Analysis of Apollo Lunar Ascent Guidance

Parth S. Shrotri*
University of Washington, Seattle, WA, 98195

*MS Student, Department of Aeronautics and Astronautics

I. Introduction

Before understanding how the Apollo program approached the power ascent guidance problem, it is necessary to understand what motivates the requirements and constraints for this algorithm. The requirements and performance of the resulting algorithm are as a result of the mission architecture and the characteristics of the Lunar Module ascent stage's propulsion and attitude control system.

A. Overview of Apollo Mission Architecture

The primary objective of NASA's Apollo program was to land humans on the moon and return them safely to Earth. The mission architecture of the program, commonly referred to as 'Lunar Orbit Rendezvous' (LOR) [1], consisted of the use of two spacecraft: the Command and Service Module (CSM) and the Lunar Module (LM). The CSM, shown in Fig. 1, would serve as primary habitation for the entire mission and provide propulsion for lunar orbit insertion and departure. The Command Module was the only spacecraft capable of atmospheric entry. After lunar orbit insertion, the CSM would remain in low-lunar orbit (LLO). The LM, shown in Fig. 1, would descend and land on the surface. The LM consisted of the descent and ascent stage. The descent propulsion system would bring the entire LM to the lunar surface. At the end of surface operations, the ascent stage (Fig. 2) would separate from the descent stage, place itself in an orbit from which it would rendezvous and dock with the CSM.[2].

B. Ascent Stage Propulsion and Attitude Control System

The ascent stage consists of a single Ascent Propulsion System (APS) engine that produces a fixed amount of thrust (15568 N) and does not have a gimbal mechanism. The LM also has four sets of four (445N) Reaction Control System (RCS) thrusters, which can be used to augment the APS thrust, but are primarily used for attitude control [3]. The effect of this design on the developed control scheme is that the entire spacecraft must be rotated to align the thrust vector in the desired direction. Since there is an inherent lag that results from this large change in attitude, the algorithm should be robust to small errors in input caused by this.

C. Trajectory and Automatic Control Phases

The problem trajectory engineers and their software counterparts faced is how to design trajectories and implement autonomous control laws that obey the following requirements. The trajectory used must minimize the risk of collision with the terrain, minimize fuel use, maximize crew visibility in critical phases, and allow aborts when possible. The trajectory also must result in a terminal state from which a rendezvous with the CSM can be performed with the remaining propellant. Bennett [3] provides an analysis of the pre-mission planning of these trajectories, the implementation of the resulting guidance and control schemes, and their real-time performance on Apollo 11. It should also be noted that the entire powered ascent trajectory, from the surface to insertion into a specified rendezvous orbit, can only be automatically controlled by the Primary Guidance Navigation and Control System (PGNCS) or the Abort Guidance System (AGS) [3]. The only axis that can be manually controlled is rotation about the body X-axis (Fig. 3).

Bennett describes the lunar ascent program as being divided into two phases: vertical rise and orbital insertion. The vertical rise phase of the ascent program is used to move the ascent stage above obstacles at the start of the powered ascent, after which the fuel-optimal orbital insertion control scheme is used. This results in a trajectory such that the spacecraft is in the desired orbit at the end of the powered ascent. Nominally, this initial orbit that the LM is placed in is 16.7-by-83.4km (perilune altitude-by-apolune altitude) with the CSM being in a circular 111.1km altitude orbit. It is important to note that the desired orbit must be easily modified by the astronaut prior to the start of powered ascent. It should also be noted that the lack of an atmosphere on the Moon allows for simplifications in the guidance scheme[3].

The algorithm, referred to as the "Powered Ascent Program" or "P-12" in the LM software, solves these sets of problems and represents a vital part of the Apollo program. The analysis performed by Bennett helped verify that this first-of-its-kind ascent guidance algorithm performed as designed. Understanding its underlying theories and formulation is also key to understanding its shortcomings and motivations for the development of modern powered ascent guidance algorithms [4][5]. I describe the guidance scheme in Section II. I discuss some of the techniques and underlying theories in Section III. I also implemented the guidance law as described in [6] and a simplified version of the control laws in Python, simulated its performance using the Apollo 11 mass properties, launch location and target orbit parameters. I describe my implementation in Section IV. I present a comparison of the performance of my implementation with that of the Apollo 11 LM and its expected pre-mission simulations in Section V. This shows that the performance of the algorithm can be reproduced using a simplified model with reasonable accuracy.

II. Mathematical Formulation

While Bennett outlines the powered ascent guidance algorithm at a high level, the complete set of reference frames and equations that were implemented in P-12 is given in the Operations Plan submitted by MIT Instrumentation Lab, the original developers of the LM software[6]. Note that the guidance law is unique to lunar ascent because of the lack of atmosphere, meaning that orbital insertion can start much earlier in the ascent compared to ascent from a body with atmosphere. It is also unique to the constraints of the LM's APS. They begin by defining a LM comoving reference frame, called the Local Vertical Coordinate System (LVCS): $\{\boldsymbol{u_R}, \boldsymbol{u_Y}, \boldsymbol{u_Z}\}$ where

$$\begin{aligned} \boldsymbol{u_R} &= \tfrac{\boldsymbol{r}}{\|\boldsymbol{r}\|} \\ \boldsymbol{u_Z} &= \boldsymbol{u_R} \times \boldsymbol{Q} \\ \boldsymbol{u_Y} &= \boldsymbol{u_Z} \times \boldsymbol{u_R}, \end{aligned} \tag{1}$$

where $\boldsymbol{Q}$ defines the orbital plane of the CSM:

$$\boldsymbol{Q} = \frac{\boldsymbol{v_c} \times \boldsymbol{r_c}}{\|\boldsymbol{v_c} \times \boldsymbol{r_c}\|} \tag{2}$$

where $\boldsymbol{r}$ is the position of the LM. $\boldsymbol{r_c}$ and $\boldsymbol{v_c}$ represent the position and velocity (state vector) of the CSM respectively. These are all given in the Moon-Centered-Inertial (MCI) Frame. It should be noted that the state vector of the CSM does not need to be known at all points in time during the ascent. Rather, we only need to know the state vector at at one instance in time near the initiation of powered ascent. This is because we only utilize the state vector of the CSM to define its orbital plane. The LVCS system is illustrated in Fig. 4. In order to fully define the problem, we also need the desired injection radius R_D, out-of-plane distance Y_D, radial velocity $\dot{R}_D$, cross-range velocity $\dot{Y}_D$, and down-range velocity $\dot{Z}_D$ in the LVCS.

For the first two seconds of the ascent, the ascent stage will hold its initial attitude to clear the descent stage. Then, it will pitch such that it's X-axis aligns with the LVCS radial direction and roll such that it's Z-axis aligns with the LVCS downrange direction. The LM maintains this attitude until the radial velocity reaches 12.2m/s. After this threshold is met, the guidance program switches to a "fuel-optimal orbital insertion" mode. In this mode, the following steps are repeated every two seconds in an iterative fashion. Note that because some steps may require accessing information from the previous iteration, for clarity, I have introduced the subscript k to denote a value calculated in the current iteration and $k-1$ to denote a value calculated in the previous iteration.

First, we calculate the current state of the LM in the LVCS:

$$\begin{aligned} R &= \|r\| \\ Y &= R_D \arcsin\left(\boldsymbol{u_R} \cdot \boldsymbol{Q}\right) \\ \dot{R} &= \boldsymbol{v} \cdot \boldsymbol{u_R} \\ \dot{Y} &= \boldsymbol{v} \cdot \boldsymbol{u_Y} \\ \dot{Z} &= \boldsymbol{v} \cdot \boldsymbol{u_Z} \end{aligned} \tag{3}$$

where $\boldsymbol{v}$ is the velocity vector of the LM in the MCI frame. We then calculate the velocity error in the LVCS frame:

$$\boldsymbol{v}_{\text{err}} = \begin{bmatrix} \dot{R}_D - \dot{R} & \dot{Y}_D - \dot{Y} & \dot{Z}_D - \dot{Z} \end{bmatrix}^T \tag{4}$$

We then compute the velocity-to-be-gained: v_g. For the first iteration, use an assumed $t_{\text{go},\,k-1}$ of 370 seconds. This is then used to compute the time-to-go for the current iteration ($t_{\text{go},\,k}$) using a series expansion of a reformulation of the "ideal rocket equation" terminating at the second term.

$$v_g = \left\| \boldsymbol{v}_{\text{err}} - \frac{1}{2} t_{\text{go},\,k-1} \begin{bmatrix} g_{\text{eff}} & 0 & 0 \end{bmatrix}^T \right\| \tag{5}$$

$$t_{\text{go},\,k} = \tau \frac{v_g}{v_e} \left(1 - \frac{1}{2} \frac{v_g}{v_e}\right) + \Delta t_{\text{tail-off}} \tag{6}$$

where τ is the mass of the LM divided by the mass flow rate of propellants into the APS: $\frac{m}{\dot{m}}$, v_e is the exhaust velocity of the APS, $\Delta t_{\text{tail-off}}$ is a negative value to correct for the shut-down transient time of the APS, and g_{eff} is the effective force of gravity. We use fundamental equations of rocket dynamics to attain v_e and $\dot{m}$:

$$v_e = I_{\text{sp}} g_0 \tag{7}$$

$$\dot{m} = -\frac{T}{v_e} \tag{8}$$

and

$$g_{\text{eff}} = \frac{\|\boldsymbol{r} \times \boldsymbol{v}\|^2}{\|\boldsymbol{r}\|^2} - \left\|g_{\text{avg}}\right\| \tag{9}$$

where I_{sp} and T are the specific impulse and thrust of the APS, respectively, and g_0 is the surface gravity of Earth (9.81 $\frac{\text{m}}{\text{sec}^2}$) and g_{avg} is the magnitude of the average gravitational acceleration induced on the LM during a 2 second guidance cycle period. The next step is to calculate the guidance coefficients with the following equations:

$$L = \log\left(1 - \frac{t_{\text{go},\,k}}{\tau}\right) \tag{10}$$

$$D_{12} = \tau + \frac{t_{\text{go},\,k}}{L} \tag{11}$$

$$D_{21} = t_{\text{go},\,k} - D_{12} \tag{12}$$

$$E = \frac{t_{\text{go},\,k}}{2} - D_{21} \tag{13}$$

$$B_k = \begin{cases} \frac{D_{21}(\dot{R}_D - \dot{R}) - (R_D - R - \dot{R}t_{\text{go},\,k})}{t_{\text{go},\,k}E} & t_{\text{go},\,k} \geq 10 \text{ sec} \\ 0 & t_{\text{go},\,k} < 10 \text{ sec} \end{cases} \tag{14}$$

$$D_k = \begin{cases} \frac{D_{21}(\dot{Y}_D - \dot{Y}) - (Y_D - Y - \dot{Y}t_{\text{go},\,k})}{t_{\text{go},\,k}E} & t_{\text{go},\,k} \geq 10 \text{ sec} \\ 0 & t_{\text{go},\,k} < 10 \text{ sec} \end{cases} \tag{15}$$

If $B_k > 0$ we set $B_k = 0$ and if $B_k < -0.03048\tau$ we set $B_k = -0.03048\tau$

$$A_k = \begin{cases} -D_{12}B - \frac{\dot{R}_D - \dot{R}}{L} & t_{\text{go},\,k} \geq 2 \text{ sec} \\ A_{k-1} & t_{\text{go},\,k} < 2 \text{ sec} \end{cases} \tag{16}$$

$$C_k = \begin{cases} -D_{12}D - \frac{\dot{R}_D - \dot{R}}{L} & t_{\text{go},\,k} \geq 2 \text{ sec} \\ C_{k-1} & t_{\text{go},\,k} < 2 \text{ sec} \end{cases} \tag{17}$$

We can then calculate the acceleration commands in the radial and cross-range directions at time t:

$$a_{TR} = \frac{1}{\tau}(A_k + (t - t_k)B_k) - g_{\text{eff}} \tag{18}$$

$$a_{TY} = \frac{1}{\tau}(C_k + (t - t_k)D_k) \tag{19}$$

where t_k is the time at which the coefficients $A_k, B_k, C_k,$ and D_k were calculated. We initially set the downrange acceleration a_{TZ} to zero. The commanded acceleration is defined in vector form in the LVCS as:

$$\boldsymbol{a_H} = \begin{bmatrix} a_{TR} & a_{TY} & a_{TZ} \end{bmatrix} \tag{20}$$

We then need to check that $\|\boldsymbol{a_H}\|$ is less than or equal to the maximum acceleration possible by the LM: a_T, which can be found by dividing the APS thrust by the current mass of the LM. If $\|\boldsymbol{a_H}\| > a_T$, we set:

$$\boldsymbol{a_H} = a_T \frac{\boldsymbol{a_H}}{\|\boldsymbol{a_H}\|} \tag{21}$$

We can then set a_{TZ} such that any excess acceleration capability is used in the downrange direction:

$$a_{TZ} = a_T - \|\boldsymbol{a_H}\| \tag{22}$$

We then get the direction of the acceleration vector in the LVCS frame:

$$\boldsymbol{a_C} = \frac{\boldsymbol{a_H}}{\|\boldsymbol{a_H}\|} \tag{23}$$

An target attitude can be found by finding a rotation such that the atttiude command vector $\boldsymbol{a_C}$ perfectly aligns with the body X-Axis (see Fig. 3). A "window pointing vector" (desired Z-axis pointing direction) can also optionally be given in order to define a non-zero roll angle. This cycle repeats as long as the LM is in powered ascent. Once $t_{\text{go}} \leq 4$ sec, we start a timer for 4 seconds. After the timer reaches zero, we command the APS to cut-off, concluding powered ascent.

III. Mathematical Analysis

A. Modification of the Hill Frame

One of the underlying techniques used in the guidance algorithm is the use of an adaptation of the Hill frame. In the Hill frame, the second axis points in the direction of a spacecraft's velocity vector (downrange). The LVCS third axis points in a direction which intersects the orbital plane of the CSM (desired downrange). Essentially, the LVCS is a form of the Hill frame that is rotated about the radial axis to account for the cross-range error. As the cross range error, Y, approaches zero these axes become parallel.

B. Formulation

It can be seen in the formulation that there is no explicit control of any of the downrange components of the state. This is because the LM's APS cannot be throttled. The effect of this is that explicit control of the downrange components of the states is not possible. The effect of this on the guidance algorithm is that acceleration demand in the radial and cross-range directions is determined first to minimize error in those directions and if any excess acceleration capacity exist, is used in the downrange direction. This means that acceleration commands in the radial and cross-range directions are effectively prioritized over the downrange component. In order to reach the desired downrange velocity ($\dot{Z}$) we must end powered ascent at the correct time.

C. Notes on Fuel-Optimality

The paper by Bennett claims that the orbital insertion phase is fuel-optimal, however a explanation of this is outside of his paper's scope. I was not able to find a paper that explains how this problem was posed as an optimization problem or how this resulted in the equations that are given in [6]. Regardless, if we assume that the powered ascent guidance algorithm is fuel optimal as stated, it is important to note that this results in trajectory that that is only fuel-optimal to reach the desired orbit. The problem of choosing the appropriate powered ascent initiation time (t_{ig}), $R_D, Y_D, \dot{R}_D, \dot{Y}_D$, and $\dot{Z}_D$ is separate from the powered ascent guidance problem but has a large impact on the overall optimality of the launch to docking trajectory. That is, it is possible to choose a terminal state for powered ascent guidance that does not minimize the total amount of fuel used for the launch to rendezvous trajectory or in the worst case, results in an fuel state with which a rendezvous cannot be performed. Thus, while the choice of t_{ig} and terminal state is outside of the scope of the powered ascent guidance problem, it is a vital step in the process of ensuring orbital rendezvous can be performed. This is the subject of the phasing optimization problem. Bennett does provide some insight in this, stating that out-of-plane maneuvering is limited to 0.5° with any additional out-of-plane maneuvering being combined with subsequent rendezvous maneuvers.

IV. Implementation, Simulation, and Verification

In order to verify the performance of the algorithm, I implemented a modified version in Python. I simulate the entire lunar powered ascent profile of the LM ascent stage in 6 degrees-of-freedom from lift-off to the coasting phase. The code for my implementation and simulation can be found at [7].

A. Implementation

There are some key differences between my implementation and that of the original Apollo flight software as well as differences between my simulation dynamics and that of the real spacecraft which I will highlight. In the vertical rise phase, the only difference in implementation is that the LM's guidance switches to orbital insertion logic when the radial rate exceeds 12.2m/s but does not begin the pitch-over until the radial rate exceeds 15.2m/s due to the Digital Autopilot steering command lags [3]. To reflect this in my simulation, I use 15.2m/s as the threshold to determine the switch from vertical rise to orbital insertion.

There are a few modifications of note in the orbital insertion phase. Where the LM's flight software calculates control inputs every two seconds, I have configured my algorithm to calculate control inputs every 0.1 seconds to more accurately reflect modern flight computers. However, I have chosen to retain the coefficient calculation frequency of once every two seconds. Also, since my guidance coefficient recalculation frequency is relatively high, I do not use the average gravitational acceleration over a guidance cycle period. Instead, I used the instantaneous gravitational acceleration at the computation time. I have also not implemented or accounted for the effect of $\Delta t_{\text{tail-off}}$ in the simulation

dynamics or in my implementation of the guidance law. The most significant difference between the real dynamics and my simulated dynamics is that I do not simulate the RCS. I chose not to include the RCS in my simulations as it expands the scope of the implementation well beyond the guidance law. Instead, I replace the RCS with a Proportional-Derivative controller to determine the amount of torque that is applied on the simulated LM. I have modeled fuel mass depletion but chose not to model the resulting change in moment of inertia due to complexity. If the fuel mass reaches zero, the simulated APS will stop producing thrust.

B. Simulation Configuration

I simulate the trajectory of the LM for 700 seconds sampled at an interval of 0.1 seconds. There are a number of parameters that include the launch site location, CSM's orbit, the LM's mass and propulsion properties, the target state, and the attitude control gains. Where possible, I have matched these parameters with those of Apollo 11. The launch site location is 0.67409°N, 23.47298°E [8]. The Keplerian elements that define the CSM's orbit can be found in Table 1. The mass properties of the LM can be found in Table 2. The LM's APS propulsion properties are found in Table 3 The target state that I used can be found in Table 4. The attitude control gains that I used can be found in Table 5. The powered ascent begins 5 seconds after the simulation start time. All other transitions between powered ascent guidance modes are made automatically based on the state thresholds outlined in Section II. Once powered ascent guidance is complete, the LM will coast until the end of the simulation in the LVLH attitude.

C. Verification

After running the simulation, we can examine the trajectory and error in state over time to gain a better understanding of the powered ascent guidance algorithm and its performance. A plot of the entire altitude vs. downrange distance is shown in Fig. 5. In this plot, triangles (to represent the LM) are placed at 25 second intervals and oriented to demonstrate the direction of acceleration. A plot with more detail around lift-off in Fig. 6. In this figure, the triangles representing the LM are placed at 1 second intervals. In the vertical rise phase, we see that the performance between pre-mission simulations, the real flight, and my simplified simulation matched almost exactly. This is demonstrated by the guidance in all three switching from vertical rise to orbital insertion approximately 10.0 seconds after lift-off, as shown in Fig. 6.

In the orbital insertion phase, we will first look at how the acceleration demand is distributed among the radial, cross-range, and down-range axes. A time series plot is shown in Fig. 7. Here we see that from the point at which the orbit insertion logic begins until cut-off, there is always excess acceleration capacity that is used to begin accelerating the LM in the downrange direction. This shows that even though there is no explicit control in the downrange direction, for a well-planned trajectory, acceleration in the downrange direction is still achieved so that the cut-off condition can be reached.

We can also examine the state error over time, shown in Figs. 8 and 9, which demonstrates that the guidance algorithm performs such that the error in all states is approximately zero at the same time. I have also compared the residual error in states at cut-off in the pre-mission simulation, the flight, and the simplified simulation. These results can be found in Table 6. We can see that the simplified simulation and modified guidance program that I developed produces comparable results with the pre-mission simulations. The lower residuals are likely due to the fact that the attitude control system and thrust termination tail-off time are not modeled. Since RCS thrusters produce a fixed amount of torque, the real attitude control system is inherently more coarse than the one I implement. The difference in residuals between the simulations and the real mission is likely due to some variance in the Apollo 11 APS performance.

One of the results of Bennett's analysis was on the amount of propellant used during the powered ascent. The expected usage of propellant in pre-mission simulations and planning for Apollo 11's lunar ascent was approximately 2253kg [3]. The real mission used approximately 2200kg. The usage predicted by my simulations is 2224kg, marginally lower than the pre-mission simulation and probably attributed to the lack of RCS in my simulation. A plot of fuel usage over the duration of the burn is given in Fig. 10.

The final comparison I will make between the pre-mission simulations, the real flight, and my simplified simulation is that of the burn time. That is, the duration of the powered ascent. The pre-mission simulations for Apollo 11 predicted that the powered ascent burn would take 7 minutes and 18 seconds. The duration of the burn during the mission was 7 minutes and 15 seconds, the difference of 3 seconds being attributed to a higher thrust-to-weight ratio than expected [3]. The powered ascent duration demonstrated in my simulation matches that of actual flight's at 7 minutes and 15 seconds. The difference between my simulation and the pre-mission simulation's burn time may be attributed to the fact that my implementation is modified to compute control inputs more frequently, leading to lower error accumulation.

V. Assessment

I find that my simplified analysis of the Apollo 11 powered ascent guidance largely produces results similar to Bennett's analysis. Bennett's analysis using real flight data, flight planning information, and guidance software does not seem to make assumptions. The powered ascent guidance algorithm documentation does not state what assumptions were made in it's development, so it is not possible to comment on the validity or accuracy of those assumptions or indeed the fuel-optimality of the algorithm. However, we can examine the simplifications that were made. One of the simplifications made is that the time-to-go series expansion is truncated at the second term. As this results in very low residuals in the terminal state, it is safe to state that this assumption is reasonable. Another simplification is the use of effective gravity, which is a singular term that combines the effect of orbital angular momentum and gravity. When this is used to calculate the "gravity loss" term in the velocity-to-be-gained equation, another simplification is made in that the guidance algorithm uses average gravity over a guidance update period and applies that acceleration over the entire time-to-go. This is not strictly physically accurate because of the nature of gravitational fields. However, since this is applied in low orbits over a short period of time it seems that this is a reasonable assumption. Furthermore, the guidance coefficients are updated as the spacecraft converges to its terminal state, the errors caused by this simplification will have minimal impact. The use of the average gravitational force was necessary as it enabled the algorithm to run accurately on PGNCS with its 2 second guidance update period. However, I was able to demonstrate that by reducing the computation cycle time as would be possible on modern computers, it is possible to replace this with instantaneous gravity.

Work on this algorithm has been done extensively as part of the development for several orbital vehicles. However, finding the exact formulation of the original atmosphere-free ascent guidance problem would be useful in proving the fuel-optimality argument. It would also be worth determining how/if it would be possible to adapt this algorithm to powered ascent from a body with an atmosphere, what modifications would be necessary, and how it would compare to modern ascent guidance techniques. As there are a few steps in the guidance computation that require values to be limited within a certain range after computation and there are no guarantees that a terminal state is one that is reachable, it would also be useful to research convex optimization approaches that would result in control laws that automatically satisfy the maneuvering or propulsion constraints of a spacecraft.

References

[1] Houbolt, J. C., "Lunar Rendezvous," *International Science and Technology*, February 1963.

[2] Team, M. E., and Low, G. M., "Apollo 11 Mission Report," Tech. rep., NASA, MSC-00171, November 1969.

[3] Bennet, F. V., "Apollo Lunar Descent and Ascent Trajectories," Tech. rep., NASA, TM X-58040, March 1970.

[4] McHenry, R. L., Brand, T. J., Long, A. D., Cockrell, B. F., and Thibodeau, J. R., "Shuttle Ascent Guidance, Navigation, and Control," *The Journal of the Astronautical Sciences*, January-March 1979.

[5] Dukeman, G. A., "Atmospheric Ascent Guidance for Rocket-Powered Launch Vehicles," *AIAA Guidance, Navigation, and Control Conference*, August 2002.

[6] Levine, G. M., "Guidance System Operations Plan for Manned LM Earth Orbital and Lunar Missions Using Program Luminary IE," Tech. rep., MIT Instrumentation Laboratory, December 1971.

[7] Shrotri, P. S., "Implementation and Simulation of Luminary Powered Ascent Guidance," Tech. rep., University of Washington, March 2025. `https://github.com/parthshrotri/Luminary_P12`.

[8] Jones, E. M., and Glover, K., "Apollo 11 Lunar Surface Journal," Tech. rep., NASA, November 2017. `https://www.nasa.gov/history/alsj/a11/a11ov.html`.

[9] Teague, K., and Gamble, C., "Project Apollo Technical Diagrams," Tech. rep., NASA, December 2023.

[10] Williams, D. R., "NASA Space Science Coordinate Archive," Tech. rep., NASA, October 2022. `https://nssdc.gsfc.nasa.gov/nmc/spacecraft/displayTrajectory.action?id=1969-059A`.

Appendix

A. Images

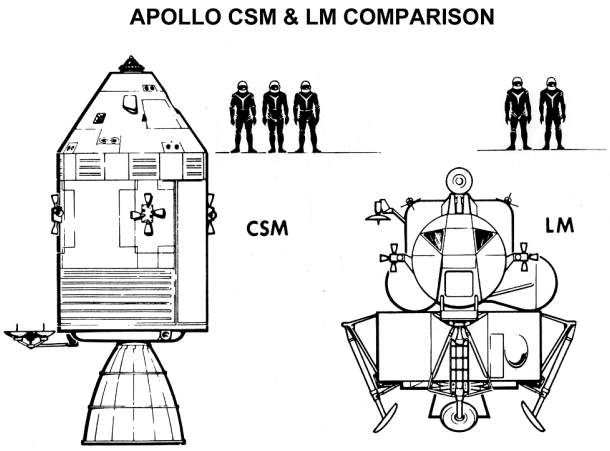

Fig. 1 Command/Service and Lunar Modules [9]

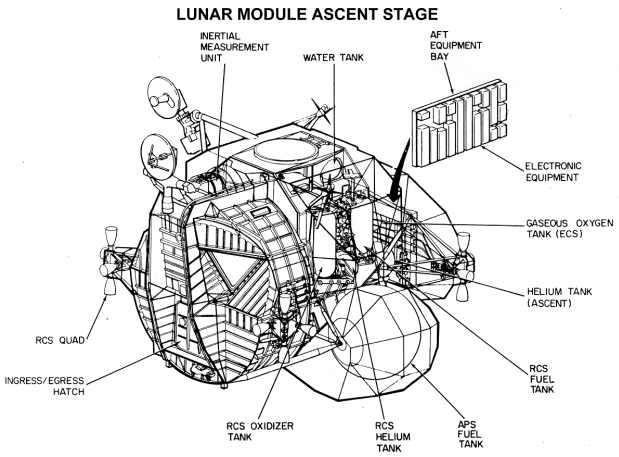

Fig. 2 Lunar Module Ascent Stage [9]

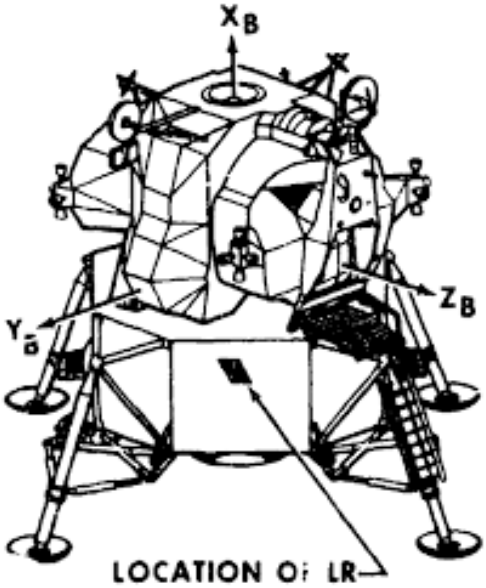

Fig. 3 Lunar Module with Body-Coordinate System Illustrated [3]

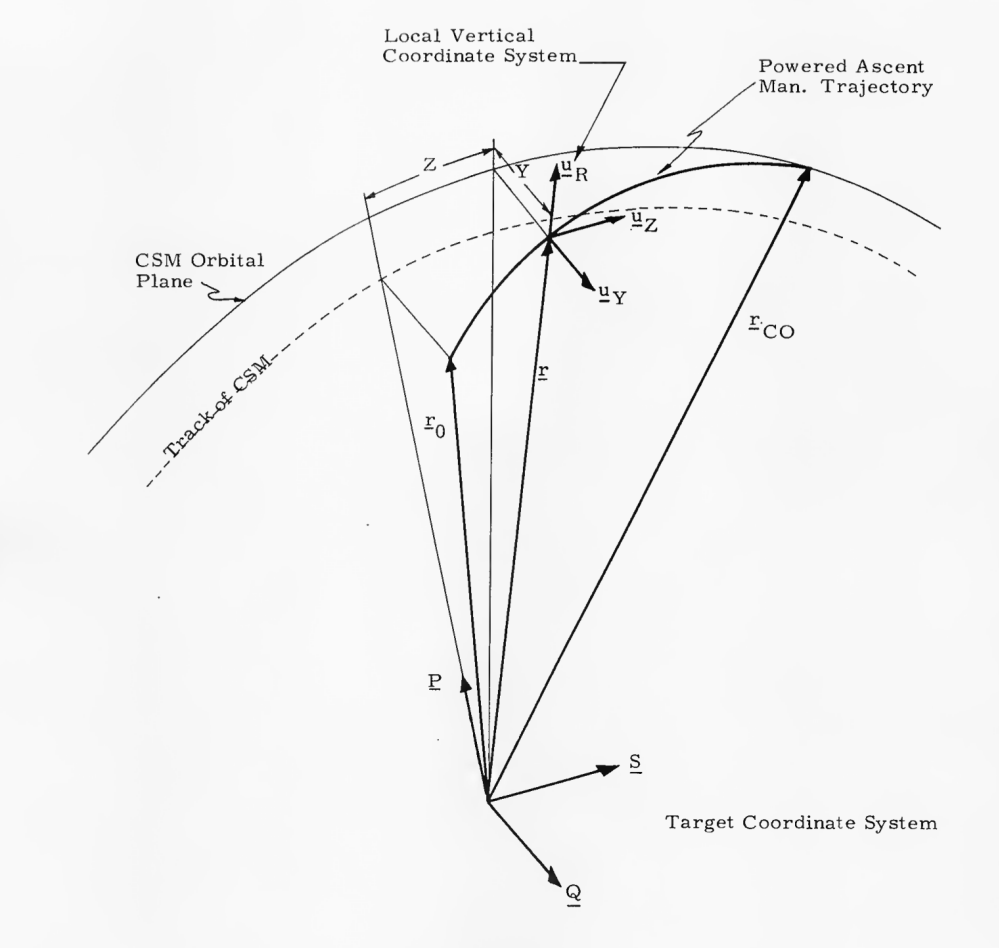

Fig. 4 Local Vertical Coordinate System [6]

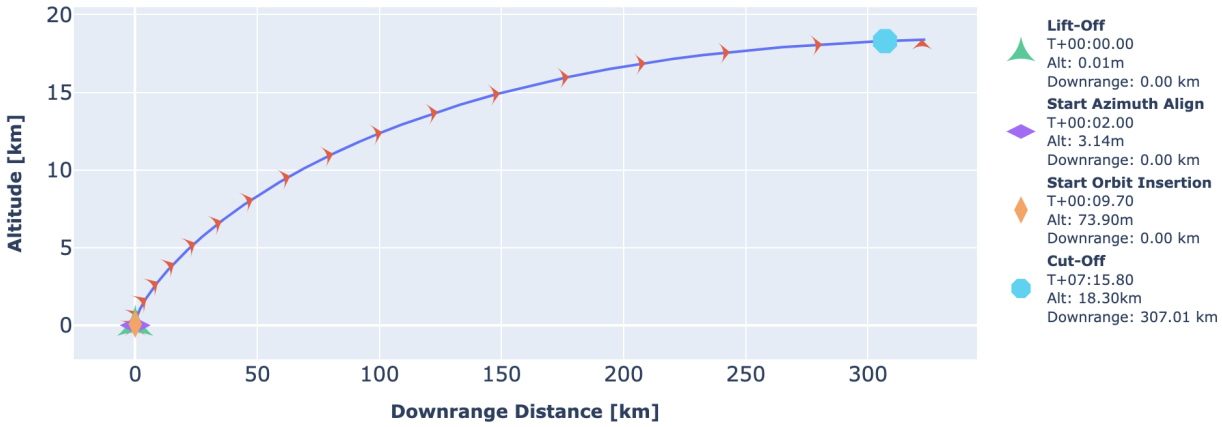

Fig. 5 Entire Ascent Trajectory

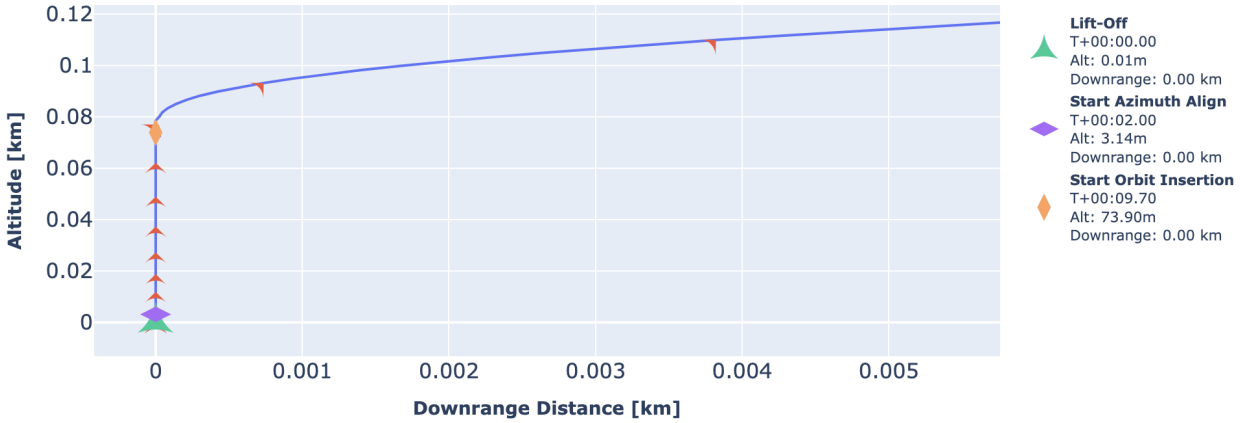

Fig. 6 Trajectory near Lift-Off

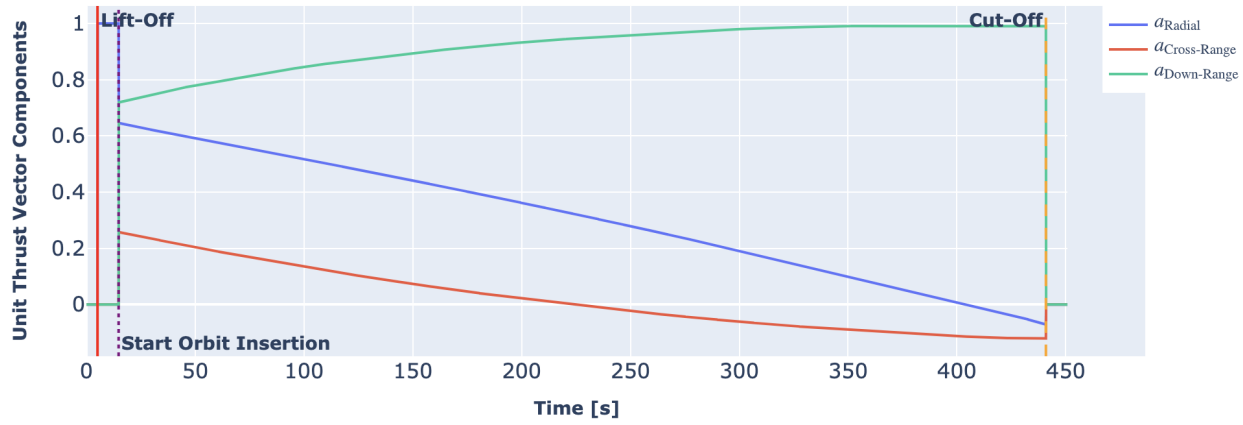

Fig. 7 Acceleration Command Distribution In Each Axis

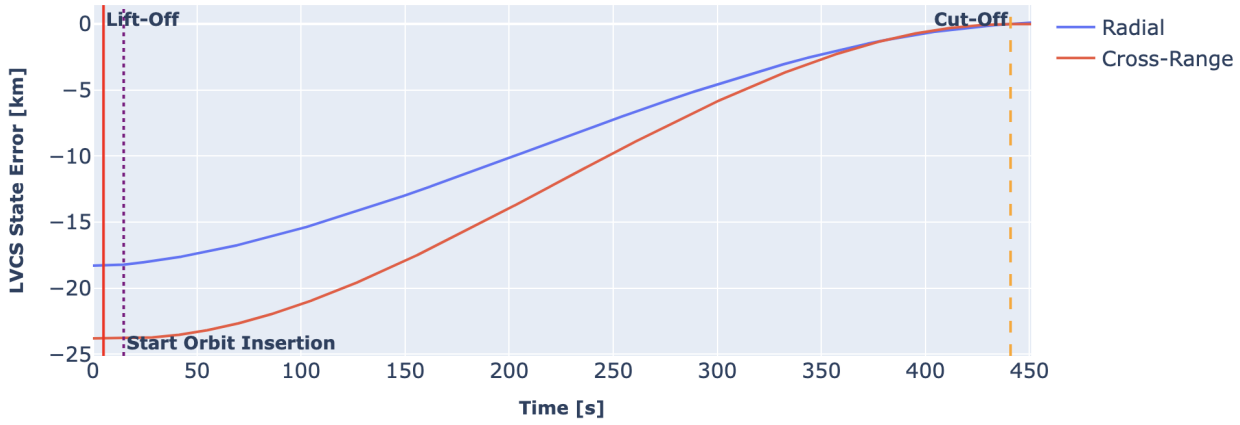

Fig. 8 State Error

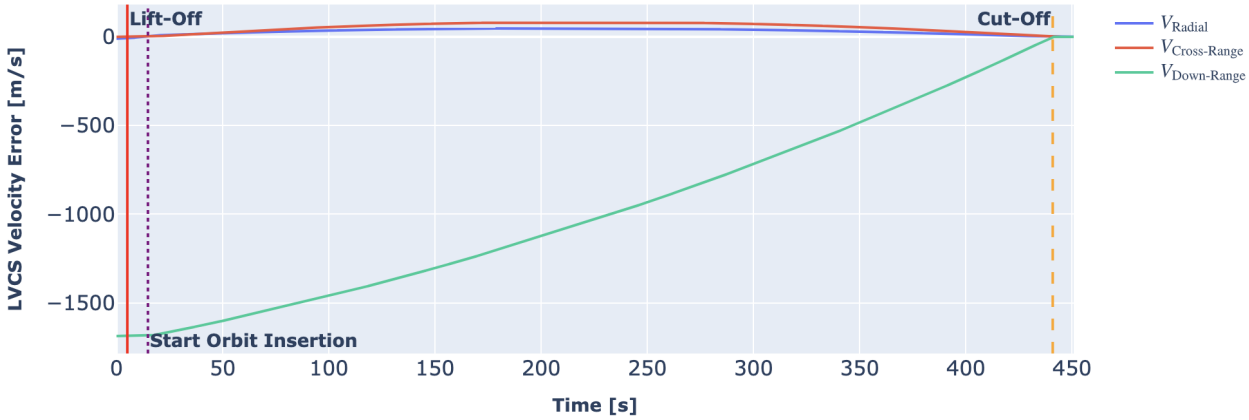

Fig. 9 Velocity Error

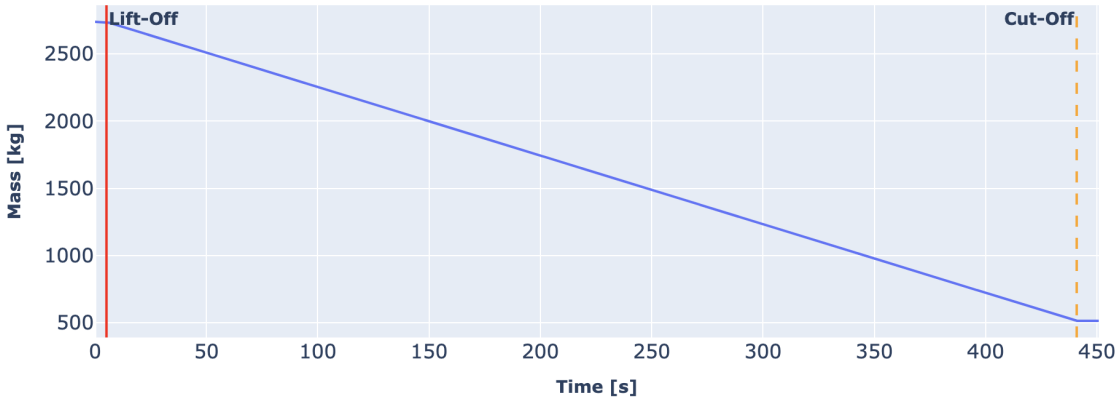

Fig. 10 Fuel Usage During Powered Ascent

B. Tables

Semimajor Axis (km)	1848.6
Eccentricity	0.00568
Inclination (°)	0.00
Longitude of Ascending Node (°)	0.00 [assumed]
Argument of Periapse (°)	0.00 [assumed]
True Anomaly (°)	0.00 [assumed]

Table 1 Command and Service Module Orbital Parameters [10]

Initial Mass (kg)	4888
Dry Mass (kg)	2150
Initial Moment of Inertia (I_{xx}, kg/m^2)	9230
Initial Moment of Inertia (I_{yy}, kg/m^2)	4711
Initial Moment of Inertia (I_{zz}, kg/m^2)	8096

Table 2 Lunar Module Mass Properties [2]

Thrust (N)	15570
Specific Impulse (sec)	311

Table 3 Lunar Module Ascent Propulsion System Properties [3]

Radial Distance (m)	1755388
Out-of-Plane Distance (m)	3148.4
Radial Rate (m/s)	9.81456
Cross-Range Rate (m/s)	0
Down-Range Rate (m/s)	1687.03752

Table 4 Lunar Module Target State [3]

Proportional (K_P)	90000
Derivative (K_D)	20000

Table 5 Attitude Control Gains

State Component	Pre-Mission Simulation	Real	Simplified Simulation
Radial(m)	25.8	91.4	11.3
Radial Rate (m/s)	0	0	0
Down-Range Rate (m/s)	0.7	0.4	0.1

Table 6 State Residual Comparison