minimizes

 $f(x) \tag{4.2}$

subject to the constraints

$$g_j(x) \le 0 \tag{4.3}$$

for $j = 1, \ldots, m$, and

$$l_j(x) = 0 \tag{4.4}$$

for j = 1, ..., p.

Objective function f(x) is to be minimized over the n-variable vector of parameters x, $g_j(x)$ are the inequality constraints and $l_j(x)$ are the equality constraints. The number of variables n and the number of constraints p + m need not be related. If p + m = 0 the problem is called an unconstrained optimization problem.

Objective function

If the objective function f(x) is convex, its local minimum is also the global minimum. If the function is nonconvex, finding a local minimum does not imply finding a global minimum. This concept is displayed in Figure 4.1. The problem of optimal descent trajectory is nonconvex [10].



Figure 4.1: Global and local minimum

Finding a minimum

For a single variable function we can find a local minimum by applying the first and second derivation of a function [12]:

Let $f : D \subseteq \mathbb{R}^n \to \mathbb{R}$. A local minimum of a function f is a point $a \in D$ such that $f(x) \geq f(a)$ for x near a. The point a then must be a critical point of f, which means that f'(c) = 0. To ensure that the point a is a local minimum we can use *Second Derivative Test*, which states that if a is a critical point of f and f''(a) > 0, then a is a local minimum. According to [12], this test can be generalized to the multivariable case as follows:

First we form the *Hessian*, which is the matrix of second partial derivatives at a. If f is a function of n variables, then the Hessian is an $n \times n$ matrix. The entry in row i and column j of H is defined by:

$$H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}(a) \tag{4.5}$$

The Second Derivatives Test now can be used. If a is a critical point of f and the Hessian H is positive definite, then a is a local minimum. To determine if the matrix H is positive definite, we can apply number of statements, including that the determinate of H is positive.

Optimal control problem

Problem of optimal descent trajectory falls into the subset of optimization problems called optimal control problems. These problems involve a controlled dynamical system. A controlled dynamical system is a dynamical system in which the trajectory can be altered continuously in time by choosing a control parameter u(t) continuously in time [9].

As stated in [9], a controlled dynamical system is usually governed by differential equation of the form:

$$\dot{x}(t) = f(t, x(t), u(t)), t > 0 \tag{4.6}$$

$$x(0) = x_0 \in \mathbb{R} \tag{4.7}$$

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 history to minimize the objective (cost) function J. Function c is the cost function and t_f is the final time [9]:

$$J(u) := \int_0^{t_f} c(t, x(t), u(t)) dt$$
(4.8)

Methods for solving optimal control problems

We can try to solve an optimal control problem analytically, but this may be difficult or even impossible. In practice, the numerical methods are used instead. These methods can be divided into two main categories: direct and indirect methods.

The *indirect methods* provide more precise results, but user have to derive and construct various equations for necessary and sufficient conditions, which may be again very difficult.

Direct methods start from an initial guess of the state and control variables and search in the feasible region for a minimum of cost function [4]. It is much easier to guess an initial state and control variables, thus direct methods are usually chosen.

Midpoint method

The used optimization software computes using a direct method approach. User can choose between a variety of different methods. In optimization software it was observed, that *Midpoint* method provides the best results within reasonable computation time.

Midpoint method is based on Euler method. It is one-step, which means we are using information from a single previous step. We are trying to solve a differential equation [8]:

$$y'(x) = f(x,y), \ y(x_0) = y_0$$
(4.9)

To compute y_{i+1} we use the following formulas:

$$k1 = f(x_i, y_i) \tag{4.10}$$

$$k2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_1)$$
(4.11)

$$y_{i+1} = y_i + hk_2 \tag{4.12}$$

for $i = 0, \ldots, n - 1$. h is step size.

Geometrically, we first find point P which lies on the line with slope $f(x_i, y_i)$ and its x coordinate is $x_i + \frac{h}{2}$. Then the approximate value of the function in x_{i+1} lies on the line with slope field in point P. This principle is displayed in Figure 4.2 retrieved from [8].



Figure 4.2: One step of midpoint method [8]

4.2 Implementation of optimal descent trajectory

In this section the optimization software is described along with own implementation of optimal descent trajectory problem.

Bocop

An open-source toolbox for solving optimal control problems - Bocop [34] was chosen for implementation of optimal descent trajectory problem. The optimal control problem is approximated by a finite dimensional optimization problem (NLP) using a time discretization. The NLP problem is solved by software *IPOPT* [40], using sparse exact derivatives computed by ADOL-C [36].

Optimal control problem in Bocop can be defined as [5]:

$$(P) = \begin{cases} \operatorname{Min} J(t_0, y(t_0), t_f, y(t_f), \pi) & \text{Objective} \\ \dot{y}(t) = f(t, u(t), y(t), z(t), \pi) & \text{Dynamics} \\ \Phi_l \leq \Phi(t_0, y(t_0), t_f, y(t_f), \pi) \leq \Phi_u & \text{Boundary Conditions} \\ y_l \leq y \leq y_u, u_l \leq u \leq u_u, z_l \leq z \leq z_u, \pi_l \leq \pi \leq \pi_u & \text{Bounds} \\ g_l \leq g(t, u(t), y(t), z(t), \pi) \leq g_u & \text{Path Constraints} \end{cases}$$

with $y(\cdot)$ the state variables, $u(\cdot)$ the control, $z(\cdot)$ the optional algebraic variables and π the optional variables to be optimized. t_0 is starting time and t_f corresponds to the final time.

Bocop code structure

The problem is saved in its own directory containing a number of files:

- 4 functions in C/C++, which corresponds to J, f, Φ, g: criterion, dynamics, boundarycond and pathcond.
- 3 definition files in plain text: problem.def for general definition of the problem problem.bounds for the bounds (no bounds, lower and/or upper bound, equality) problem.constants for optional constant values for the problem.

Program pipeline is presented in Figure 4.3 which is based on a diagram retrieved from [5]. Input files are first processed, Bocop performs a time discretization to build an objective and constraints functions. These along with a starting point are the input of IPOPT optimization tool. Derivations of objective and constraints functions are computed by ADOL-C. The results are saved in output files and presented to user in program's GUI.



Figure 4.3: Bocop pipeline [5]

Bocop GUI

Bocop provides an intuitive GUI written in Qt framework. It is organized in 4 main modules:

Problem Definition module allows user to define an optimal control problem by providing dimensions and names for variables, constants, functions for the objective, dynamics and constraints. User can also choose computation algorithm - Euler, Midpoint, Runge Kutta and others.

In *Starting Point* module the starting point values are set. They can be constant, linear or splines.

Optimization module serves as settings panel for IPOPT. Different options can be set including maximum number of iterations, output file name and single or batch optimization among other things.

Finally in *Visualization* module user can read and display the contents of the solution file generated after optimization. It includes graphs of state and control variables, constraints and other functions in time.

Figure 4.4 shows a screen-shot of the Bocop's main window, captured on the author's computer.



Figure 4.4: Main window of Bocop

Optimal descent trajectory problem

The objective of this optimization is to perform a soft landing of a lunar module, while minimizing the fuel usage. Soft landing means, that the lunar module's final pitch and radial velocity should be close to 0 deg (legs down) and 0 ms^{-1} respectively to avoid any potential damage.

Definition of the problem

The *state vector* is defined by equations of motion described in Chapter 3.2:

$$x(t) = [r(t), \theta(t), v_r(t), v_\theta(t), \psi^r(t), \omega(t), m(t)]$$
(4.13)

The *control vector* is composed of engine throttle command k and angular acceleration command α :

$$u(t) = [k(t), \alpha(t)] \tag{4.14}$$

The *objective function* is based on final mass of the lunar module:

$$J_{min} = -m(t_f) \tag{4.15}$$

The constants have been chosen from Tables A.2 and 2.1. T_{max} and V_{ex} are the Lunar Module engine maximum thrust limit and engine exhaust velocity respectively. R_{eq} is the Moon equatorial radius and μ its gravitational parameter. Values are in SI units. The equations of motion operate with radians. All degrees were therefore converted to radians. Table 4.1 shows the constants used in the optimal descent trajectory problem.

Table 4.1: Constants used in the optimal descent trajectory problem

T_{max}	$45 \ 040 \ N$
V_{ex}	$3050 \ ms^{-1}$
R_{eq}	$1.7381 \times 10^6 m$
μ	$4.902800066 \times 10^{12} \ m^3 s^{-2}$

Table 4.2 shows the *bounds* for state and control variables.

Table 4.2: Boundaries for state and control variables

Variable	Lower Bound	Upper Bound
r(t)	$1\ 738\ 100\ m$	$1\ 753\ 340\ m$
$\theta(t)$	$0 \ deg$	180 <i>deg</i>
$v_r(t)$	-	$0 m s^{-1}$
$v_{\theta}(t)$	$0 \ ms^{-1}$	-
$\psi^r(t)$	$-90 \ deg$	0 deg
$\omega(t)$	$-10 \ deg \ s^{-1}$	$10 \ deg \ s^{-1}$
m(t)	$6855 \ kg$	$15 \ 103 \ kg$
k(t)	0	1
$\alpha(t)$	$-0.5 \ deg \ s^{-2}$	$0.5 \ deg \ s^{-2}$

The limits on angular acceleration command and angular velocity were set according to [10] to limit the rotational motion of LM to the reasonable levels. r represents magnitude of

the radius vector. Since it originates in the center of Moon gravity, zero altitude is equal to Moon's equatorial radius and represents the lower bound. Upper bound is then computed as Moon's radius + starting altitude of 15.24 km. Central angle θ is limited to 180 deg. Radial velocity of lunar module is upper limited to $0 ms^{-1}$, because LM should perform descent. Analogically the lower bound for tangential velocity is set to $0 ms^{-1}$ to ensure the forward movement. Bounds of ψ^r angle represent the LM pitch of [-90, 0] deg. Mass of the vehicle without the propellant was computed to be 6855 kq and it represents the lower bound for the LM's mass m. Throttle command is set within the interval [0,1] where 1 represents full throttle and 0 the engine turned off.

Finally, the initial and final conditions have to be constructed:

Initial conditions

$$r_0 = 1 \ 753 \ 340 \ m \tag{4.16}$$

$$v_{\theta_0} = 1630 \ ms^{-1} \tag{4.17}$$

$$\psi_o^r = -90 \ deg \tag{4.18}$$

$$m_0 = 15 \ 103 \ kg \tag{4.19}$$

Powered descent starts in altitude of 15.24 km with tangential velocity 1630 ms^{-1} . Pitch angle of lunar module is set to $-90 \ deg$ and its initial mass is at its maximum value 15 103 kg (counting with crew of three astronauts as in Apollo 11 mission).

Final conditions

$$r_{tf} = 1\ 738\ 100\ m \tag{4.20}$$

$$r_{tf} = 1\ 738\ 100\ m \tag{4.20}$$

$$v_{\theta_{tf}} = 0\ ms^{-1} \tag{4.21}$$

$$0 \le v_{r_{tf}} \le 0.5 \ ms^{-1} \tag{4.22}$$

$$-0.5 \le \psi_{tf}^r \le 0.5 \ deg$$
 (4.23)

The lunar module should land on the ground, with zero tangential velocity. Final radial velocity constraints are relaxed, because problem didn't converged when equality constraint was set. Same applies to the final thrust angle - interval between -0.5 deq and 0.5 deq is reasonable if near vertical landing is desired. LM was constructed to remain stable in even greater pitch angle [10].

Results of the optimization

Results of the optimization are presented below. As seen on graphs, the lunar module smoothly descends from the initial altitude of $15.34 \ km$ while performing a pitch maneuver to reach the vertical position. Throttle command is kept on maximum while the lunar lander performs a soft landing with the final radial velocity of 0 ms^{-1} and pitch at 0 deg. We can observe, that the vehicle mass is well above the minimal value upon landing, thus the optimization can be considered successful.

Figure 4.5 shows the optimal descent trajectory altitude in time. In Figure 4.6 we can observe the central angle of the lunar module in time. Figure 4.7 shows the LM's radial velocity. At its maximum it reaches $-60 m s^{-1}$. Tangential velocity is displayed in Figure 4.8. Figure 4.9 shows the LM's pitch angle in time. Lunar module gradually reaches the pitch angle of $-60 \ deg$ and then changes its attitude to almost $0 \ deg$. Figure 4.10 shows the LM's mass in time. In Figure 4.11 we can see the throttle command of the LM's engine in time. It is kept on the maximum up until the end of the landing maneuver.





Figure 4.5: Optimal descent trajectory altitude in time



Figure 4.6: Optimal descent trajectory central angle in time



velocity in time

Figure 4.7: Optimal descent trajectory radial Figure 4.8: Optimal descent trajectory tangential velocity in time





module pitch angle in time

Figure 4.9: Optimal descent trajectory, lunar Figure 4.10: Optimal descent trajectory, lunar module mass in time



Figure 4.11: Optimal descent trajectory, lunar module throttle command in time

Chapter 5

Design and implementation of visualization environment for descent maneuver interpretation

In this chapter the process of implementation of 3D application for visualization of lunar module descent trajectory is described. Reader can learn about tools and methods used for creating this application. The process of creating input dataset containing trajectory coordinates is also explained.

5.1 Application implementation

The idea of this application is to provide the user with an intuitive graphic way to visualize a descent trajectory from the dataset of coordinates. These coordinates were obtained from the computed optimal trajectory (see Chapter 4) and also from the real mission data of Apollo 11 mission [25]. Creation of the input dataset is explained in detail in the next section.

Godot Engine

The application was created using open source (MIT license) 3D engine Godot [13]. It is cross-platform, which means you can deploy your application to any major operating system as well as to mobile and web (HTML5). Throughout the development, its documentation was extensively used [14]. A brief explanation of the engine work-flow follows, along with description of own implementation.

Two main building blocks of Godot Engine are **nodes** and **scenes**. A node always has the following attributes [14]:

- It has a name.
- It has editable properties
- It can receive a callback to process every frame.
- It can be extended (to have more functions).
- It can be added to another node as a child.

A scene is composed of a group of nodes. Running an application means running a scene. This allows for creation of different program views.

Application composition

Application is composed of two scenes - menu and main scene. A simple menu allows the user to load a dataset of coordinates. File dialog is presented and after selecting the dataset file, the program switches to the main scene and animation of descent trajectory starts playing. Figure 5.1 shows the application's main menu.



Figure 5.1: Screenshot of the application menu

For the main scene, various assets have been downloaded or created. This includes the Lunar Module model [32] and Apollo 11 landing site model [7]. The Moon was created as a sphere mesh with seamless surface texture [2] applied to it. Space skybox generator [35] was used for the environment creation. It generates a panorama texture which is wrapped around the entire scene for visualization of space beyond the 3D geometry.

In the main view there are several control elements. User can play or pause descent animation, change its speed and also play it backwards. Camera distance and angle can be changed either by mouse scroll wheel or by clicking on *cockpit* button to view the descent from the astronaut's perspective. The real-time flight data are displayed in the left corner. These include yaw, pitch and roll angles, distance of LM from landing site and its altitude. The trajectory itself is displayed as a curve. Figure 5.2 shows the main view of application.



Figure 5.2: Screenshot of application main view

Implementation details

Descent trajectory is created with help of the Path and the PathFollow nodes. The Path-Follow node takes its parent node - Path and returns the coordinates of a point within it, given a distance from the first vertex [14]. The lunar module movement can then be animated by setting the offset of PathFollow node. It also performs a cubic interpolation, so the movement is smoothed.

The Path node serves as container for Curve3D node. Curve3D describes a Bézier curve in 3D space[14]. Modified coordinates from the input dataset are points of the curve.

The modification of dataset points is necessary, because of the curvature of the Moon mesh. The method of Ray-casting is used to exactly determine the Y axis value in 3D space.

Next the Bézier curve segments have to be created. Godot defines a point of Bézier curve as:

$$K = [pos, in, out] \tag{5.1}$$

where *pos* is position of the point in 3D space. *in* and *out* are control points of the curve. All parameters are stored in Vector3 data structure which is composed of (x, y, z) coordinates. Control point position is defined locally from the *pos* vector. In order to create curve segments, the coordinates of control points have to be computed. Method was implemented for this purpose. It is based on the article about creation of smooth Bézier spline through prescribed points [28].

The cubic Bézier curve is defined as [28]:

$$B(t) = (1-t)^{3}P_{0} + 3(1-t)^{2}tP_{1} + 3(1-t)t^{2}P_{2} + t^{3}P_{3}, t \in [0,1]$$
(5.2)

where P_0 , P3 are knot points and P_1 , P_2 are control points.

First derivative of cubic is expressed as [28]:

$$B'(t) = -3(1-t)^2 P_0 + 3(3t^2 - 4t + 1)P_1 + 3(2t - 3t^2)P_2 + 3t^2 P_3$$
(5.3)

Second derivative is equal to [28]:

$$B''(t) = 6(1-t)P_0 + 3(6t-4)P_1 + 3(2-6t)P_2 + 6tP_3$$
(5.4)

To make a sequence of individual Bézier curves to be a spline, the control points have to be calculated so that the spline curve has two continuous derivatives at knot points [28]. Continuous derivative means, that the derived function is continuous in the same interval as the original function.

From these assumptions we can build several conditions. The first derivative continuity condition states [28]:

$$B_{i-1}'(1) = B_i'(0) \Leftrightarrow \tag{5.5}$$

$$P1_i + P2_{i-1} = 2Pi - 1; \dots (i = 2, ..., n)$$
(5.6)

where *i* is *i*th knot point, $i = 1, \ldots, n$.

The second derivative continuity condition can be expressed as [28]:

$$B_{i-1}''(1) = B_i''(0) \Leftrightarrow \tag{5.7}$$

$$P1_{i-1} + 2P1_i = P2_i + 2P2_{i-1}; \dots (i = 2, ..., n)$$
(5.8)

Next the conditions for ends of intervals have to be constructed [28]:

$$B_1''(0) = 0 \Leftrightarrow 2P1_1 - P2_1 = P_0 \tag{5.9}$$

$$B_n''(1) = 0 \Leftrightarrow 2P2_n - P1_n = P_n \tag{5.10}$$

From this system of condition equations, the individual control points of curve segments can be computed. Figure 5.3 shows a Bézier curve representation.



Figure 5.3: Bézier curve representation

Curve3D class cannot store additional information about lunar module pitch. For the purpose of setting the module's pitch at given point another method was implemented. It computes the current point on the trajectory curve by finding the shortest distance between the lunar module and points of the curve. The corresponding pitch is then set from the input dataset.

5.2 Creation of the input dataset

The coordinates for visualization application were created from two sources. First, the real Apollo 11 descent trajectory was reconstructed from the mission report graphs [25]. Second, data from the optimization were used to create the coordinates of optimal descent trajectory. Description of creating the input datasets follows.

Apollo 11 descent trajectory dataset

Since the graphs of descent trajectory in Apollo mission report were available only in the form of images, it was necessary to perform their digitization. For this purpose, the WebPlotDigitizer [29] tool was used. It is an open source web application built in HTML5, which allows to extract exact numerical data from various types of graph images.

It uses several algorithms for the data extraction. The X Step with Interpolation algorithm has been chosen. As mentioned in user manual [30], it can identify data points at regular intervals on the X axis, that fall between X_{min} , X_{max} and Y_{min} , Y_{max} respectively. The important feature of this algorithm is that the data points are spaced at an interval ΔX units apart. This provides the user a better control of the final dataset. Algorithm interpolates over missing data using cubic splines, which makes it suitable even for curves consisting only of data points.

First the graph axes have to be manually calibrated. This allows for program to correctly map the image pixels to the corresponding data values. Next, the user selects data points on the image. Several tools like Box, Pen and Erase are available to mark the region containing the required data. After this selection, the data extraction algorithm can be run. After the computation, user is presented with dataset of (x, y) values.

Figure 5.4 shows the automatically identified points of the Apollo 11 descent trajectory graph.



Figure 5.4: Visual output from WebPlotDigitizer application containing one part of the Apollo 11 descent trajectory graph

Several graphs were processed in this manner, including the graph of lunar module pitch in time. This allows for better visualization of descent maneuver.

In the final step of input dataset creation, the individual outputs of digitization application have been merged together and file containing the range from the landing point, altitude and pitch angle of the lunar module was created. The input file contains individual values in CSV format, meaning that each value is separated by a comma. Figure 5.5 shows the final digitized trajectory of Apollo 11.



Figure 5.5: Digitized of Apollo 11 descent trajectory, altitude vs. distance from the landing site

Optimal trajectory dataset

Optimization software output provided numerical data which could be easily used to create trajectory dataset. In simple text files user can find discretized time units and corresponding variable values. A spreadsheet was created to collect this data and to create necessary graphs.

In order to create the optimal trajectory dataset, the distance from landing site in each time point had to be calculated. For this purpose the following formula was used:

$$\Delta d = \Delta \theta (R_{eq} + alt) \tag{5.11}$$

where Δd represents a travelled distance increment, $\Delta \theta$ an angular increment, R_{eq} is the Moon equatorial radius and *alt* is lunar module altitude in current time point. Total travelled distance can be computed as sum of travelled distance increments.

It should be noted, that distance from the landing site is only an approximation. The error can be caused by various elements, including a round off error of the angular increment.



Figure 5.6: Optimal descent trajectory altitude vs. distance from the landing site

The pitch data was processed in the same manner. Figure 5.6 shows the optimal descent trajectory altitude according to the distance from the landing site. In Figure 5.7 we can see the optimal descent trajectory pitch according to the distance from the landing site.



Figure 5.7: Optimal trajectory pitch vs. distance from the landing site

Chapter 6

Evaluation of achieved results

In this chapter the results of the final trajectories are evaluated and potential further improvements of both optimization task and visualization application are suggested.

Comparison of descent trajectories

As seen in Figure 6.1, total travel distance of the optimal trajectory is approximately 61 km shorter then the total travel distance of the Apollo 11. The throttle of the lunar module engine is kept on the maximum until the end of the landing maneuver. LM reaches a considerable radial velocity of $-60 m s^{-1}$ and is thus able to reach the landing site quicker than in the real Apollo 11 mission. The computed optimal trajectory reached its initial goal to perform a soft landing while minimizing the fuel consumption. Propellant mass of 6528.63 kg was burned during descent. That leaves the lunar module with 1719.37 kg of the remaining fuel.

The real Apollo 11 descent started with the engine turned on. Throttle was reduced after approximately 386 s after ignition. The lunar module reached the high gate altitude of about 2 km at descent rate of approximately 38 ms^{-1} . Just before landing, the Apollo guidance computer indicated that the approach path was leading into a large crater. Manual intervention was necessary to extend range and avoid the crater. This maneuver guided the lunar module 335 m from the initial landing point. Vertical velocity upon landing was $1.7 ms^{-1}$. 316.6 kg of usable propellant remained at landing [25], significantly less than estimated, mainly due to the crash avoidance maneuver.

Further improvements

The optimization of descent trajectory could be done in 3D space instead of one plane of motion. This would allow for the more realistic simulation of the descent maneuver. Another potential improvement would be to add different constraints for different altitudes and thus more accurately simulate individual phases of descent maneuver. This is not possible in the current version of optimization software.

The graphics of the application would benefit from the precise topological terrain of the lunar surface. The topological data from the Lunar Reconnaissance Orbiter Camera are available [31], but since the current version of Godot Engine does not support a displacement mapping technique, it was not possible to implement without modifying the source code. Currently this would be both very difficult and out of scope of this thesis. Another possible improvement of the visualization application is to implement full six degrees of freedom

movement of the lunar module. This was not implemented in the current version because the available data was only in one plane of the motion. In the future version of the application, the proper physical model of the lunar module could be implemented. This would allow for more realistic animation and additional parameters of the flight could be simulated.



-Apollo 11 trajectory - Optimal trajectory

Figure 6.1: Comparison of Apollo 11 and optimal descent trajectories

Chapter 7

Conclusion

The goal of this thesis was to perform a computation of optimal descent trajectory of lunar module and visualize it in 3D environment. First step to achieve this goal was to research the history of Apollo 11 mission. For this purpose various literature was read including Apollo 11 mission report [25]. All necessary historical data was extracted, including trajectory description and lunar module technical parameters. With this knowledge it was possible to approach the optimization problem. It was crucial to place the lunar module into coordinate system and derive the equations of motion. Next the optimization theory was researched and optimal descent trajectory was modeled as optimal control problem. The author got acquainted with optimization software Bocop and implemented this problem within its environment. In order to visualize the descent maneuver the 3D engine Godot was chosen. Application logic was implemented. This includes designing the input format for the application, parsing the input dataset and creating an interactive 3D animation. The input dataset was created by digitization of various graphs from Apollo 11 mission and by converting the optimization data. Finally, the optimal and real descent trajectory were compared.

Further work may include generation of the Moon topology from satellite data in order to enhance the visual quality of the presentation. Additionally the computation of optimal trajectory could be extended to the 3D space. This would enable creation of more realistic simulation system. Another possible future work could be done on implementation of realistic physical model in 3D engine.

Bibliography

- [1] Allen, B.: NASA Langley Research Center's Contributions to the Apollo Program.
 [Online; visited 12.02.2019].
 Retrieved from: https://www.nasa.gov/centers/langley/news/factsheets/Apollo.html
- [2] Anonymous: Seamless Moon texture. [Online; visited 09.04.2019]. Retrieved from: https://imgur.com/a/aw3nD
- [3] Astolfi, A.: Optimization An introduction. 09 2006. Retrieved from: http://www3.imperial.ac.uk/pls/portallive/docs/1/7288263.PDF
- Betts, J. T.: Practical methods for optimal control and estimation using nonlinear programming. 2nd ed. 01 2010.
 Retrieved from: https://epdf.tips/queue/practical-methods-for-optimalcontrol-and-estimation-using-nonlinear-programming.html
- [5] Bonnan, F.; Martinon, P.; Giorgi, D.; et al.: BOCOP 2.1.0 User Guide. Retrieved from: https://files.inria.fr/bocop/UserGuide-BOCOP.pdf
- [6] Colasurdo, G.: Astrodynamics. 2006. Retrieved from: http://dma.ing.uniroma1.it/users/lss_mo/MATERIALE/ AvanziniColasurdoAstrodynamics.pdf
- [7] Erickson, K.: Apollo 11 Landing Site 3D Model. [Online; visited 4.03.2019]. Retrieved from: https://nasa3d.arc.nasa.gov/detail/Apollo11-Landing
- [8] Fajmon, B.; Hlavičková, I.; Novák, M.; et al.: Numerická matematika a pravděpodobnost (Informační technologie). 2014.
 Retrieved from: https://www.vutbr.cz/studenti/predmety/detail/174531?apid=174531
- [9] Ghosh, M. K.: Optimal Control Theory. Retrieved from: http://www.iisertvm.ac.in/assets/uploads/downloads_buffer/ 2019_05_15_15_21_06/optimal-control.pdf
- [10] Hawkins, A. M.: Constrained trajectory optimization of a soft lunar landing from a parking orbit. 03 2006.
- [11] Johnston, L.; Williamson, S. H.: What Was the U.S. GDP Then? [Online; visited 12.02.2019].
 Retrieved from: http://www.measuringworth.org/usgdp/

- [12] Lambers, J.: Maximum and Minimum Values. 2009. Retrieved from: http://www.math.sci.hokudai.ac.jp/~s.settepanella/ teachingfile/Calculus/Calculus1/pagine/lecture8.pdf
- [13] Linietsky, J.; Manzur, A.: Godot Engine. [Online; visited 14.11.2018]. Retrieved from: https://godotengine.org/
- [14] Linietsky, J.; Manzur, A.: Godot Engine Documentation. [Online; visited 14.11.2018]. Retrieved from: https://docs.godotengine.org/
- [15] LLC, E. I.: Saturn V (1:100th scale). [Online; visited 4.03.2019]. Retrieved from: https://estesrockets.com/product/001969-saturn-v/
- [16] Mařík, R.: Polar coordinates. [Online; visited 20.03.2019]. Retrieved from: http://user.mendelu.cz/marik/wiki/in-mat-web/in-mat-webse21.html
- [17] Murray, C. A.; Cox, C. B.: Apollo: The Race to the Moon. Simon and Schuster. 1989. ISBN 0671611011.
- [18] NASA: Apollo 11 Quick Reference Data. [Online; visited 4.03.2019]. Retrieved from: https://www.hq.nasa.gov/alsj/LM04_Lunar_Module_ppLV1-17.pdf
- [19] NASA: Apollo Lunar Module Propulsion Systems Overview. [Online; visited 4.03.2019].
 Retrieved from: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20090016298.pdf
- [20] NASA: Command Module Overview. [Online; visited 4.03.2019]. Retrieved from: https://www.hq.nasa.gov/alsj/CSM06_Command_Module_Overview_pp39-52.pdf
- [21] NASA: Full Earth image. [Online; visited 4.03.2019]. Retrieved from: https://web.archive.org/web/20160112123725/http: //grin.hq.nasa.gov/ABSTRACTS/GPN-2000-001138.html
- [22] NASA: Saturn V news reference. [Online; visited 4.03.2019]. Retrieved from: https://history.msfc.nasa.gov/saturn_apollo/documents/Introduction.pdf
- [23] NASA: Service Module Overview. [Online; visited 4.03.2019]. Retrieved from: https://www.hq.nasa.gov/alsj/CSM07_Service_Module_Overview_pp53-60.pdf
- [24] NASA: Apollo 11 Press Kit. Technical report. NASA. 1969. Retrieved from: https://www.nasa.gov/specials/apollo50th/pdf/A11_PressKit.pdf
- [25] NASA: Apollo 11 Mission Report. Technical report. NASA Manned Spacecraft Center. 1971.
 Retrieved from: https://www.hq.nasa.gov/alsj/a11/a11MIssionReport_1971015566.pdf

- [26] Northon, K.: NASA Unveils Sustainable Campaign to Return to Moon, on to Mars. [Online; visited 10.02.2019]. Retrieved from: https://www.nasa.gov/feature/nasa-unveils-sustainablecampaign-to-return-to-moon-on-to-mars
- [27] Revera, G. H.: Full Moon photograph. [Online; visited 4.03.2019]. Retrieved from: https://commons.wikimedia.org/wiki/File:FullMoon2010.jpg
- [28] Rohatgi, A.: Smooth Bézier Spline Through Prescribed Points. 2012. Retrieved from: https://www.particleincell.com/2012/bezier-splines/
- [29] Rohatgi, A.: WebPlotDigitizer. 2019. Retrieved from: https://apps.automeris.io/wpd/
- [30] Rohatgi, A.: WebPlotDigitizer User Manual. 2019. Retrieved from: https://automeris.io/WebPlotDigitizer/userManual.pdf
- [31] Scholten, F.; Oberst, J.; Matz, K.-D.; et al.: LROC WAC DTM GLD100. 2014. Retrieved from: https://astrogeology.usgs.gov/search/map/Moon/LRO/ LROC_WAC/Lunar_LROC_WAC_GLD100_79s79n_118m_v1_1
- [32] SolCommand: Apollo Lunar Module 3D Model. [Online; visited 4.03.2019]. Retrieved from: https: //www.sharecg.com/v/67738/view/5/3D-Model/Apollo-Lunar-module-and-ALSEP
- [33] Teague, K.: Project Apollo Drawings and Technical Diagrams. [Online; visited 4.03.2019].
 Retrieved from: https://history.nasa.gov/diagrams/apollo.html
- [34] Team Commands; Saclay, I.: BOCOP: an open source toolbox for optimal control. 2019.
 Retrieved from: http://bocop.org
- [35] Terrell, R.: Space 3D Skybox generator. [Online; visited 09.04.2019]. Retrieved from: http://wwwtyro.github.io/space-3d/
- [36] Walther, A.; Griewank, A.: ADOL-C. Retrieved from: https://projects.coin-or.org/ADOL-C
- [37] Williams, D. R.: Apollo 11 Lunar Module / EASEP. [Online; visited 4.03.2019]. Retrieved from: https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=1969-059C
- [38] Williams, D. R.: Earth Fact Sheets. [Online; visited 4.03.2019]. Retrieved from: https://nssdc.gsfc.nasa.gov/planetary/factsheet/earthfact.html
- [39] Williams, D. R.: Moon Fact Sheet. [Online; visited 4.03.2019]. Retrieved from: https://nssdc.gsfc.nasa.gov/planetary/factsheet/moonfact.html
- [40] Wächter, A.; Biegler, L. T.: Interior Point OPTimizer. Retrieved from: https://github.com/coin-or/Ipopt

 [41] Zupp, G. A.: An Analysis and a Historical Review of the Apollo Program Lunar Module Touchdown Dynamics. Technical report. NASA Johnson Space Center -Structural Engineering Division. 2013.
 Retrieved from: https://www.lpi.usra.edu/lunar/documents/SP-2013-605.pdf

Appendix A

Astronomical objects and spacecrafts in Apollo 11 mission

The Earth



Figure A.1: The Earth [21]

Table A.1: The Earth's physical characteristics [38]

Equatorial radius	$6378.137 \ km$
Mass	$5.9723 \times 10^{24} \ kg$
Surface gravity	$9.798 \ ms^{-2}$
Escape velocity	$11.186 \ kms^{-1}$
Gravitational	$3.986004418 \times 10^{14} \ m^3 s^{-2}$
parameter	

The Moon



Figure A.2: The Moon [27]

Table A.2: The Moon's physical characteristics [39]

Equatorial radius	$1738.1 \ km$
Mass	$0.07346 \times 10^{24} \ kg$
Surface gravity	$1.62 \ ms^{-2}$
Escape velocity	$2.38 \ kms^{-1}$
Gravitational	$4.902800066 \times 10^{12} \ m^3 s^{-2}$
parameter	

Saturn V rocket



Table A.3: Saturn V technical specifications [22]

Function	Apollo lunar program
	launcher
Stages	3
Height	110.6 m
Diameter	10.1 m
Mass	$2 \ 970 \ 000 \ kg$
Payload to translunar	$48\ 600\ kg$
injection	
1 st stage thrust	$35\ 100\ kN$ at sea level
2 nd stage thrust	5141 kN in vacuum
3 rd stage thrust	$1033.1 \ kN$ in vacuum

Figure A.3: The model of the Saturn V rocket[15]

Appendix B

Content of the included CD

- /xfiloj01-BP.pdf electronic version of the thesis
- /doc.zip compressed source files of the thesis
- /bocop/ directory containing the source files of the optimization problem
- /src/ directory containing the source files of the implemented application
- /bin/ directory containing the executable files of the implemented application
- /video.mp4 sample video of the implemented application