Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 2886312, 14 pages
https://doi.org/10.1155/2022/2886312

Research Article

Soft Landing Parameter Measurements for Candidate Navigation Trajectories Using Deep Learning and AI-Enabled Planetary Descent

Janhavi H. Borse,[1] **Dipti D. Patil**,[2] **Vinod Kumar**,[3] **and Sudhir Kumar**[4]

[1]*Department of Computer Engineering, SKNCOE, Savitribai Phule Pune University, Pune, India*
[2]*Department of Information Technology, MKSSS's Cummins College of Engineering for Women, Savitribai Phule Pune University, Pune, India*
[3]*U. R. Rao Satellite Centre, Indian Space Research Organization, Bangalore, India*
[4]*PCPS College, Lalitpur, Nepal*

Correspondence should be addressed to Sudhir Kumar; sudhir.kumar@patancollege.org

Received 16 May 2022; Accepted 28 July 2022; Published 27 August 2022

Academic Editor: Amandeep Kaur

Smart instruments, sensors, and AI technologies are playing an important role in many fields such as medical science, Earth science, astronomy physics, and space study. This article attempts to study the role of sensors, instruments, and AI (artificial intelligence) based smart technologies in lunar missions during navigation of trajectories. Lunar landing missions usually divide the power descent phase into three to four sub-phases. Each sub-phase has its own set of initial and final constraints for the desired system state. The landing systems depend on human competencies for making the most crucial landing decisions. Trajectory planning and designing are very significant in lunar missions, and it requires inputs with precision. The manual systems may be prone to errors. In contrast, AI and smart sensor-based measurements give an accurate idea about the trajectory paths and make appropriate decisions where manual systems may turn into disasters. The manual systems are either pre-fed or have manual controls to guide the trajectory. For autonomous landing problems, trajectory design is a very crucial task. The automated trajectories play a vital role in the measurement and prediction of landing state parameters of the space rocket. Nowadays, sensors, intelligent instruments, and the latest technologies go hand in hand to devise measurement methods for accurate calculations and make appropriate decisions during landing space rockets at the designated destination. Space missions are very expensive and require huge efforts to design smart systems for navigation trajectories. This paper attempts to design all possible candidates of reference navigation trajectories for autonomous lunar descent by employing 3D non-linear system dynamics with randomly chosen initial state conditions. The generated candidates do not rely on multiple hops and thus exhibit an ability to serve autonomous missions. This research work makes use of smart sensors and AI federated techniques for smartly training the system to serve the ultimate purpose. The trajectories are simulated in an automated simulating environment to perform exhaustive analyses. The results accurately approximate the trajectories analogous to their numerical counterparts and converge to their measured final state estimates. The generation rate of feasible trajectories measures the accuracy of the algorithm. The algorithm's accuracy is near 0.87 for 100 sec flight time, which is reasonable.

1. Introduction

1.1. Background Details. Nowadays, the bulk of data is available through various space missions. Technological advancements are also taking place in domains like artificial intelligence and machine learning. These new technologies can be leveraged to find autonomous solutions to the safe planetary landing problem. AI-ML systems may use this mission data for performing numerous tasks specific to Guidance, Navigation, and control of space vehicles. Out of the recent technological advancements [1, 2], terrain relative navigation of NASA [3], and CE'4 [4] landing mission of

China showed flawless descent using AI-ML techniques and previous mission DEM data. It proves the capability of ML models to guide a spacecraft for navigation through space without the help of a GPS-like facility.

Along with the DEM data, analytical trajectory data are another powerful resource for the efficient training of such systems. Plenty of robust and qualitative trajectory data for autonomous systems must be available. Current lunar descent systems follow multi-hop pre-fed trajectory paths. But for self-driven systems, such data are not suitable. To solve this problem, we present an approach that takes care of terminal zero velocity constraints before touchdown and generates a single-hop trajectory from any randomized starting state. This attempt generates analytical reference trajectory data valid for the lunar power descent module. A complete 3-dimensional non-linear system dynamics and randomized initial system state parameters are used. Zero touchdown velocity is guaranteed by posing terminal velocity constraints on the guidance law.

A handful of literature is studied to understand trajectory estimation techniques, and experiments are done based on the knowledge gained. Research [4] shows that machine learning techniques are used to solve non-linear system problems. Few rely on this vast data and are called data-driven designs out of these systems. These typical supervised systems try to find a correlation between input and output. Such data-driven system designs need input and past trajectory information for efficient system training.

Few literature pieces also use non-linear system dynamics [5, 6] with few design variations. Researchers in [7] have solved dynamic system problems using physics laws through embedded programming solutions. Few researchers used a more straightforward 2D [8] system dynamics. At the same time, few dealt with the complete 3D system dynamics to solve non-linear state estimation problems. Few research studies concentrated on the optimization part of the non-linear dynamics. Ramanan and Lal [9] analyzed different trajectory optimization techniques and provided a clear overview. The hybrid optimization concept in trajectory analysis is introduced by [10]. Zhang and Barczyk focused on collision-free trajectory design using Non-linear Model Predictive Control (NMPC) for drone dynamics. Zhang, Xiaoxue et al. used a variant of Gaussian Mixture Models for trajectory generation to predict the uncertainty in moving obstacles. Barczyk and Martin employed a closed feedback loop for real-time trajectory prediction in the non-linear MPC system. Many valuable works [11, 12] describe guidance laws for lunar landing missions. The work in [13, 14] introduced guidance law for planetary missions, which tends to find the optimal path from the current position by solving a 2-point boundary problem. Research in [15] defines a crew-initiated explicit lunar descent guidance based on ground-based implicit law. The work in [16, 17] describes the guidance methods for hazard-free landing and analyses implicit video navigation requirements for autonomous missions. Reference [18] describes the hybrid propulsion design of the lunar lander and analyses its effect on the descent system as a whole. Few pieces of research converted optimal control problem to minimum fuel problem and accordingly posed constraints on guidance law. Research [19] demonstrates a trajectory generation algorithm for crewless missions with the help of simulation studies. Few researchers solved quadcopter dynamics in 3-dimensional state space for hazard avoidance in a cluttered environment, while few used non-linear dynamics for multi-robot trajectory generation and planning. 3-dimensional trajectory planning for terrain hazard avoidance and optimal control of a spacecraft is demonstrated in work [20].

1.2. Related Works. Survey [21] in the literature provides complete information about the working of AI-ML algorithms and their dependencies right from the data inception to the model existence. This work is a comprehensive survey of autonomous guidance methods analyzing planetary power descent constraints. Few works describe existing vision-based deep learning navigation techniques for orbital and landing missions. The analysis in [22, 23] discusses state-of-the-art feature extraction methods for deep learning applications. Depending on data availability, AI-based autonomous landing systems use deep learning approaches or deep reinforcement learning [24]. AI-ML approaches may use control software like General Pseudo-spectral OPtimal control Software (GPOPS) for generating trajectories for spacecraft guidance and control applications. A deep learning approach like [25] uses trajectory data for carrying out further investigation. The approach [26] discusses feasibility criteria for generating optimal fuel trajectories using deep learning techniques. Research [27] introduces dual-constrained guidance for hypersonic vehicles using a deep learning network. A deep reinforcement learning approach like [28] also depends on trajectory data. MATLAB tools like DIDO [29] and extended pseudo-spectral tools like GPOPS [30] are registered for commercial availability. These are not freely available to all. Certain open-source tools like Pyomo [31] are used to design research-specific frameworks for solving optimal control problems. A few of these tools and their applicability to the current problem are discussed below.

1.2.1. GPOPS. GPOPS is initially designed to solve non-linear, multi-phase optimal control problems using pseudo-spectral methods. Later on, GPOPS II used a Gaussian variant of pseudo-spectral methods. Basically, both the versions are written in MATLAB and possess leveled structure to allow user-defined specifications to be fed for each phase. It uses non-linear problem-solvers like SNOPT or IOPT to solve differential equations. The collocation method allows a user-defined number of nodes to incorporate multiple phases into the system, and the cost function can also be defined at the user level. Many researchers use this tool to generate trajectory data by manually providing initial guesses about the system's initial state.

1.2.2. DIDO. DIDO has been a part of NASA's first actual flight demonstration for solving optimal control problems since 2006. It provides a minimalist approach that uses

Pontryagin's principle as a baseline analytical tool. An early version of DIDO works on initial guesses. Still, later versions do not have that constraint as they contain multifaceted algorithms. These algorithms focus on spectral acceleration and provide feasible, robust, and optimal solutions. It is designed to meet the challenges of global space maneuvers. DIDO has fewer memory requirements and can be directly embedded into the mission hardware. It is not suitable for generating a complete dataset for training purposes as we need to train AI-ML systems.

1.2.3. Pyomo. Pyomo is a modeling language used by an open-source framework for dynamic optimization using pseudo-spectral methods. Although it is an open-source framework, it works on the same principles as GPOPS.

Almost all of these tools are not open-source, and their focus is on solving multi-phase control problems. The initial guess requirement is mandatory for generating a single optimal trajectory in many available tools. It introduces a subjective bias in each generated solution, although it produces one of the best optimal solutions. Human efforts are also required to generate many trajectories, which imposes hard constraints on the system. The data generated through this method misses the realistic random behavior of the trajectories. The ML models so trained may also involve subjective bias.

The existing approaches introduce subjectiveness in the generated trajectories, which is not desired for real-time applications. For autonomous missions, huge amounts of data must be produced with lower costs. These solutions are costly enough for researchers to generate such a large quantity of data. It motivates us to propose an autonomous trajectory generation solution by removing subjective bias with a zero-cost programmatic approach. The proposed algorithm works with randomized inputs adjusted automatically to address this drawback. The algorithm removes any subjective bias by incorporating all possible solutions that might not be optimal but are feasible in their respective ranges. When AI-ML models are trained on possible solutions, they perform well for favorable pre-known and unfavorable unknown inputs.

1.3. Research Contributions. The literature covered so far talks about how non-linear system dynamics are taken care of and how exactly physics-informed systems work. Existing physical systems use mathematical formulations and provide analytical solutions to planetary landing problems. With the advent of AI-ML technologies, new approaches to solving non-linear system dynamics for accurate planetary landing have come up. Intelligent techniques need more and more data to work. For instance, for supervised landing problems, accurate mission data with system state information must be available to make accurate predictions in real-time. This trajectory data has every piece of information about the physical system's past, current, and future state. It is not just landing problems for which an AI-ML solution would need trajectory data. All kinds of orbital and interplanetary missions also need such data. This data is not directly available to the researchers who work on such problems. Although few software solutions are available, they are licensed versions and hence not affordable to all. These solutions are detailed in Section 1.1 under the label of the related works.

The trajectory data requirement for implementing different AI-based systems is different. For example, missions like interplanetary, planetary landing, and orbital maneuvering follow different trajectories with varying constraints to reach their destinations. In essence, each problem carries its own challenges. Hence, the change in initial and final state parameters for each problem domain drifts the solution in different directions. Hence, an effort is made to limit our scope to the lunar landing problem, which would help AI-ML algorithms to be trained to make accurate state predictions. Before proceeding further, let us examine the challenges of the existing AI-ML solutions to the lunar landing problem as follows:

(i) Supervised learning techniques like deep neural networks need a foundation of accurate data to build an accurate model for state prediction.

(ii) Need for trajectory data with accurate past, current, and future state information comparable with natural physical systems.

(iii) Trajectories should be realistic. It means that, in case of any adverse event, a spacecraft following those trajectories should land softly on the ground.

(iv) Unexpected trajectory behavior should not end up in infeasible solutions.

(v) Unavailability of system state tracking system like GPS (Global Positioning System).

This paper attempts to meet those challenges through a programmatic approach. It generates accurate and feasible trajectory database solutions, which can be used to train AI-ML algorithms and help build predictive models for real-time applications. This database can also render the image data for providing real-time context information for the vision-based solutions. This is useful for creating a valid and labeled dataset, which further can be employed to train a neural network. One of the significant contributions of this work is that it tries to remove the subjective bias in training the AI-ML systems. All the trajectories are randomly generated and do not require any initial guesses. The focus is not on generating the optimal trajectories because such data trains the system only for favorable inputs. The data generated has all possible candidate trajectories, and at the same time, they are all feasible within acceptable ranges. When an AI-ML system comes across such a mixing of all possibilities, it makes the models more robust to unexpected environments. To be specific, the major contributions of this research work are as follows:

(i) The data generation removes the subjective bias and is hence more suitable for AI-ML tasks.

(ii) The generated trajectories are helpful for state prediction tasks.

(iii) The proposed takes into account 3-dimensional system dynamics as in real-time systems for generating data.

(iv) Existing work considers the multiple phases of the dynamic physical system. It hence needs initial guesses of landing parameters for every phase. The proposed design removes this multi-phase barrier, and thus possible system initialization delays are reduced through randomization.

(v) The proposed design considers only the final touchdown velocity constraint, eliminating all other constraints that are present in existing guidance laws, which guarantee soft landing trajectory computation.

1.4. Paper Organization. The entire paper is organized as follows. Section 1.1 introduces the topic with an overview of the background. Section 1.2 discusses prior AI-related works that need trajectories as the basis for training AI algorithms. Section 1.3 discusses the research contribution of this work and its benefits over existing techniques. Section 2.1 details the prerequisites needed to understand the mathematical formulation of the system better. A complete mathematical formulation and design of the proposed algorithm for autonomous trajectory generation is described in Section 2.2. Section 3 discusses results and simulations concerning the lunar power descent. Section 4 concludes the paper by describing the most significant achievements of this work.

2. Dynamic System Identification

Before formulating the system, it is necessary to understand a few concepts related to the 3-dimensional dynamics of any moving object. Every moving object possesses 3D motion dynamics about some stationary reference frame. This frame is often called the inertial frame of reference. In this case, the object is a Lunar Lander, and it is assumed to have the Moon as a stationary frame of reference. For each position, velocity parameters are calculated for this reference frame. Apart from the 3D frame of reference, once a lander starts descending, the objective is to track its decreasing altitude on the X-Y plane as it approaches the Moon's surface. This frame seems to have only 2 dimensions and is called the local vertical and local horizon frame of reference. Both these reference frames are explained in the following subsections. The following Section 2.1 is intended to make the reader understand the background and feel more comfortable reading upcoming sections.

2.1. Prerequisites

2.1.1. Inertial Frame of Reference. Newton's First Law of motion governs an inertial frame of reference. It states: "An object at rest remains at rest, and an object that is moving will continue to move straight and with constant velocity, if and only if there is no net external force acting on that object."

It means if $F_{\text{net}} = 0$ implies $v = 0$ or $\mathrm{d}v/\mathrm{d}t = 0$.

In particular, a spacecraft moving around a celestial body is assumed to move with constant velocity if no external forces are acting upon it. In the case of the Moon, there is no atmosphere, and hence there is no chance of external forces. The Moon's gravitational pull is the only force that keeps spacecraft tied to the Moon. Here the Moon is assumed to be stationary, and the spacecraft is in the inertial reference frame of the Moon.

For instance, in Figure 1, inertial reference frame M acts as a stationary observer for the spacecraft. The spacecraft with reference frame S moves at constant velocity unless other external forces acted. The spacecraft frame S is assumed to follow 3-dimensional motion dynamics and three axes (x', y', and z') of its inertial reference frame. In contrast, frame M's axes (x, y, and z) is assumed to be stationary. It is the 3D dynamics of the spacecraft and needs attention while computing trajectories.

2.1.2. Local Horizontal Local Vertical (LVLH) Frame. When a space vehicle starts descending toward the Moon, it is necessary to track its altitude and downward velocity components. This helps the vehicle control the final touchdown velocity and altitude. The assumption is to keep the X-Y plane as a constant Lunar surface to land on. The Z values keep on changing until the vehicle achieves its final altitude. The local vertical is a perpendicular axis for the Local Horizon as the lunar surface. For instance, Figure 2 shows such an LVLH frame where a vehicle is descending from altitude z_s to z_6. When it approaches the ground, it has traversed a downrange of $x_{\text{I}} + x_{\text{II}} + x_{\text{III}}$ km.

2.2. Mathematical Formulation. The lunar landing problem is solved in 3-dimensional state space governed by the physics laws. Each state parameter is a 3-dimensional vector along three axes of inertial reference frame according to the mean Earth coordinate system of the Moon. An autonomous lunar power descent phase exhibits dynamic behavior and is non-linear.

The governing motion equations for 3D non-linear system dynamics as given in [32] are given by equations as follows–:

$$\dot{r} = V, \tag{1}$$

$$\dot{\theta} = \frac{u}{r \cos \varnothing}, \tag{2}$$

$$\dot{\varnothing} = -\frac{w}{r}, \tag{3}$$

$$\dot{u} = -\frac{uv}{r} - \frac{vw}{r}\tan \varnothing + \frac{T}{m}\cos \alpha \cos \beta, \tag{4}$$

$$\dot{w} = \frac{u^2}{r}\tan \varnothing - \frac{vw}{r} + \frac{T}{m}\sin \alpha, \tag{5}$$

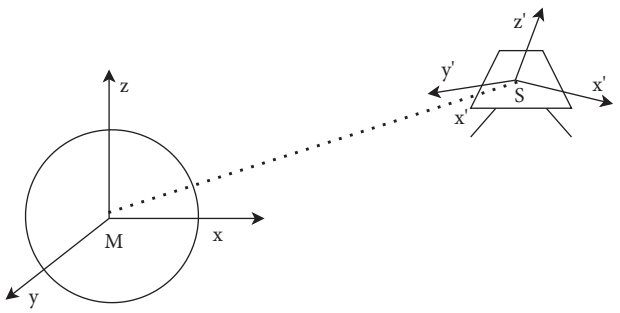


Figure 1: Inertial reference frame of spacecraft with respect to moon II.

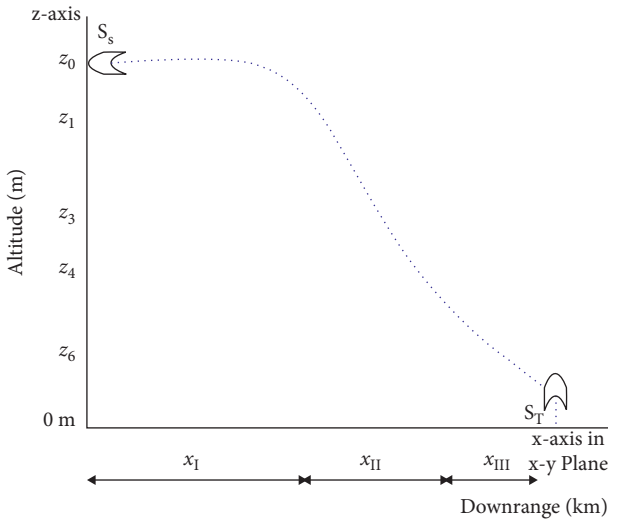


Figure 2: Local vertical local horizon (LVLH) frame.

$$\dot{v} = \frac{u^2}{r} + \frac{w^2}{r} - \frac{g_m}{r^2} - \frac{T}{m}\cos\,\alpha\,\sin\,\beta, \tag{6}$$

$$\dot{m} = -\frac{T}{I_{sp}g_0}. \tag{7}$$

The nomenclature followed with respect to the moon-centered reference frame for equations (1)–(7) is as follows:

$r = \left[\, x \;\; y \;\; z \,\right]^T$ is the distance between the current position of spacecraft and moon center (m).

θ, $\varnothing$ are the down range angle and cross range angle (degree).

$V = \left[\, u \;\; w \;\; v \,\right]^T$ are velocities along x, y, and z axes of 3D moon-centered reference frame (m/s).

$\dot{V} = \left[\, \dot{u} \;\; \dot{w} \;\; \dot{v} \,\right]^T$ is the net acceleration components along x, y, and z axes of 3D moon-centered reference frame (m/s^2).

$\dot{m}$ is the propellent mass flow rate (kg/s).

m is the mass of the spacecraft (kg).

T is the thrust magnitude (N).

α, β is the thrust Direction angles in LVLH frame.

At the start of the power descent phase, the spacecraft's altitude is less than 18 km. The gravitational acceleration is assumed to be constant, and the downrange angle is assumed to be very small in the range of 0.5 degrees. For the computation of trajectory state parameters, this simplified dynamics (equations (8)–(12)), as suggested in [33], is used.

$$\dot{r} = V, \tag{8}$$

$$\dot{V} = a - g_m, \tag{9}$$

$$a_x = \frac{T}{m}\cos\,\alpha\,\cos\,\beta, \tag{10}$$

$$a_y = \frac{T}{m}\sin\,\alpha, \tag{11}$$

$$a_z = \frac{T}{m}\cos\,\alpha\,\sin\,\beta, \tag{12}$$

where

$a = \left[\, a_x \;\; a_y \;\; a_z \,\right]^T$ is the acceleration vector.

$g_m = \left[\, 0 \;\; 0 \;\; 1.62 \,\right]^T$ is the acceleration due to the gravity of the Moon.

$\dot{r}$ is the time derivative of displacement = Velocity vector V.

$\dot{V}$ is the time derivative of velocity = Net Acceleration $(a - g_m)$.

The jerk is a time derivative of the acceleration vector as stated in

$$\begin{aligned} \dot{a} &= U, \\ U &= \left[\, u_x \;\; u_y \;\; u_z \,\right]^T. \end{aligned} \tag{13}$$

The minimum jerk guidance design [34] with constrained terminal velocity is employed to generate the acceleration command for the candidate trajectories. The minimum jerk guidance is used to minimize the following cost function as in the following equation:

$$\min_U \frac{1}{2}\int_0^{T_f} U^T U\,dt, \tag{14}$$

where T_f is the total time of flight.

2.3. Constrained System Design. Following initial values (equation (15)) are used to compute trajectories,

$$\begin{aligned} m &= 1000\,\text{kg}, \\ T &= 440\,\text{N}, \\ I_{sp} &= 310\,\text{s}, \\ g_m &= 1.62\,\text{m/s}^2, \end{aligned} \tag{15}$$

where I_{sp} is the specific impulse (s).

After solving the mass flow (16) we get mass flow rate, $\dot{m}$ = 0.1448 kg/s.

$$\dot{m} = -\frac{T}{I_{sp}g_0}. \tag{16}$$

For developing trajectories, we have randomly selected initial values of position, velocity, and final constraints are posed as given in Table 1.

Table 1: Initial position and initial velocity conditions and touchdown velocity constraints.

State parameter	Along axis	Initial condition range	Final condition range and velocity constraints
Position (m)	X	Random	Random
	Y	Random	Random
	Z	$Z < 15000$	$0 < Z < 1$
Velocity (m/s)	X	Random	0
	Y	Random	0
	Z	Random	0

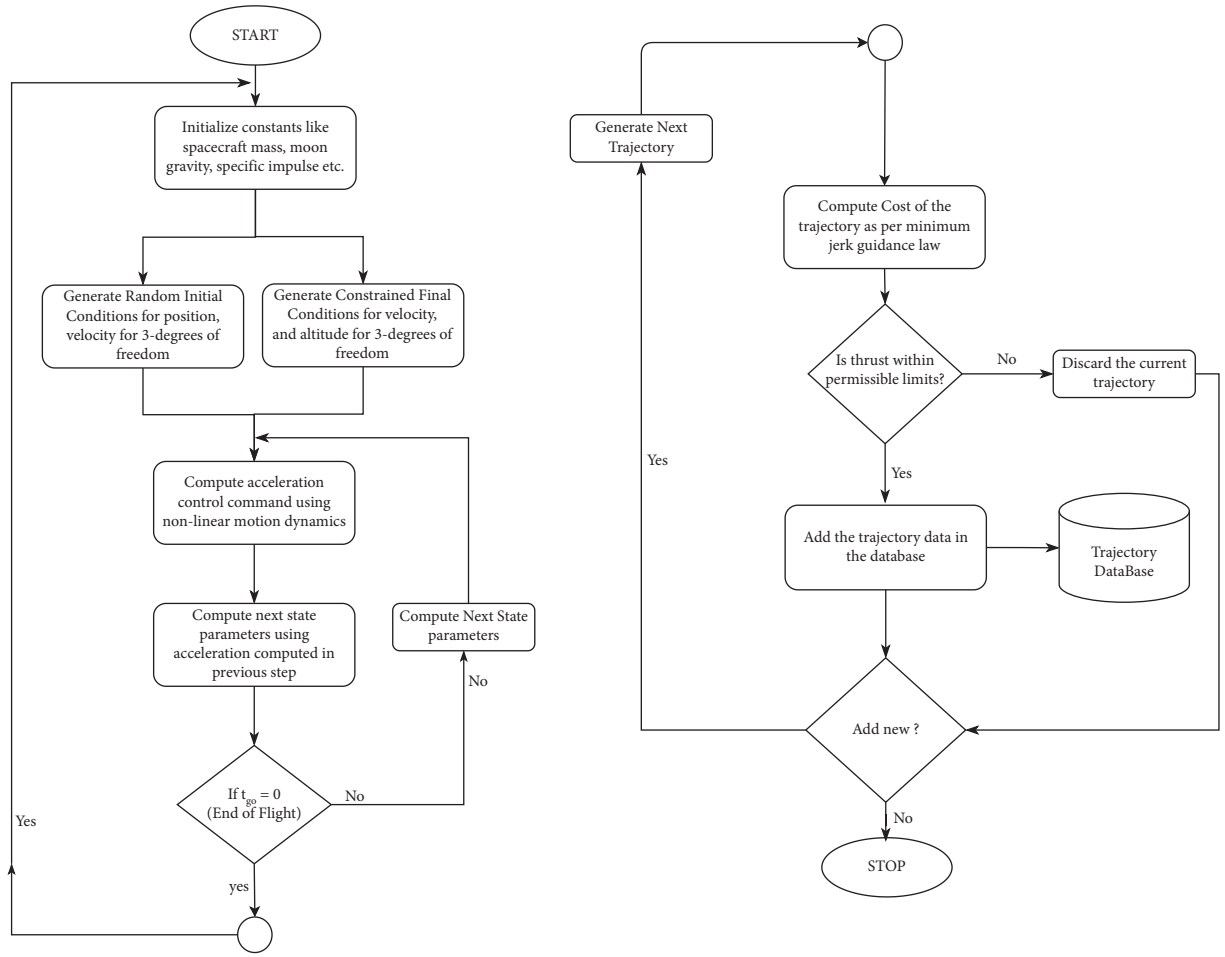


Figure 3: Trajectory generation flowchart.

The fundamental postulate regarding any physical landing system requires the final touchdown velocities to be as minimum as possible, near-zero, to minimize any hard landing possibility. This assumption forced us to put near-zero terminal velocities as a hard constraint while devising this algorithm. Along with that, a complete erect posture of the space vehicle is the one desired. It avoids the possibility of tumbling down due to imbalance. The perpendicular final thrust direction guarantees it such that the space vehicle lands softly on the landing surface. The feasibility of each generated trajectory is ascertained only after the validation of all these assumptions.

2.4. Autonomous Trajectory Generation Algorithm. Figure 3 shows a flow of the algorithm used for generating the trajectory database. This database is helpful for training machine learning models for autonomous lunar landing missions. It started with defining all constants mentioned in (15). The positions and velocities along x, y, and z directions are selected randomly from the range specified in Table 1. As the intention is to generate a hop-less trajectory, only starting and terminal constraints are known. The benefit of this algorithm is twofold: first, it is used to generate a one-shot trajectory; second, it can also be used to generate multi-hop trajectories using multiple hop-less trajectories depending on the mission demands.

Table 2: Position and velocity as a part of space vehicle state with randomized input and zero terminal velocity the constraint imposed by the algorithm for two samples.

Trajectory sample	State parameter	Along axis	Starting state parameters	Terminal state parameters
Sample[1]	Position (m)	X	14383	−29
		Y	350	−4
		Z	7799	0.26
	Velocity (m/s)	X	180	$4.5 \times e^{-12}$
		Y	9	0
		Z	−32	$0.4 \times e^{-12}$
Sample[2]	Position (m)	X	11946	−59
		Y	54	−7
		Z	3307	0.47
	Velocity (m/s)	X	208	0
		Y	7	0
		Z	−74	$0.3 \times e^{-12}$

After selecting the random positions and velocities, the next task is to compute the acceleration control command using non-linear simplified dynamics given by equations (8)–(12). Next-state parameters are computed using the previously computed acceleration. The whole process repeats till the total time of flight is reached. After the candidate trajectory is approximated, it passes through two checks: first, its thrust profile must be within the allowable non-inclusive limits i.e., $T_{\max} = 2000\,\mathrm{N}$ and $T_{\min} = 0\,\mathrm{N}$; second, the trajectory, which satisfies the minimum jerk command, is selected. Thus, a dataset with feasible reference trajectories is generated.

At each timestamp, the end of flight time is checked. For simplicity, only two values of total flight time are considered equal to 100 sec and 150 sec. The values for the total flight time are empirically chosen, giving more stable results and generating more feasible trajectories. If the end of flight time is reached, the vehicle has reached its destination. Accordingly, the current trajectory is assembled, and its cost is computed according to the minimum jerk guidance law. After computing the trajectory's cost, it has undergone a feasibility test to check if the thrusts are within permitted limits or not. Once the generated trajectory passes those two tests, it is stored in the database. Then, the initial and final constraints are initialized through the next trajectory's random function, and the procedure is repeated.

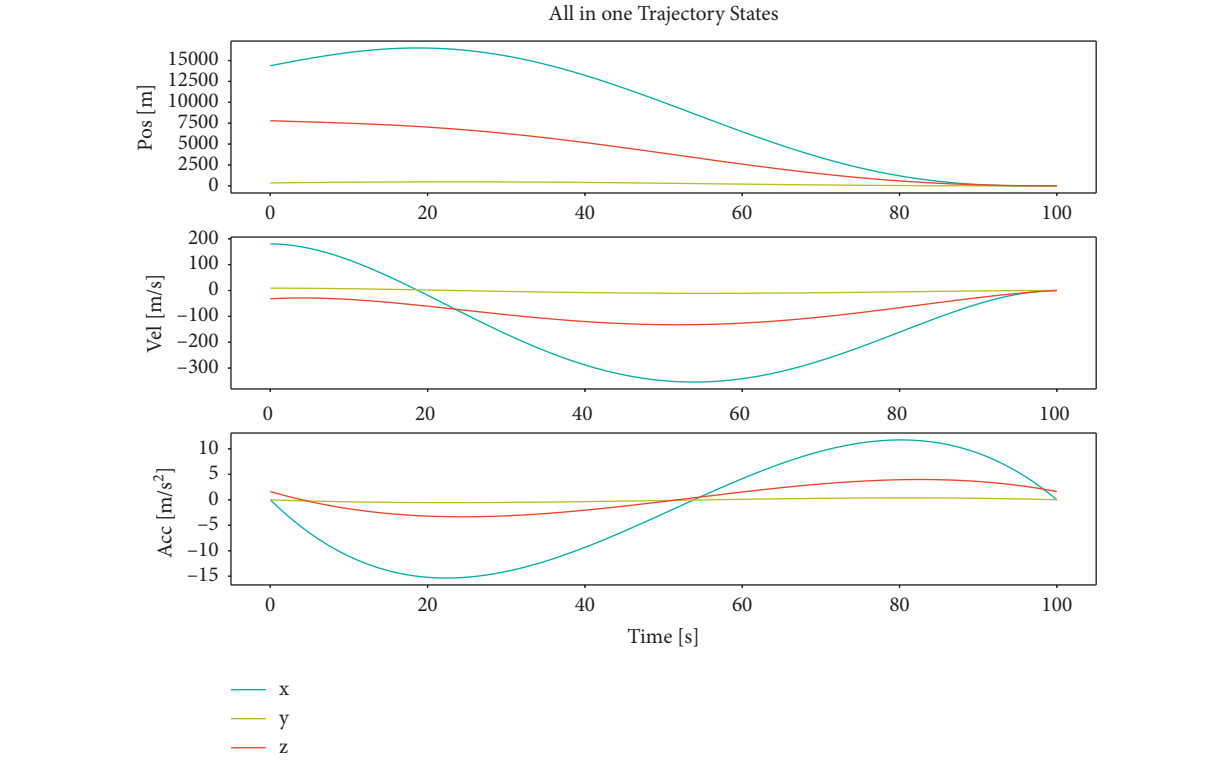
3. Results and Discussion

Experiments are performed to implement the proposed algorithm using a Python programming language. The generated trajectories are simulated to validate them analytically. A comprehensive analysis of sample trajectories is discussed in subsequent sections.

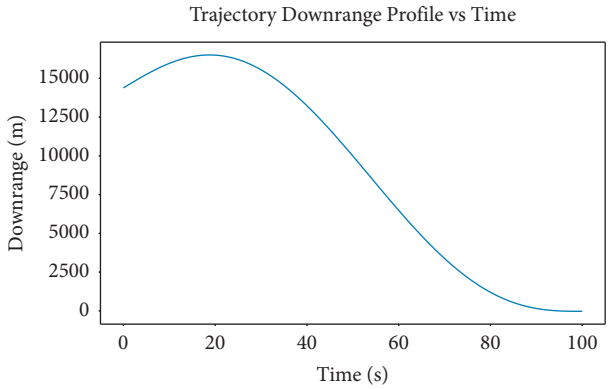
3.1. Single Trajectory Analysis. Table 2 enlists initial and final values of the position and velocity for two sample trajectories generated as an outcome of the trajectory generation algorithm. The initial and final position values are entirely random within the specified range as mentioned in Table 1. Velocity constraints are imposed while constructing these trajectories. This constraint results in the experimental touchdown velocities for samples, Sample[1] and Sample[2], almost approaching zero as demanded. The altitude of the space vehicle is also less than 1 m when the velocities approach zero. Hence, the decision can be made to allow the vehicle to free fall by switching off all engine thrusters. Thus, the generated analytical trajectories attain the best possible approximation of numerical constraints and are suitable for defining autonomous guidance law.

Further, to analyze the feasibility of the generated trajectories, simulation profiling of each state parameter is performed with respect to time. Figure 4 shows these simulations for Sample[1] trajectory from Table 2. Figure 4(a) shows position, velocity, and acceleration details along the x, y, and z axes of the moon-centered inertial reference frame. This sample trajectory exhibits the time of flight equal to 100 seconds. The convergence of all velocities and hence acceleration to numerical zero is apparent from the figure. Figure 4(b) shows the downrange profile with respect to the time axis. The altitude profile in the inertial reference frame is shown in Figure 4(c). Figure 4(c) says that the altitude is continuously decreasing and does not show an upper trend to decrease the horizontal velocity of the lander as in Figure 4(d). The altitude vs. downrange in the local vertical local horizontal (LVLH) frame is depicted in Figure 4(d). A local horizontal is an x-y plane, and the local vertical is perpendicular to this plane. Here, the space vehicle attitude is assumed along this perpendicular axis in the terminal descent phase. Figures 4(e) and 4(f) show profiles of horizontal and vertical velocity components. The velocities get reduced continuously and remain negative for the maximum period, which ascertains the decrease in altitude profile. Figures 4(g) and 4(h) are thrust and mass profiles that a trajectory follows throughout the time of the flight. The thrust profile shows thrust values along the time axis. Thrust is an external force applied to the lander that either accelerates or decelerates it to follow a specific trajectory.

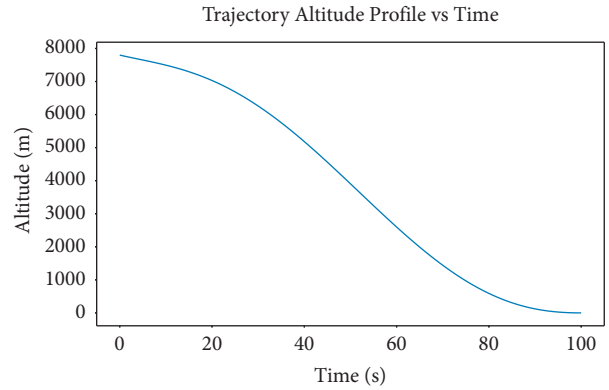
3.2. Multiple Trajectory Analysis. Multiple trajectory analysis is performed to understand their state parameters' trends and compare them with other trajectory data as depicted in Figures 5–10. Figures 5–10 show altitude, downrange, angular velocity, thrust magnitude, horizontal velocity, and vertical velocity trends, respectively. 10 trajectories are randomly selected from the generated trajectory dataset, and trends are analyzed for each state parameter. Altitude and downrange profiles (Figures 5 and 6), starting with random input values, converge to their desired target values throughout the flight. The maximum change in angular velocity (Figure 7) is far less than 90 degrees and is proportional to the thrust magnitude (Figure 8) as the desired profile. A low rate of change in angular velocity guarantees less thrust application and lower fuel consumption. Horizontal velocity (Figure 9) reduces for the first few seconds as in the rough braking phase of power descent. In the end, it increases gradually to attain the zero touchdown



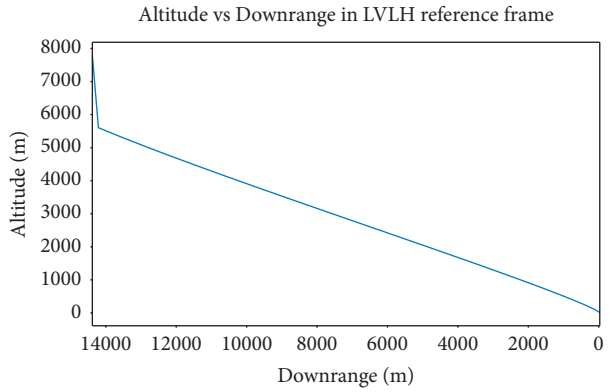
(a)



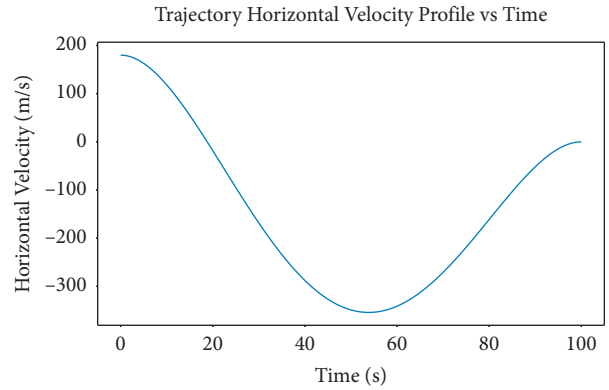
(b)



(c)



(d)



(e)

Figure 4: Continued.

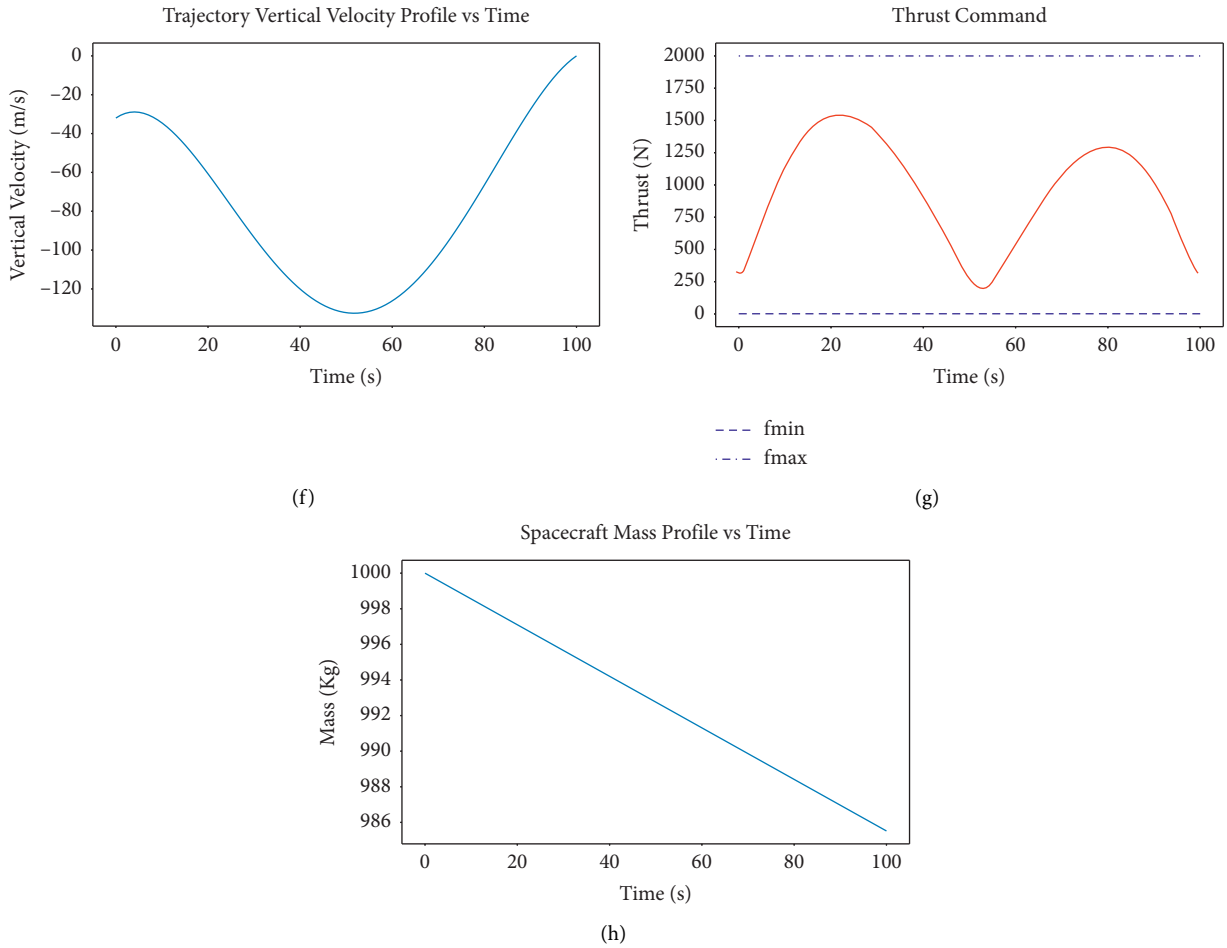


FIGURE 4: (a)–(h) Trajectory profiles for sample[1] (Table 2) at $T_f = 100$ sec. (a) Complete simulated trajectory. (b) Downrange profile. (c) Altitude profile. (d) Altitude in LVLH frame. (e) Horizontal velocity profile. (f) Vertical velocity profile. (g) Thrust profile. (h) Mass profile.

convergence. Vertical velocities (Figure 10) are changing as per hovering or maneuvering requirements of the lander but are obeying the zero touchdown constraint imposed.

3.3. Feasibility Measurement in Trajectory Computation. The algorithm tries to keep only the feasible trajectories by discarding the nonfeasible ones. We tried to measure the rate of generation of feasible trajectories in each algorithm run. The rate is defined by (17). As discussed in Section 2, the feasible candidates are found by applying two checks. The proposed algorithm is run for two values of time of flights $T_f = 100$ sec & $T_f = 150$ sec. For the first value, 4 times the algorithm is run, while 2 times it is run for the second value. The results of the feasibility rate are tabulated in Table 3 for each run. The feasible candidate generation rate of the algorithm for 100 sec flight time is 0.87, and for 150 sec flight time is 0.67. It means that the chance of getting feasible trajectories for greater flight time is the least. Regarding the power descent phase, flight time is also a tunable parameter. It needs more exhaustive experimentation to arrive at the optimal solution. It is again a new optimization problem, and it is beyond the scope of this research.

$$\text{rate}^{i}_{\text{feasible}} = \frac{\text{number of feasible candidates}^{i}}{\text{total candidates}^{i}}. \tag{17}$$

3.4. AI-ML Use Case for Usefulness of the Generated Data

3.4.1. State Prediction for Dynamic Systems. A typical AI-ML system for predicting the next state S_{i+1} of the system using the current state S_i is shown in Figure 11. The trajectory data generated through our proposed algorithm is fed to an AI-ML algorithm for training. The trajectory data contains information about all the system states S_i, where $i = T_0$ to T_f. In short, it contains all the training data with all the ground truth, which is what a supervised AI-ML system desires for training. This learning outcome is a state prediction model used to predict the next expected state of the system in real-time. An autonomous landing vehicle can use this state prediction model for making accurate maneuvering decisions.

The effectiveness of AI-ML technology is based on the quality of data used for training. This paper introduces an approach for generating such good quality data.

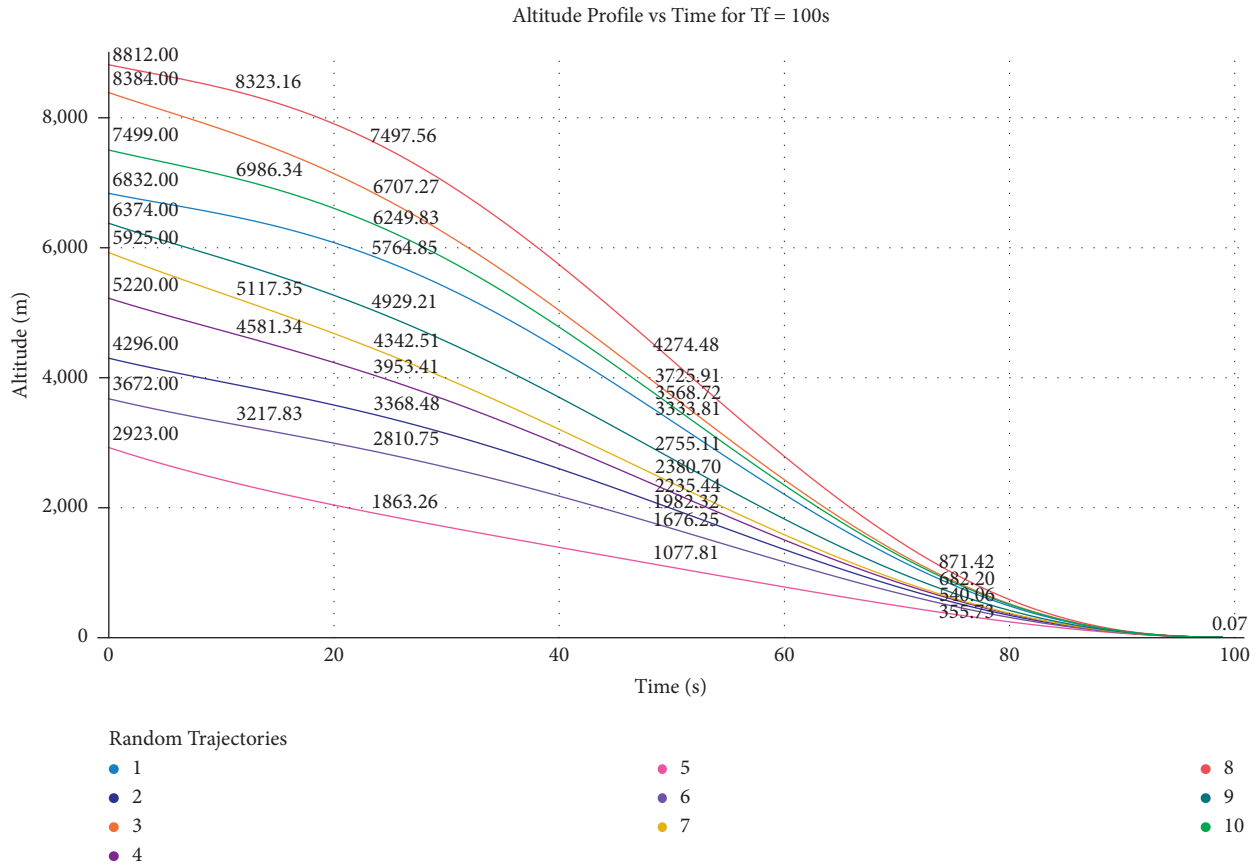


Figure 5: Trend analysis of altitude profiles for randomly selected 10 trajectories at $T_f = 100$ sec.

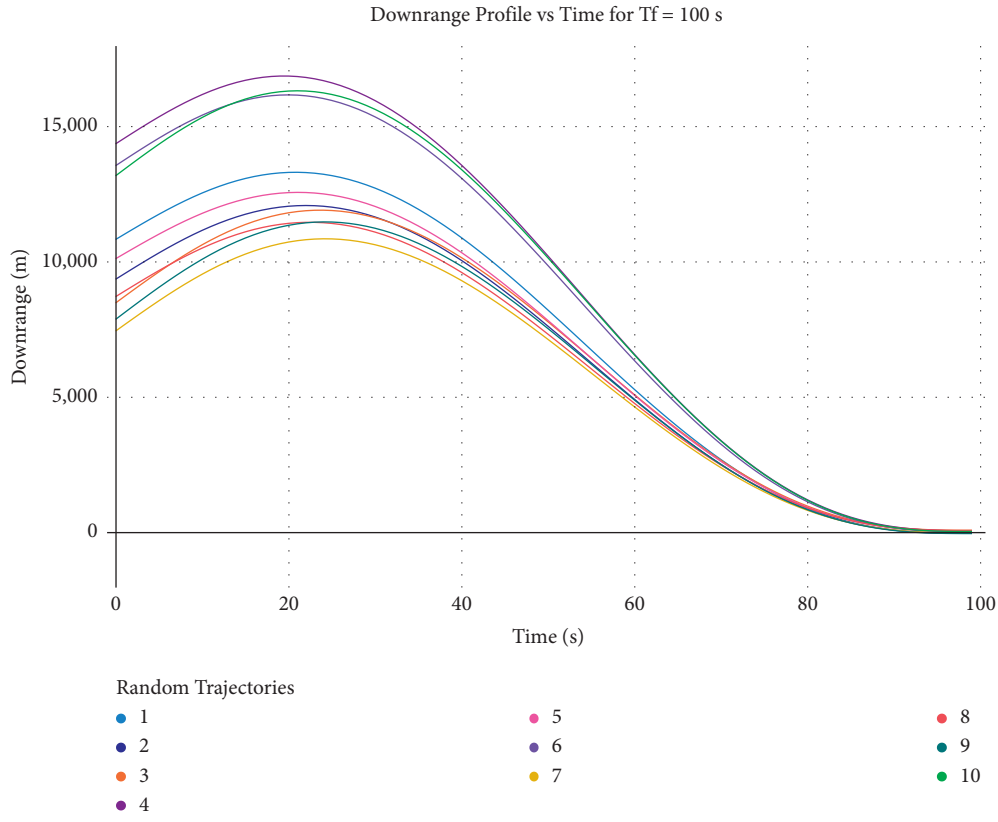


Figure 6: Trend analysis of downrange profile for randomly selected 10 trajectories at $T_f = 100$ sec.

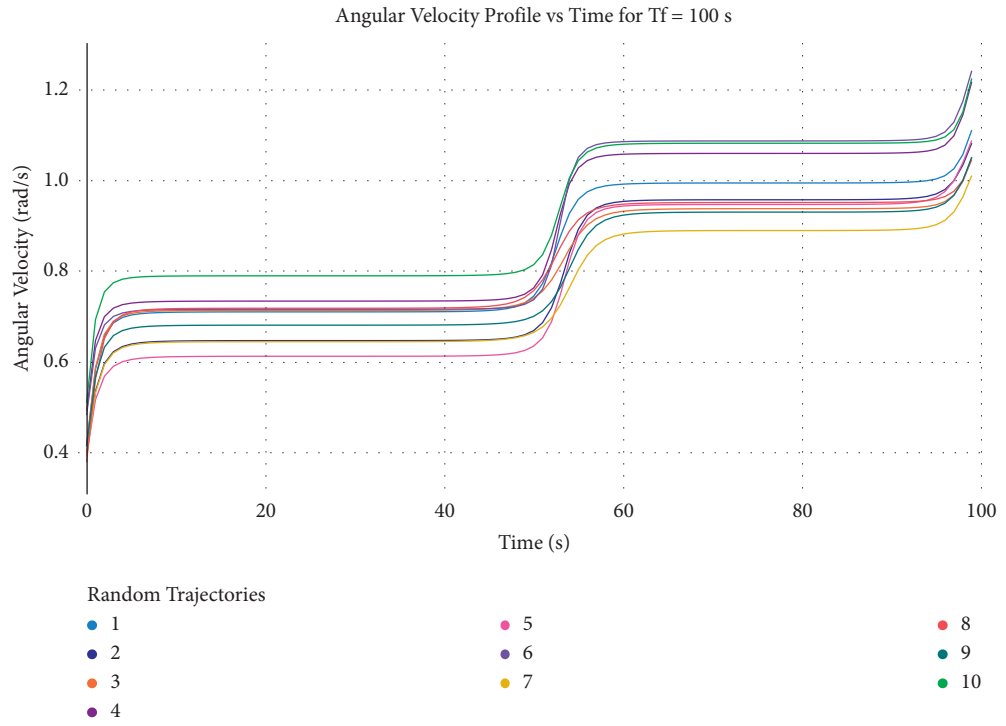


Figure 7: Trend analysis of angular velocity profile for randomly selected 10 trajectories at $T_f = 100$ sec.

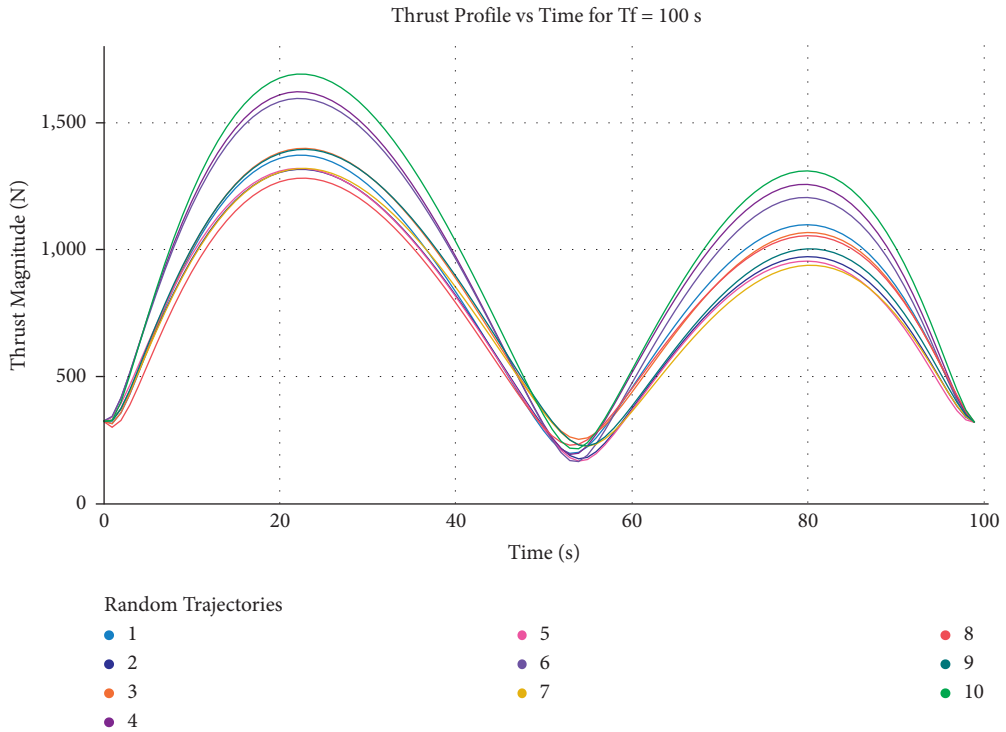


Figure 8: Trend analysis of thrust magnitude for randomly selected 10 trajectories at $T_f = 100$ sec.

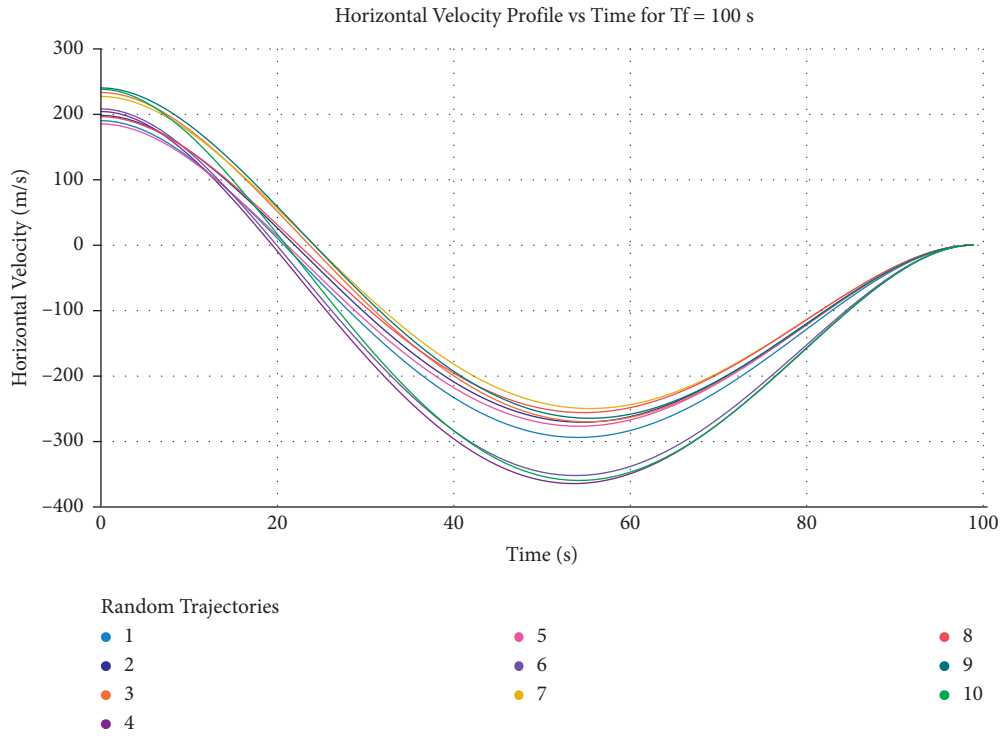


Figure 9: Trend analysis of horizontal velocity for randomly selected 10 trajectories at $T_f = 100$ sec.

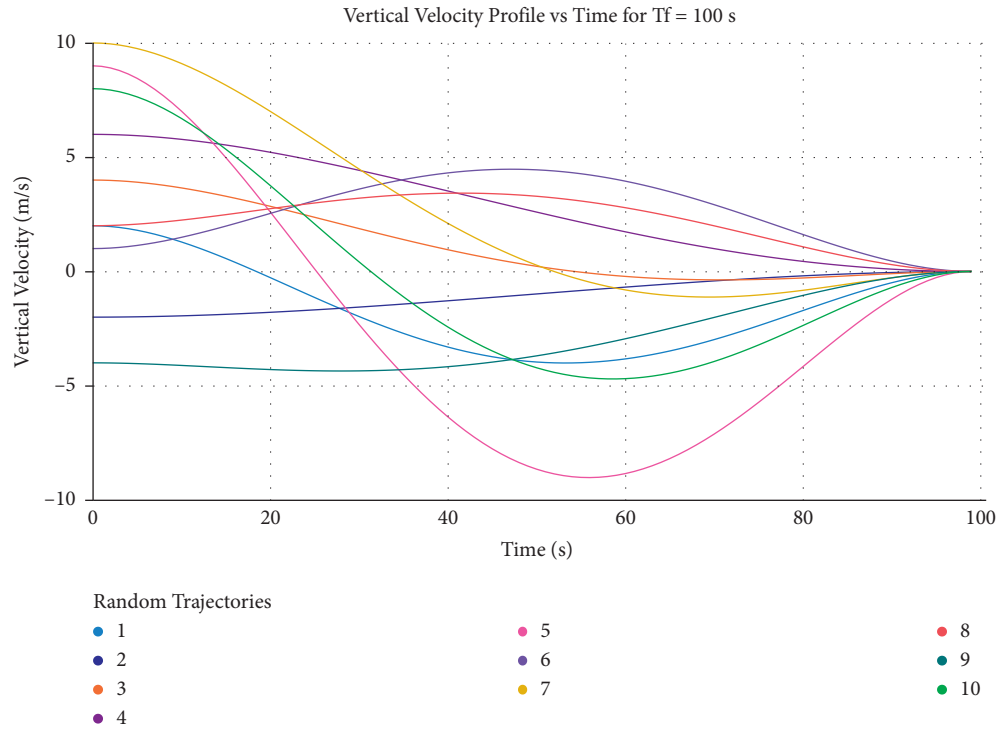


Figure 10: Trend analysis of vertical velocity for randomly selected 10 trajectories at $T_f = 100$ sec.

Table 3: Analysis of feasible trajectory generation rate of the proposed algorithm.

Time of flight	T_f = 100 sec				T_f = 150 sec	
i^{th} run	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 1$	$i = 2$
$rate^i_{feasible}$	0.89	0.86	0.87	0.86	0.70	0.65
Average	0.87				0.67	

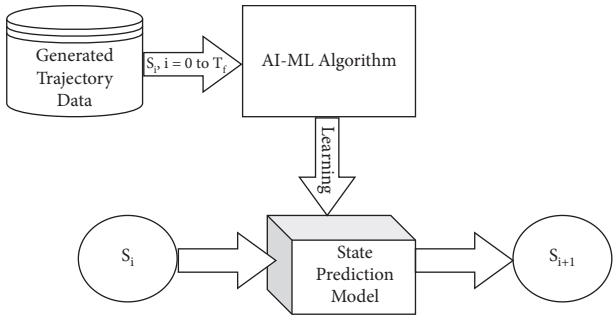


Figure 11: State prediction using AI-ML.

4. Conclusions and Future Scope

This paper presents an autonomous algorithm for generating possible candidates of feasible trajectory paths which a spacecraft would follow from a random initial state to a zero velocity end state. The algorithm is useful for training AI systems in autonomous landing missions. The presented study concentrated specifically on the lunar landing. However, the algorithm can be fine-tuned to generate trajectories for any autonomous mission. The algorithm uses 3-dimensional system dynamics with randomized inputs. It enforces a constrained terminal velocity guidance law to ensure a soft landing on the lunar surface. The actual engine design is taken into consideration while developing the proposed algorithm. Hence, the point-mass vehicle assumption of existing systems is discarded. A comprehensive analysis of generated trajectories is presented using simulation studies. It is found that the algorithm approximates the trajectories almost similar to their numerical counterparts and converges to their measured final state estimates. The generation rate of feasible trajectories measures the accuracy of the algorithm. The rate is near 0.87 for 100 sec flight time, which is reasonable accuracy. But it is also found that higher flight time significantly affects the accuracy. Flight time optimization is a further scope of research and can be targeted in future works. The overall impact of the trajectory generation algorithm is significant. It treats each state parameter with three degrees of freedom in the nonlinear sense that real-time missions require. These trajectories provide analytical measurements of dynamic state parameters and are further useful in future state predictions. It also removes the multi-hop barrier in the trajectory construction. The generated data consists of all possible solutions for proper training of AI-ML systems. The introduction of randomness in the trajectory computation minimizes the subjective bias in the training process. This research is a part of an autonomous landing project. It requires vast data to be prepared. The generated data can be used to generate prediction models for the spacecraft descent maneuver as a future task [6, 23, 24].

Data Availability

The data are available on request.

Conflicts of Interest

The authors do not have any conflicts of interest to declare.

Authors' Contributions

Janhavi H. Borse wrote the paper, Dipti D. Patil simulated the method, Vinod Kumar supervised the work, and Sudhir Kumar analyzed the method.

References

[1] M. Kaur, S. Kadam, and N. Hannoon, "Multi-level parallel scheduling of dependent-tasks using graph-partitioning and hybrid approaches over edge-cloud," *Soft Computing*, vol. 26, no. 11, pp. 5347–5362, 2022.

[2] J. H Borse, D. D Patil, and V. Kumar, "Tracking keypoints from consecutive video frames using CNN features for space applications," *Teh. Glas*, vol. 15, no. 1, pp. 11–17, 17, Mar. 2021.

[3] J. Song, D. Rondao, and N. Aouf, "Deep learning-based spacecraft relative navigation methods: a survey," *Acta Astronautica*, vol. 191, pp. 22–40.

[4] J. Liu, X. Ren, W. Yan et al., "Descent trajectory reconstruction and landing site positioning of Chang'E-4 on the lunar farside," *Nature Communications*, vol. 10, no. 1, pp. 1–10, 2019.

[5] D. Izzo and G. C. H. E. de Croon, "Nonlinear model predictive control applied to vision-based spacecraft landing," *EuroGNC*, pp. 1–17, 2013.

[6] X. Zhang, J. Ma, Z. Cheng, S. Huang, S. S. Ge, and T. H. Lee, "Trajectory generation by chance-constrained nonlinear MPC with probabilistic prediction," *IEEE Transactions on Cybernetics*, vol. 51, no. 7, pp. 3616–3629, 2021.

[7] Y. Al Younes and M. Barczyk, "Nonlinear model predictive horizon for optimal trajectory generation," *Robotics*, vol. 10, no. 3, p. 90, 2021.

[8] B. G. Park, J. S. Ahn, and M. J. Tahk, "Two-dimensional trajectory optimization for soft lunar landing considering a landing site," *International Journal of Aeronautical and Space Sciences*, vol. 12, no. 3, pp. 288–295, 2011.

[9] R. V. Ramanan and M. Lal, "Analysis of optimal strategies for soft landing on the moon from lunar parking orbits," *Journal of Earth System Science*, vol. 114, no. 6, pp. 807–813, 2005.

[10] Q. B. Peng, H. Y. Li, H. X. Shen, and G. J. Tang, "Hybrid optimization of powered descent trajectory for manned lunar mission," *Transactions of the Japan Society for Aeronautical and Space Sciences*, vol. 56, no. 3, pp. 113–120, 2013.

[11] J. Y. Zhou, K. L. Teo, D. Zhou, and G. H. Zhao, "Optimal guidance for lunar module soft landing," *Nonlinear Dynamics and Systems Theory*, vol. 10, no. 2, pp. 189–201, 2010.

[12] W. Zhang and M. Kaur, "A novel QACS automatic extraction algorithm for extracting information in blockchain-based systems," *IETE Journal of Research*, pp. 1–13, 2022.

[13] B. Mareschal, M. Kaur, V. Kharat, and S. S Sakhare, "Convergence of smart technologies for digital transformation," *Tehnički glasnik*, vol. 15, no. 1, 2021.

[14] C. N. D'Souza and C. D'Souza, "An optimal guidance law for planetary landing," *Guidance, Navigation, and Control Conference*, pp. 1376–1381, 1997.

[15] A. K. Automatica, *Apollo Lunar Descent Guidance,* Elsevier 1974, HYPERLINK ".https://www.sciencedirect.com/science/article/pii/0005109874900193."\o".https://www.sciencedirect.com/science/article/pii/0005109874900193".

[16] A. Jadhav, M. Kaur, and F. Akter, "Evolution of software development effort and cost estimation techniques: five decades study using automated text mining approach," *Mathematical Problems in Engineering*, vol. 2022, pp. 1–17, 2022, https://doi.org/10.1155/2022/5782587.

[17] J. Borse and V. K. Dipti Patil, *Deep Semantic Classification Of Visual Inputs For Hazard-Free Lunar Landing*, June, vol. 3, pp. 14–18, 2021.

[18] C. Schmierer, K. Tomilin, M. Kobald, J. Steelant, and S. Schlechtriem, "Analysis and Preliminary Design of a Hybrid Propulsion Lunar Lander," in *Proceedings of the Space Propulsion 2016Sp. Propuls*, Rome, Italy, May, 2016.

[19] M. S. Islam and I. M. Mehedi, "Landing trajectory generation and energy optimization for unmanned lunar mission," *Mathematical Problems in Engineering*, vol. 2021, pp. 2021–11.

[20] A. R. Babaei and H. Maghsoudi, "Aircraft three-dimensional hard-constrained trajectory planning using pseudospectral optimization method," *Journal of Aerospace Technology and Management*, vol. 13, p. 2021.

[21] D. Izzo, M. Märtens, and B. Pan, "A survey on artificial intelligence trends in spacecraft guidance dynamics and control," pp. 1–13, Astrodynamics, 2018.

[22] J. H. Borse and D. D. Patil, "Empirical analysis of feature points extraction techniques for space applications," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 9, pp. 81–87, 2021.

[23] R. Furfaro et al., "Deep learning for autonomous lunar landing," in *Proceedings of the 2018 AAS/AIAA Astrodynamics Specialist Conference*, pp. 1–22, Snowbird, Utah, U.S.A, Augest2018.

[24] R. Furfaro, I. Bloise, M. Orlandelli, P. Di, F. Topputo, and R. Linares, *A Recurrent Deep Architecture for Quasi-Optimal Feedback Guidance in Planetary Landing*, pp. 1–24.

[25] C. Sánchez-sánchez and D. Izzo, "Real-time Optimal Control via Deep Neural Networks," *Study on landing Problems*, pp. 1–35.

[26] Y. Song, X. Miao, L. Cheng, and S. Gong, "The feasibility criterion of fuel-optimal planetary landing using neural networks," *Aerospace Science and Technology*, vol. 116, p. 2021, Article ID 106860.

[27] L. Cheng, F. Jiang, Z. Wang, and J. Li, "Multiconstrained real-time entry guidance using deep neural networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 1, pp. 325–340, 2021.

[28] G. Ciabatti, S. Daftry, and R. Capobianco, "Autonomous planetary landing via deep reinforcement learning and transfer learning," in *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2031–2038, Nashville, TN, USA, June2021.

[29] M. Kaur, S. R. Sakhare, K. Wanjale, and F. Akter, "Early stroke prediction methods for prevention of strokes," *Behavioural Neurology*, pp. 1–9, 2022.

[30] S. L. Rexius, T. E. Rexius, T. R. Jorris, and A. V. Rao, "Advances in highly constrained multi-phase trajectory generation using the General Pseudospectral Optimization Software GPOPS," *AIAA Guidance, Navigation, and Control (GNC) Conference*, vol. 298, 2013.

[31] M. Kaur, A. Jadhav, and F. Akter, "Resource Selection from Edge-Cloud for IIoT and Blockchain-Based Applications in Industry 4.0/5.0," *Security and Communication Networks,Hindawi*, 2022.

[32] M. D. Li, M. MacDonald, C. R. McInnes, and W. X. Jing, "Analytical landing trajectories for embedded autonomy," *Proceedings of the Institution of Mechanical Engineers - Part G: Journal of Aerospace Engineering*, vol. 224, no. 11, pp. 1177–1191, 2010.

[33] D. H. Cho, D. Kim, and H. Leeghim, "Optimal lunar landing trajectory design for hybrid engine," *Mathematical Problems in Engineering*, vol. 2015, pp. 1–8, 2015.

[34] K. Uchiyama, Y. Shimada, and K. Ogawa, "Minimum-jerk guidance for lunar lander," *Transactions of the Japan Society for Aeronautical and Space Sciences*, vol. 48, no. 159, pp. 34–39, 2005.