Exploring Deep Learning Techniques in Planetary Soft Landing

Vano Mazashvili (Student)

Sapienza University of Rome, Rome, Italy

ARTICLE INFO

Keywords:
Autonomous Landing
Deep Learning
Hazard Detection
Trajectory Optimization
Space Robotics
Multi-Modal Sensor Fusion
Space Exploration

ABSTRACT

Safe and precise landings on various planetary bodies present a challenge of utmost importance for the future of space exploration. Recent developments in parallel computing, machine learning, and mass data processing led to the emergence of powerful tools for advancing and honing the capabilities of autonomous planetary landing systems. This survey provides an overview of the state-of-the-art approaches in deep learning already existent and developing for the autonomous planetary landing problem, namely, we focus on trajectory and fuel consumption optimization, hazard detection, obstacle avoidance, and effective landing site selection.

In this survey, we classify existing methods based on their architectures, training approaches, and performance metrics. Then we compare and analyze the differences and effectiveness of the given methodologies and deduct their strengths, weaknesses, and limitations. Therefore, following the aforementioned, we state the open research challenges. Addressing these problems will pave a path to the new generation of deep learning algorithms for more efficient and optimized autonomous planetary landing techniques.

1. Introduction

Space exploration is one of the most important scientific and engineering endeavors of humankind. Recent advances in space technology and robotics created the need for optimal autonomous planetary soft landing solutions. Although already existing algorithms, for example, the ones used for the Apollo lunar landing, are being used with success, their design does not accommodate the dynamic and unpredictable environment of outer space, thus providing poor adaptability.

The objective is to create an efficient system with great adaptability - performing well in uncertain and unpredictable environments with minimal human intervention.

Deep learning - a sub-field of artificial intelligence, has found great success in data-driven applications. The ability to extract and learn upon intricate patterns from given datasets has massively progressed fields like medical diagnosis, natural language processing, computer vision, and so on. That being said, advancements in data acquisition techniques and aforementioned challenges in space exploration, birthed the applications of deep learning in planetary landing problems.

In this survey, we review 13 papers on autonomous planet landing using deep learning and aim to compare and analyze those approaches on different metrics. With a focus on Hazard detection, trajectory optimization, and landing site selection, we classify their effectiveness, robustness, and applicability. Stating their advantages and weaknesses will be a segue for the discussion regarding the remaining hurdles and open research questions in the field.

This survey presents insight into the current state of autonomous planetary soft landing solutions and their applications, comments on their effectiveness based on their modularity, safety, and efficiency, and states their shortcomings to inspire further research.

2. Problem Description

The given problem of the planetary soft landing will be deemed solved once we optimally knock down the issues regarding safety, efficiency, and adaptability. Therefore, this multifaceted challenge can be divided into several parts; this can involve the input generation from multiple information sources, preprocessing, hazard map, or any other data structure (containing the information about the landing environment) generation using various neural network architectures. Problems can be related to fuel efficiency, acceleration and velocity constraints, changes in gravitational forces, lighting, winds, etc. Essentially, if we group these requirements, the given problem boils down to two main components.

(a) Landing site selection through hazard detection

For this subproblem, the lander chooses the safe landing zone. That means that it should successfully identify and avoid hazards such as craters, cliffs, boulders, unstable ground, or just rough terrain. Leveraging sensor data such as digital elevation maps, coming from LiDAR, or video/images taken with onboard cameras, the main objective is to create a robust algorithm capable of hazard detection, segmentation, and safest zone calculation/evaluation. This task can be done as a preliminary, or, ideally, in real-time, to enhance flexibility, adaptability, and safety in dynamic environments.

(b) Trajectory optimization and guidance

Once the algorithm has obtained the information regarding hazards, the lander's job is to guide the spacecraft to touchdown. The key points are safe, efficient, and adaptable, coming from the fact that the algorithm has to calculate fuel optimal and feasible landing trajectory while accounting for the sensor error margins, malfunctions, dynamic environmental variables, and delays/errors in communication systems. All these constraints depend on each other, as the lander may be able to calculate the safe landing trajectory successfully. Still, if fuel efficiency is not taken into consideration, the lander might run out of propellant if it takes too much time for the calculations, or chooses a non-fuel-optimal route.

To sum up, safe and fast, real-time autonomous landing site selection based on hazard identification, and fuel-efficient, adaptable trajectory optimization lander algorithms are the cornerstone of the frontier of space exploration. In the following sections, we will delve into the cutting-edge Machine Learning solutions offered by the dozen academic papers, trying to solve the given problem with unique and novel approaches.

3. Existing Methods Categorized

In pursuit of developing an efficient and safe solution for the autonomous planetary soft-lander algorithm, researchers are considering the two main domains of the challenge. Addressing every aspect should give us a good and comprehensive knowledge of the overall structure of the existing and desired approaches.

3.1. Hazard Detection

Hazard detection problem plays a crucial role in the selection of landing sites. The autonomous lander is responsible for successfully recognizing and classifying existing hazardous locations online in real time to manage evasive maneuvers and/or land on an optimal landing site. The task is usually done between 0.5 and 2 km above the surface.

A deep learning approach is presented to solve the given problem using LIDAR scans of the lunar surface (8). The paper takes advantage of a cross-NASA-developed Hazard Detection System (HDS) which is a component of the Autonomous Landing and Hazard Avoidance Technology (ALHAT) sensor suite. The following technology enables us to generate a Digital Elevation Map (DEM) using just LIDAR sensors, which can be fed to the Convolutional Neural Networks to classify safe and feasible landing locations. For this Challenge, Semantic Segmentation, popularized in computer vision applications is used to analyze DEMs, meaning that, it is an image classification problem. (8).

Although the Lunar DEM data are abundant, the paper still uses data augmentation techniques and introduces noise to closely simulate on-board sensor distortions (8). For this particular case, LIDAR scans are strongly preferred over camera inputs as DEMs are robust in all lighting scenarios. The resulting data are used to train a UNet-like network architecture as it is characteristic of semantic segmentation. (8)

As a result, CNN returns a mean pixel accuracy of 92%, as the paper states, on the testing dataset. (8)It is noteworthy that we have a similarity between validation and testing dataset accuracy, meaning that the network is neither over nor under-fitted.

One of the more interesting points in this paper is the proposed weighted Jaccard loss function which prioritizes the False Safe states (FS, as opposed to False Hazardous, True Safes, or True Hazardous). In the given function (1),

$$L_{FS}(Y_{pred}, Y_{true}) = s\left(1 - \frac{TS + s}{TS + K * FS + FH + s}\right) \quad (1)$$

K is one of the more noteworthy weights controlling the importance of the FS while training. In practice, as the weight K is increased, the performance of the CNN is sacrificed for the improvement of the FS value. This trend continues until K = 35, at this point, increasing K weight doesn't affect the False State percentage values much.

Although the paper claims that increasing the weights can get the False Safe percentage down to $>1\%$. (8)In that case, the K value must be sufficiently large, affecting performance, although misclassifications occur at the class boundaries (8). Another drawback of the proposed solution is that the landing site must be chosen sufficiently far from the safe/hazardous class boundary to avoid the selection of the hazardous landing location (8).

Overall, the paper proposes a straightforward semantic segmentation solution for the hazard detection problem with a robust design for various lighting conditions. The added robustness gives the algorithm some degree of adaptability, although, given that the network is trained exclusively on Lunar DEMs, it will be interesting to see how efficient Transfer Learning will be to make the model more suitable for broad types of environment with different variables.

Another approach to the problem would be a network consisting of an encoder and decoder. (9) The combination of these elements gives us U-Net. The input comes from the camera, making the solution a Semantic Segmentation. While the output of the network is a labeled image with safe and/or unsafe areas. From here, the algorithm calculates the safest landing spot based on the longest average minimum distance from the hazard.(9) As the landing spot is recalculated for every frame, the suboptimal behavior of the agent is possible, which is solved with the adaptive searching window centered in the center of the frame. (9) As the lander approaches the ground, the search window decreases in size. (9)

Training is done via supervised learning, on the Digital Terrain Models dataset composed of 18000 images. Although the data are light-sensitive, the shadows are considered hazards and thus lack robustness in dynamic lighting conditions. (9) Otherwise, giving the other constraints to consider the spot (image) safe/unsafe yields the binary labeling of the given image as an output. (9) The mentioned output is generated through the actor-critic method, which seems to be a prominent method for the hazard detection neural network training phase.

3.2. Trajectory Optimization

For the trajectory optimization subproblem, we can approach the issue from different angles. As an introduction, in order to better see the effectiveness, advantages, and disadvantages of the different architectures, we can analyze the solution of OpenAI's Gym 2D LunarLander-v2 problem. As we are ditching 3rd dimension and thus some DOFs, the variable space is small and simplified. The 2 reinforcement learning techniques are considered: SARSA and deep Q-learning, On and Off policies, respectively. (6).

(a) Sarsa

For the Sarsa approach, state generalization is done to shrink the state space (there are 400,000 of them), after which the exploration policy is implemented in several stages, switching from exploration to exploitation after each batch of states.

(b) Deep Q-learning

DQN method uses a multilayer perceptron model; the current state is an input, while the outputs are the Q values for the state-action pairs for the given state (6).

It is observed that both architectures perform well with the regular problem and the one with the introduced uncertainties. Although the DQN agent gets consistently better rewards even with the uncertainties, although both agents experience drops in their rewards, DQN is more unstable which may be connected to the introduced randomness. The better benchmark of the DQN agent is connected to the fact that no state discretization is used to train the network, unlike for Sarsa (in this case we sacrifice some information).

It is noteworthy that the retrained Sarsa doesn't perform well when the sensor noise is introduced, instead, Partially Observable Markov Decision Process agents are used with more success, as we can frame the problem as a POMDP. Therefore, noise and uncertainties are better represented as the agent can use the belief vector to model a distribution over the possible next states and make more informed decisions. Although the success rate of the POMDP agent depends on the uncertainties, it is not useful for all situations.

To improve robustness and adaptability, some papers introduce Reinforcement Meta-Learning to train Guidance, Navigation, and Control (GNC) policy. (10). First, reinforcement learning enables us to describe the environment as a Markov Decision Process (MDP) where the current state depends on the previous state only. If we consider a continuous state space and a probability distribution describing the probability of changing the state depending on the reward function and action, we can assume the existence of the partially observable MDP. In this case, we can work with the sensor noise and limited, or nonexistent (malfunctioned) access to the modules.

RNN-LSTM-CNN architecture is also tested for the planar lunar lander problem to imitate the optimal trajectory and high fuel efficiency (1). CNN-LSTM tandem is used to:

(*a*) Classify the image from the onboard camera data stream through the neural network, and

(*b*) Act, learn to recognize an important input, learn to store and preserve it for as long as it is needed, and learn to extract information whenever is needed (1).

In this case, the given networks, although being computationally expensive, achieved an accuracy of 98.51% on thrust level prediction and RMSE of 0.76 o on the thrust direction, (1), proving that, the Deep Neural Networks can be successfully used in planetary lander problem.

Significant improvements are made in reinforcement learning autonomous planetary descent algorithms using proximal policy optimization (PPO) and different discount rates for the terminal and shaping rewards. (7) PPO leverages a heuristic to keep the KL divergence in an optimal state so that the monotonic improvement is guaranteed between the policy updates. (7) On top of that, the authors ditch the generalized advantage estimation in favor of multiple discount rates to better approximate the importance of future state rewards. By employing different discount rates for the terminal and shaping rewards, the algorithm can concentrate on intermediate goals, rather than tunnel vision on the end goal.

It is noteworthy that one of the main objectives for the given paper is implementability. 3-DOF and 6-DOF policies are trained with realistic hardware constraints in mind. On a 2.3GHz processor, it takes 1 ms for the 6-DOF policy to map the input to action, while using a more realistic, 100mhz processor can take 23 ms. It is worth saying that further optimization can be done using C++ to code the guidance and control system. (7)

The approach synthesizing the GNC policy as an RNN can also successfully deploy the classical actor-critic method: The critic evaluates the policy based on the generated sample trajectories at each global iteration. (9)

Contrary to the training phase, the architecture is different while testing, as the policy is tested as if it was deployed on an autonomous lander. Therefore, there is no need for the critic and instead of random landing site selection, we have a safe landing site selection and hazard detection/avoidance subroutine. (9)

This avoids using the sparse reward approach for the reward function. The principle behind it is that the agent is given hints to follow a good trajectory. In this occasion, the hint represents a gaze heuristic potential function, which achieves good spatial accuracy but cannot dictate the landing velocity. To solve this issue, the agent will try to minimize or reduce the targeted velocity as the ratio between the range and the velocity decreases. (9)

Besides the use of different discount rates for terminal and shaping rewards (7), or the use of the sparse reward approach for the reward function (9), the inherent shortcomings of the Deep reinforcement learning algorithm, those being poor convergence property and difficult reward function design, are being addressed with the actor-indirect method. The result is excellent convergence and high computational efficiency. This is achieved with an actor learning the optimal parameters from the classical indirect methods by deploying 5 deep neural subnetworks used for various costate variables, guaranteeing good initial shooting guesses. The unavoidable error during the terminal landing phase is solved using another feedback controller gradually taking over the command by gradually increasing the given weight of its actions as the spacecraft approaches the goal. (2)

The result is a high computational efficiency given that the actor-indirect method does not need to solve OCPs on board, unlike the actor-critic methods. Although the provided DNN performs poorly on its own, with a landing success of just 10%. However, the problem is solved using the DNN+Feedback architecture, since both of those techniques are independent, applying them as a compound proves to be highly successful in reaching the goal (although the fuel and time errors are tolerably higher). (2)

To improve robustness and adaptability, some articles introduce reinforcement metalearning to train the Guidance, Navigation, and Control (GNC) policy(9)(10). First of all, Reinforcement Learning enables us to describe the environment as a Markov Decision Process (MDP) where the current state depends only on the previous state only.(9)(10)(6)If we consider a continuous state space and a probability distribution describing the probability of changing the state dependent on the reward function and action, we can assume the existence of the partially observable MDP. In this case, as we mentioned above, we can work with the sensor noise and limited, or nonexistent (malfunctioned) access to the modules. The adaptability is achieved through the implementation of Meta-Learning with the following structure: (10)

- Sample task and collect data
- adapt the policy according to the collected data
- with a modified policy, gather new data
- update the parameters according to new data and the function of the sampled tasks MDP.

Besides robustness, there is another issue of training time. As RL algorithms take many trials to learn new information, the paper introduces Reinforcement Meta-Learning (RML) as a new framework. In that case, instead of starting over from zero every time we start training, the model already has some knowledge based on "experience". This is achieved with the recurrent neural networks thanks to their internal (or hidden) states; As the hidden states are constant through the task, weights learn the task using RL algorithms.(10)

One of the more unique approaches uses the already existing Zero-Effort-Miss/Zero-Effort-Velocity (ZEM/ZEV) guidance algorithm, which was a popular method for autonomous precise landings. Although simple and robust, the algorithm suffers from not being able to enforce thrust/flight constraints, therefore lacking flexibility. The new approach creates a close-loop, incorporating the ZEM/ZEV algorithm with machine learning to solve the given limitations. The overall idea is that the guidance gains and the time-to-go can be adapted during the powered descent phase to satisfy specific constraints while maintaining quasi-fuel optimality and close-loop characteristics. (4). Typical to the theme, the algorithm uses an actor-critic architecture, (4)(3) (10) (9) where the actor learns to optimally choose the aforementioned ZEM/ZEV guidance algorithm parameters, while critic evaluates the decision made by the actor. The resulting A-ZEM/ZEV algorithm is shown to outperform ZEM/ZEV and GPOPS algorithms while being scalable, implementable on-board, and flexible in different constraint scenarios. (4)

Improvements in adaptability are addressed by applying transfer learning to the Deep Deterministic Policy Gradient using the actor-critic model. (3) Similarly to other papers, the DDPG algorithm leverages the Ornstein-Uhlenbeck process (3), which is a stationary Gauss–Markov process,(9)(10)(6) to improve exploration efficiency in physical control problems.(3) That being said, the transfer learning task is a new improvement to the already existing, similar approaches to the autonomous planetary lander problem. It is implemented with noteworthy success during the tests done in simulations for several different planetary surfaces.

4. Open Challenges and Future Directions

As we have explored the various papers intending to analyze and summarize the existing Deep Learning techniques used to tackle the autonomous planetary lander problem, it becomes evident that there are several challenges left to address.

First, improvements in robustness in uncertain environments are always welcome, as only a few approaches are implemented in simulation, whereas most of the papers disregard the extremely uncertain and dynamic conditions and inherent hardware constraints. More detailed, dynamic simulators and/or Meta-learning can be one way to improve the robustness in miss-modeled dynamic environments.

Another approach would be XAI to decipher the inner workings of the algorithm to improve safety and decision-making. This information could be helpful in developing a new solution where an individual/group of neurons fail due to radiation-related hardware damage or Single Event Upsets, which are more probable in space.

Second, the model needs to be computationally efficient and lightweight, as most of the spacecrafts will not be equipped with strong and capable onboard CPUs. Using specifically designed lightweight libraries could be essential.

5. Conclusion

To sum up this survey, Deep Learning greatly enhances the autonomy of the planetary descent problem with high adaptability; pushing the frontiers of space exploration and efficient guidance. The new approaches redefine the classical autonomous guidance algorithms and make them accessible, highly adaptable, and optimal. Harnessing the power of machine learning can streamline and accelerate the multifaceted task of planetary exploration, cargo delivery, and settlement.

References

[1] Bloise, I., and Orlandelli, M. PhD thesis, 2017.

[2] Cheng, L., Wang, Z., and Jiang, F. Real-time control for fuel-optimal moon landing based on an interactive deep reinforcement learning algorithm. *Astrodynamics 3*, 4 (2019), 375–386.

[3] Ciabatti, G., Daftry, S., and Capobianco, R. Autonomous planetary landing via deep reinforcement learning and transfer learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2021), pp. 2031–2038.

[4] Furfaro, R., and Linares, R. Deep learning for autonomous lunar landing.

[5] FURFARO, R., SCORSOGLIO, A., LINARES, R., AND MASSARI, M. Adaptive generalized zem-zev feedback guidance for planetary landing via a deep reinforcement learning approach. *Acta Astronautica 171* (2020), 156–171.

[6] GADGIL, S., XIN, Y., AND XU, C. Solving the lunar lander problem under uncertainty using reinforcement learning. *CoRR abs/2011.11850* (2020).

[7] GAUDET, B., LINARES, R., AND FURFARO, R. Deep reinforcement learning for six degree-of-freedom planetary landing. *Advances in Space Research 65*, 7 (2020), 1723–1741.

[8] MOGHE, R., AND ZANETTI, R. A deep learning approach to hazard detection for autonomous lunar landing. *The Journal of the Astronautical Sciences 67*, 4 (2020), 1811–1830.

[9] SCORSOGLIO, A., D'AMBROSIO, A., GHILARDI, L., FURFARO, R., GAUDET, B., LINARES, R., AND CURTI, F. Safe lunar landing via images: a reinforcement meta-learning application to autonomous hazard avoidance and landing.

[10] SCORSOGLIO, A., FURFARO, R., LINARES, R., AND GAUDET, B. Image-based deep reinforcement learning for autonomous lunar landing. *AIAA Scitech 2020 Forum* (2020).

[11] SHIROBOKOV, M., TROFIMOV, S., AND OVCHINNIKOV, M. Survey of machine learning techniques in spacecraft control design. *Acta Astronautica 186* (2021), 87–97.

[12] SULLIVAN, C., AND BOSANAC, N. Aas 20-689 using multi-objective deep reinforcement learning to uncover a pareto front in multi-body trajectory design, 08 2020.

[13] XU, X., CHEN, Y., AND BAI, C. Deep reinforcement learning-based accurate control of planetary soft landing. *Sensors 21*, 23 (2021).