International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Comprehensive Survey on Agent Based Deep Learning Techniques for Space Landing Missions

Utkarsh R. Moholkar*[1], Dipti D. Patil[2]

Submitted: 12/12/2023 **Revised**: 16/01/2024 **Accepted**: 01/02/2024

Abstract: Spacecraft landing is a complex and challenging task that requires precise control and decision making. In recent years, reinforcement learning (RL) has emerged as a promising approach for spacecraft landing, enabling autonomous and adaptive control strategies. This literature survey paper presents an overview of the existing research on spacecraft landing using RL. We examine various RL algorithms, simulation environments, and evaluation metrics employed in this domain. Furthermore, we discuss the challenges, limitations, and future directions for applying RL to spacecraft landing. This survey aims to provide researchers and practitioners with a comprehensive understanding of the current state-of-the-art in this field and inspire further advancements in spacecraft landing using RL.

Keywords: *Reinforcement Learning (RL), Digital Terrain Model (DTM), Lunar Reconnaissance Orbiter (LROC), Deep Learning (DL), Deep Reinforcement Learning (DRL), Deep Q-Network (DQN), Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C), Deep Deterministic Policy Gradient (DDPG), Trust Region Policy Optimization (TRPO)*

1. Introduction

Spacecraft landing refers to the controlled descent[43] and touchdown of a spacecraft on a planetary surface or other celestial bodies. It is a critical phase of a mission and requires precise control[45] and decision-making to ensure a safe and successful landing. Spacecraft landing poses several unique challenges due to the harsh and dynamic nature of the space environment, as well as the complexities involved in navigating and controlling a vehicle in such conditions. One of the primary challenges in spacecraft landing is the uncertainty and variability of the landing site. Each planetary body has its own specific characteristics, such as surface topography, composition, and atmospheric conditions. These factors can significantly impact the landing process, making it difficult to predict and plan the landing trajectory accurately. Additionally, the presence of hazards such as craters, boulders, slopes, and rough terrain further complicates the landing process.[6]

Another challenge is the limited availability of real-time information during the descent and landing phase. Due to communication delays between the spacecraft and the mission control center on Earth, spacecraft often must rely on onboard sensors and systems for navigation and control. These sensors provide incomplete and delayed information, making it necessary for the spacecraft to possess autonomous decision-making capabilities. Furthermore, spacecraft landing requires precise control of various vehicle parameters, such as velocity, attitude, and descent rate. Achieving the desired landing conditions while considering the limitations of the spacecraft's propulsion system and control mechanisms is a challenging task.[44] Factors like fuel consumption and structural integrity must be carefully managed to ensure a safe landing. Traditional control approaches for spacecraft landing typically rely on pre-defined mathematical models and control algorithms. These approaches often struggle to adapt to the uncertainties and dynamic nature of the landing environment [7]. This is where reinforcement learning (RL) has gained prominence as a potential solution. RL allows spacecraft to learn and adapt their landing strategies based on interactions with the environment, making it a promising approach for autonomous and adaptive spacecraft landing. By using RL, spacecraft can learn optimal landing policies through trial and error, considering both long-term goals (such as touchdown accuracy and safety) and short-term constraints (such as fuel consumption and trajectory limitations). RL enables spacecraft to make real-time decisions based on sensor inputs and learn from the outcomes of previous landing attempts, improving performance and robustness. Addressing the challenges of spacecraft landing using RL involves designing suitable RL algorithms, developing accurate simulation environments, defining appropriate state representations and reward functions, and ensuring safe and reliable learning processes. Research in this field aims to enhance the autonomy and adaptability of spacecraft during the landing phase, ultimately improving the success rates and safety of space missions.

Motivatsion for using reinforcement learning:

The motivation for using reinforcement learning (RL) in spacecraft landing stems from several key advantages and

[1] *Department of Computer Engineering, Smt. Kashibai Navale College of Engineering, 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*
* *Corresponding Author Email: utkarshmoholkar.research@gmail.com*

unique capabilities that RL offers. These factors make RL an appealing approach for addressing the challenges and complexities associated with spacecraft landing.

1. Adaptive Decision-Making: RL enables spacecraft to learn and adapt their landing strategies through interactions with the environment. Instead of relying on pre-defined control algorithms or fixed landing policies, RL agents can continuously update their decisions based on real-time feedback and learn from the outcomes of previous landing attempts. This adaptability allows spacecraft to adjust their behavior and responses to varying environmental conditions, uncertainties, and unexpected events during the landing process. [8]

2. Autonomous Learning: RL facilitates autonomous learning by allowing spacecraft to acquire landing skills and policies without relying heavily on human expertise or explicit programming. By providing an RL agent with a reward signal that reflects the desired landing objectives, spacecraft can learn to optimize their behavior through trial and error. This autonomy reduces the dependence on ground control and enables spacecraft to make real-time decisions during the landing phase, even in situations where communication delays prevent immediate human intervention.[9]

3. Handling Complex and Dynamic Environments: Spacecraft landing involves dealing with complex and dynamic environments, such as varying terrains, atmospheric conditions, and hazards. RL provides a framework for spacecraft to learn and adapt to such complexities by exploring different actions and their consequences. RL algorithms can discover effective landing strategies by considering the uncertainties and variabilities of the environment and learning from the resulting experiences [16]. This capability makes RL well-suited for handling the intricacies of spacecraft landing, which are often challenging to capture with traditional control approaches.

4. Optimization of Multiple Objectives: Spacecraft landing requires the optimization of multiple objectives, such as achieving accurate touchdown, minimizing fuel consumption, maintaining structural integrity, and ensuring the safety of the spacecraft [10]. RL allows for the incorporation of multiple objectives into a unified reward function. By appropriately designing the reward function, RL agents can balance trade-offs between different landing objectives and learn to make decisions that optimize a combination of these objectives. This capability enables spacecraft to achieve landing performance that meets various mission requirements and constraints simultaneously.

5. Generalization and Transfer Learning: RL offers the potential for generalization and transfer learning, which are crucial for spacecraft landing. RL agents can learn landing policies in simulated environments and then transfer the learned knowledge to real-world scenarios. This ability to generalize across different landing conditions, terrains, and planetary bodies is essential for enabling spacecraft to adapt to new and unexplored environments. By training in diverse simulation environments, RL agents can acquire robust landing policies that can be applied to real missions with limited data and prior knowledge.[11]

6. Continuous Improvement and Iterative Refinement: RL allows for continuous improvement and iterative refinement of landing strategies. As spacecraft collect more data and gain experience, RL agents can update their policies and enhance their performance over time. This iterative learning process enables spacecraft to adapt to changing conditions, incorporate new information, and refine their decision-making strategies. RL provides a framework for spacecraft to continuously learn and improve their landing performance, leading to increased success rates and enhanced mission outcomes.

The motivation for using RL in spacecraft landing arises from its adaptive decision-making capabilities, autonomous learning, ability to handle complex environments, optimization of multiple objectives, generalization and transfer learning, and the potential for continuous improvement. By harnessing these advantages, RL has the potential to revolutionize spacecraft landing by enabling autonomous, adaptive, and robust landing strategies that can enhance mission success and safety.

1.1. Abbreviations and Acronyms

Reinforcement Learning(RL), Digital Terrain Model (DTM), Lunar Reconnaissance Orbiter (LROC), Deep Learning(DL), Deep Reinforcement Learning (DRL), Deep Q-Network (DQN), Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C), Deep Deterministic Policy Gradient (DDPG), Trust Region Policy Optimization (TRPO)

2. Literature Review

1. Autonomous Planetary Landing via Deep Reinforcement Learning and Transfer Learning [1]

This paper discusses the challenge of autonomous landing and navigation in space exploration and how recent research has made progress in this field using Deep Learning and Meta-Reinforcement Learning. The paper aims to tackle the problem of autonomous planetary landing using Deep Reinforcement Learning and Transfer Learning. The authors have developed a real-physics simulator using the Bullet/PyBullet library as shown in Figure 1 & 2, and trained a Deep Reinforcement Learning model using DDPG to autonomously land on the lunar environment. The results show that the model can learn a good landing policy, which can be transferred to other environments.

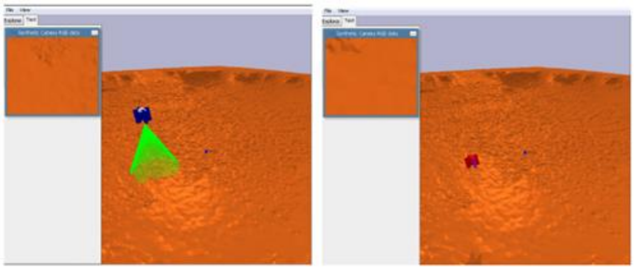

Fig 1. Testing transfer learning for landing on Mars during (a) approach phase, and (b) at touchdown[1]

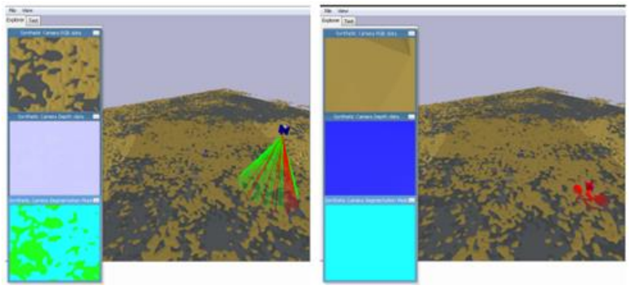

Fig 2. Testing transfer learning for landing on Mars during (a) approach phase, and (b) at touchdown[1]

2. Image-based Deep Reinforcement Learning for Autonomous Lunar Landing[2]

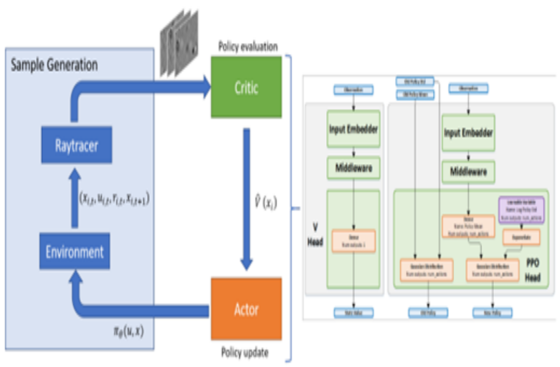

Fig 3. RL Framework[2]

This paper discusses the importance of soft landing on planetary bodies for future human exploration. Soft landing requires advanced navigation and guidance algorithms to achieve precision and zero-velocity touchdown. The paper presents an adaptive landing algorithm that utilizes image-based deep reinforcement learning to derive optimal thrust for a lunar pinpoint landing problem. The algorithm learns from experience using radar altimeter and optical sensor data, enabling autonomous landing on planetary bodies.

As shown in Figure 3, the simulator uses a Digital Terrain Model (DTM) obtained from the Lunar Reconnaissance Orbiter (LROC) database. The DTM, with a resolution of 1791x1791 pixels, contains elevation data used for rendering ground images without the need for actual 3D shapes.

Raytracing Technology: The use of raytracing technology in the simulator ensures quick rendering of observation images without compromising accuracy. Rendering a 16x16-pixel observation with 5 light bounces and 20 samples takes an average of 0.015 seconds.

Blender Integration: The simulator leverages Blender, a Python-based ray tracer, enabling the authors to execute the entire learning algorithm within the renderer framework. This eliminates the need to save image observations on the hard drive, expediting the policy rollout phase.

3. Deep Reinforcement Learning for Six Degree-of Freedom Planetary Powered Descent and Landing [3]

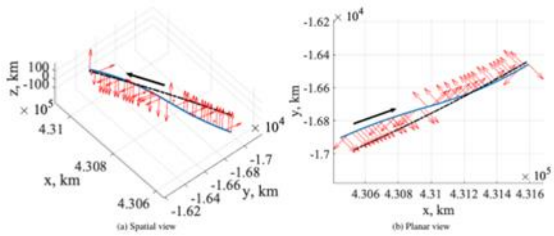

Fig 4. 6-DOF and 3-DOF Fuel-Mass Performance[3]

This paper addresses the critical need for advanced guidance and control algorithms during the powered descent phase of future Mars missions, aiming to achieve precise landing accuracy. The study focuses on developing a groundbreaking integrated guidance and control algorithm that utilizes reinforcement learning theory. This novel approach maps the lander's estimated state directly to a commanded thrust for each engine, resulting in both accuracy and fuel efficiency. Such precise trajectories are crucial for exploring regions on planets and satellites with high scientific potential. The paper presents a novel integrated guidance and control algorithm for the powered descent phase of Mars missions, which uses reinforcement learning theory and proximal policy optimization as a policy gradient method to learn a policy mapping the lander's estimated state directly to a commanded thrust for each engine. The algorithm was found to result in accurate and fuel-efficient trajectories during the powered descent phase, achieving a landing error ellipse of less than 5 meters in radius. The paper concludes that the integrated guidance and control algorithm presented in this paper has the potential to improve the accuracy and efficiency of powered descent and landing for future Mars missions. The policy was shown to be robust to noise and parameter uncertainty. Compared to proposed systems such as that described in figure [4], this system has the advantage of not requiring a cone-shaped glideslope constraint, allowing the targeting of locations such as the bottom of a deep crater.

4. Using Reinforcement Learning to Design a Low-Thrust Approach into a Periodic Orbit in a Multi-Body System [4]

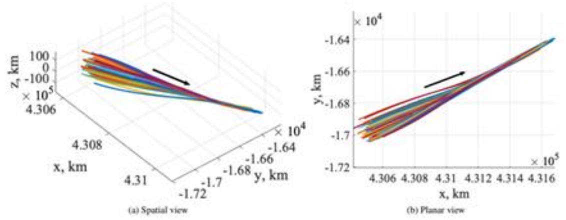

Fig 5. 100 trajectories guided to a reference trajectory (black) using neural networks trained to minimize position and velocity differences, evaluated using an isochronous correspondence.[4]

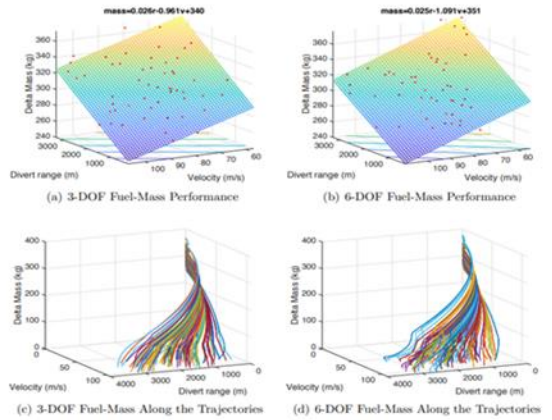

Fig 6. Low-thrust trajectory from a highly perturbed initial condition in blue guided towards the reference trajectory, denoted by the black dashed line, with thrust directions depicted by red arrows, derived from the neural networks trained to reduce state differences, evaluated using an isochronous correspondence.[4]

This paper addresses the challenges of designing trajectories and maneuvers for spacecraft operating in chaotic multi-body systems, with a specific focus on low-thrust-enabled Small Sats facing limited propulsive capabilities, scheduling constraints, and fixed initial conditions. To overcome these hurdles, the authors propose a novel approach based on deep reinforcement learning. The method aims to design a control profile that enables a low-thrust-enabled small satellite to approach a periodic orbit efficiently and autonomously over a short time horizon. By developing robust and autonomous design strategies, this study opens new scientific opportunities for exploring multi-body systems with low-thrust-enabled small satellites. The study highlights that guiding a low-thrust-enabled Small Sat towards a periodic orbit using a reference trajectory along a stable manifold is feasible. However, designing trajectories and maneuver profiles for such small satellites with limited propulsive capabilities presents challenges to trajectory and maneuver designers. The proposed deep reinforcement learning (DRL) method proves (Figure 5 & 6) effective in iteratively learning dynamics and objectives to create a control profile that yields locally optimal solutions, with the formulation of a suitable reward function being crucial to reflect design objectives and operational constraints. The versatility of this approach extends to exploring multi-body systems with low-thrust-enabled small satellites, offering new scientific opportunities. Despite the computational intensity during neural network training in the DRL method, reevaluating a trained network is computationally trivial, making it a practical choice for trajectory design. Moreover, the method's adaptability makes it applicable for designing control profiles for other spacecraft with limited control authority. Overall, this research opens promising avenues for trajectory design and control in the realm of small satellite missions and beyond.

5. Reinforcement Learning for Spacecraft Maneuvering Near Small Bodies[5]

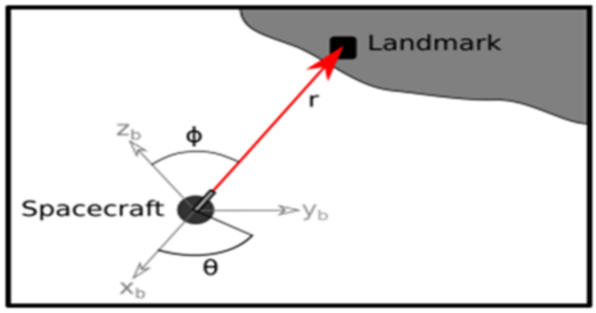

Fig 7. Acquisition of relative position offset to landmark on small body surface.[5]

This paper introduces a novel neural reinforcement learning approach for spacecraft control around small celestial bodies with unknown gravity fields (Figure 7). These bodies have weak gravity compared to Earth, and other factors like solar radiation pressure play a significant role. Their irregular shapes create complex gravity fields that are not easily predictable. To tackle this challenge, the authors use neural reinforcement learning and direct policy search with a genetic algorithm to find control policies for the spacecraft. The feed-forward neural network architecture is chosen due to its ability to handle continuous state and action spaces efficiently.

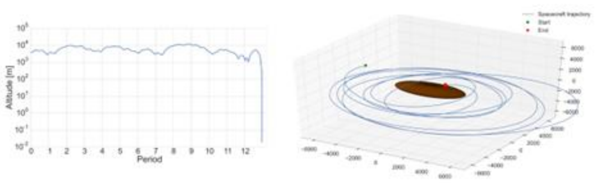

Fig 7. Example of a simulated trajectory of an uncontrolled spacecraft around a tumbling tri-axial small body. The altitude over the surface is shown (left) against the simulation time normalized by the angular velocity period P together with the actual trajectory (right).[5]

The results of the study showcase the effectiveness of the proposed approach. Despite the uncertainty of the asteroid's gravity field and limited perception capabilities, the spacecraft successfully hovers above the asteroid surface with minimal residual drift (Figure 8). This demonstrates the potential of the neural reinforcement learning method combined with lightweight neuromorphic systems for spacecraft maneuvering in low-gravity environments. The study highlights the adaptability of the approach to challenging conditions and uncertain environments, making it promising for future space missions involving spacecraft maneuvering around small celestial bodies like asteroids and comets.

3. Fundamentals of Reinforcement Learning

Definition of reinforcement learning and its components (agent, environment, actions, rewards):

Reinforcement Learning (RL) is a subfield of machine learning that focuses on developing algorithms and techniques for an agent to learn optimal behavior through interactions with an environment. RL is inspired by the concept of how humans and animals learn from feedback and rewards to make decisions and improve their performance over time [12].

The main components of RL include:

1. Agent: The agent is the entity that learns and takes actions within the environment. It is the learner or decision-maker that interacts with the environment to achieve a specific goal. The agent receives observations (state information) from the environment, selects actions, and learns from the feedback (rewards) received from the environment
2. Environment: The environment represents the external context with which the agent interacts. It can be a simulated environment, a physical system, or a combination of both. The environment provides the agent with information about its current state and responds to the agent's actions by transitioning to a new state and providing rewards or penalties.
3. Actions: Actions are the choices made by the agent to influence the environment. The agent selects actions based on its policy, which defines the mapping from states to actions. The action space can be discrete (a finite set of possible actions) or continuous (an infinite set of possible actions). The agent's goal is to learn a policy that maximizes the cumulative rewards received over time.
4. Rewards: Rewards are the feedback signals that the agent receives from the environment. They indicate the desirability or quality of the agent's actions in each state. The agent's objective is typically to maximize the cumulative rewards it receives over a series of interactions with the environment. Rewards can be positive, negative, or zero, and they can be immediate or delayed. The agent uses the reward signal to evaluate and update its policy to improve its future actions. The RL process involves the agent repeatedly observing the current state of the environment, selecting actions based on its policy, receiving rewards from the environment, and updating its policy based on the received rewards. Through this iterative process of exploration and exploitation, the agent learns to make better decisions over time, leading to improved performance in achieving its goals.

Reinforcement learning algorithms, such as Q-learning, policy gradients, and actor-critic methods, are designed to guide the agent's learning process by optimizing the trade-off between exploration (learning from new experiences) and exploitation (leveraging known information). RL algorithms employ various techniques to handle the exploration-exploitation dilemma and learn effective policies in complex and uncertain environments.

Overview of RL algorithms: Q-learning, Deep Q-Network (DQN), Proximal Policy Optimization (PPO), etc.:

Reinforcement learning (RL) encompasses a wide range of algorithms that aim to enable agents to learn optimal behavior through interactions with an environment. Here is an overview of some prominent RL algorithms:

1. Q-Learning: Q-learning is a model-free RL algorithm that is particularly effective for discrete action spaces. It uses a value function called the Q-function to estimate the expected cumulative rewards for taking an action in each state. Q-learning updates the Q-values based on the Bellman equation and uses an exploration-exploitation strategy, such as epsilon-greedy, to balance between exploring new actions and exploiting the current best actions.[13]
2. Deep Q-Network (DQN): DQN is an extension of Q-learning that utilizes deep neural networks to approximate the Q-function for high-dimensional state spaces. It combines Q-learning with a deep neural network as a function approximator, allowing the agent to learn directly from raw sensory inputs. DQN incorporates experience replay and target networks to stabilize and improve learning.[14]
3. Proximal Policy Optimization (PPO): PPO is a model-free, policy optimization algorithm that operates directly on policy functions. It aims to find the optimal policy by iteratively updating it in a way that maximizes the expected cumulative rewards. PPO employs a surrogate objective function and a clipped surrogate objective to ensure stable and reliable policy

updates. It also uses multiple epochs of optimization on collected experience to improve sample efficiency.[15]

4. Actor-Critic Methods: Actor-Critic methods combine the advantages of both value-based methods (e.g., Q-learning) and policy-based methods (e.g., PPO). They utilize both a value function (critic) to estimate the expected rewards and a policy function (actor) to determine the actions. Actor-Critic algorithms leverage the gradient information from the value function to update the policy parameters. Examples of actor-critic algorithms include Advantage Actor-Critic (A2C), Advantage Actor-Critic with Generalized Advantage Estimation (A2C-GAE), and Asynchronous Advantage Actor-Critic (A3C) [16].

5. Deep Deterministic Policy Gradient (DDPG): DDPG is an off-policy actor-critic algorithm specifically designed for continuous action spaces. It combines an actor network, which approximates the policy function, with a critic network, which estimates the action-value function. DDPG employs the concept of experience replay and utilizes target networks to stabilize learning.[17]
6. Trust Region Policy Optimization (TRPO): TRPO is a policy optimization algorithm that seeks to maximize the expected rewards while maintaining a constraint on policy updates to prevent large deviations. It optimizes the policy iteratively within a trust region framework, ensuring that the new policy is close to the previous policy. TRPO utilizes conjugate gradient optimization to solve for the policy updates [18].

These are just a few examples of RL algorithms, each with its unique characteristics, advantages, and application domains. The choice of algorithm depends on factors such as the problem domain, action space, state space, and specific requirements of the problem at hand. Researchers and practitioners select the most appropriate algorithm based on these considerations to address the challenges of spacecraft landing using reinforcement learning.

Exploration vs. exploitation trade-off in RL [19]:

The exploration-exploitation trade-off is a fundamental concept in reinforcement learning (RL) that addresses the challenge of balancing between exploring new actions to learn more about the environment and exploiting the current knowledge to maximize the cumulative rewards.

Exploration:

Exploration refers to the process of actively seeking new information about the environment by trying out different actions and observing their outcomes. In RL, exploration is necessary to discover potentially better actions or states that may lead to higher rewards. By exploring, an RL agent can gather more data and learn about the dynamics of the environment, uncovering hidden opportunities or better policies. It allows the agent to avoid prematurely settling for suboptimal policies.

Exploitation:

Exploitation involves leveraging the agent's current knowledge and making decisions that are based on the current best-known actions or policies. Exploitation aims to maximize the immediate rewards by choosing actions that are deemed to be optimal based on the agent's existing knowledge. Exploitation is crucial to capitalize on the learned information and make the most of the agent's current understanding of the environment.

Finding the Right Balance:

Finding the right balance between exploration and exploitation is essential for effective RL. If an agent only focuses on exploitation, it may get stuck in suboptimal policies and miss out on discovering better options. On the other hand, excessive exploration can lead to wasted time and resources, delaying the agent's ability to exploit the best-known actions and policies.

Strategies to Address the Trade-off:

1. Epsilon-Greedy: One common approach is the epsilon greedy strategy, where the agent chooses the best-known action (exploitation) most of the time but occasionally selects a random action (exploration) with a small probability epsilon. This strategy allows the agent to explore while still favoring actions that have shown to be more rewarding.[20]

2. SoftMax Action Selection: SoftMax action selection is another strategy that introduces stochasticity into the action selection process. The agent assigns probabilities to each possible action based on their estimated values and then selects actions according to these probabilities. This approach promotes exploration by allowing the agent to choose actions with lower estimated values, although actions with higher values are still more likely to be selected.[21]
3. Upper Confidence Bound (UCB): UCB-based algorithms, such as UCB1 or UCB-Tuned, balance exploration and exploitation by assigning a measure of uncertainty to each action's value estimation. Actions with high uncertainty are given priority for exploration to reduce uncertainty and gain more knowledge about their true values.[22]
4. Thompson Sampling: Thompson Sampling is a Bayesian-based strategy that assigns probabilities to each action based on its estimated value distribution. The agent samples actions based on

these probabilities, which allows for exploration while favoring actions with higher estimated values.[23]

4. Reinforcement Learning Approaches for Spacecraft Landing and Control

Review of RL techniques used in spacecraft landing:

Reinforcement learning (RL) techniques have been widely explored and applied to address the challenges of spacecraft landing. Here is a review of RL techniques commonly used in spacecraft landing research:

1. Model-Free Q-Learning: Q-learning is a popular model-free RL algorithm used in spacecraft landing. It involves estimating the action-value function (Q-function) to determine the best action in each state. Q-learning has been applied to discrete action spaces, where the spacecraft learns an optimal policy through exploration and exploitation of actions. It can be combined with exploration strategies, such as epsilon-greedy, to balance exploration and exploitation during the landing process.[24]

2. Deep Q-Network (DQN): DQN extends Q-learning to handle high-dimensional state spaces by employing deep neural networks as function approximators. DQN approximates the Q-function using a deep neural network and leverages experience replay and target networks to stabilize learning. DQN has been applied to spacecraft landing research, allowing agents to learn directly from raw sensory inputs, such as images or sensor readings.[25]

3. Actor-Critic Methods: Actor-critic methods combine the advantages of value-based methods (e.g., Q-learning) and policy-based methods to improve learning efficiency and stability. These methods employ both a policy network (actor) to select actions and a value function (critic) to estimate the expected rewards. Actor-critic algorithms leverage the gradient information from the critic to update the policy parameters. Advantage Actor -Critic (A2C), Advantage Actor-Critic with Generalized, Advantage Estimation (A2C-GAE), and Asynchronous Advantage Actor-Critic (A3C) are commonly used in spacecraft landing research.[26]

4. Proximal Policy Optimization (PPO): PPO is a policy optimization algorithm that operates directly on policy functions. It maximizes the expected cumulative rewards while ensuring stable and reliable policy updates. PPO employs a surrogate objective function and a clipped surrogate objective to control the magnitude of policy updates. PPO has shown promising results in spacecraft landing tasks, providing improved sample efficiency and stable learning.[27]

5. Deep Deterministic Policy Gradient (DDPG): DDPG is an off-policy actor-critic algorithm specifically designed for continuous action spaces. It combines an actor network, which approximates the policy function, with a critic network, which estimates the action-value function. DDPG employs the concept of experience replay and utilizes target networks to stabilize learning. DDPG has been applied to spacecraft landing tasks requiring continuous control of descent parameters.[28]

6. Trust Region Policy Optimization (TRPO): TRPO is a policy optimization algorithm that seeks to maximize the expected rewards while maintaining a constraint on policy updates. It optimizes the policy within a trust region framework, ensuring that the new policy is close to the previous policy. TRPO has been used in spacecraft landing research to enhance the safety and stability of the landing process.[29]

State representation and sensor fusion techniques:

State representation and sensor fusion techniques are crucial aspects of spacecraft landing using reinforcement learning (RL). They involve the representation of the environment and the fusion of information from different sensors to provide an accurate and comprehensive understanding of the spacecraft's state. Here is an overview of state representation and sensor fusion techniques used in spacecraft landing:

State Representation:

State representation involves encoding the relevant information about the environment, spacecraft dynamics, and other relevant factors into a format that can be input to the RL algorithm. The state representation aims to capture the essential aspects that influence the spacecraft's behavior and decision-making during landing. Common techniques for state representation in spacecraft landing include:

1. Raw Sensor Readings: In some cases, the raw sensor readings, such as camera images or LIDAR point clouds, can serve as the state representation. Deep RL algorithms, such as Deep Q-Networks (DQNs), can directly process these sensory inputs to learn complex representations and make decisions based on them.

2. Feature Extraction: Feature extraction involves extracting relevant features or descriptors from raw sensor data. It reduces the dimensionality of the input and focuses on the most informative aspects of the data. Feature extraction techniques, such as image processing algorithms or filtering methods, can be applied to sensor data to extract useful features for the state representation.

Derived Variables: Derived variables are calculated based on sensor data or other environmental information. These variables can include altitude, velocity, orientation, fuel level, or other derived quantities that capture important aspects of the spacecraft's state. Derived variables provide a more concise representation of the state and can facilitate learning and decision-making processes.

Sensor Fusion Techniques:

Sensor fusion techniques involve combining information from multiple sensors to obtain a more accurate and comprehensive perception of the environment. The fusion process aims to leverage the strengths of different sensors while compensating for their individual limitations. Sensor fusion techniques commonly used in spacecraft landing include:

1. Kalman Filtering: Kalman filtering is a widely used technique for sensor fusion that estimates the true state of the system by combining measurements from different sensors with an underlying dynamic model. Kalman filters are effective in dealing with sensor noise and uncertainties and can provide a robust estimation of the spacecraft's state.[30] [47]

2. Particle Filtering: Particle filtering, also known as Monte Carlo filtering, is a non-parametric Bayesian filtering technique. It uses a set of particles to represent the posterior distribution of the system's state. Particle filters can handle nonlinear and non-Gaussian systems and are suitable for spacecraft landing scenarios with complex dynamics.[31] [46]

3. Extended Kalman Filtering: Extended Kalman filtering (EKF) extends the Kalman filter to handle non-linear system dynamics by approximating them with linear models through Taylor series expansion. EKF is commonly used in spacecraft landing scenarios where the system's dynamics deviate from linearity.[32]

4. Sensor Weighting and Fusion Rules: Sensor fusion can involve assigning weights to different sensors based on their reliability, accuracy, or relevance to the landing task. Fusion rules determine how the measurements from different sensors are combined to obtain the overall state estimate.

Weighting and fusion rules can be based on statistical methods, information theory, or expert knowledge.[33] By appropriately representing the state and fusing information from multiple sensors, spacecraft landing systems can obtain a comprehensive understanding of the environment and the spacecraft's position, orientation, and dynamics. These techniques enable RL algorithms to make informed decisions and navigate the spacecraft safely during the landing process.

Action space design and control strategies:

Action space design and control strategies play a crucial role in spacecraft landing using reinforcement learning (RL). They determine the set of actions available to the RL agent and the mechanisms for controlling the spacecraft's descent and landing. Here is an overview of action space design and control strategies used in spacecraft landing:

Action Space Design:

The action space represents the set of actions that the RL agent can choose from during the landing process. The design of the action space depends on the specific requirements and constraints of the spacecraft landing task. Common approaches for action space design include:

1. Discrete Actions: Discrete action spaces define a finite set of actions that the agent can choose from. Examples of discrete actions in spacecraft landing can include thrust commands (e.g., increase, decrease), attitude adjustments (e.g., pitch, yaw), or control mode switches (e.g., hover, descent). Discrete action spaces are suitable when the spacecraft's control system operates in a stepwise manner or has limited control options.[34]

2. Continuous Actions: Continuous action spaces allow for a wide range of real-valued actions. Continuous actions are appropriate when precise control over continuous variables is required. Examples of continuous actions in spacecraft landing can include throttle settings, thruster gimbal angles, or control surface deflections. Continuous action spaces enable fine-grained control and can provide more flexibility in controlling the spacecraft's descent.[34]

Control Strategies:

Control strategies determine how the RL agent utilizes the selected actions to guide the spacecraft's descent and landing. These strategies aim to ensure safe, accurate, and efficient landing. Some common control strategies in spacecraft landing include:

1. Proportional-Integral-Derivative (PID) Control: PID controls a classic control strategy that uses feedback to regulate system behavior. It calculates control outputs based on proportional, integral, and derivative terms. PID controllers can be used to stabilize and control the spacecraft's attitude, velocity, or position during landing.[35]

2. Model Predictive Control (MPC): MPC is a control strategy that utilizes a predictive model of the system dynamics to optimize control actions over a finite time horizon. MPC predicts the future system states and optimizes control inputs to achieve desired objectives while considering constraints. MPC can be applied to spacecraft landing to generate control commands that optimize landing accuracy, fuel efficiency, or other objectives.[36]

3. Adaptive Control: Adaptive control strategies adjust control parameters or policies based on real time measurements or estimation of the system dynamics. Adaptive control allows the spacecraft to adapt to changing conditions, uncertainties, or disturbances during the landing process. It can enhance the robustness and adaptability of the control system.[37]

4. Trajectory Planning: Trajectory planning involves generating a desired trajectory for the spacecraft's descent and landing. It defines the path that the spacecraft should follow to achieve landing objectives while considering constraints such as fuel consumption or obstacle avoidance. Trajectory planning can be combined with RL to learn optimal trajectories or to refine and adjust planned trajectories based on real-time feedback.[38]

5. Hybrid Control: Hybrid control combines different control strategies or techniques to achieve the desired landing performance. It may involve switching between different controllers or using a combination of feedback-based control and RL-based control. Hybrid control strategies leverage the strengths of multiple approaches and can enhance the overall control performance during spacecraft landing.

Reward engineering and shaping:

Reward engineering and shaping are techniques used in reinforcement learning (RL) to design and shape the reward signals that guide the learning process of the RL agent. By carefully designing the reward structure, these techniques can influence the behavior of the RL agent and expedite the learning process. Here's an overview of reward engineering and shaping in spacecraft landing:

Reward Engineering [39]:

Reward engineering involves designing the reward function to provide appropriate feedback to the RL agent based on the desired behavior or objectives of the spacecraft landing task. The reward function influences the RL agent's policy by assigning positive or negative rewards based on the agent's actions and their outcomes. Key considerations in reward engineering for spacecraft landing include:

1. Landing Accuracy: Rewarding the agent for achieving accurate and precise landings is a common objective in spacecraft landing. The reward function can include terms that measure the distance or deviation from the target landing site, penalizing larger deviations, and incentivizing accurate touchdown.

2. Fuel Efficiency: Fuel consumption is a critical factor in spacecraft missions. Rewarding the agent for minimizing fuel consumption encourages the RL agent to learn efficient control strategies and trajectories that conserve fuel during descent and landing.

3. Safety Considerations: Safety is paramount in spacecraft landing. The reward function can incorporate safety-related metrics such as avoiding collisions, staying within specified limits or boundaries, or adhering to operational constraints to ensure safe landings.

4. Time Efficiency: Rewarding the agent for completing the landing task within a specified time frame can encourage the RL agent to learn expedited descent and landing strategies, optimizing for timely mission execution.

Reward Shaping [40]:

Reward shaping involves adding additional reward signals or shaping functions to guide the RL agent's learning process. These shaped rewards can provide more informative feedback to the agent, accelerate learning, and promote desirable behavior. Key aspects of reward shaping in spacecraft landing include:

1. Sparse Reward Augmentation: In some spacecraft landing scenarios, the rewards associated with successful landings may be sparse, meaning that the agent receives a reward only at the end of a successful landing. Reward shaping techniques can provide intermediate rewards during the landing process, encouraging the agent to learn incremental progress towards successful landings.

2. Potential-Based Reward Shaping: Potential-based reward shaping uses a shaping potential function to provide additional rewards based on the agent's progress towards a goal. It guides the agent by assigning rewards proportional to the potential progress made towards a successful landing. Potential-based shaping can help overcome sparse rewards and accelerate the learning process.

3. Expert Demonstrations: Expert demonstrations involve providing pre-recorded or pre-trained expert trajectories as additional reward information. The RL agent can learn from these demonstrations and imitate the expert behavior, enabling faster convergence and improved performance.

4. Curriculum Learning: Curriculum learning involves gradually increasing the difficulty or complexity of the landing task over time. By starting with simpler scenarios and gradually introducing more challenging environments, the RL agent can learn in a more structured and progressive manner, leveraging reward shaping to guide the learning process.

Reward engineering and shaping techniques allow researchers to guide the RL agent's learning process in spacecraft landing tasks. By carefully designing the reward function and shaping rewards, RL agents can learn efficient and accurate landing policies, enhance safety, optimize fuel consumption, and adapt to different mission objectives and constraints. These techniques help overcome challenges related to sparse rewards and provide valuable guidance for RL-based spacecraft landing.

Training procedures and algorithms:

Training procedures and algorithms play a critical role in spacecraft landing using reinforcement learning (RL). These procedures and algorithms guide the learning process of the RL agent and enable it to acquire optimal landing policies. Here's an overview of the training procedures and algorithms commonly used in spacecraft landing:

1. Off-Policy Training:

-Off-policy training allows the RL agent to learn from a mixture of data generated by its current policy and data collected from other policies or sources. This approach enables the agent to learn from a diverse set of experiences and promotes sample efficiency.

-Algorithms: Off-policy algorithms such as Q-learning, Deep Q-Network (DQN), and Deep Deterministic Policy Gradient (DDPG) are commonly used for spacecraft landing. These algorithms utilize experience replay, where the agent stores and samples from a replay buffer to break temporal correlations and improve learning stability.

2. On-Policy Training:

- On-policy training involves learning from the interaction of the RL agent with the environment using the current policy. The agent explores the environment and updates its policy based on the collected experiences.

-Algorithms: On-policy algorithms such as Proximal Policy Optimization (PPO), Trust Region Policy Optimization (TRPO), and Actor-Critic methods (A2C, A3C) are popular for spacecraft landing. These algorithms optimize the policy directly to maximize the expected cumulative rewards, while often using techniques like advantage estimation and policy gradients to improve learning efficiency.

. Exploration Strategies:

- Exploration is critical for RL agents to discover new states and actions and avoid getting stuck in suboptimal policies. Various exploration strategies are employed to balance exploration and

exploitation, enabling the RL agent to learn more effectively.

- Epsilon-Greedy, SoftMax Action Selection, Upper Confidence Bound (UCB), and Thompson Sampling are commonly used exploration strategies in spacecraft landing. These strategies promote exploration by encouraging the agent to select less-explored actions or visit unexplored regions of the state space.

4. Curriculum Learning [41]:

-Curriculum learning involves gradually increasing the difficulty or complexity of the training tasks. In spacecraft landing, this can involve starting with simpler landing scenarios and gradually introducing more challenging terrains or environmental conditions.

- Algorithms: RL algorithms can be combined with curriculum learning techniques to enhance learning efficiency. The difficulty of the landing task can be progressively increased, allowing the agent to acquire knowledge and skills in a more structured and gradual manner.

5. Transfer Learning:

-Transfer learning leverages knowledge gained from previously learned tasks or domains to accelerate learning in new tasks or domains. Pre-training the RL agent on related tasks or using pre trained models can provide valuable initialization for the spacecraft landing task.

- Algorithms: Algorithms like Deep Q-Networks (DQN) and Proximal Policy Optimization (PPO) can be combined with transfer learning approaches to bootstrap the learning process and improve convergence speed [42].

6. Reward Shaping and Engineering [40]:

- Reward shaping and engineering techniques are employed to design appropriate reward functions that guide the RL agent towards desired behaviors. These techniques provide informative feedback and accelerate learning by shaping the reward signals.

- Potential-based reward shaping, sparse reward augmentation, and expert demonstrations are commonly used reward shaping techniques in spacecraft landing. These techniques provide additional rewards or guidance to facilitate learning. The selection of specific training procedures and algorithms depends on the characteristics of the spacecraft landing task, available resources, and desired performance objectives. Researchers and practitioners often experiment with different combinations of algorithms, exploration strategies, reward engineering, and transfer learning techniques to achieve effective RL-based spacecraft landing.

5. Conclusion

In conclusion, the survey findings highlight the significant progress and potential of using reinforcement learning (RL) in spacecraft landing. Here's a recap of the current state-of-the-art and the promising opportunities for future research in spacecraft landing using RL:

1. Current State-of-the-Art: RL has emerged as a promising approach for spacecraft landing, offering advantages such as adaptability, learning from data, and handling complex dynamics. Key components of RL, including state representation, action space design, training procedures, and evaluation metrics, have been extensively studied and applied in the context of spacecraft landing.

2. RL Algorithms and Techniques: Various RL algorithms have been explored, including Q-learning, Deep Q-Networks (DQN), Proximal Policy Optimization (PPO), and model-based RL. The integration of deep neural networks and hierarchical RL has further enhanced the capabilities of RL based spacecraft landing systems.

3. Challenges and Limitations: Several challenges and limitations exist, including sample efficiency, defining appropriate evaluation metrics, generalization to varying

environmental conditions, safety concerns, and the integration of RL with other control techniques. Addressing these challenges requires further research, algorithmic advancements, collaboration with domain experts, and real world testing and validation.

4. Future Research Opportunities: The survey identifies several promising opportunities for future research in spacecraft landing using RL. These include the exploration of advanced RL algorithms, multi-modal sensor fusion, hybrid control approaches, transfer learning and domain adaptation, safety-critical RL, real-world implementation challenges, and autonomy in complex landing scenarios. Collaboration between researchers, space agencies, industry partners, and regulatory bodies is crucial for driving progress in these areas.

The current state-of-the-art in spacecraft landing using RL showcases the potential of RL algorithms to improve the performance, efficiency, and safety of spacecraft landing systems. However, further research and development are needed to address challenges, ensure safety, and enable the practical implementation of RL-based landing systems in real-world missions. The field presents promising opportunities for advancements in algorithmic techniques, integration with other control methods, and the exploration of autonomous landing capabilities in complex and challenging scenarios.

References

[1] G. Ciabatti, S. Daftry and R. Capobianco, "Autonomous Planetary Landing via Deep Reinforcement Learning and Transfer Learning," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Nashville, TN, USA, 2021, pp. 2031- 2038, doi: 10.1109/CVPRW53098.2021.00231.

[2] Scorsoglio, Andrea & Furfaro, Roberto & Linares, Richard & Gaudet, Brian. (2020). Image based Deep Reinforcement Learning for Autonomous Lunar Landing. 10.2514/6.2020-1910.

[3] Gaudet, Brian & Linares, Richard & Furfaro, Roberto. (2020). Deep Reinforcement Learning for Six Degree-of-Freedom Planetary Landing. Advances in Space Research. 65. 10.1016/j.asr.2019.12.030.

[4] Sullivan, Christopher & Bosanac, Natasha. (2020). Using Reinforcement Learning to Design a Low-Thrust Approach into a Periodic Orbit in a Multi-Body System. 10.2514/6.2020-1914.

[5] Gaudet, Brian & Furfaro, Roberto. (2012). Robust Spacecraft Hovering Near Small Bodies in Environments with Unknown Dynamics Using Reinforcement Learning. 10.2514/6.2012- 5072.

[6] Russo, Antonia, and Gianluca Lax. 2022. "Using Artificial Intelligence for Space Challenges: A Survey" Applied Sciences 12, no. 10: 5106. https://doi.org/10.3390/app12105106

[7] Wilson, C., Riccardi, A. Improving the efficiency of reinforcement learning for a spacecraft powered descent with Q-learning. Optim Eng 24, 223–255 (2023).

[8] Yen, Gary & Hickey, Travis. (2004). Reinforcement learning algorithms for robotic navigation in dynamic environments. ISA transactions. 43. 217-30. 10.1016/S0019- 0578(07)60032-9.

[9] Tipaldi, Massimo & Iervolino, Raffaele & Massenio, Paolo. (2022). Reinforcement learning in spacecraft control applications: Advances, prospects, and challenges. Annual Reviews in Control. 10.1016/j.arcontrol.2022.07.004.

[10] Silvestrini, Stefano, and Michèle Lavagna. 2022. "Deep Learning and Artificial Neural Networks for Spacecraft Dynamics, Navigation and Control" Drones 6, no. 10: 270. https://doi.org/10.3390/drones6100270

[11] Trisolini, Mirko & Colombo, Camilla & Lewis, Hugh. (2017). Multi-objective optimization for spacecraft design for demise and survivability.

[12] Chen, X. (2020). Data-Efficient Reinforcement and Transfer Learning in Robotics (PhD dissertation, KTH Royal Institute of Technology). Retrieved from https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva285993

[13] Sutton, Richard S. and Barto, Andrew G.. Reinforcement Learning: An Introduction. Second: The MIT Press, 2018.

[14] Habib, Nazia. (2019) 2019. Hands-On Q-Learning with Python. 1st ed. Packt Publishing. https://www.perlego.com/book/960445/handson-qlearning-with-python-practical-qlearningwith-openai-gym-keras-and-tensorflow-pdf

[15] Winder, Phil. Reinforcement Learning. O'Reilly Media, Inc., 2020.

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," arXiv:1707.06347 [cs.LG], Jul. 2017.

[17] Haarnoja, T., Zhou, A., Abbeel, P. & Levine, S.. (2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. Proceedings of the 35th International Conference on Machine Learning, in Proceedings of Machine Learning Research 80:1861-1870Availablefrom

https://proceedings.mlr.press/v80/haarnoja18b.html.

[18] T. Tiong, I. Saad, K. T. K. Teo and H. b. Lago, "Deep Reinforcement Learning with Robust Deep Deterministic Policy Gradient," 2020 2nd International Conference on Electrical, Control and Instrumentation Engineering (ICECIE), Kuala Lumpur, Malaysia, 2020, pp. 1-5, doi: 10.1109/ICECIE50279.2020.9309539.

[19] W. Meng, Q. Zheng, Y. Shi and G. Pan, "An Off-Policy Trust Region Policy Optimization Method With Monotonic Improvement Guarantee for Deep Reinforcement Learning," in IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 5, pp. 2223-2235, May 2022, doi: 10.1109/TNNLS.2020.3044196.

[20] Chadi, M. A., & Mousannif, H. (2023). Understanding Reinforcement Learning Algorithms: The Progress from Basic Q-learning to Proximal Policy Optimization. arXiv preprint arXiv:2304.00026.

[21] Powell, W. B., & Frazier, P. (2008). Optimal learning. In State-of-the-Art Decision-Making Tools in the Information-Intensive Age (pp. 213-246). Informs.

[22] Cruz, F., Dazeley, R., Vamplew, P., & Moreira, I. (2021). Explainable robotic systems: Understanding goal-driven actions in a reinforcement learning scenario. Neural Computing and Applications, 1-18

[23] Yang, J., Hou, X., Hu, Y. H., Liu, Y., & Pan, Q. (2020). A reinforcement learning scheme for active multi-debris removal mission planning with modified upper confidence bound tree search. IEEE Access, 8, 108461-108473.

[24] Russo, D. J., Van Roy, B., Kazerouni, A., Osband, I., & Wen, Z. (2018). A tutorial on thompson sampling. Foundations and Trends® in Machine Learning, 11(1), 1-96. Braun, D., Marb, M. M., Angelov, J., Wechner, M., & Holzapfel, F. (2023). Worst-Case Analysis of Complex Nonlinear Flight Control Designs Using Deep Q-Learning. Journal of Guidance, Control, and Dynamics, 1-13.

[25] Chen, Y., & Kulla, E. (2019). A Deep Q-Network with Experience Optimization (DQN-EO) for Atari’s Space Invaders. In Web, Artificial Intelligence and Network Applications: Proceedings of the Workshops of the 33rd International Conference on Advanced Information Networking and Applications (WAINA-2019) 33 (pp. 351-361). Springer International Publishing.

[26] Shi, X. N., Zhou, D., Chen, X., & Zhou, Z. G. (2021). Actor-critic-based predefined-time control for spacecraft attitude formation system with guaranteeing prescribed performance on SO (3). Aerospace Science and Technology, 117, 106898.

[27] Miller, D., Englander, J. A., & Linares, R. (2019, August). Interplanetary low-thrust design using proximal policy optimization. In 2019 AAS/AIAA Astrodynamics Specialist Conference (No. GSFC-E-DAA-TN71225)

[28] Zhang, J., Wu, F., Zhao, J., & Xu, F. (2019). A method of attitude control based on deep deterministic policy gradient. In Cognitive Systems and Signal Processing: 4th International Conference, ICCSIP 2018, Beijing, China, November 29-December 1, 2018, Revised Selected Papers, Part II 4 (pp. 197-207). Springer Singapore.

[29] Miller, D., & Linares, R. (2019, February). Low-thrust optimal control via reinforcement learning. In 29th AAS/AIAA Space Flight Mechanics Meeting (Vol. 168, pp. 1817-1834). American Astronautical Society Ka’anapali, Hawaii.

[30] Chen, S. Y. (2011). Kalman filter for robot vision: a survey. IEEE Transactions on industrial electronics, 59(11), 4409-4420.

[31] Rohilla, R., Sikri, V., & Kapoor, R. (2017). Spider monkey optimisation assisted particle filter for robust object tracking. IET Computer Vision, 11(3), 207-219.

[32] Fujii, K. (2013). Extended kalman filter. Refernce Manual, 14, 41.

[33] Dong, J., Zhuang, D., Huang, Y., & Fu, J. (2009). Advances in multi-sensor data fusion: Algorithms and applications. Sensors, 9(10), 7771-7784.

[34] Capra, L., Brandonisio, A., & Lavagna, M. (2023). Network architecture and action space analysis for deep reinforcement learning toward spacecraft autonomous guidance. Advances in Space Research, 71(9), 3787-3802

[35] Peng, X., & Gong, Y. (2022). Globally stable proportional-integral-derivative control for spacecraft pose tracking via dual quaternions. IET Control Theory & Applications, 16(18), 1847-1859.

[36] Weiss, A., Baldwin, M., Erwin, R. S., & Kolmanovsky, I. (2015). Model predictive control for spacecraft rendezvous and docking: Strategies for handling constraints and case studies. IEEE Transactions on Control Systems Technology, 23(4), 1638-1647.

[37] Hu, W., Li, Z., Dai, M. Z., & Huang, T. (2023). Robust adaptive control for spacecraft attitude synchronization subject to external disturbances: A performance adjustable event- triggered mechanism. International Journal of Robust and Nonlinear Control, 33(3), 2392- 2408.

[38] Chai, R., Tsourdos, A., Savvaris, A., Chai, S., Xia, Y., & Chen, C. P. (2019). Six-DOF spacecraft optimal trajectory planning and real-time attitude control: a deep neural network-based approach. IEEE transactions on neural networks and learning systems, 31(11), 5005- 5013.2408.

[39] Dewey, D. (2014, March). Reinforcement learning and the reward engineering principle. In 2014 AAAI Spring Symposium Series.

[40] Grzes, M. (2017). Reward shaping in episodic reinforcement learning.

[41] Narvekar, S., Peng, B., Leonetti, M., Sinapov, J., Taylor, M. E., & Stone, P. (2020). Curriculum learning for reinforcement learning domains: A framework and survey. The Journal of Machine Learning Research, 21(1), 7382-7431.

[42] Zhu, Z., Lin, K., Jain, A. K., & Zhou, J. (2023). Transfer learning in deep reinforcement learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[43] Janhavi H. Borse, Dipti D. Patil, Vinod Kumar, Sudhir Kumar, "Soft Landing Parameter Measurements for Candidate Navigation Trajectories Using Deep Learning and AI-Enabled Planetary Descent", Mathematical Problems in Engineering, vol. 2022, Article ID 2886312, 14 pages, 2022. https://doi.org/10.1155/2022/2886312 2021-22 (6)

[44] Utkarsh R. Moholkar, Dipti D. Patil, Vinod Kumar, Archana Patil, “Deep Learning Approach for Unmanned Aerial Vehicle Landing”, in International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075 (Online), Volume-11 Issue-10, September 2022 Doi: 10.35940/ijitee.J9263.09111022

[45] Janhvi Borse, Dipti D. Patil, “Empirical Analysis of Feature Points Extraction Techniques for Space Applications”, in International Journal of Advanced Computer Science and Applications, ISSN: 2158-107X, Vol. 12, No. 9, 2021https://dx.doi.org/10.14569/IJACSA.2021.0120910

[46] J. Borse, D. Patil and V. Kumar, "Tracking Keypoints from Consecutive Video Frames Using CNN Features for Space Applications", Technical Journal, vol.15, no. 1, pp. 11-17, 2021. [Online]. https://doi.org/10.31803/tg-20210204161210

[47] Chindhe, B., Ramalingam, A., Chavan, S., Hardas, S., Patil, D. (2023). Advances in Vision-Based UAV Manoeuvring Techniques. In: Thampi, S.M., Mukhopadhyay, J., Paprzycki, M., Li, KC. (eds) International Symposium on Intelligent Informatics. ISI 2022. Smart Innovation, Systems and Technologies, vol 333. Springer, Singapore. https://doi.org/10.1007/978-981-19-8094-7_35