Project Report On

Automation of a Lunar Lander using Reinforcement Learning

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

By

Rohan Ranjith (U2003172)

Sebastian K Suresh (U2003217)

Sneha Sarah John (U2003203)

Varun Pradeep (U2003211)

Under the guidance of

Ms. Mehbooba P Shareef

Department of Computer Science and Engineering
Rajagiri School of Engineering & Technology (Autonomous)
(Parent University: APJ Abdul Kalam Technological University)
Rajagiri Valley, Kakkanad, Kochi, 682039
April 2024

CERTIFICATE

This is to certify that the project report entitled ***"Automation of a Lunar Lander using Reinforcement Learning"*** *is a bonafide record of the work done by* ***Rohan Ranjith (U2003172), Sebastian K Suresh (U2003217), Sneha Sarah John (U2003203), Varun Pradeep (U2003211)****, submitted to Rajagiri School of Engineering & Technology (RSET) (Autonomous) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Engineering during the academic year 2023-2024.*

Ms. Mehbooba P Shareef	Ms. Anita John
Project Guide	Project Coordinator
Asst. Professor	Asst. Professor
Dept. of CSE	Dept. of CSE
RSET	RSET

Dr. Preetha K.G.
Professor and HoD
Dept. of CSE
RSET

ACKNOWLEDGMENT

We wish to express our sincere gratitude towards **Dr. Sreejith P S**, Principal of RSET, and **Dr. Preetha K.G.**, Head of the Department and Professor of Computer Science and Engineering for providing us with the opportunity to undertake our project, "Project Title".

We are highly indebted to our project coordinators, **Ms. Anita John**, Assistant Professor, Department of Computer Science and Engineering, and **Dr. Mary Priya Sebastian**, Associate Professor, Department of Computer Science and Engineering for their valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our project guide **Ms. Mehbooba P Shareef** for her patience and all the priceless advice and wisdom she has shared with us.

Last but not the least, We would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Rohan Ranjith

Sebastian K Suresh

Sneha Sarah John

Varun Pradeep

Abstract

One of the key areas in space exploration is to ensure successful planetary soft landings, which require a paramount amount of precision as well as accuracy for the safety of these missions. However, uncertain terrains, dynamic environments, and restrictions in communication have made this task quite difficult. While many traditional methods are still being employed, they face limitations with respect to adaptability and autonomy. Hence, a fuel efficient, safe and accurately controlled descent still remains as a challenge to this day.

An approach utilizing Reinforcement Learning (RL) in the powered descent phase, has been proposed to counter the limitations of adaptability and improve the accuracy of planetary landings. Terrain data obtained are used to create Digital Elevation Models (DEMs), which are then used to generate the required safety maps. An RL algorithm based on a policy network aids in the creation of optimal control signals for a controlled descent at a safe landing site.

The algorithm is then trained in a simulated environment created using DEMs until it learns to adapt to new environments and real world systems. The environment simulation helps cut down the costs of prototyping and testing. This novel approach increases the reliability of existing systems, paving a new standard in planetary soft landings in deep space exploration missions.

Contents

List of Abbreviations

Abbreviation	Expansion
DEM	Digital Elevation Model
DNN	Deep Neural Network
ML	Machine Learning
RL	Reinforcement Learning
DDPG	Deep Deterministic Policy Gradient
TD3	Twin Delayed DDPG
SAC	Soft Actor Critic
PPO	Proximal Policy Optimization
DOF	Degrees of Freedom
SDK	Software Development Kit
GIS	Geographic Information System
LiDAR	Light Detection And Ranging
ALHAT	Autonomous Landing Hazard Avoidance Technology
GDAL	Geospatial Data Abstraction Library
USGS	United States Geological Survey
DRL	Deep Reinforcement Learning
OCP	Optimal Control Policy

List of Abbreviations and Their Expansions

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

This report provides a detailed insight into the novel method of utilizing RL algorithms to facilitate a planetary soft landing. The primary motivation behind this project is to address communication restrictions, dynamic environments, and various uncertainties involved in deep space exploration missions, particularly during the powered descent phase. RL contributes to bringing aspects of autonomy and adaptability, which are highly critical requirements for a safe and efficient spacecraft landing.

This project aims to develop a prototype capable of simulating a planetary soft landing using the Unity game engine. DEMs of real-world terrain are utilized to generate the simulated terrain, while the Unity physics engine supports dynamics such as gravity and lander movement. The ML agents toolkit is used define and integrate RL into the lander.

The lander scans the terrain and creates a DEM which is employed to develop a safety map based on the aspects of roughness and slope. The system uses this map to select a safe landing site and guide the lander for a fuel-efficient and safe descent.

As a part of this study, an investigation into certain existing methods, including the applications of AI in Chandrayaan 3, ALHAT, and PDG systems, is conducted. Furthermore, parallels between the various existing systems and the proposed systems are drawn through an extensive literature survey.

In conclusion, this study aims to highlight the benefits of an approach to planetary soft landing that uses computer vision and RL algorithms. The prototype for this project seeks to achieve a cost-effective manner of training landers before space exploration missions, positioning it as a key element in future innovations.

1.2 Problem Definition

The problem definition states that the project intends to create a system that is tasked with autonomously and adaptively guiding lunar landers to perform safe, precise, and efficient soft landings on the lunar surface.

1.3 Scope and Motivation

Scope of the project includes implementing adaptive trajectory adjustments through RL enabling the spacecraft to achieve precise targeting during descent, crucial for controlled landings. RL fosters adaptability, allowing spacecraft to respond to dynamic conditions such as atmospheric changes or terrain features for safe landings. Also, RL aids in fuel optimization by adjusting trajectories to conserve fuel, ensuring reserves for unforeseen circumstances or additional objectives post-landing. Real-time decision-making during descent, facilitated by RL enables spacecraft to respond swiftly to unexpected challenges for safe landings.

The fundamental motivation behind this project was to improve adaptability and develop a more autonomous system that would enhance safety and mission success. Autonomous systems could provide increased accuracy and precision during descent, enabling more mission flexibility. When the best possible choice for the descent trajectory is made, it also helps to reduce fuel consumption, making missions more fuel-optimal and efficient.

1.4 Objectives

- Create a simulation for planetary landing in Unity game engine, including the lander and the terrain.
- Develop a hazard detection algorithm that creates a safety map from a given DEM to spot the safe landing sites.
- Choose a trajectory that leads to fuel optimality in the mission.

- Ensure the system can make real-time decisions at all times.
- Ensure the system is robust to small uncertainties in the thrusters.

1.5 Challenges

Challenges in applying reinforcement learning (RL) to spacecraft lander automation include the risk of inaccurate simulations for model training, potentially affecting performance in real lunar conditions. Technical hurdles arise from diverse lander designs, necessitating fine-tuning for variations in size, shape, and thruster layouts. Computation challenges stem from the need for high efficiency in executing calculations and observations on the lander module. Unfavorable environmental conditions, like low visibility and atmospheric occlusion, further complicate accurate terrain scanning. Additionally, maintaining confidence in RL decision-making, especially in critical situations, remains an ongoing challenge for autonomous spacecraft operations.

1.6 Assumptions

Assumptions for the spacecraft landing automation include a constant gravitational effect during descent and generally non-extreme environmental conditions, such as minimal wind and a non-corrosive atmosphere. For the lander, it is assumed that sensors and onboard computing systems will not fail, and these systems possess sufficient power to perform the necessary calculations. Additionally, the assumption is made that Digital Elevation Model (DEM) data of known planets is freely available for training purposes.

1.7 Societal / Industrial Relevance

Applying RL to automate spacecraft landers has a big impact on both our society and industries. Firstly, it significantly boosts the chances of mission success by helping spacecraft make quick and smart decisions in changing space conditions. This is crucial for tasks like scientific exploration and deploying satellites. Secondly, it can make space missions more cost-effective by reducing the need for human control and adjusting to changes on its own, like using fuel more efficiently. Safety is also improved, as automation reduces the chance of human errors during important parts of a mission. Thirdly, by optimizing the use of resources like fuel, space missions become more sustainable and economically sensible. The technology behind this doesn't just benefit space; it also sparks innovation that can influence other industries. Fourthly, it allows spacecraft to operate more independently, especially in deep space where communication delays are common. Furthermore, it encourages collaboration between different countries in space exploration, with shared and standardized landing procedures. In the realm of commercial space activities, using automation gives companies a competitive edge, ensuring efficiency in tasks like satellite launches and space tourism. The flexibility of using RL is a game-changer, allowing spacecraft to handle various missions and unexpected challenges. Lastly, these advancements in space technology help capture the public's interest, inspiring curiosity and interest in science and engineering among future generations.

1.8 Organization of the Report

The report commences with an abstract, which offers a concise overview of the topic. The introduction delves into the background, problem definition, scope, motivation, objectives, challenges, assumptions, and social/industrial relevance of employing RL to automate a spacecraft lander. A literature survey is conducted to review existing knowledge. Hardware, software, and functional requirements are outlined followed by the system architecture section which encompasses the overview, architectural design, module division, and a work schedule - Gantt chart. System implementation is discussed, followed by results and discussions related to the project. The report concludes by summarizing key findings and providing references for further exploration.

In this chapter, we have outlined the background, problem definition, scope and motivation, objectives, challenges, assumptions, and social/industrial relevance of utilizing RL for autonomous spacecraft landing. The usage of RL in the Unity game engine presents an approach to address challenges in deep space exploration, particularly during the powered descent phase. The objectives encompass the development of a simulation, hazard detection algorithm, and RL model for safe and efficient landings.

Chapter 2

Literature Survey

2.1 Deep Reinforcement Learning-Based Accurate Control of Planetary Soft Landing [1]

The paper's major contributions include introducing velocity-tracking-based rewards and fuel consumption penalties and constraints. Initially, it outlines the preliminaries of formulating the soft landing problem and discusses various RL models capable of faster and more efficient convergence. Deep Deterministic Policy Gradient (DDPG) is an actor-critic algorithm learning deterministic policies in continuous action spaces, while Twin Delayed DDPG (TD3) extends DDPG with twin critics and target policy smoothing. Soft Actor-Critic (SAC) incorporates entropy regularization, making it effective for tasks with sparse rewards as it encourages exploration.

Various reward settings are also discussed to index the agents' behavior. The goal-achieving reward rewards the agent when altitude, downward velocity, attitude angle, and angular rate meet specified criteria. The velocity tracking reward provides feedback based on the lander's deviation from reference velocity, obtained via real-time position. The crash penalty penalizes the lander if its tilt or speed deviation exceeds defined thresholds. The fuel consumption penalty discourages excessive fuel usage, while the constant reward provides constant reinforcement independent of task completion.

2.2 Deep Reinforcement Learning for Safe Landing Site Selection with Concurrent Consideration of Divert Maneuvers [2]

The paper references previous work on LiDAR-based hazard avoidance for safe landing on Mars. It also discusses the assumptions used in modeling the HDA phase and shows that the sequence of action choices can be formulated as a partially observable Markov decision process (POMDP).

The proposed framework leverages model-free reinforcement learning techniques to learn a policy that concurrently selects the landing site and guidance strategy to the target, by interacting with the simulator environment and learning how to maximize the total probability of successful landing with minimal fuel consumption. The architecture of the policy network and critic network is also discussed.

2.3 Real-Time Control for Fuel-Optimal Moon Landing based on An Interactive Deep Reinforcement Learning Algorithm [3]

This study presents an inventive approach to real time optimal control aimed at increasing the autonomy and adaptability of lunar landing missions. Given the challenges posed by communication restrictions and environmental uncertainties, it has become imperative that we look into more advanced landing techniques. Traditional solutions, both direct and indirect methodologies, often lack autonomy and adaptability despite offering computational efficiency. DRL techniques have emerged as a promising solution to the OCP, but still face issues regarding slow convergence and intricate reward design. To counter these challenges, a DRL framework utilizing indirect method is harnessed to generate optimal control actions, with certain enhancements such as costate normalization and DNN-based initial guess generation strategies. Simultaneously, an interactive actorindirect structured DRL algorithm is introduced to train a DNN-driven controller. Additionally, a non-linear feedback controller is developed and combined with the DNN-driven controller, to improve landing accuracy. Several numerical simulations are conducted to validate the effectiveness of the proposed interactive DRL algorithm. In summary, this study underscores the potential of interactive DRL algorithms in the realm of deep space exploration. It contributes to future Moon landing missions by integrating traditional as well as modern control techniques.

2.4 Summary and Gaps Identified

2.4.1 Summary

Title	Advantages	Disadvantages
Accurate control of planetary soft landing [1]	• DRL is more adaptable to 6DOF dynamics to ensure a more realistic simulation	• Some RL algorithms are highly sensitive to hyper-parameters and hence require manual tuning
Safe Landing Site Selection with Consideration of Divert Maneuvers [2]	• Use of RL in landing site selection ensures flexibility in diverting options if required	• Computationally heavy algorithms might limit real-time adjustments to the trajectory
Fuel-Optimal Moon Landing based on An Interactive DRL Algorithm[3]	• A non-linear feedback controller is developed and combined with the DNN-driven controller to improve landing accuracy. • Fuel optimal trajectory is chosen.	• Overfitting: DNN controllers may perform well on the training data but poorly on new, unseen data.

Table 2.1: Review of existing literature

2.4.2 Gaps Identified

- **Reduced Autonomy:** Factors like environmental restrictions and communication limitations necessitate the need for an immediate decision-making mechanism. Existing systems have reduced autonomy, creating challenges in making timely decisions. Delays in spacecraft maneuvers often result from communication delays with Earth. Therefore, it is vital to implement a system capable of making autonomous decisions in split seconds when necessary.

- **Difficulty Handling Varied Terrains:** Different planets have diverse topographic features and gravitational conditions. Current systems are often unable to adapt to these different terrains and need to be fine-tuned for a particular mission. Hence adaptability remains as a challenge that is yet to be overcome by existing systems.

- **Inefficient Resource Utilization:** Existing systems may employ pre-programmed algorithms that might lead to suboptimal decisions regarding resource utilization, particularly in the case of fuel consumption. In this budget-constrained economy, it is crucial to ensure that we can develop sustainable systems wherever feasible.

- **Safety over Scientific Collection:** Due to the reduced autonomy and adaptability of the current systems, a lot of energy is spent to ensure that the mission makes decisions that prioritize the safety of the spacecraft over scientific data collection.

Chapter 3

Requirements

3.1 Hardware and Software Requirements

3.1.1 Hardware Requirements

Computer vision and deep learning algorithms are computationally intensive and memory-heavy. Suitably powerful hardware is required for running the required software tools for the development and operation of the programs.

- **CPU:** Ryzen 7 or higher processor.
- **GPU:** GTX 1660 (4GB VRAM) or higher graphics card.
- **RAM:** Minimum 16GB of RAM.
- **Storage:** At least 100GB of available storage space. Prefer SSD for speed.

3.1.2 Software Requirements

Tools, frameworks, and modules required for development.

- **Windows 11 OS:** Operating system for running the simulation and software tools.
- **Unity Game Engine:** Platform for the creation and execution of simulation environments and RL agents.
- **ML Agents Toolkit for Unity:** Integrates ML models and algorithms with Unity.
- **QGIS:** GIS software for viewing and operating on DEM/DTM files.
- **Python:** Programming language.
- **Google Colab:** Cloud compute service to develop and execute code.
- **VS Code:** IDE for development.

3.2 Functional Requirements (Numbered List/ Description in Use Case Model)

1. Descent Control:
   - Precise descent control is a vital part of the landing phase, actively managing thrust, orientation, and velocity to ensure a controlled and safe touchdown.
2. Autonomous Navigation:
   - This is required to ensure a system capable of processing real-time sensor data to determine the lander's position and adjust its trajectory accordingly.
3. Terrain Analysis:
   - A terrain analysis function that assesses the lunar surface is essential to avoid obstacles that may be present on the selected trajectory.
4. Landing Site Adaptability:
   - It implements algorithms that allow the lander to adapt its landing site based on real-time data assessments of the lunar surface.
5. Trajectory Adjustment:
   - These adjustments are made during descent in response to dynamic environmental conditions, ensuring optimal fuel usage and hazard avoidance.
6. Fuel Optimization Objective:
   - Define an RL-based fuel optimization objective, where the RL algorithm seeks to maximize fuel efficiency during the descent phase.

These requirements maintain a focus on autonomous control, adaptability, and mission success.

Chapter 4

System Architecture

The introduction to this chapter provides a comprehensive overview of the spacecraft landing automation system, delving into its structure and organization. The system overview offers a high-level perspective, outlining the key components and objectives of the automation system. Following this, the architectural design section delves into the detailed structure and arrangement of these components, providing insights into how they interact to achieve the system's goals. The module division section further dissects the system into distinct modules, elucidating their functionalities and roles within the overall architecture. To provide a roadmap for implementation, the work schedule is presented in the form of a Gantt chart, offering a visual representation of the planned timeline and milestones for the development and deployment of the spacecraft landing automation system. This comprehensive introduction sets the stage for a detailed exploration of each aspect in subsequent sections, providing a clear understanding of the system's design and development.

4.1 System Overview

1. Scanning Module:

The Scanning Module employs simulated LIDAR sensors[4] within the Unity game engine to capture terrain details during the descent, generating a Digital Elevation Model (DEM). To simulate realistic sensor limitations, the module selects a section of the main terrain DEM within the sensor's field of view, introducing noise to account for sensor inaccuracies. The DEM undergoes resizing to a fixed value, simulating changing spatial resolution with altitude. This processed DEM serves as the input for the subsequent Safety Map Generation Module.

2. Safety Map Generation Module:

The Safety Map Generation Module utilizes the processed DEM from the Scanning Module and employs GDAL for segmentation, creating a binarized safety map where white represents safe areas and black signifies unsafe regions. The resulting safety map undergoes dimension reduction through an autoencoder, optimizing it for input into the Reinforcement Learning (RL) model employed in the Landing Site Selection Module.

3. Landing Site Selection Module:

The Landing Site Selection Module integrates the current state of the lander (position, velocity, acceleration, rotation) and the reduced safety map to determine target landing site coordinates. The PPO RL algorithm present in the ML agents toolkit is employed to train the model. This module is pivotal as multiple safe landing regions exist, and the selection is influenced by the lander's trajectory, considering factors such as divert maneuvers and the dynamic changes in safety due to increasing spatial resolution during descent.

4. Control Module:

The Control Module takes the current state of the lander and the target landing site position as inputs, utilizing the PPO algorithm to generate thruster control signals for a safe landing. This iterative process ensures the adaptation of control strategies based on evolving conditions during descent.

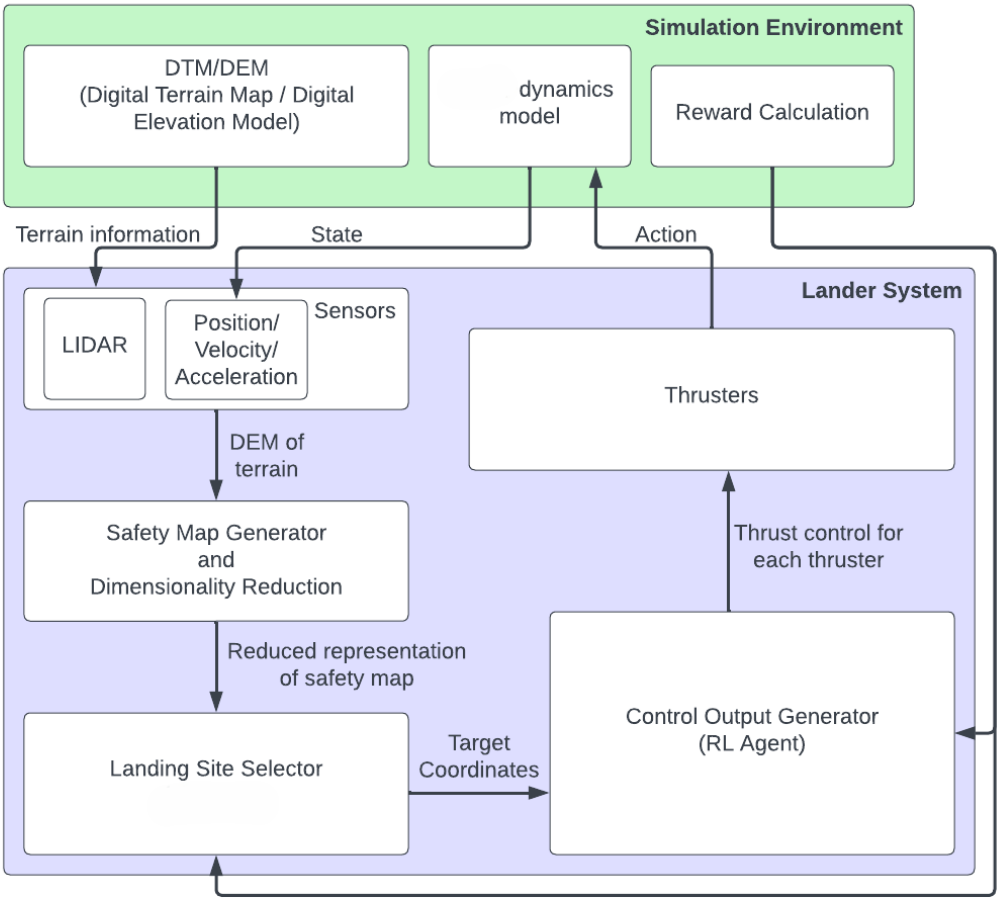


Figure 4.1: System Architecture diagram

4.2 Architectural Design

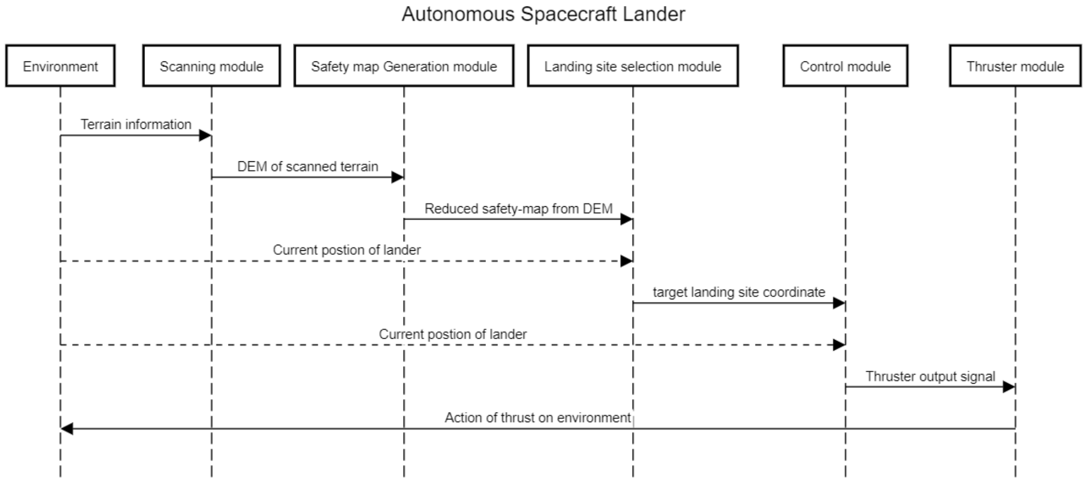


Figure 4.2: Sequence diagram

4.3 Module Division

1. Terrain Scanning Module

- Scans terrain and produces the corresponding DEM.
- To reduce computational complexity, the LIDAR scanning[4] is only simulated. The terrain in the field of view of the scanner will be identified, and the corresponding region will be cut out from the base terrain DEM for use.
- To account for spatial resolution differences at different altitudes, the cropped section will be resized to a fixed size, i.e., that of the sensor array (64 x 64).
- The resized DEM is then passed to the safety map generation module.

2. Safety Map Generation Module

- Using safety parameters from the ALHAT project[5], an algorithm is used to predict safety based on slope and roughness values of the DEM.
- Slope and roughness maps of the DEM are calculated, and pixels falling under the specified threshold values for both slope and roughness are classified as safe.
- The output of the algorithm is a safety map indicating safe and unsafe regions.

3. **Landing Site Selection Module**
   - Takes the current safety map and selects landing site coordinates centered on the largest safe region.
4. **Control Module**
   - Takes sensor readings of the lander and deviation from the landing site as input and generates the required thruster controls
   - Uses (PPO) RL Algorithm.

4.3.1 Module Division among Group Members

- **Rohan:**
  - Terrain Scanning
  - Safety map generation
- **Sebastian:**
  - Terrain scanning
  - Control module
- **Sneha:**
  - Landing site selection module
  - Control module
- **Varun:**
  - Terrain Scanning
  - Safety map generation
  - Landing site selection module
  - Control module

4.4 Work Schedule - Gantt Chart

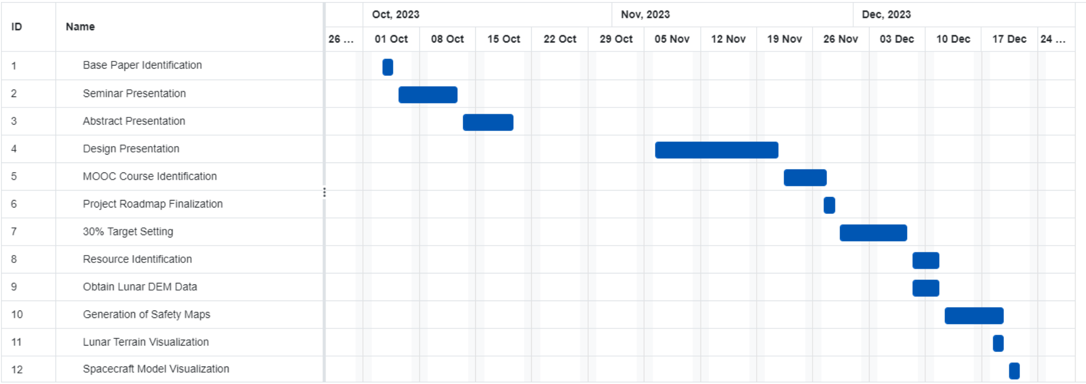


Figure 4.3: Phase 1 Gantt chart

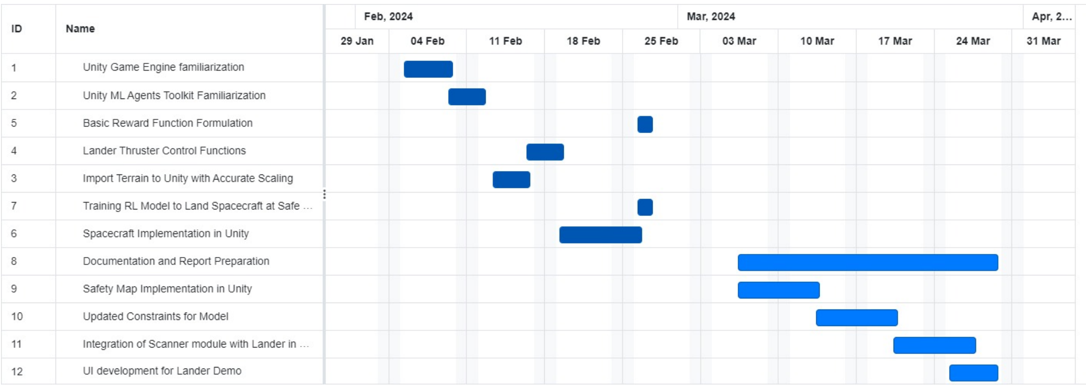


Figure 4.4: Phase 2 Gantt Chart

In conclusion, this chapter has provided an insight into the spacecraft landing automation system, beginning with a comprehensive system overview. The architectural design section provided a comprehensive understanding of the system's structure and the interrelated functionalities of its components. The module division further refined our understanding by breaking down the system into individual modules, highlighting their specific roles and contributions. To guide the implementation journey, the work schedule in the form of a Gantt chart has been presented, offering a visual representation of planned activities and milestones. As we proceed through the subsequent sections, a deeper exploration of each element will unveil the intricacies of the spacecraft landing automation system, ultimately contributing to a comprehensive understanding of its design, development, and anticipated outcomes.

Chapter 5

System Implementation

This chapter details the implementation of our lunar terrain scanning and spacecraft control system. We begin by discussing the datasets utilized, specifically focusing on the lunar surface data obtained from USGS. Subsequently, we outline the proposed methodology and algorithms employed for terrain scanning, safety map generation, target site selection, and system control. Additionally, we provide insights into the implementation strategies adopted, including the utilization of software libraries and frameworks to facilitate the development process.

5.1 Datasets Identified

We have sourced our lunar surface data from USGS Astrogeology Science Center (DEM source). The acquired lunar surface data is a shape-from-shading (SfS or photoclinometry) DEM for the Haworth area. The Haworth area is beyond the north rim of the Haworth crater and centered at 338.0 (-21.0) E, -86.8 S on the Moon in Float32 image format with a .tif file extension. The DEM accurately depicts the Haworth area at a spatial resolution of 1m per pixel.

5.2 Proposed Methodology/Algorithms

5.2.1 Terrain Scanning

The terrain scanning process employs LiDAR as the lander descends. In our implementation, to mitigate computational costs, we simulate this process by cropping the section of the Digital Elevation Model (DEM) corresponding to the lander's field of view (FOV). This section is then resized to a fixed dimension of 64x64, utilizing nearest neighbor interpolation to accommodate variations in spatial resolution with altitude.

Algorithm 1 Terrain Scanning

Input: LanderPosition (coordinates), Altitude (height above ground), BaseDEM (2000x2000 matrix)

Output: ScannedDEM (64x64 matrix)

$Range \leftarrow 2 \times \tan(15°) \times Altitude$ ▷ Calculate the range of LiDAR view

$CroppedDEM \leftarrow$ extract_subregion$(BaseDEM, LanderPosition, Range, Range)$ ▷ Crop DEM within LiDAR's view

$ResizedDEM \leftarrow$ resize_image$(CroppedDEM, 64, 64)$ ▷ Resize cropped DEM

Return $ResizedDEM$

5.2.2 Safety Map Generation

The Scanned DEM serves as the basis for generating a safety map of the area. Through the DEM, the slope and roughness of each terrain point are computed, and based on predefined threshold values, each point is classified into either safe or unsafe regions. Additionally, the safety map undergoes a morphological opening operation to eliminate very small safe sites and ensure a minimum distance from unsafe regions.

Algorithm 2 Safety Map Generation

Input: DEM (64x64 matrix)

Output: SafetyMap (64x64 matrix)

$GradientMap \leftarrow$ compute_gradient_map(DEM) ▷ Compute gradient map

$SlopeMap \leftarrow$ calculate_slope$(GradientMap)$ ▷ Calculate slope map

$RoughnessMap \leftarrow$ perform_convolution$(DEM, 3 \times 3$ kernel$)$ ▷ Compute roughness map

$SafetyMap \leftarrow$ create_empty_array$(64, 64)$ ▷ Initialize safety map

for each pixel in $SafetyMap$ **do**

    if $SlopeMap[pixel] < 5°$ and $RoughnessMap[pixel] <$ SpatialResolution **then** ▷ Check slope and roughness

        $SafetyMap[pixel] \leftarrow 1$ ▷ Set as safe

    else

        $SafetyMap[pixel] \leftarrow 0$ ▷ Set as unsafe

Return $SafetyMap$

5.2.3 Target Site Selection

For landing site selection, the algorithm identifies safe regions from the safety map. The centroid of the largest safe region is then chosen as the landing site. Selecting the largest area is advantageous as even small deviations from the target would still fall within the safe region, ensuring a safe landing.

Algorithm 3 Target Site Selection

Input: SafetyMap (64x64 matrix), LanderPosition (coordinates)

Output: TargetSite (coordinates), Deviation (distance)

1: $SafeRegions \leftarrow$ find_connected_components($SafetyMap$) ▷ Identify connected safe regions

2: $LargestSafeRegion \leftarrow$ find_largest_region($SafeRegions$) ▷ Select largest safe region

3: $Centroid \leftarrow$ compute_centroid($LargestSafeRegion$) ▷ Compute centroid of largest safe region

4: $Deviation \leftarrow$ distance($LanderPosition, TargetSite$) ▷ Calculate deviation from target site

5: **Return** $Centroid$, $Deviation$

5.2.4 System Control

The system control is governed by a Reinforcement Learning (RL) agent, which is trained to make decisions based on the altitude, deviation from the target landing site, velocity, attitude (tilt), and angular velocity along each axis. These parameters serve as inputs to the RL agent, enabling it to learn the optimal control policies. The lander has three possible output signals, which include firing the main engine, pitch control Reaction Control System (RCS) thrusters, and yaw control RCS thrusters.

To facilitate training, various reward functions are implemented to guide the learning process. These include functions for vertical velocity tracking, attitude tracking, and target tracking based on velocity. Additionally, a penalty for fuel consumption is enforced to encourage fuel-optimal descent strategies. By iteratively adjusting these reward functions, the RL agent learns to navigate the descent process efficiently and effectively.

Vertical Reference Velocity:

$f(x) = 0.3 - 0.5(x+1)^{0.5} \quad \text{for } 0 \leq x \leq 1000$

(where x represents altitude)

Vertical Velocity Tracking Reward:

$$f(x) = \begin{cases} 1 - |x| & \text{for } 0 < |x| < 1 \\ \frac{1-|x|}{10} & \text{otherwise} \end{cases}$$

(where x represents deviation from reference velocity)

Attitude Tracking Reward:

$$f(x) = -2\left|\tanh\left(\frac{x}{10}\right)\right| + 1$$

(where x represents angle from vertical)

Target Tracking Reference Velocity:

$$f(x) = \begin{cases} -0.45|x|^{0.45} & \text{for } 0 < x < 250 \\ 0.45|x|^{0.45} & \text{otherwise} \end{cases}$$

(where x represents distance from target)

Target Velocity Tracking Reward:

$$f(x) = \begin{cases} 1 - |x| & \text{for } 0 < |x| < 1 \\ \frac{1-|x|}{5} & \text{otherwise} \end{cases}$$

(where x represents deviation from reference velocity)

Target Deviation Distance Reward:

$$f(x) = \begin{cases} 1 - 0.2|x| & \text{for } 0 < |x| < 5 \\ \frac{1-0.2|x|}{25} & \text{otherwise} \end{cases}$$

(where x represents distance from target)

5.3 Description of Implementation Strategies

- **Safety Map Calculation:** We employed the Geospatial Data Abstraction Library (GDAL), a robust software library designed for reading and writing various geospatial data formats, to compute the safety map from the terrain data.
- **Reinforcement Learning (RL) Algorithm:** To implement the RL algorithm, specifically Proximal Policy Optimization (PPO), we utilized The Unity Machine Learning Agents Toolkit (ML-Agents). This open-source toolkit allows us to train intelligent agents within game environments.
- **Simulation Environment:** Our simulation environment, including both the terrain and spacecraft visualization, was developed using the Unity Game Engine. Unity's built-in physics engine effectively handles the dynamics of the simulation. We imported a Digital Elevation Model (DEM) into Unity to simulate the lunar terrain, adjusting scaling as needed. The spacecraft model, based on the Apollo XI lander, was created using freely available assets. Within this environment, we integrated the PPO algorithm from the ML agents toolkit to train the spacecraft for autonomous landing.

In conclusion, this chapter has provided a comprehensive overview of our system implementation for lunar terrain scanning and spacecraft control. We began by discussing the datasets sourced from USGS, particularly the detailed DEMs that formed the basis of our operations. We detailed the methodologies and algorithms we developed for terrain scanning, safety map generation, target site selection, and system control, outlining the step-by-step processes involved in each aspect of our mission. Finally, we described the practical implementation strategies we employed, such as utilizing tools like GDAL and ML-Agents

Chapter 6

Results and Discussions

In this section, we present the results and discussions of our study on the performance of the proposed spacecraft landing control system. We examine the accuracy, training progress, convergence rates based on different reward functions, performance metrics such as target deviation over time, and the effectiveness of safety map generation. These results will provide insights into the effectiveness and robustness of the developed system.

6.1 Overview

The trained RL control agent was deployed into velocity and target tracking simulations. Over 3883 testing episodes, the agent has achieved a success rate of 90.11% at landing safely at the selected target landing site, and 98.66% if the target site is not considered a requirement for success.

Metric	**Success rate (%)**	**Number of episodes**
Safe landing at Target Landing site	90.11%	3499
Safe Landing Without Target Tracking	98.66%	332 + 3499

Table 6.1: Success rate

6.2 Safety Map Generation

The terrain is scanned regularly as the lander descends, each scan results in a DEM that is processed to produce a safety map of the region.

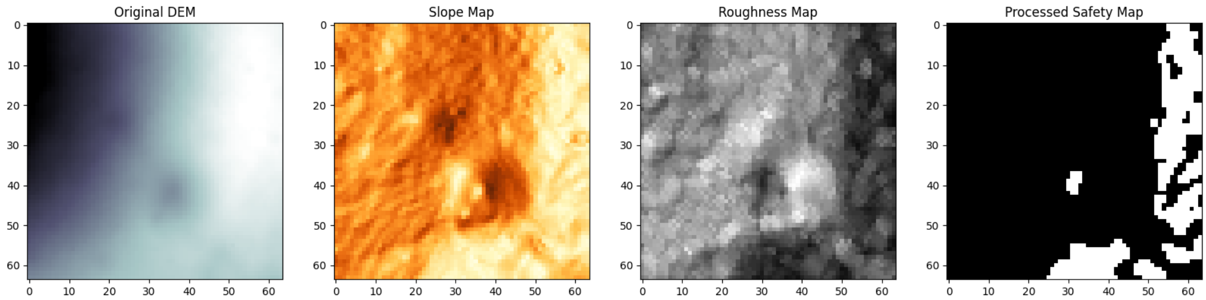


Figure 6.1: Safety map generation process showing intermediate outputs

6.3 Training

The RL control agent underwent training in simulation utilizing the PPO algorithm. The Mean Cumulative Reward graph is presented, illustrating the training progress and providing insights into the convergence properties of the agent.

The training was conducted in phases of 3 Million timesteps, each with increasing difficulty

In phase 1 the starting altitude was set to 200m and the target was given a max distance from start as 0m. This was to ensure the model learns to perform safe landing.

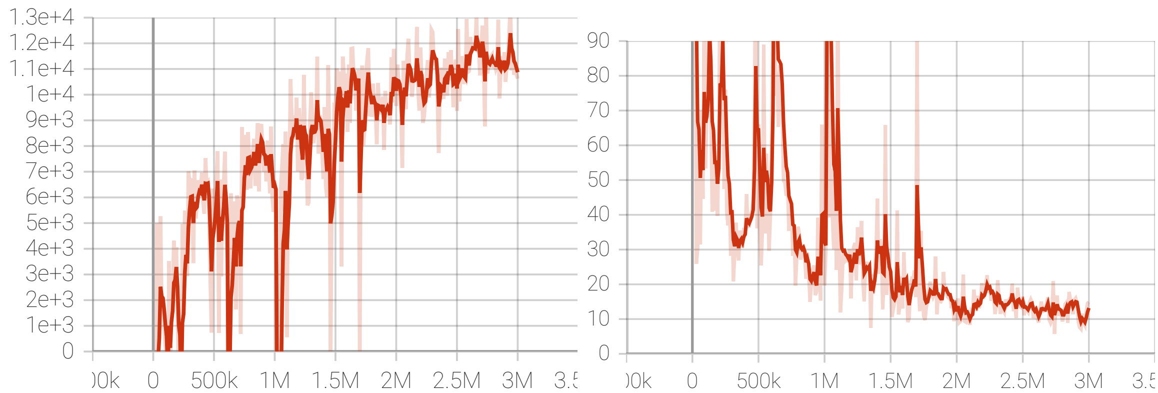


Figure 6.2: Phase 1 reward(left) and mean target deviation(right)

The training was initially unstable but has converged.

In phase 2 the starting altitude was set to 600m and the target was given a max distance from start as 30m. Here the target tracking task was made more challenging.

The reward initially decreased as the model had overfit slightly to the phase 1 starting conditions. But over time the the model has learnt to generalize and has converged

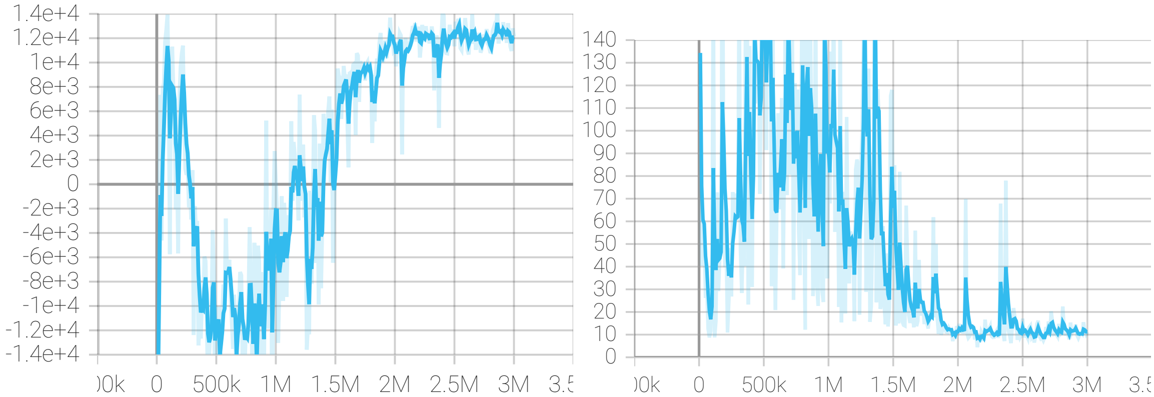


Figure 6.3: Phase 2 reward(left) and mean target deviation(right)

6.4 Impact of Reward Functions on Convergence

By comparing the convergence rate when trained with different reward functions, such as a deviation-based penalty for target tracking or a velocity-based reward for target tracking, fuel usage penalty, and handling of malfunction cases, we observe the impact of reward function selection on the learning process and overall performance.

6.4.1 Target Tracking

Two functions were considered as rewards for target tracking.

1. Deviation-based penalty: Reward is given based on current deviation from the target landing site
2. Velocity-based reward: Reward is given based on deviation from reference horizontal velocity required to track target landing site.

We compare convergence with the cumulative reward using each function

6.4.2 Fuel usage penalty

To ensure fuel-optimal descent, the agent was penalized for using thrusters.

The agent converged faster and had better performance with a fuel usage penalty. The fuel usage penalty ensured that the thrusters were not fired in a suboptimal state which could lead to further deviation from the target. This leads to less random movement allowing the lander to maintain ideal states from the start, allowing training to converge quickly.

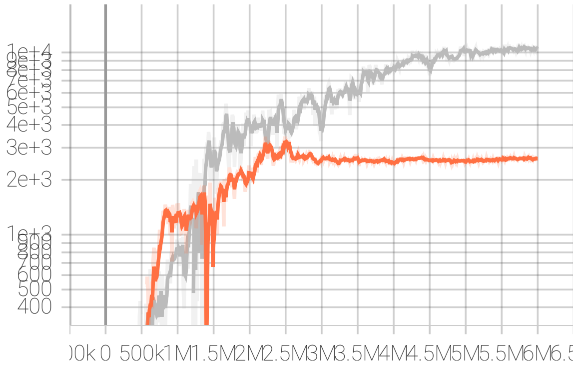


Figure 6.4: Deviation-based(orange) Velocity-based(grey)

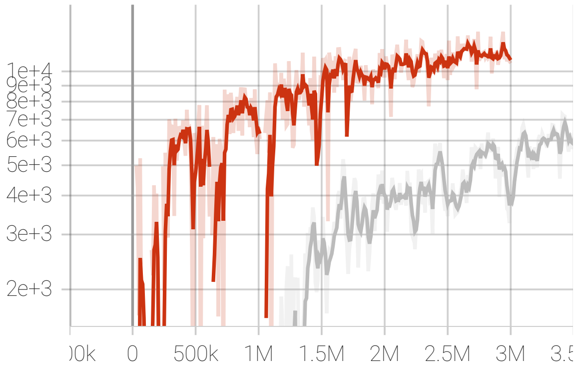


Figure 6.5: No Fuel usage penalty(grey) Training with Fuel usage penalty(red)

6.5 Malfunction Cases

In order for the RL agent to handle varying situations, it must also be robust against variations in lander performance. To test the ability of the agent to tolerate malfunctions. A test case was performed with each activation of the RCS thrusters having a 20$ chance to fail. The trained agent was further improved under this condition for 3 Million time steps.

The agent had no difficulty adapting to the new conditions maintaining consistent performance with very low instability in training performance.

In this section, we presented the results and discussions of our study on the performance of the proposed spacecraft landing control system. We examined various aspects including accuracy, training progress, convergence rates based on different reward functions, performance metrics such as target deviation over time, and the effectiveness of safety map generation. The success rate of the system in safely landing at the selected target landing site was found to be 90.11%, with an overall success rate of 98.66% without

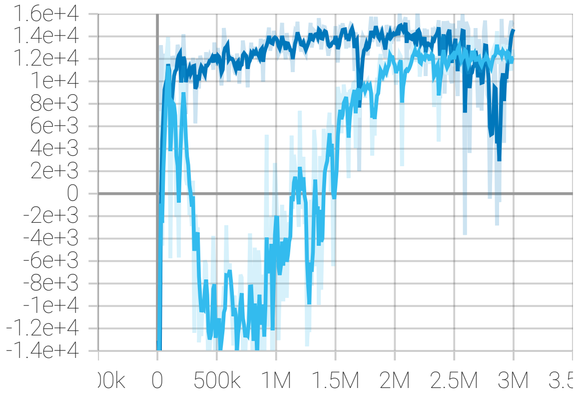


Figure 6.6: training under no malfunction(light blue) Training with possibility of malfunction(dark blue)

considering the target landing site. The training progress was depicted through Mean Cumulative Reward and Mean Target Deviation graphs, showing convergence properties over different training phases. Moreover, the impact of reward functions on convergence was analyzed, highlighting the effectiveness of a fuel usage penalty in ensuring a faster convergence rate. The system also demonstrated robustness against malfunctions, maintaining consistent performance even under adverse conditions. These findings underscore the effectiveness and robustness of the developed spacecraft landing control system, offering promising prospects for autonomous landing in planetary exploration missions.

Chapter 7

Conclusions & Future Scope

A method that leverages Reinforcement Learning, to analyze and identify safe landing zones based on terrain in view and current trajectory with considerations for divert maneuvers, thus providing adaptability to changing situations is proposed. A reinforcement learning agent is employed to generate accurate and adaptive thruster control to maintain target velocity, minimize fuel consumption and perform safe and accurate soft landing at the selected site.

The future scope of the spacecraft landing automation project includes advancing adaptability to diverse planetary environments, incorporating cutting-edge sensing technologies for improved perception, and implementing machine learning for anomaly detection. Further developments may explore multi-agent systems for collaborative missions, enabling coordinated efforts between multiple spacecraft. Real-time learning capabilities could be integrated to enhance decision-making on-the-fly, while the project's potential integration with human space exploration activities presents exciting possibilities. Additionally, the project could contribute to the exploration of spacecraft swarms, fostering collaborative and distributed sensing missions. Addressing hardware failures, encouraging global collaborations, and promoting public engagement and education emerge as essential facets for the project's continued evolution and impact.

References

[1] X. Xu, Y. Chen, and C. Bai, "Deep reinforcement learning-based accurate control of planetary soft landing," *Sensors*, vol. 21, no. 23, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/23/8161

[2] K. Iiyama, K. Tomita, B. A. Jagatia, T. Nakagawa, and K. Ho, "Deep reinforcement learning for safe landing site selection with concurrent consideration of divert maneuvers," *arXiv preprint arXiv:2102.12432*, 2021.

[3] L. Cheng, Z. Wang, and F. Jiang, "Real-time control for fuel-optimal moon landing based on an interactive deep reinforcement learning algorithm," *Astrodynamics*, vol. 3, pp. 375 – 386, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:199009111

[4] J. A. Keim, S. Mobasser, D. Kuang, Y. Cheng, T. Ivanov, A. E. Johnson, H. R. Goldberg, G. Khanoyan, and D. B. Natzic, "Field test implementation to evaluate a flash lidar as a primary sensor for safe lunar landing," in *2010 IEEE Aerospace Conference*, 2010, pp. 1–14.

[5] T. Ivanov, A. Huertas, and J. M. Carson, "Probabilistic hazard detection for autonomous safe landing," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013, pp. 1–12.

Appendix A: Presentation

Automation of a Lunar Lander using RL

100% code evaluation

Rohan Ranjith, Sebastian K Suresh, Sneha Sarah John, Varun Pradeep

April 15, 2024

Overview

Problem Definition

To create a system that is tasked with autonomously and adaptively guiding lunar landers to perform safe, precise, and efficient soft landings on the lunar surface.

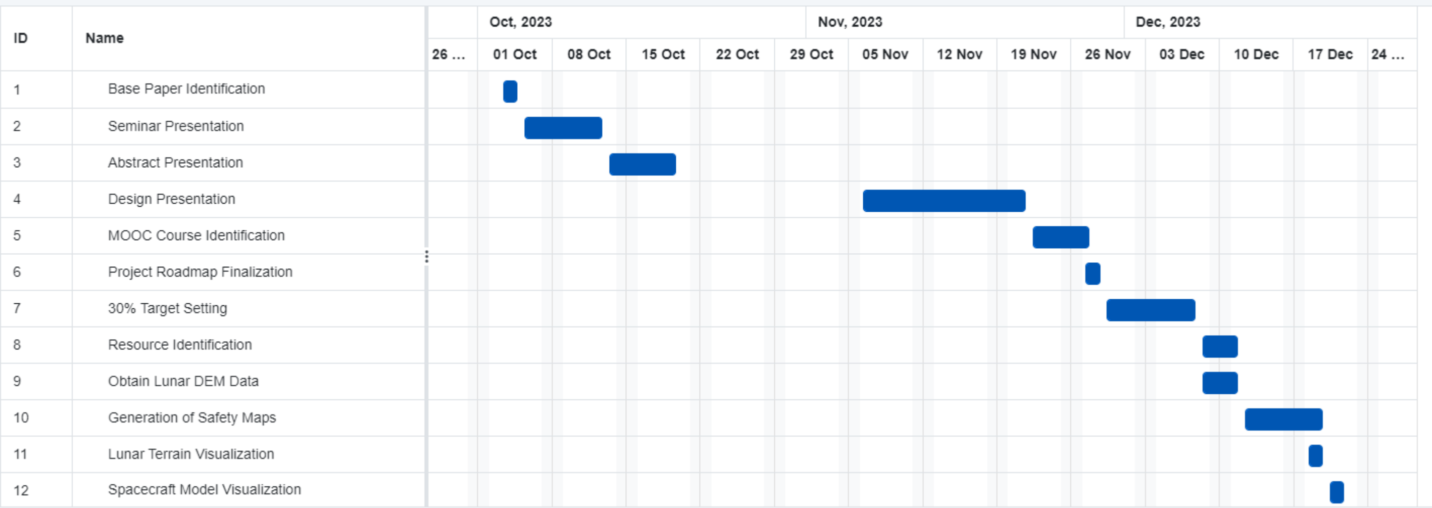
Project Objectives

- Hazard detection and finding the optimal landing spot
- Precise navigation to landing spot
- Optimizing fuel consumption

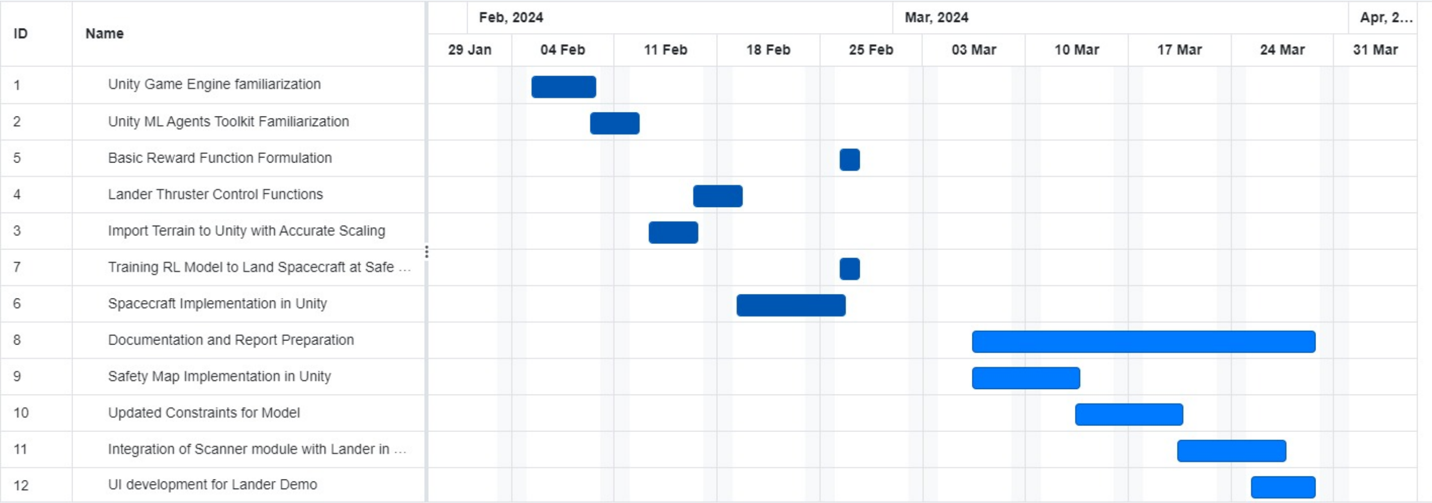
Novelty of Idea and Scope of Implementation

- Novelty:
  1. The ability of the algorithm to adapt to new terrains.
  2. Ability of algorithm to compensate for minor technical malfunctions and unavailable data.
  3. Added benefit of fuel optimization when the best possible choice for landing site is made.
- Scope of implementation:
  1. Encourages more autonomy and adatability in space tech.
  2. Expansion of RL into terrain navigation by the rover.
  3. Educational tool as it utilizes a simulation.

Project Gantt Chart-Phase 1



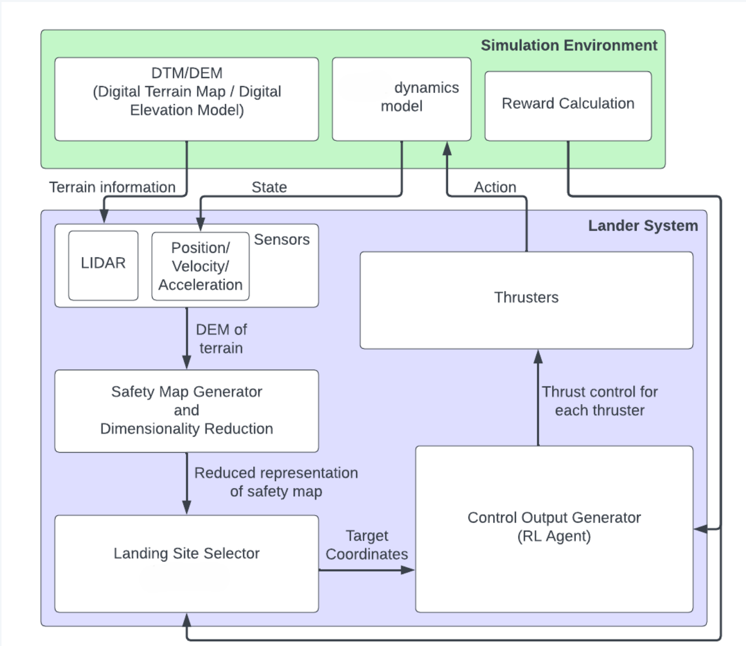
Project Gantt Chart-Phase 2



System Module

1. Terrain scanning
2. Safety map creation
3. Target site selection
4. Control module

Architecture Diagram



Work Done (30% Evaluation)

Work done (30% evaluation)

1. Obtain Lunar DEM source data
   - Obtained DEM from the LRO mission present in NASA's website
2. Generation of safety maps
   - Used the formulae in GDAL and implemented an algorithm to create slope and roughness maps which in turn is used to produce safety maps
3. Lunar Terrain Visualization
   - Taking source DEM and try to visualize
4. Spacecraft model visualization
   - Used the assets of the lander present on the Apollo 11 spacecraft

Work Done (60% Evaluation)

Work done (60% evaluation)

1. Thruster control functions
   - Activation and deactivation of main thruster. (45000N)
   - Activation and deactivation of pitch and yaw control thrusters. (450N)
2. Basic Reward function formulation
   - Reward function for velocity tracking. (thresholded)
   - Reward function for attitude tracking. (with multiple thresholds)
3. Training the RL model to land the spacecraft to a safe site.
   - Training model using PPO algorithm using ML Agents toolkit.
   - Trained for 25,000,000 time steps for an increase in reward from 300 to 1300.
4. Imported terrain to Unity with accurate scaling.

Work Done (100% Evaluation)

Work done (100% evaluation)

1. Incorporating the safety map generation module into unity
   - Set up Python-Unity interprocess communication.
   - Communicate FoV of lander (coordinates) to the python program.
   - Perform processing and send the image data of the safety map to Unity.
2. Lander improvements to be made
   - Refine reward functions for better training/faster convergence (removes the threshold functions and make them continuous)
   - Add more sensors with respect to knowledge of current mass and to detect the tilt of the terrain

Results

Interim results (30%)

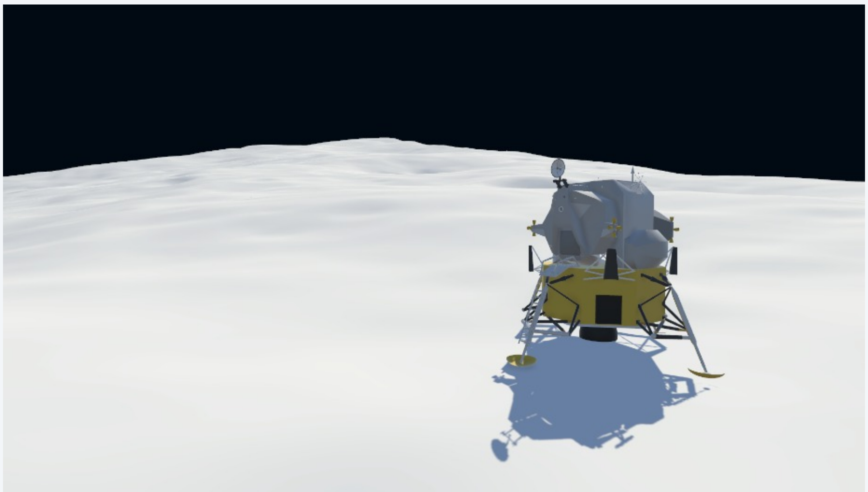


Figure: Lander and Terrain in Unity

Interim results (30%)

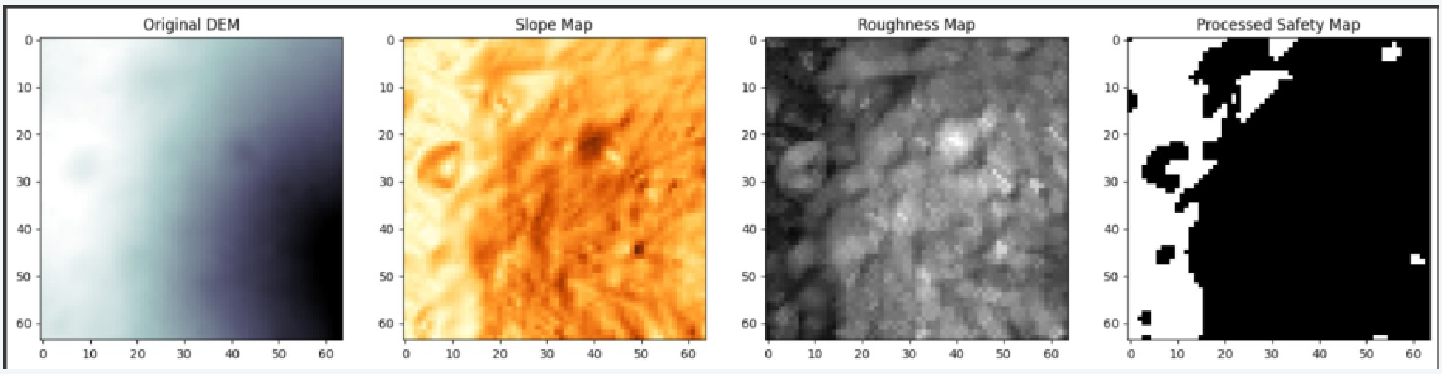
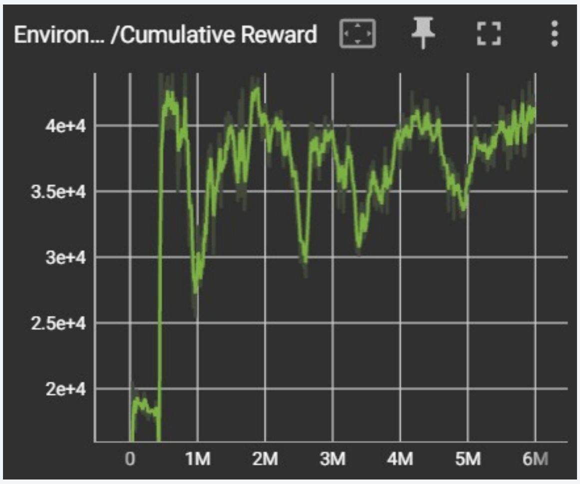


Figure: Safety map generation process

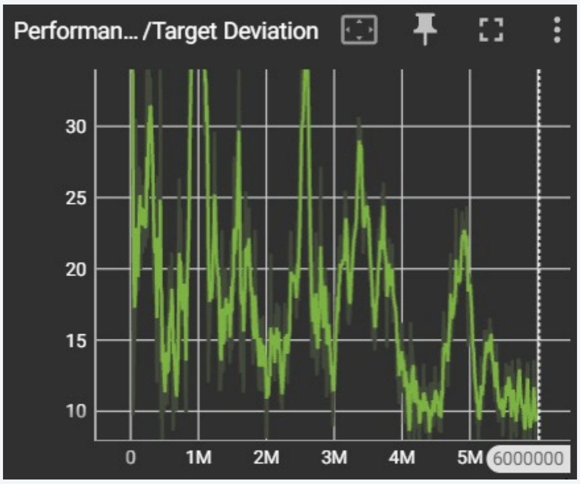
Interim results (60%)

Figure: RL model Training Reward

Results(100%)



Results(100%)



Future Scope

Future Scope

- Better thruster layout
- Include training on more planetary terrain
- Include more cases of malfunction
- Expand the model (Include factors such as fuel)

Task Distribution

Task Distribution

Varun
- Generation of Safety Map
- Lander Model Training
- Python to Unity connection

Rohan
- Terrain data import and scaling
- Field of View Calculation
- Python to Unity connection

Sneha
- Spacecraft model visualization
- Lander model controls
- Reward function formulation

Sebastian
- Terrain conversion
- Terrain data import and scaling
- Reward function formulation

Conclusion

In summary, we propose a method that leverages computer vision and reinforcement learning to analyze and identify safe landing zones based on terrain in view and current trajectory with considerations for divert maneuvers, thus providing adaptability to changing situations. A reinforcement learning agent is employed to generate accurate and adaptive thruster control to maintain target velocity, minimize fuel consumption and perform safe and accurate soft landing at the selected site.

References

References

- Deep Reinforcement Learning for Safe Landing Site Selection with Concurrent Consideration of Divert Maneuvers
- Deep Reinforcement Learning-Based Accurate Control of Planetary Soft Landing

Status of Paper Publication

Status of Paper Publication

- Planning to submit the paper in the Control Engineering Practice Journal under 'New trends in Spacecraft and Launch Vehicles Control'
- Submission Deadline: 31st May 2024

Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes

Vision, Mission, Programme Outcomes and Course Outcomes

Institute Vision

To evolve into a premier technological institution, moulding eminent professionals with creative minds, innovative ideas and sound practical skill, and to shape a future where technology works for the enrichment of mankind.

Institute Mission

To impart state-of-the-art knowledge to individuals in various technological disciplines and to inculcate in them a high degree of social consciousness and human values, thereby enabling them to face the challenges of life with courage and conviction.

Department Vision

To become a centre of excellence in Computer Science and Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

Department Mission

To inspire and nurture students, with up-to-date knowledge in Computer Science and Engineering, ethics, team spirit, leadership abilities, innovation and creativity to come out with solutions meeting societal needs.

Programme Outcomes (PO)

Engineering Graduates will be able to:

1. **Engineering Knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and Team work: Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

10. Communication: Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Programme Specific Outcomes (PSO)

A graduate of the Computer Science and Engineering Program will demonstrate:

PSO1: Computer Science Specific Skills
The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

PSO2: Programming and Software Development Skills
The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

PSO3: Professional Skills
The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur.

Course Outcomes (CO)

Course Outcome 1: Model and solve real world problems by applying knowledge across domains (Cognitive knowledge level: Apply).

Course Outcome 2: Develop products, processes or technologies for sustainable and socially relevant applications (Cognitive knowledge level: Apply).

Course Outcome 3: Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks (Cognitive knowledge level: Apply).

Course Outcome 4: Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms (Cognitive knowledge level: Apply).

Course Outcome 5: Identify technology/research gaps and propose innovative/creative solutions (Cognitive knowledge level: Analyze).

Course Outcome 6: Organize and communicate technical and scientific findings effectively in written and oral forms (Cognitive knowledge level: Apply).

Appendix C: CO-PO-PSO Mapping

CO-PO AND CO-PSO MAPPING

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO 1	2	2	2	1	2	2	2	1	1	1	1	2	3		
CO 2	2	2	2		1	3	3	1	1		1	1		2	
CO 3									3	2	2	1			3
CO 4					2			3	2	2	3	2			3
CO 5	2	3	3	1	2							1	3		
CO 6					2			2	2	3	1	1			3

3/2/1: high/medium/low

JUSTIFICATIONS FOR CO-PO MAPPING

MAPPING	LOW/MEDIUM/ HIGH	JUSTIFICATION
100003/ CS722U.1-PO1	M	Knowledge in the area of technology for project development using various tools results in better modeling.
100003/ CS722U.1-PO2	M	Knowledge acquired in the selected area of project development can be used to identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions.

100003/ CS722U.1-PO3	M	Can use the acquired knowledge in designing solutions to complex problems.
100003/ CS722U.1-PO4	M	Can use the acquired knowledge in designing solutions to complex problems.
100003/ CS722U.1-PO5	H	Students are able to interpret, improve and redefine technical aspects for design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
100003/ CS722U.1-PO6	M	Students are able to interpret, improve and redefine technical aspects by applying contextual knowledge to assess societal, health and consequential responsibilities relevant to professional engineering practices.
100003/ CS722U.1-PO7	M	Project development based on societal and environmental context solution identification is the need for sustainable development.
100003/ CS722U.1-PO8	L	Project development should be based on professional ethics and responsibilities.
100003/ CS722U.1-PO9	L	Project development using a systematic approach based on well defined principles will result in teamwork.
100003/ CS722U.1-PO10	M	Project brings technological changes in society.

100003/ CS722U.1-PO11	H	Acquiring knowledge for project development gathers skills in design, analysis, development and implementation of algorithms.
100003/ CS722U.1-PO12	H	Knowledge for project development contributes engineering skills in computing & information gatherings.
100003/ CS722U.2-PO1	H	Knowledge acquired for project development will also include systematic planning, developing, testing and implementation in computer science solutions in various domains.
100003/ CS722U.2-PO2	H	Project design and development using a systematic approach brings knowledge in mathematics and engineering fundamentals.
100003/ CS722U.2-PO3	H	Identifying, formulating and analyzing the project results in a systematic approach.
100003/ CS722U.2-PO5	H	Systematic approach is the tip for solving complex problems in various domains.
100003/ CS722U.2-PO6	H	Systematic approach in the technical and design aspects provide valid conclusions.

100003/ CS722U.2-PO7	H	Systematic approach in the technical and design aspects demonstrate the knowledge of sustainable development.
100003/ CS722U.2-PO8	M	Identification and justification of technical aspects of project development demonstrates the need for sustainable development.
100003/ CS722U.2-PO9	H	Apply professional ethics and responsibilities in engineering practice of development.
100003/ CS722U.2-PO11	H	Systematic approach also includes effective reporting and documentation which gives clear instructions.
100003/ CS722U.2-PO12	M	Project development using a systematic approach based on well defined principles will result in better teamwork.
100003/ CS722U.3-PO9	H	Project development as a team brings the ability to engage in independent and lifelong learning.
100003/ CS722U.3-PO10	H	Identification, formulation and justification in technical aspects will be based on acquiring skills in design and development of algorithms.
100003/ CS722U.3-PO11	H	Identification, formulation and justification in technical aspects provides the betterment of life in various domains.

100003/ CS722U.3-PO12	H	Students are able to interpret, improve and redefine technical aspects with mathematics, science and engineering fundamentals for the solutions of complex problems.
100003/ CS722U.4-PO5	H	Students are able to interpret, improve and redefine technical aspects with identification formulation and analysis of complex problems.
100003/ CS722U.4-PO8	H	Students are able to interpret, improve and redefine technical aspects to meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
100003/ CS722U.4-PO9	H	Students are able to interpret, improve and redefine technical aspects for design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
100003/ CS722U.4-PO10	H	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools for better products.
100003/ CS722U.4-PO11	M	Students are able to interpret, improve and redefine technical aspects by applying contextual knowledge to assess societal, health and consequential responsibilities relevant to professional engineering practices.

100003/ CS722U.4-PO12	H	Students are able to interpret, improve and redefine technical aspects for demonstrating the knowledge of, and need for sustainable development.
100003/ CS722U.5-PO1	H	Students are able to interpret, improve and redefine technical aspects, apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
100003/ CS722U.5-PO2	M	Students are able to interpret, improve and redefine technical aspects, communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
100003/ CS722U.5-PO3	H	Students are able to interpret, improve and redefine technical aspects to demonstrate knowledge and understanding of the engineering and management principle in multidisciplinary environments.
100003/ CS722U.5-PO4	H	Students are able to interpret, improve and redefine technical aspects, recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.
100003/ CS722U.5-PO5	M	Students are able to interpret, improve and redefine technical aspects in acquiring skills to design, analyze

		and develop algorithms and implement those using high-level programming languages.
100003/ CS722U.5-PO12	M	Students are able to interpret, improve and redefine technical aspects and contribute their engineering skills in computing and information engineering domains like network design and administration, database design and knowledge engineering.
100003/ CS722U.6-PO5	M	Students are able to interpret, improve and redefine technical aspects and develop strong skills in systematic planning, developing, testing, implementing and providing IT solutions for different domains which helps in the betterment of life.
100003/ CS722U.6-PO8	H	Students will be able to associate with a team as an effective team player for the development of technical projects by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
100003/ CS722U.6-PO9	H	Students will be able to associate with a team as an effective team player to Identify, formulate, review research literature, and analyze complex engineering problems
100003/ CS722U.6-PO10	M	Students will be able to associate with a team as an effective team player for designing solutions to complex engineering problems and design system components.

100003/ CS722U.6-PO11	M	Students will be able to associate with a team as an effective team player, use research-based knowledge and research methods including design of experiments, analysis and interpretation of data.
100003/ CS722U.6-PO12	H	Students will be able to associate with a team as an effective team player, applying ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
100003/ CS722U.1-PSO1	H	Students are able to develop Computer Science Specific Skills by modeling and solving problems.
100003/ CS722U.2-PSO2	M	Developing products, processes or technologies for sustainable and socially relevant applications can promote Programming and Software Development Skills.
100003/ CS722U.3-PSO3	H	Working in a team can result in the effective development of Professional Skills.
100003/ CS722U.4-PSO3	H	Planning and scheduling can result in the effective development of Professional Skills.
100003/ CS722U.5-PSO1	H	Students are able to develop Computer Science Specific Skills by creating innovative solutions to problems.
100003/ CS722U.6-PSO3	H	Organizing and communicating technical and scientific findings can help in the effective development of Professional Skills.

CO - PO Mapping

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
1												
2												
3												
4												
5												

CO - PSO Mapping

CO	PSO 1	PSO 2	PSO 3
1			
2			
3			
4			
5			

Justification

Mapping	Justification
CO1 - PO1	Reason
CO2 - PO2	Reason