Deep Neural Networks Based Real-time Optimal Control for Lunar Landing

Lingchao Zhu\textsuperscript{1,2}, Jian Ma\textsuperscript{1,2} and Shuquan Wang\textsuperscript{1,*}

\textsuperscript{1}Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences, Beijing 100094, China
\textsuperscript{2}University of Chinese Academy of Sciences, Beijing 100049, China

\textsuperscript{*}Corresponding author: shuquan.wang@csu.ac.cn

Abstract. Recent research on deep learning control, a new control algorithm based on machine learning able to learn deep architectures, has shown excellent performance on robots and drones. With the development of intelligent control like deep learning and reinforcement learning, accuracy, real-time, adaptability, robustness and autonomy of control algorithm have been achieved by the intelligent controls. Traditional control methods have difficulties to achieve nice performance in complex situations. Deep learning offers powerful algorithms to real-time search near-optimal controllers of lunar landing spacecraft with nonlinear dynamics. In terms of lunar landing control system, deep architectures offer the possibility to get an approximate solution of co-state equation without time-consuming iterative process. Furthermore, real-time optimal thrust during lunar landing may be derived directly through deep neural networks. As a single infrastructure for machine learning in both production and research, TensorFlow is chosen for training the deep artificial neural networks in this paper. Numerical simulations demonstrate the effectiveness of deep neural networks. The results of deep neural networks based optimal control are contrasted with traditional optimal algorithm, whose main idea is to track the pre-designed optimal trajectory by ground station. This research provides an effective approach to cope with the lunar landing problem.

1. Introduction
The Moon is an astronomical body that orbits planet Earth and is Earth's only permanent natural satellite. Soft landing on large planetary bodies is a critical step in space exploration that enables future manned moon landing, lunar lander detection, and lunar base construction. The rise of strong willingness for sending humans back to the Moon within the next decade has also caused a lot of research boom. In particular, soft landing technology will continue to evolve to meet the demand for increasingly accurate and complex space exploration missions. Guidance and control are two independent systems for the mainstream landing system currently, wherein the guidance system designs an optimal trajectory by solving the Hamilton-Jacobi-Belmann (HJB) equations and the two point boundary value problem [1], and the goal of the control system is to control the spacecraft to track the optimal reference trajectory accurately. The improvement of direct and indirect methods [2] is the main research hotspot of optimal control problems (OCPs), the homotopy approach [3] could increase the probability of indirect methods in solving the optimal solutions. Additionally, a robust and precise control techniques such as PID algorithm, sliding-mode control [4] and terminal sliding-mode control [5] are still needed to track the reference trajectory if the optimal trajectory is solved. Thanks
to Moore’s Law that computing power has been boosted by several orders of magnitude and the cost has been greatly reduced over the years, it is possible to train a deep neural networks for real-time optimal control. Deep learning [6] has shown great excellent performance in feature extraction and model fitting in recent years. Its recognition rate in traditional recognition tasks has a significant increase [7].

For control systems with high dimensional data input, the introduction of deep neural networks (DNN) has a natural advantage. DNNs are artificial neural networks with an input layer, several hidden layers and an output layer, which can map the nonlinear relationship between input and output correctly [8]. Recently, some studies have focused on the application of deep learning in the field of aerospace control include two-dimensional moon optimal landings [9], two-dimensional quadcopter fuel optimal control, two-dimensional optimal orbit transfers, and asteroid irregular gravitational filed estimation using deep neural networks. A finite-time attitude tracking control scheme for spacecraft using Chebyshev neural network was proposed as the earliest research on introducing neural networks into spacecraft control [10]. An RBF neural network was used to adaptively compensate for the plant uncertainties or estimate unknown nonlinearities in an adaptive output feedback control scheme [11,12]. Stefan used neural reinforcement learning to control a spacecraft around a small celestial body with unknown gravity field [13], Dario proposed an early research of deep artificial neural networks to represent the optimal guidance profile of an interplanetary mission [14]. A deep Recurrent Neural Network and Long-Short Term Memory (RNN-LSTM) architecture was designed for predicting the fuel-optimal thrust from sequence of spacecraft’s states during a powered planetary descent phase [15].

In this paper, the landing scenario is researched assuming overall information on the landing spacecraft state. Due to the assumptions considered in the models, a real-time fuel optimal control to the moon soft landing is proposed by utilizing the indirect methods with homotopy approach and training the feed-forward DNNs architectures directly in a supervised manner on a large amount of optimal trajectory data. The trained DNNs are suitable for the on-board generation of optimal control signal with no need solve OCPs on-board.

The paper is structured as follows: in Section II we introduce the dynamic model of spacecraft and mathematical form of the fuel OCP for lunar soft landing problems, then we give the definition of the fuel optimal control policy to be learned by the DNNs. In the following Section III, the two-point boundary value problem (TPBVP) is derived and solved by means of single shooting and continuation techniques. The training data was filtered before saved and normalized for DNNs. In the following section IV the network architectures and training procedures used to approximate the optimal solutions are trained with Tensorflow. A numerical simulation scenario based on MATLAB is proposed and the results are compared with the optimal trajectories. This paper is concluded in Section V.

2. Spacecraft dynamics and problem formulation

In this section, we consider a Lunar landing problem, in which the state comprises three components: the spacecraft dynamical state position and velocity, and mass. The control signal will have two component, one is the control thrust amplitude and one is the thrust direction . The equations of motion are the following:

$$\dot{r} = v, \quad \dot{v} = c_1 \frac{u}{m} \hat{r} - \mu_m \frac{r}{r^3}, \quad \dot{m} = -c_2 u_1$$  (1)

Here, \(r\) and \(v\) are position and velocity vector of spacecraft in Lunar inertial coordinate system. \(m\) is the mass of the landing spacecraft, \(\mu_m = 4.901793455 \times 10^{12} \text{ m}^3/\text{s}^2\) is gravitational constant of Lunar. The constant \(c_2 = c_1 / I_{sp} g_0\) represents the rocket engine efficiency in terms of its impulse \(I_{sp} = 311 \text{s}\) and \(g_0 = 9.81 \text{m/s}^2\). The control \(u_1 \in [0,1]\) models a thrust amplitude applied along the direction \(\hat{r}\). For fuel optimization problem, \(u_1\) is thrust switch sequence that \(u_1 = 0\) means engine off and \(u_1 = 1\) means engine on with the maximum thrust magnitude \(c_1\).
Where these equations have been integrated to minimize the following cost function:

$$ J = (1 - \alpha) \int_0^{t_f} \gamma_1 c_i c_i u_i^2 dt + \alpha \int_0^{t_f} c_i u_i dt $$

(2)

where $\gamma_1 = 1/[1/N]$ is a trade-off coefficient for adjusting cost function between the two contributions. The parameter $\alpha \in [0, 1]$, defines a continuation between a fuel optimal control problem (MOC) $\alpha = 1$ and a quadratic optimal control problem (QC) $\alpha = 0$. Consider the following Hamiltonian to solve the optimal control problem:

$$ H = \dot{\lambda}_v + \lambda_m (c_1 u_1 - \mu) + \lambda_m c_2 u_1 + (1 - \alpha) \gamma_1 c_i c_i u_i^2 + \alpha c_i u_i $$

(3)

where we introduce several auxiliary co-state functions $\dot{\lambda}_v(t), \dot{\lambda}_m(t)$ and $\lambda_m(t)$. From the maximum principle, the thrust direction must be in the opposite direction of $\lambda$ for Hamiltonian function to be minimized, writing as follows:

$$ \dot{\lambda}_v = -\dot{\lambda}_v / \lambda_v $$

(4)

Note that in the corner case $\alpha = 1$, which correspond to a mass optimal control in this paper, the above expression results to be singular, it is necessary to rewrite $\dot{\lambda}_v$ by introducing the switching function as follows:

$$ \dot{\lambda}_v = \min \left( \max \left( \frac{c_i \lambda_v / c_i m + \lambda_m - \alpha}{2c_i (1 - \alpha)}, 0 \right), 1 \right) $$

(5)

Eventually, the following two points boundary value problem (TPBVP) is obtained by bring above equations into the dynamics model:

$$ \ddot{r} = v, \quad \ddot{v} = c_i \frac{u_i}{m} \dot{i}_\theta - \mu \frac{r}{r^3}, \quad \dot{m} = -c_i u_i $$

(8)

with boundary conditions $r_0, v_0, m_0$ at $t = 0$ and $r_f = R_{\text{land}}, v_f = 0$ and $\dot{\lambda}_m = 0$ at $t = t_f$, where $R_{\text{land}}$ is the position of the landing site.

3. Data generation

A fuel optimal trajectories data is obtained via continuation over parameter $\alpha$ through homotopy approach. Each optimal trajectory consists of a list of pairs $(x^*, u^*)$ where $x^* = [\Delta r, \Delta v, m]$ is the position error, velocity error and the mass of spacecraft, $u^*$ is the corresponding optimal thrust switch
sequence and its direction. For Lunar landing fuel-optimum control problem, to find the MOC solution, the corresponding quadratic control problem ($\alpha = 0$) is solved first and then a homotopy path is followed by continuously increasing $\alpha$ from 0 to 1, resulting in smooth changes of the control action. An initialization area $\mathcal{R}$ is defined to draw the initial conditions from $x_0 \in \mathcal{R}$. The initialization position area of $\mathcal{R}$ is a cuboid whose three-axis length are $[400,400,1000]$ meters, The initialization velocity area of $\mathcal{R}$ is a cuboid whose three-axis length are $[60,60,60]$ m/s. 80,000 optimal trajectories are generated, from each fuel-optimum trajectory 200 state-control pairs are uniformly selected along the trajectory and inserted in the training data.

It is worth noting that a good initial guess can greatly reduce the iterative solution time of the value of the co-state $\lambda_0$ and optimal control time $t_f$, thus it is worthy training an independent deep neural network with $\lambda_0$ and $t_f$ as the output of DNN and initial state $x_0$ as input. An initial value $\lambda_0$ and $t_f$ close to the optimal solution can be given directly through a well-trained DNN, this DNN will greatly reduce the time for data generation. Limited by the length of the paper, this part will only shows a brief results. The DNN for co-state estimation is trained by a feedforward, fully-connected neural network with 5 hidden layers, 32 units/layer, Tanh activation function, linear output layer and MSE loss function, The results of the MSE after 10000 epoch is less than $1.6 \times 10^{-5}$. It could satisfactorily map the nonlinear relationship between initial spacecraft’s state $x_0$ and initial co-state $\lambda_0$.

After we generate the data containing 16,000,000 optimal state action pairs, the distribution of position error in the data are uneven as illustrated by Figure 1. The rest of the data is also unevenly distributed. This is because the data is too concentrated near the landing site. To accelerate the training process of deep neural networks, data pruning can smooth data distribution. It should be noted that the data input and output are simultaneously pruned, the principle of data pruning has two aspects. The first is to pruning the trajectory far away from most trajectories, that is, to remove the position and velocity error of the corresponding trajectory. The other is to perform random resections in places where a large number of data are concentrated at the end of the landing.

![Figure 1. Position error distribution of raw data](image1)

The distribution of position error in the data after data pruning is illustrated in Figure 2. Normalize the data after pruning, then the finial state-action pairs are ready for DNNs to learn.

![Figure 2. Position error distribution after data pruning](image2)
4. DNN training and simulation results

The based on the fuel-optimum data generated by the indirect method with homotopy approach, DNN are used to learn the nonlinear relationship between the real-time state and optimal actions. Architectures with different numbers of layers and neurons per layer are considered. The control actions comprises two components, one is the magnitude of the thrust and the second one to the direction in which the thrust is applied. For the thrust magnitude, there are only two states (on for \( u_i = 1 \) and off for \( u_i = 0 \)) as it has a bang-bang control action profile. Accordingly, the deep neural network architecture should be designed as a classification neural network. Moreover, the thrust direction is smooth and continuous. Consequently, the network shall be designed as a regression network to learn the optimal thrust direction.

The selection of activation function is one of the most important factors of DNNs, there are several choices such as ReLUs, Tanh, sigmoid, ELU, Leaky ReLU, PReLU etc., after repeated iteration simulations, we conclude that a proper architecture can give quite accurate results, the architectures of classification neural network and regression network are can be found in Table 1. The accuracy of classification deep neural network can reach 99.55\%, test on data can at most reach a precision of 99.82\%. The MSE of regression deep neural network can be as low as \( 6 \times 10^{-4} \) after 5000 epochs.

|          | layers | activation function | Output layer | Loss function      | Batch size | Learning rate | Optimizer |
|----------|--------|---------------------|--------------|--------------------|------------|---------------|-----------|
| classification | 6-30   | ReLU                | ReLU         | Cross-entropy      | 10000      | 0.001         | Adam      |
| regression   | 4-12   | tanh                | Linear       | MSE                | 1000       | 0.001         | SGD       |

The DNN-driven trajectories are evaluated as described in this section to demonstrate the performance. The landing site \( \mathbf{R}_{\text{land}} \) is chosen as the North Pole of the moon, the expected velocity on the landing site is zero for a soft landing demand. The simulation parameters are given in section 2. Landing simulations are shown in Figure 3.

**Figure 3.** The optimal and DNN-driven trajectories

*Figure 3* shows the optimal and DNN-driven trajectories, the left subgraph displays the trajectories with equal ratio of axis, the right subgraph displays the magnified view. As we can see from the two subgraphs, the approximate optimal solutions obtained by the trained DNNs are very close to the
optimal trajectory. We further quantitatively analyze the DNN-driven trajectory and find that the trajectory deviates from the optimal path with a maximum error of 5.963m. Therefore, a multi-scale DNN driven and nonlinear control cooperation strategy is proposed for lunar landing. Figure 4 show the position and velocity error where the blue line is the DNN driven phase and the red line is the nonlinear control, the switching strategy is to switch to a non-linear control phase when the spacecraft reaches five meters above the landing site. Figure 5 displays the switch sequence and its direction of trust, the statistical results of terminal fight errors is illustrated by Table 2.

Figure 4. Position and velocity error of cooperation strategy

Figure 5. Switch sequence and its direction of cooperation strategy

| Position error(m) | -1.732×10^{-7} | -6.303×10^{-8} | -4.657×10^{-9} |
|-------------------|----------------|----------------|----------------|
| Velocity error(m/s) | -1.733×10^{-7} | 6.38×10^{8} | 4.66×10^{9} |

As we can see from above simulation results that the DNN-based landing controller cooperated with nonlinear control is robust and achieves an almost fuel-optimum trajectory. The proposed DNN-driven controller can replace the time-consuming homogeneous polyhedra method and significantly improve the computational efficiency.

5. Conclusions
In this study, deep neural network technology is introduced to improve the computational efficiency for the fuel-optimum lunar landing problem. Generally, fuel-optimum problem can be found only for open-loop trajectories, closed-loop guidance algorithms are not available. By taking advantage of classification and regression neural networks with excellent generalization ability executed on Tensorflow, a real-time multi-scale DNN-driven and nonlinear control cooperation strategy is
proposed for the landing problem. This DNN-driven optimal controller is trained off-line and outputs the real-time optimal control actions in the on-line application stage. Terminal landing accuracy has been greatly improved by introducing a control switching strategy. Simulation results of fuel-optimum lunar landing problems are given to substantiate the effectiveness of real-time performance and illustrate the reliability of the multi-scale DNN driven and nonlinear control cooperation control strategy.

Acknowledgement
This research was supported by the National Natural Science Foundation of China (under grant No. 11672294) and the Advanced Research Project China's Manned Space Program.

References
[1] Todorov, E., “Optimality principles in sensorimotor control,” Nature neuroscience, Vol. 7, No. 9, 2004, pp. 907–915, doi:DOI:10.1038/nn1309.
[2] J. T. Betts, “Survey of numerical methods for trajectory optimization,” Journal of guidance, control, and dynamics, vol. 21, no. 2, pp. 193–207, 1998.
[3] C. Zhang, F. Topputo, F. Bernelli-Zazzera, and Y.-S. Zhao, “Low-thrust minimum-fuel optimization in the circular restricted three-body problem,” Journal of Guidance, Control, and Dynamics, vol. 38, no. 8, pp. 1501–1510, 2015.
[4] Xiao, Bing, Qinglei Hu, and Youmin Zhang. "Adaptive sliding mode fault tolerant attitude tracking control for flexible spacecraft under actuator saturation." IEEE Transactions on Control Systems Technology 20.6 (2012): 1605-1612.
[5] Lee, Daero, and George Vukovich. "Adaptive finite-time control for spacecraft hovering over an asteroid." IEEE Transactions on Aerospace and Electronic Systems 52.3 (2016): 1183-1196.
[6] Schmidhuber, Jürgen. "Deep learning in neural networks: An overview." Neural networks 61 (2015): 85-117.
[7] Hunt, K. Jetal, et al. "Neural networks for control systems—a survey." Automatica 28.6 (1992): 1083-1112.
[8] Liu, Weibo, et al. "A survey of deep neural network architectures and their applications." Neurocomputing 234 (2017): 11-26.
[9] Sánchez-Sánchez, Carlos, and Dario Izzo. "Real-time optimal control via Deep Neural Networks: study on landing problems." Journal of Guidance, Control, and Dynamics 41.5 (2018): 1122-1135.
[10] Zou, An-Min, et al. "Finite-time attitude tracking control for spacecraft using terminal sliding mode and Chebyshev neural network." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 41.4 (2011): 950-963.
[11] Seshagiri, Sridhar, and Hassan K. Khalil. "Output feedback control of nonlinear systems using RBF neural networks." IEEE Transactions on Neural Networks 11.1 (2000): 69-79.
[12] Li, Yahui, et al. "Robust and adaptive backstepping control for nonlinear systems using RBF neural networks." IEEE Transactions on Neural Networks 15.3 (2004): 693-701.
[13] Willis, Stefan, Dario Izzo, and Daniel Hennes. "Reinforcement learning for spacecraft maneuvering near small bodies." AAS/AIAA Space Flight Mechanics Meeting. 2016.
[14] Izzo, Dario, Christopher Sprague, and Dharmesh Tailor. "Machine learning and evolutionary techniques in interplanetary trajectory design." arXiv preprint arXiv:1802.00180 (2018).
[15] Furfaro, Roberto, et al. "A Recurrent Deep Architecture for Quasi-Optimal Feedback Guidance in Planetary Landing." IAA SciTech Forum on Space Flight Mechanics and Space Structures and Materials. 2018.