FEATURE-BASED LEARNING FOR OPTIMAL ABORT GUIDANCE

by

Vinay Kenny

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science

School of Aeronautics and Astronautics

West Lafayette, Indiana

August 2022

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Ran Dai, Chair

School of Aeronautics and Astronautics

Dr. Kathleen Howell

School of Aeronautics and Astronautics

Dr. James Garrison

School of Aeronautics and Astronautics

Approved by:

Dr. Gregory Blaisdell

Dedicated to my Mom, Pushpalatha Kenny

and

Dad, Vilas Kenny

ACKNOWLEDGMENTS

Firstly, I would like to thank my mom and dad for giving me the opportunity to follow my dreams and continuously supporting me. I wouldn't be where I am today if not for your sacrifices. You have always provided me confidence in the most difficult of times and I am eternally grateful for that.

Next, I would like to extend my immense gratitude to my advisor, Dr. Ran Dai, for believing in me and agreeing to guide me through this research. Your suggestions and feedback were very valuable to me in completing this research. I would also like to thank my committee members, Dr. Kathleen Howell and Dr. James Garrison, for their feedback and support.

I am also grateful for the support from my friends at Purdue. I would like to thank Monish, Abbas, Harshit, Pei, Sean, Myungjin for continuously supporting me through this period. Lastly, I would like thank the Department of Aeronautics and Astronautics at Purdue and Dr. Ran Dai for providing me with a teaching and research assistantship to overcome my financial obligations at Purdue.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

$\mathbf{g}$	gravitational acceleration vector, m/s^2
m	landing vehicle mass, kg
$\mathbf{r}$	position vector, m
r_{pm}	primary mass radius, m
t	time, s
$\mathbf{T}$	thrust vector, N
$\mathbf{1_T}$	unit thrust vector direction, N
$\mathbf{v}$	landing vehicle velocity vector, m/s
H	hamiltonian
L	lagrangian
ϕ	endpoint cost

ABBREVIATIONS

OAG	optimal abort guidance
TOAG	time optimal abort guidance
FOAG	fuel optimal abort guidance
LM	lunar module
CSM	command service module
PMP	pontriyagin's maximum principle
GLCC	generalized legendre-clebsh condition
BVP	boundary value problem
TPBVP	two point boundary value problem
DNN	deep neural network
EDL	entry, descent and landing
TRL	technology readiness level
NLP	non-linear programming

ABSTRACT

The abort mission refers to the mission where the landing vehicle needs to terminate the landing mission when an anomaly happens and be safely guided to the desired orbit. Missions involving crew on board demands for a robust and efficient abort strategy. This thesis focuses on solving the time-optimal abort guidance (TOAG) problem in real-time via the feature-based learning method. First, according to the optimal control theory, the features are identified to represent the optimal solutions of TOAG using a few parameters. After that, a sufficiently large dataset of time-optimal abort trajectories is generated offline by solving the TOAG problems with different initial conditions. Then the features are extracted for all generated cases. To find the implicit relationships between the initial conditions and identified features, neural networks are constructed to map the relationships based on the generated dataset. A successfully trained neural network can generate solution in real time for a reasonable initial condition. Finally, experimental flight tests are conducted to demonstrate the onboard computation capability and effectiveness of the proposed method.

1. INTRODUCTION

Technological advancements in space science have made much progress in the last few decades, ranging from robotic exploration missions to sample-return missions. Still, it has been almost five decades since humans have stepped foot on the Moon. With a bold move, Mission Artemis, plans to take humans to the Moon by 2024 and then to Mars and beyond [1]. The mission would be similar to the Apollo program but with a larger crew. With many human lives involved in the mission, the crew's safety is of prime importance. This makes the abort phase a crucial part of any space flight in case of anomalies during the mission. The recent travel of the perseverance rover to Mars had the terminal section of the mission termed as "6 minutes of terror", i.e., the entry, descent, and landing (EDL) phases, as the sequence happens faster than the time radio signals take to reach Earth from Mars [2]. The base station cannot communicate with the spacecraft in time to remotely correct course errors. In situations like this, the onboard computer must be able to handle the landing process autonomously. Missions with crew onboard become more critical and challenging and demand guidance laws that can safely guide the space vehicle to the desired terminal conditions without any human interventions. The most difficult abort scenario is during the powered descent phase, where the landing vehicle has to be steered from a nominal descent trajectory to an ascent trajectory, which often involves aggressive maneuvers. With such a short mission duration for the powered descent phase along with the uncertainty associated with the abort time, it is required to have a robust guidance method that can quickly generate abort control commands in real time and autonomously guide the vehicle to the desired orbit. This thesis aims to address the challenges described above by developing a robust and efficient method to solve an optimal abort guidance (OAG) problem in real time. Typically, the fuel or time consumption is considered as an objective function to minimize in the OAG problem.

1.1 Apollo Abort Guidance Logic

The Apollo guidance logic for ascent was used to guide the Lunar Module (LM). The logic just utilizes attitude control for guidance and does not require engine throttling. Three

types of ascent guidance programs have been used, namely P12, P70 and P71. P12 guidance was used to ascent from ground, while P70 and P71 were guidance laws used for ascent aborts during descent. The major difference between the abort systems is that P70 uses descent propulsion system, while P71 utilizes ascent propulsion system (APS). These systems comprise of two phases, the vertical rise and insertion phase. In the vertical rise phase, the current attitude is held for 2 seconds to clear the powered descent stage. Then heading is moved to the desired azimuth. The phase terminates when the attitude rate is greater than 40 fps upwards or the altitude is greater than 25,000 feet. The insertion phase requires to achieve the desired orbit from which the LM can rendezvous with the command and service module (CSM). The guidance law was determined based on the current state and the target [3]. In scenarios when a failure occurs during the descent stage, the ascent propulsion system is used to inject the LM into a safe orbit to rendezvous with the CSM.

1.2 Optimal Control Problem

Optimal control refers to determining the control strategy for a dynamical system under some system constraints while optimizing the system objective [4]. Any optimal control problem can be defined as the set of expressions stated by

$$\begin{aligned} \min_{\mathbf{u},t_f} \quad & J = \phi(t_f, \mathbf{x}(t_f)) + \int_{t_0}^{t_f} L(t, \mathbf{x}, \mathbf{u}) dt, \\ \text{s.t.} \quad & \dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{u}), \\ & \mathbf{h}_{\mathrm{i}}(t, \mathbf{x}, \mathbf{u}) \leq 0, \, \mathrm{i} = 1, 2, \ldots, h_p, \\ & \mathbf{x}(0) = \mathbf{x}_0, \;\; \mathbf{x}(t_f) = \mathbf{x}_f, \end{aligned} \tag{1.1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $\mathbf{u} \in \mathbb{R}^m$ is the control vector, t refers to time, J is the objective function, ϕ represents end point cost, L represents the lagrangian, $\mathbf{f}$ refers to the system dynamics, $\mathbf{x}_0$ represents value of the initial state vector, $\mathbf{x}_f$ represents value of the final state vector and, $\mathbf{h}_{\mathrm{i}}(t, \mathbf{x}, \mathbf{u}) \leq 0, \mathrm{i} = 1, 2, \ldots, h_p$, represents the path constraints and p is the number of path constraints. The path constraints can be either state or control constraint. The objective is to find the history of control variable(s) that will minimize or

maximize a given cost function J, while satisfying the system dynamics, path and boundary constraints.

The methods for solving optimal control problems are broadly classified into two major categories, namely, direct and indirect methods [5], [6]. Indirect methods utilize the concept of the calculus of variations to obtain the first order optimality conditions, leading to a two-point-boundary-value (TPBVP) problem [7]. For this method, analytical solutions are available only for a few exceptional cases due to the complexities involved in dynamics. Several numerical techniques have been developed to solve this type of problem, but the convergence of those methods cannot be guaranteed due to the sensitivity of the adjoint variables. Optimal abort guidance strategies have been developed based on indirect methods. For example, authors in [8] divide the abort scenario into two separate phases, pull-up phase and ascent phase. In the pull-up phase the vehicle is guided from its initial state obtained from the powered descent phase to a state having a positive flight path angle, where a constant velocity is maintained during the pull-up phase. The ascent phase is to guide the vehicle to ascent from the end of the pull-up phase to the injection orbit using minimum fuel. The guidance algorithm for the ascent phase is based on an indirect method that can handle diverse targeting conditions [9]. The authors in [8] also observed some convergence issues at a lower altitude, which was addressed in [10].

In contrast, for the direct methods, the states and the controls are discretized, and the original problem is converted into a finite dimension nonlinear programming (NLP) problem that is solved using nonlinear optimization techniques [11], [12]. The solution from direct methods gives a good approximation of the optimal solution and is locally optimal [13]–[15]. Even though direct methods based on NLP have be implemented in many applications, it often requires a good initial guess of unknown parameters. The obtained discrete form solution requires further processing, such as interpolation, for the onboard implementation.

A comparison between both of these methods with regards to several attributes are tabulated in Table 1.1.

Table 1.1. Comparison between direct and indirect methods of optimization.

Attribute	Direct Methods	Indirect Methods
Ease of implementation	Easy	Difficult
Sensitivity to initial guess	Low	High
Handling path constraints	Easy	Difficult
Convergence accuracy	Can be inaccurate	Accurate

1.3 Optimal Control Theory

For the indirect method, Hamiltonian is utilized to generate the necessary conditions of optimality.

$$H = \lambda^T(t)\mathbf{f}(t, \mathbf{x}, \mathbf{u}) - L(t, \mathbf{x}, \mathbf{u}). \tag{1.2}$$

$$\dot{\lambda} = -\frac{\partial H}{\partial \mathbf{x}}. \tag{1.3}$$

$$\frac{\partial H}{\partial \mathbf{u}} = 0. \tag{1.4}$$

Equation (1.2) defines the Hamiltonian, where λ is the costate vector and having its dynamics governed by (1.3). Equation (1.4) is used to determine the optimal control law. Equations (1.3)–(1.4) define the necessary conditions of optimality.

Pontryagin's Maximum Principle (PMP) is used to solve the TPBVP to find the optimal control solution, and is expressed as

$$H(t, \mathbf{x}^*(t), \mathbf{u}^*(t), \lambda^*(t)) \geq H(t, \mathbf{x}^*(t), \mathbf{u}(t), \lambda^*(t)), \tag{1.5}$$

where * refers to the optimal value [7]. The principle states that the optimal control will maximize the Hamiltonian, H, for all admissible sets of controls.

In general, the Hamiltonian is not an explicit function of time which implies the value of the Hamiltonian at the terminal point should be constant for the entire time. The transversality condition is used to determine the constant value of the Hamiltonian, written as

$$\left[H(t_f) + \frac{\partial \phi}{\partial t_f}\right] dt_f + \left[\lambda(t_f)^T + \frac{\partial \phi}{\partial \mathbf{x}_f}\right] dx_f = 0. \tag{1.6}$$

When the Hamiltonian includes a scalar control, u, shown below and the control is bounded between -k and +k [7].

$$H = H_0(t, \mathbf{x}(t), \lambda(t)) + H_1(t, \mathbf{x}(t), \lambda(t))u, \tag{1.7}$$

As, H is linear with respect to u, (1.4) leads to

$$\frac{\partial H}{\partial u} = H_1. \tag{1.8}$$

Any choice of u cannot influence $H_0(t, \mathbf{x}^*(t), \lambda(t))$ and therefore is ignored in the process of obtaining the solution. If H_1 is a positive number, then the solution has the maximum value of u to maximize H. Similarly, if H_1 is less than 0, we choose the minimum value of u. In summary, the PMP gives the control law shown below,

$$u = \begin{cases} k & \text{if } H_1 > 0, \\ \text{unknown} & \text{if } H_1 = 0, \\ -k & \text{if } H_1 < 0. \end{cases} \tag{1.9}$$

The coefficient $H_1(t, \mathbf{x}^*(t), \lambda(t))$ is often referred to as the switching function. When H_1 oscillates between positive and negative values, the control law switches back and forth between its upper and lower bounds, resulting in a "bang-bang" structure. If $H_1 = 0$ for a non-trivial time interval between [t_0,t_f], then a singular solution is obtained for the interval when $H_1 = 0$ and the control no longer influences H, implying the solution is not unique.

$$\frac{d}{dt}(H_u) = 0 = \frac{\partial H_u}{\partial t} + \frac{\partial H_u}{\partial x}\dot{x} + \frac{\partial H_u}{\partial \lambda}\dot{\lambda} + \frac{\partial H_u}{\partial u}\dot{u}. \tag{1.10}$$

$$(-1)^q \frac{\partial}{\partial u}\left(\frac{d^{2q}H_u^*}{dt^{2q}}\right) \geq 0. \tag{1.11}$$

For determining unique candidate singular controls, we differentiate (1.4) with respect to time as shown in (1.10). Here, the last term in (1.10) vanishes because the derivative of hamiltonian with respect to the control is a constant.

The solution generates $\frac{d^2}{dt^2}H_u = 0$, and if u appears explicitly, we use (1.10) along with the Generalized Legendre-Clebsch Condition (GLCC) (1.11) to determine the singular control law. In case if u does not appear explicitly, the solution process takes even time derivatives of H_u until the process satisfies GLCC along with the conditions specified by odd time-derivatives of H_u [7] to obtain the singular control law.

1.4 Machine Learning

Machine learning methods, a branch of artificial intelligence [16], have been applied to solve real-time optimal control problems [17], [18]. Machine learning is broadly classified into three categories, namely, supervised learning, unsupervised learning, and reinforcement learning. Supervised learning has been used to solve several optimal control problems and is based on a simple idea to learn from the examples [19], [20]. A deep neural network (DNN) in [21] has been applied to map the relationship between the initial conditions and the optimal solution for the powered descent problem. Our prior work in [22] demonstrate the ability of the supervised learning method to solve an optimal control problem real time along with the onboard implementation for the powered descent guidance problem. In addition, image-based deep reinforcement learning has been applied for autonomous lunar landing [23].

In this thesis report, a feature-based learning method that combines supervised learning and optimal control theory is applied to solve the OAG problem in real time. Inspired by the learning-based optimal control method developed in [24], combing the supervised learning and optimal control theory greatly reduces the learning space while obtaining optimality supported by the Pontryagin's maximum principle. By formulating the time-minimum abort problem into a TPBVP, the features can be identified via the Pontryagin's maximum prin-

ciple. By generating a sufficiently large dataset and then constructing DNNs, we aim to find the relationship between the initial conditions of the spacecraft along the powered descent phase and the values of identified features that represent the time-optimal solution of abort guidance associated with a corresponding initial condition. We start with the time-optimal abort guidance (TOAG) problem and prove that under certain conditions, the TOAG is equivalent to the fuel-optimal abort guidance (FOAG) problem.

To verify the real time computational performance of the proposed method, an experimental flight testbed that leverages the existing consumer drone market ecosystem was constructed to mimic the powered descent and abort operations. The customized drone testbed drives down observed design complexity and cost while providing a flyable platform for testing the optimal abort guidance algorithm. Furthermore, it closely matches the dynamics of real rocket-powered vehicle in landing and abort missions even in the Earth environments. At the end, both simulation and experimental results are provided and compared with the state-of-art method.

This thesis is based on my previous publication [25]. The thesis organization is as follows. Section 2 presents the formulation of the TOAG problem. Section 3 explains the steps involved in feature based learning method. The experimental setup is described in Section 4. Section 5 provides both virtual simulation and experimental results of the TOAG problem. Finally, the conclusions are presented in 6.

2. PROBLEM FORMULATION

The Moon abort mission is considered here, where the three-dimensional point mass equations of motion for a rocket-powered vehicle in the Cartesian coordinate system, fixed at the Moon's center, are expressed as

$$\dot{\mathbf{r}} = \mathbf{v}, \tag{2.1a}$$

$$\dot{\mathbf{v}} = \mathbf{g}(\mathbf{r}) + T\frac{\mathbf{1_T}}{m}, \tag{2.1b}$$

$$\dot{m} = -\frac{T}{v_{ex}}, \tag{2.1c}$$

where $\mathbf{r} \in \mathbb{R}^3$ and $\mathbf{v} \in \mathbb{R}^3$ represent the position and the velocity vector, respectively, m is the vehicle mass at any time, T is the thrust magnitude generated by the rocket, and $\mathbf{1_T}$ represents the unit vector along the thrust direction. The exhaust velocity of the rocket is given by v_{ex} and calculated as $v_{ex} = I_{SP} \times g_e$, where I_{SP} is the specific impulse of the rocket in seconds and g_e is the acceleration magnitude due to gravity on Earth ($9.81 m/s^2$). The gravity acceleration vector, denoted by $\mathbf{g} \in \mathbb{R}^3$, is a function of $\mathbf{r}$, expressed as

$$\mathbf{g} = -\mu\mathbf{r}/\|\mathbf{r}\|^3, \tag{2.2}$$

where μ is the gravitational parameter of the primary mass. Additionally, the thrust magnitude at any point is bounded, such that

$$0 \leq T_{min} \leq T \leq T_{max}, \tag{2.3}$$

where T_{min} and T_{max} are the minimum and maximum thrust magnitudes, respectively. The thrust vector $\mathbf{T} \in \mathbb{R}^3$ can also be written as $\mathbf{T} = [T_x, T_y, T_z]^T$.

The abort mission can begin at any point during the powered decent phase, and the initial position, velocity, and mass at the beginning of abort can be expressed as

$$\mathbf{r}(t_0) = \mathbf{r_0},\ \mathbf{v}(t_0) = \mathbf{v_0},\ m(t_0) = m_0, \tag{2.4}$$

where t_0 refers to the starting time of the abort mission. The abort mission has to safely guide the space vehicle to a desired final orbit. The required final orbital conditions can be expressed as,

$$(\mathbf{r}_f \times \mathbf{v}_f)^T(\mathbf{r}_f \times \mathbf{v}_f) - \mu a^*(1 - \mathrm{e}^{*2}) = 0, \tag{2.5a}$$

$$\frac{\|\mathbf{v_f}\|^2}{2} - \frac{\mu}{\|\mathbf{r_f}\|} + \frac{\mu}{2a^*} = 0, \tag{2.5b}$$

$$\mathbf{1}_z^T(\mathbf{r}_f \times \mathbf{v}_f) - \|\mathbf{r}_f \times \mathbf{v}_f\| \cos \mathrm{i}^* = 0, \tag{2.5c}$$

where $\mathbf{1_z}$ is a unit vector along the primary mass's polar axis and pointing toward the north, and $\mathbf{r_f}$ and $\mathbf{v_f}$ refers to the terminal position and velocity of the abort mission. Besides, a^*, e^* and i^* are semi-major axis, eccentricity and inclination of the final orbital, respectively. In addition, the mass at the terminal time, denoted as m_f, needs to satisfy $m_f \geq m_{\text{dry}}$, where m_{dry} denoting the structure mass. The objective of the abort guidance is to reach the desired orbit with minimum time while satisfying the dynamical and operational constraints. Then the cost is determined by the terminal time, t_f. Therefore, the TOAG problem can be written as

$$\min_{\mathbf{T}, t_f} \; t_f, \tag{2.6a}$$

$$s.t., \; \dot{\mathbf{r}} = \mathbf{v}, \; \dot{\mathbf{v}} = \mathbf{g}(\mathbf{r}) + T\frac{\mathbf{1_T}}{m(t)}, \; \dot{m} = \frac{-T}{v_{ex}}, \tag{2.6b}$$

$$0 \leq T_{\min} \leq T \leq T_{\max}, \tag{2.6c}$$

$$m(t_0) = m_0, \; \mathbf{r}(t_0) = \mathbf{r}_0, \; \mathbf{v}(t_0) = \mathbf{v}_0, \tag{2.6d}$$

$$(\mathbf{r}_f \times \mathbf{v}_f)^T(\mathbf{r}_f \times \mathbf{v}_f) - \mu a^*(1 - \mathrm{e}^{*2}) = 0, \tag{2.6e}$$

$$\frac{\|\mathbf{v_f}\|^2}{2} - \frac{\mu}{\|\mathbf{r_f}\|} + \frac{\mu}{2a^*} = 0, \tag{2.6f}$$

$$\mathbf{1}_z^T(\mathbf{r}_f \times \mathbf{v}_f) - \|\mathbf{r}_f \times \mathbf{v}_f\| \cos \mathrm{i}^* = 0, \tag{2.6g}$$

$$m_f \geq m_{\text{dry}}. \tag{2.6h}$$

In the following section, the feature-based learning method for solving the TOAG problem formulated in (2.6) is introduced in detail.

3. FEATURE-BASED LEARNING METHOD

In this section, the feature-based learning method is described in detail. First, via the optimal control theory, the features of the optimal solutions for the TOAG problem are identified. Then, a dataset is generated offline, where the features are extracted from the offline optimal solutions of problem (2.6). After that, neural networks are constructed to map the relationship between inputs and outputs. Finally, the framework of the proposed method is summarized.

3.1 Feature Identification for TOAG

Based on the optimal control theory, the Hamiltonian of the TOAG problem in (2.6) is expressed as

$$H = {\lambda_{\mathbf{r}}}^T \mathbf{v} + {\lambda_{\mathbf{v}}}^T \left(\mathbf{g}(\mathbf{r}) + \frac{T}{m} \mathbf{1_T} \right) - \lambda_m \frac{T}{v_{ex}} - 1, \tag{3.1}$$

where $\lambda_{\mathbf{r}} \in R^3$, $\lambda_{\mathbf{v}} \in R^3$, $\lambda_m \in R$ are associated costate variables. Then the necessary conditions for the costates are written as

$$\dot{\lambda}_{\mathbf{r}} = -\frac{\partial H}{\partial \mathbf{r}} = \frac{\mu}{r^3} \lambda_{\mathbf{v}} - \frac{3\mu}{r^5} ({\lambda_{\mathbf{v}}}^T \mathbf{r}) \mathbf{r}, \tag{3.2a}$$

$$\dot{\lambda}_{\mathbf{v}} = -\frac{\partial H}{\partial \mathbf{v}} = -\lambda_{\mathbf{r}}, \tag{3.2b}$$

$$\dot{\lambda}_m = -\frac{\partial H}{\partial m} = \frac{T {\lambda_{\mathbf{v}}}^T \mathbf{1_T}}{m^2}. \tag{3.2c}$$

The optimal direction of the unit thrust direction '$\mathbf{1_T}$' to maximize H is given by [8]

$$\mathbf{1_T} = \frac{\lambda_{\mathbf{v}}}{\|\lambda_v\|}. \tag{3.3}$$

The switching function S, that governs the magnitude of thrust at any instant, is defined as

$$S = \frac{\partial H}{\partial T} = \frac{1}{m} {\lambda_{\mathbf{v}}}^T \mathbf{1_T} - \frac{\lambda_m}{v_{ex}} = \frac{\|\lambda_{\mathbf{v}}\|}{m} - \frac{\lambda_m}{v_{ex}}. \tag{3.4}$$

As S is independent of the control T, the optimal thrust magnitude can be expressed as

$$T(t) = \begin{cases} T_{min}, & S < 0, \\ T_{max}, & S > 0, \\ \in [T_{min}, T_{max}], & S = 0. \end{cases} \tag{3.5}$$

For the TOAG problem, to show that the optimal thrust magnitude is a bang-bang profile, and the optimal thrust $T = T_{\max}$ during the whole abort mission, we have the following proposition:

Proposition 1. *For the TOAG problem (2.6), the optimal thrust magnitude T is a bang-bang profile, and $T = T_{max}$ exists during the entire abort mission.*

Proof. First, we will prove that the optimal thrust magnitude T is a bang-bang profile, which is obtained by showing that the singular arc with $S = 0$ is not optimal for the TOAG problem (2.6). Assume $S = 0$ exists in a non-trivial time, then (3.4) can be expressed as

$$S = \frac{\|\lambda_{\mathbf{v}}\|}{m} - \frac{\lambda_m}{v_{ex}} = 0. \tag{3.6}$$

As from (3.3), we have $\dot{\lambda}_m = \dfrac{T\|\lambda_{\mathbf{v}}\|}{m^2} \geq 0$, which indicates that λ_m is monotonically increasing. Additionally, according to the first-order transversality condition of optimality, we have

$$\lambda_m(t_f) = \frac{\partial \Phi}{\partial m} = 0, \tag{3.7}$$

where Φ refers to endpoint cost. In conclusion, λ_m should be non-positive during the entire abort mission, which can be expressed as $\lambda_m \leq 0$. Therefore, $S = 0$ if and only if $\lambda_m = 0$ and $\|\lambda_{\mathbf{v}}\| = 0$. Combined with (3.2), we can derive that $\lambda_r = \mathbf{0}$. Therefore, once $S = 0$, all costate vectors will be zero, which violates the pontryagin's maximum principle [26]. In conclusion, a singular arc where $S = 0$ is not an optimal solution, which indicates that the optimal thrust magnitude is a bang-bang profile is proved.

Then, we will prove that the optimal thrust of the time-minimum abort problem should always be $T = T_{\max}$. As $S = \dfrac{\|\lambda_{\mathbf{v}}\|}{m} - \dfrac{\lambda_m}{v_{ex}} \geq 0$ and a singular arc is not an optimal solution, the optimal thrust magnitude can only be $T = T_{\max}$. Therefore, This completes the proof of Proposition 1. □

Therefore, once we know the thrust direction $\mathbf{1_T}$ and optimal terminal time t_f, we can reconstruct the optimal control solution, where the optimal control magnitude is always the maximum value, T_{max}. From (3.3), it can be found that thrust direction $\mathbf{1_T}$ is parallel to the vector direction of $\lambda_{\mathbf{v}}$. It indicates that once $\lambda_{\mathbf{v}}$ is solved, the thrust direction $\mathbf{1_T}$ is also obtained. For the TOAG problem (2.6), it is not straightforward to calculate $\lambda_{\mathbf{v}}$, as the adjoint equations in (3.2) are nonlinear and coupled with each other. As the altitude of the space vehicle, $\|r\| - r_{pm} << r_{pm}$, where r_{pm} denotes the radius of primary mass, the magnitude of position vector can be approximated to be the radius of the primary mass, i.e., $\|r\| \approx r_{pm}$. Therefore, an assumption based on this fact is applied to the TOAG problem.

Assumption 1. *In the gravity acceleration vector expression, we assume that* $\|r\| \approx r_{pm}$. *Then, we have* $\mathbf{g}(\mathbf{r}) = -\alpha\mathbf{r}$, *where* $\alpha = \mu/{r_{pm}}^3$ *is a constant.*

Based on Assumption 1, the expressions of $\dot{\lambda}_{\mathbf{r}}$ and $\dot{\lambda}_{\mathbf{v}}$ in (3.2) are reduced to

$$\dot{\lambda_{\mathbf{r}}} = -\frac{\partial H}{\partial \mathbf{r}} = \alpha\lambda_v, \tag{3.8a}$$

$$\dot{\lambda_{\mathbf{v}}} = -\frac{\partial H}{\partial \mathbf{v}} = -\lambda_{\mathbf{r}}. \tag{3.8b}$$

By integrating (3.8), we obtain

$$\lambda_{\mathbf{r}} = \mathbf{c_1}\cos\sqrt{\alpha}t + \mathbf{c_2}\sin\sqrt{\alpha}t, \tag{3.9a}$$

$$\lambda_{\mathbf{v}} = \mathbf{c_3}\cos\sqrt{\alpha}t + \mathbf{c_4}\sin\sqrt{\alpha}t, \tag{3.9b}$$

where $\mathbf{c_1}$, $\mathbf{c_2}$, $\mathbf{c_3}$ and $\mathbf{c_4}$ are constant vectors, $\lambda_\mathbf{r} = [\lambda_{r_x}, \lambda_{r_y}, \lambda_{r_z}]$, $\lambda_\mathbf{v} = [\lambda_{v_x}, \lambda_{v_y}, \lambda_{v_z}]$. Expanding (3.9) into component-wise expression leads to

$$\lambda_{r_x} = A_1 \cos\sqrt{\alpha}t + A_2 \sin\sqrt{\alpha}t, \tag{3.10a}$$

$$\lambda_{r_y} = A_3 \cos\sqrt{\alpha}t + A_4 \sin\sqrt{\alpha}t, \tag{3.10b}$$

$$\lambda_{r_z} = A_5 \cos\sqrt{\alpha}t + A_6 \sin\sqrt{\alpha}t, \tag{3.10c}$$

$$\lambda_{v_x} = A_7 \cos\sqrt{\alpha}t + A_8 \sin\sqrt{\alpha}t, \tag{3.10d}$$

$$\lambda_{v_y} = A_9 \cos\sqrt{\alpha}t + A_{10} \sin\sqrt{\alpha}t, \tag{3.10e}$$

$$\lambda_{v_z} = A_{11} \cos\sqrt{\alpha}t + A_{12} \sin\sqrt{\alpha}t, \tag{3.10f}$$

where A_1 to A_{12} are scalar constants.

Since the optimal thrust direction $\mathbf{1_T}$ is parallel to $\lambda_\mathbf{v}$ according to (3.3), the thrust direction angles θ and ϕ can be expressed by the components of $\lambda_\mathbf{v}$,

$$\tan(\theta)\sin(\phi) = \frac{T_y}{T_z} = \frac{\lambda_{v_y}}{\lambda_{v_z}} = \frac{A_9 \cos\sqrt{\alpha}t + A_{10} \sin\sqrt{\alpha}t}{A_{11} \cos\sqrt{\alpha}t + A_{12} \sin\sqrt{\alpha}t}, \tag{3.11a}$$

$$\tan(\theta)\cos(\phi) = \frac{T_x}{T_z} = \frac{\lambda_{v_x}}{\lambda_{v_z}} = \frac{A_7 \cos\sqrt{\alpha}t + A_8 \sin\sqrt{\alpha}t}{A_{11} \cos\sqrt{\alpha}t + A_{12} \sin\sqrt{\alpha}t}, \tag{3.11b}$$

where θ represents the angle between the thrust vector and z axis, and ϕ represents the angle between the projection of the thrust vector on x-y plane and x axis, as shown in Fig. 3.1.

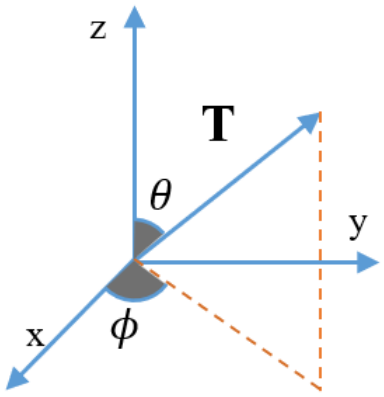


Figure 3.1. Thrust vector and the corresponding direction angles

By dividing the numerator and denominator of (3.11a) and (3.11b) by A_{11} and rearranging, we obtain

$$\frac{T_z}{T_y} = \frac{\cos\sqrt{\alpha}t + I_1 \sin\sqrt{\alpha}t}{I_2 \cos\sqrt{\alpha}t + I_3 \sin\sqrt{\alpha}t}, \tag{3.12a}$$

$$\frac{T_z}{T_x} = \frac{\cos\sqrt{\alpha}t + I_1 \sin\sqrt{\alpha}t}{I_4 \cos\sqrt{\alpha}t + I_5 \sin\sqrt{\alpha}t}, \tag{3.12b}$$

where I_1 through I_5 are constants. The simplified expressions of $\frac{T_z}{T_y}$ and $\frac{T_z}{T_x}$ reduce the number of parameters required to define the thrust direction angles from 6 to 5, which reduces the learning space consequently. Therefore, on determining the final time t_f and parameters I_1 through I_5 to represent the thrust angles, the optimal control profile can be determined. Using the dynamics and the obtained control profile, the states can be reconstructed by integrating the dynamics from the initial time to any desired time interval. The identified features for the TOAG problem are:

(1) The duration of abort phase: t_f.

(2) Parameters for thrust direction: I_1, I_2, I_3, I_4 and I_5.

For the FOAG problem to minimize the fuel consumption during the abort mission, we simply replace the objective function in (2.6a) with $J = \min_{T,t_f} \int_{t_0}^{t_f} \frac{T(t)}{v_{ex}} dt$. When the thrust has a fixed magnitude $T = T_{max}$, it is equivalent to the TOAG problem.

Proposition 2. *For the FOAG problem with a fixed thrust magnitude at the upper bound, i.e., $T = T_{max}$, the TOAG and FOAG problems yield the same optimal control profile.*

Proof. For the FOAG problem, the cost function is given by

$$J = \min_{T,t_f} \int_{t_0}^{t_f} \frac{T(t)}{v_{ex}} dt. \tag{3.13}$$

If the thrust magnitude is T_{max}, $\forall$ t $\in [t_0, t_f]$, the cost function becomes

$$J = \frac{T_{max}}{v_{ex}} \min_{t_f} \int_{t_0}^{t_f} dt, \tag{3.14}$$

where $\frac{T_{max}}{v_{ex}}$ is a given constant. Then the cost function in (3.14) is determined by the time used during the abort mission. This indicates that the FOAG and TOAG are identical when $T = T_{max}$. □

The above analysis also indicates that the FOAG problem with $T = T_{max}$ will have the same set of features for the optimal solution as that of the TOAG problem. In the following, we will focus on solving the TOAG problem.

3.2 Dataset Generation

The feature-based learning method requires a sufficient dataset to be generated offline. Before solving the TOAG problem offline, the initial conditions for aborting need to be determined. Since the abort starts along the powered descent phase, various sets of powered descent trajectories with different initial conditions, including varied initial altitude and velocities, are generated first. The fuel-optimal powered descent guidance problem is solved using an NLP solver [27]. The readers are referred to [24] for details on the fuel optimal powered descent problem formulation.

After the powered descent solutions are generated, the initial conditions for the TOAG problem, denoted by $\mathbf{r_0}$, $\mathbf{v_0}$, m_0, are obtained using the powered descent solutions and the abort time is selected during the powered descent phase. For each case with the above stated initial conditions, the TOAG problem is solved via an NLP solver [27].

However, the optimal solutions from NLP do not yield the values of identified features directly. Therefore, after collecting sufficient optimal solutions offline, the next step is to extract the identified features of the optimal solution discussed in section 3.1. The final time t_f of the abort phase is determined directly from the NLP solution. To extract features I_1 to I_5, a curve fitting method is applied, where the fitting functions are shown in (3.12). Finally, the data training is used to map the relationship between the inputs $\{\mathbf{r_0}, \mathbf{v_0}, m_0\}$ and the outputs $\{t_f, I_1, I_2, I_3, I_4, I_5\}$.

3.3 Constructing Neural Networks

After the dataset is generated, the next step is to apply the neural networks to map the relationship between initial states and identified features. In this work, the feedforward neural network is adopted. The feedforward neural network is comprised of three distinct layers, namely, input layer, hidden layer and output layer, which is shown in Fig. 3.2. Every layer has several activation functions. An appropriate combination of linear and nonlinear function can reduce the error between the actual data and the learnt solution along with a faster rate of convergence. A neural network with four layers is constructed and the number of neurons in each layer is 15, 10, 10, and 15, respectively. Additionally, the activation functions included three hyperbolic function, and one ReLU function. The algorithm used for training the neural network is Bayesian regularization back-propagation. In addition, from the generated dataset of the features, 80% of them are selected as training data, 10% of them are selected as test data, and the remaining 10% are selected as the validation data. Finally, the feedforward neural network is trained to map the relationship between the initial conditions and extracted features of the optimal solution.

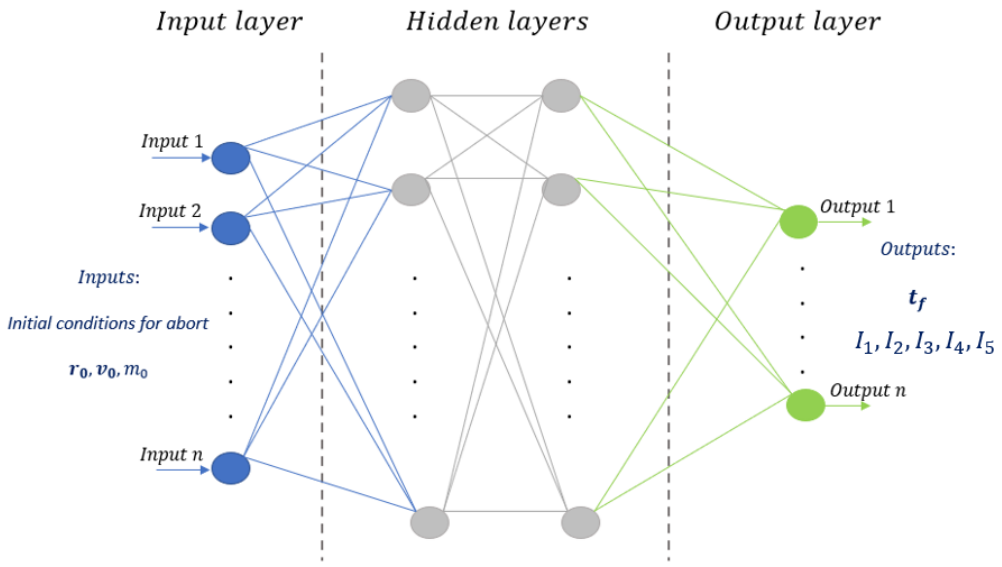


Figure 3.2. Structure of a neural network

3.4 The Framework of Feature-based Learning Method

The steps included in the offline part are data generation, feature extraction, and training of the neural network. On the other hand, the online part refers to utilizing the trained neural network onboard to calculate the values of identified features for a given initial condition within a reasonable range. Then, the optimal control solution can be reconstructed online based on feature values. The procedures of the proposed method are detailed in Table 3.1.

Table 3.1. Flowchart of Feature based learning for time-optimal abort

Algorithm: Feature-based learning method for TOAG
1: Off-line part
1.1) Identify the features for TOAG via the optimal control theory;
1.2) Generate a dataset of time-optimal abort trajectories with different initial conditions;
1.3) Extracting feature values of the optimal abort solution for each generated case;
1.4) Applying neural network to map the relationship between the initial conditions of abort and the values of identified features;
2: On-line part
2.1) Calculate the feature values according to the trained neural networks for a given initial condition of abort;
2.2) Reconstruct the optimal control solution using the obtained features values.

4. CONSTRUCTION OF EXPERIMENTAL TEST BED AND TESTING SCENARIO

The purpose of the experimental tests is to replicate the dynamics of an abort spacecraft that is originally planned to land on the Moon and then verify the computational performance of the abort guidance algorithm in a constructed scenario. Our work in [22] built a customized quadcopter to validate the computational performance of a fuel-optimal powered descent guidance algorithm. Although scale-model rocket powered vehicles have been developed for testing EDL missions [28], [29] with the purpose of increasing the technology readiness level (TRL), the complexities and costs involved in the scale-model tests make them not viable until the late stages of a mission. The testbed based on a customized quadcopter allows rapid validation of the proposed method in real-world experiments without sacrificing project budgets or safety to the degree that a scale-model would.

To mimic the dynamics of a rocket powered vehicle in a powered descent landing mission, the general design is comprised of a large vehicle mass atop a gimbaling point source of thrust. Given the limited thrust control of modern rocket motors and analysis of the TOAG optimal control profile, a thrust with a constant maximum magnitude must be used for time/fuel optimal results. However, in the case of a quadcopter/drone, the direction of thrust is always perpendicular to the quadcopter's frame. To closely mimic the dynamics of a rocket-powered vehicle, the thrust vectoring is achieved by orienting the quadcopter to align with the required thrust direction. Therefore, in the case of quadcopter the guidance strategy involves using both thrust and attitude commands. The attitude commands are realized by the yaw, pitch, and roll angles provided to orient the quadcopter.

The experimental test considers a flat Earth model with a constant gravity vector. In order to ensure the time-optimal abort using a quadcopter has the same number of features of the optimal solution as that of a rocket-powered vehicle, the identification of features are re-derived from the optimal control theory. The terminal conditions in case of spherical planet problem are defined using the orbital elements, including the orbit size and the inclination angle. To mimic the terminal conditions of TOAG, in the case of a flat planet model,

the terminal condition requires attaining a final altitude h_f, circular velocities in x and y directions, and $v_{zf} = 0$, as shown in Fig. 4.1.

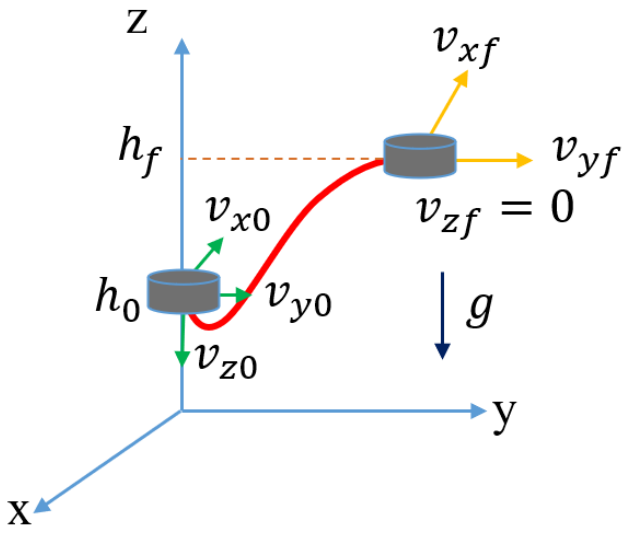


Figure 4.1. Abort test settings using a drone testbed with a flat planet model

The 3D equations of motion of a quadcopter is given by

$$\dot{\mathbf{r}} = \mathbf{v}, \tag{4.1a}$$

$$\dot{\mathbf{v}} = \mathbf{g} + T\frac{\mathbf{1_T}}{m_0}, \tag{4.1b}$$

where $\mathbf{g} = [0, 0, -g]$, $g = 9.81\ m/s^2$, is the uniform gravitational acceleration of Earth, and m_0 is the mass of the quadcopter which remains constant during the entire flight duration. Compared with the TOAG dynamics in (2.1), the difference is the gravitational acceleration vector and the mass of the vehicle, which are set as constant during the experiments. With the same assumption of bounded trust (2.3), the TOAG problem to minimize the flight time also leads to the same conclusion that the optimal thrust magnitude is T_{max} for the entire flight duration. Similar to (3.12), the optimal thrust direction can be derived from the necessary conditions of optimality.

The Hamiltonian for the TOAG problem with the quadcopter dynamics is expressed as

$$H = \lambda_{\mathbf{r}}{}^T \mathbf{v} + \lambda_{\mathbf{v}}{}^T \left(\mathbf{g} + \frac{T}{m_0} \mathbf{1_T} \right) - 1. \tag{4.2}$$

where $\lambda_{\mathbf{r}} \in R^3$, $\lambda_{\mathbf{v}} \in R^3$ are associated costate variables. The optimal direction of thrust is in the unit control direction maximizing H

$$\mathbf{1_T} = \frac{\lambda_{\mathbf{v}}}{\|\lambda_v\|}. \tag{4.3}$$

The necessary conditions for adjoint variables

$$\dot{\lambda}_{\mathbf{r}} = 0, \tag{4.4a}$$

$$\dot{\lambda}_{\mathbf{v}} = -\lambda_{\mathbf{r}}. \tag{4.4b}$$

Integrating the above equations we obtain

$$\lambda_{\mathbf{r}} = \mathbf{c_1}, \tag{4.5a}$$

$$\lambda_{\mathbf{v}} = -\mathbf{c_1} t + \mathbf{c_2}. \tag{4.5b}$$

where $\mathbf{c_1}$ and $\mathbf{c_2}$ are constant vectors. The expression for $\frac{\partial H}{\partial T}$ is

$$\frac{\partial H}{\partial T} = \frac{1}{m_0} \lambda_{\mathbf{v}}{}^T \mathbf{1_T} = \frac{\|\lambda_v\|}{m_0}. \tag{4.6}$$

As the expression for $\frac{\partial H}{\partial T}$ is independent of T, the optimal solution is of the form

$$T(t) = \begin{cases} T_{min}, \frac{\partial H}{\partial T} < 0 \\ T_{max}, \frac{\partial H}{\partial T} > 0 \\ [T_{min}, T_{max}], \frac{\partial H}{\partial T} = 0 \end{cases}$$

From (4.4) and (4.5), if $\|\lambda_v\| = 0$, $\lambda_{\mathbf{r}}$ has to be zero. Then, all the costates go to $\mathbf{0}$, which violates the Pontryagin's maximum principle. Thus, $\|\lambda_v\| \neq 0$. Equation (4.6) along with $\|\lambda_v\| > 0$ and $m_0 > 0$ imply that $\frac{\partial H}{\partial T} > 0$. Therefore, for the TOAG problem using the quadcopter dynamics, the thrust is always at the maximum value T_{max}. The component-wise solution form for (4.5) is given as

$$\lambda_{r_x} = -B_1, \tag{4.7a}$$
$$\lambda_{r_y} = -B_2, \tag{4.7b}$$
$$\lambda_{r_z} = -B_3, \tag{4.7c}$$
$$\lambda_{v_x} = B_1 t + B_4, \tag{4.7d}$$
$$\lambda_{v_y} = B_2 t + B_5, \tag{4.7e}$$
$$\lambda_{v_z} = B_3 t + B_6, \tag{4.7f}$$

where B_1 to B_6 are scalar constants of integration. Since (4.3) governs the optimal thrust direction $\mathbf{1_T}$, the functions of the thrust direction angles θ and ϕ can be expressed using (4.7), written as

$$\tan(\theta)\sin(\phi) = \frac{T_y}{T_z} = \frac{\lambda_{v_y}}{\lambda_{v_z}} = \frac{B_2 t + B_5}{B_3 t + B_6}, \tag{4.8a}$$
$$\tan(\theta)\cos(\phi) = \frac{T_x}{T_z} = \frac{\lambda_{v_x}}{\lambda_{v_z}} = \frac{B_1 t + B_4}{B_3 t + B_6}. \tag{4.8b}$$

By dividing B_3 to numerator and denominator of right side in (4.8a) and (4.8b) and rearranging, we obtain

$$\frac{T_z}{T_y} = \frac{t + D_1}{D_2 t + D_3}, \tag{4.9a}$$

$$\frac{T_z}{T_x} = \frac{t + D_1}{D_4 t + D_5}, \tag{4.9b}$$

where D_1 to D_5 are scalar constants.

The identified features for the TOAG problem using the customized quadcopter in the Earth testing environments are summarized as:

(1) The duration of abort phase: t_f

(2) Parameters for thrust direction: D_1, D_2, D_3, D_4 and D_5.

As the number of features of the optimal solution of the experimental TOAG problem effectively remained the same with the same structure of the optimal control solution, it can be used to verify the effectiveness of the proposed learning-based method.

The components used for the quadcopter are listed in Table 4.1. The Pixhawk 4 (PX4) is used as a flight controller and offers advantages such as sensor logging, proportional–integral–derivative control loops, and ability to program the navigation command. A Raspberry Pi 4 module, working as the secondary computer, has a Quad core Cortex-A72 (ARM v8) 64-bit processor with a clock speed of 1.5 GHz and 4 GB of RAM. It was used to issue MAVLINK (protocol used to communicate with drone) commands for the control actions. The detailed description of critical components and the functionality of the customized drone can be found in [22]. The labelled quadcopter components are shown in Fig. 4.2.

The operational sequences for the experimental test include: 1. Takeoff from home position; 2. Hover to a safe altitude to calculate offset; 3. Generate a random initial condition during hover; 4. Move to the initial position and provide the initial velocity; 5. Send the initial conditions to the neural network and obtain values of identified features onboard; 6. Reconstruct control profile using the obtained features; 7. Execute the control commands to match the thrust magnitude with corresponding orientation; and 8. Return home position

Table 4.1. List of components for the customized quadcopter

Item Type	Item Name	Quantity
Frame	Readytosky S500	1
Motors	Hobbypower 2212 920KV	4
Propellers	Readytosky 10x4.5 1045	4
Electronic Speed Controllers's	ARRIS 2-6S 30AMP 30A SimonK firmware OPT	4
Batteries	Zeee 3S 120C 2200mAh LiPo	4
Power Management Board	Pixhawk 4 PM07	1
Flight Controller	Pixhawk 4	1
Offboard Computer	Raspberry Pi 4 4GB	1
GPS Module	Pixhawk 4 GPS Module	2
Control Radio	Flysky FS-i6X RC Transmitter	1
Telemetry Radio	HolyBro Transceiver Telemetry Radio V3	1

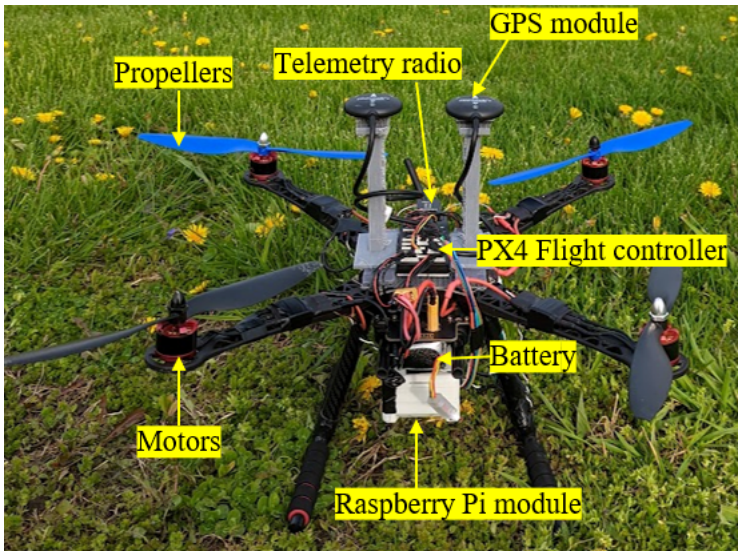


Figure 4.2. Labelled quadcopter components

after abort mission execution. The visual flowchart of the operational sequences is shown in Fig. 4.3.

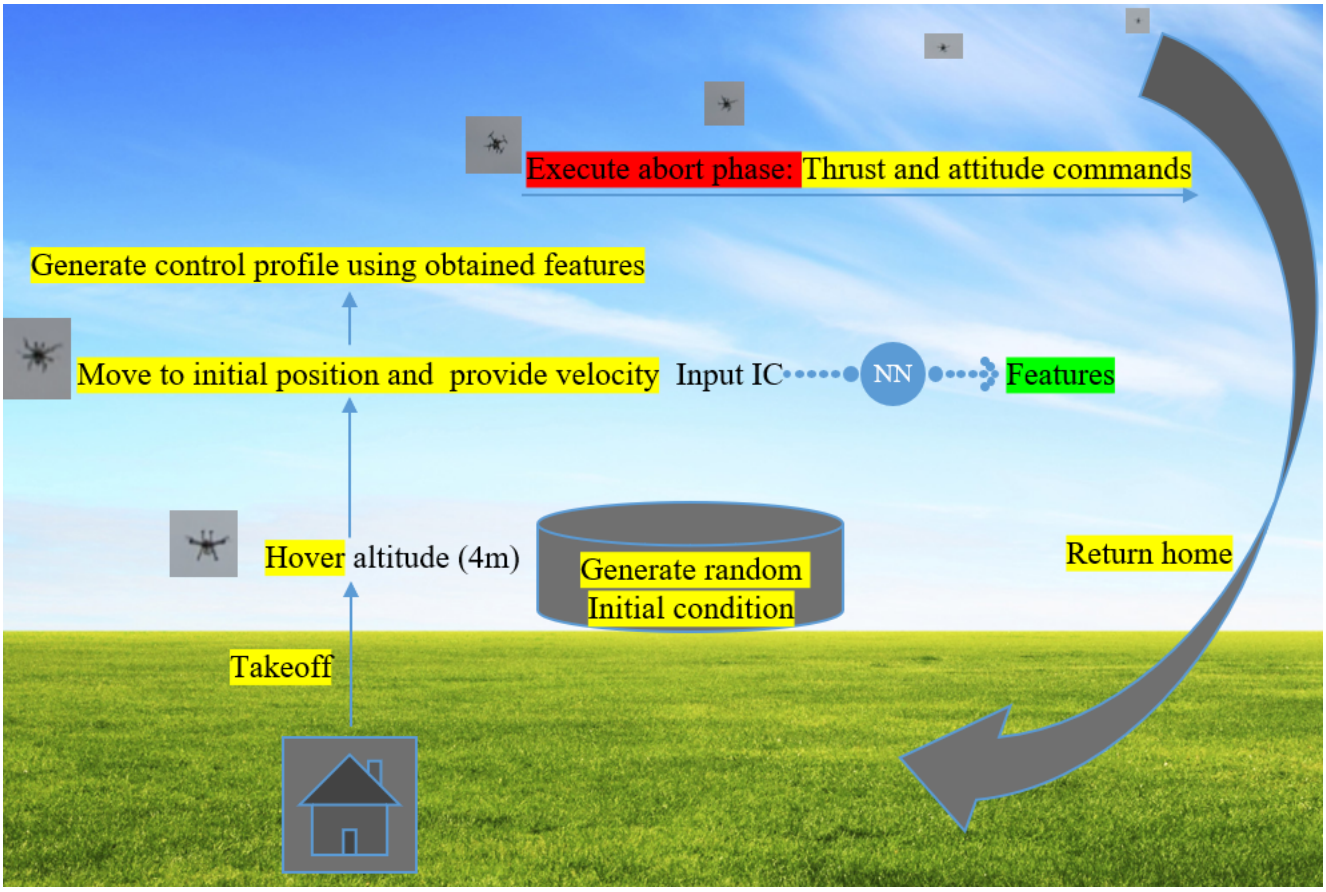


Figure 4.3. operational sequences for the experimental test

5. SIMULATION AND EXPERIMENTAL RESULTS

5.1 Lunar Abort Simulation

To show the effectiveness of the proposed method, the human-scale Moon abort mission is simulated and analysed in this section. The adopted vehicle data was obtained from the Apollo 11 mission [8]. The descent phase of the lunar module has a maximum thrust of 45 kN and a specific impulse of 311 seconds. The landing position considered is the lunar south pole. Table 5.1 summarizes the initial conditions for the mission. All the simulations were conducted on a laptop with i7-8750H CPU @ 2.20GHz and 16GB of RAM.

Table 5.1. Initial conditions for the Apollo 11 mission

Initial conditions	**Value**
Altitude (km)	15.24
Longitude (deg)	41.85
Latitude (deg)	-71.6
Initial velocity (m/s)	1,698.3
Initial topocentric flight path angle (deg)	0.0
Initial topocentric zaimuth angle (deg)	180.0
Total vehicle mass (kg)	15,103
Total fuel mass (kg)	10,624
Vehicle dry mass (kg)	4,479

When generating the dataset, the initial conditions of the powered descent trajectories are set within the initial altitude range of $[12, 18]$ km and the difference between two adjacent cases is 1 km. The initial velocity range is $[1650, 1750]$ m/s with a 10 m/s difference between two adjacent cases. The remaining initial conditions are shown in Table 5.1. Overall, 70 sets of powered descent trajectories were generated by solving the fuel-optimal powered descent guidance problem using an NLP solver with initial settings stated above. The time histories of altitude for each of the powered descent trajectories are shown in Fig. 5.1.

The initial conditions for the TOAG problem, denoted by $\mathbf{r_0}$, $\mathbf{v_0}$, m_0, are obtained by choosing an abort time every 10 seconds during the powered descent phase. For the lunar abort mission, the lunar vehicle needs to be guided into a 10 km $\times$ 150 km polar orbit. Using

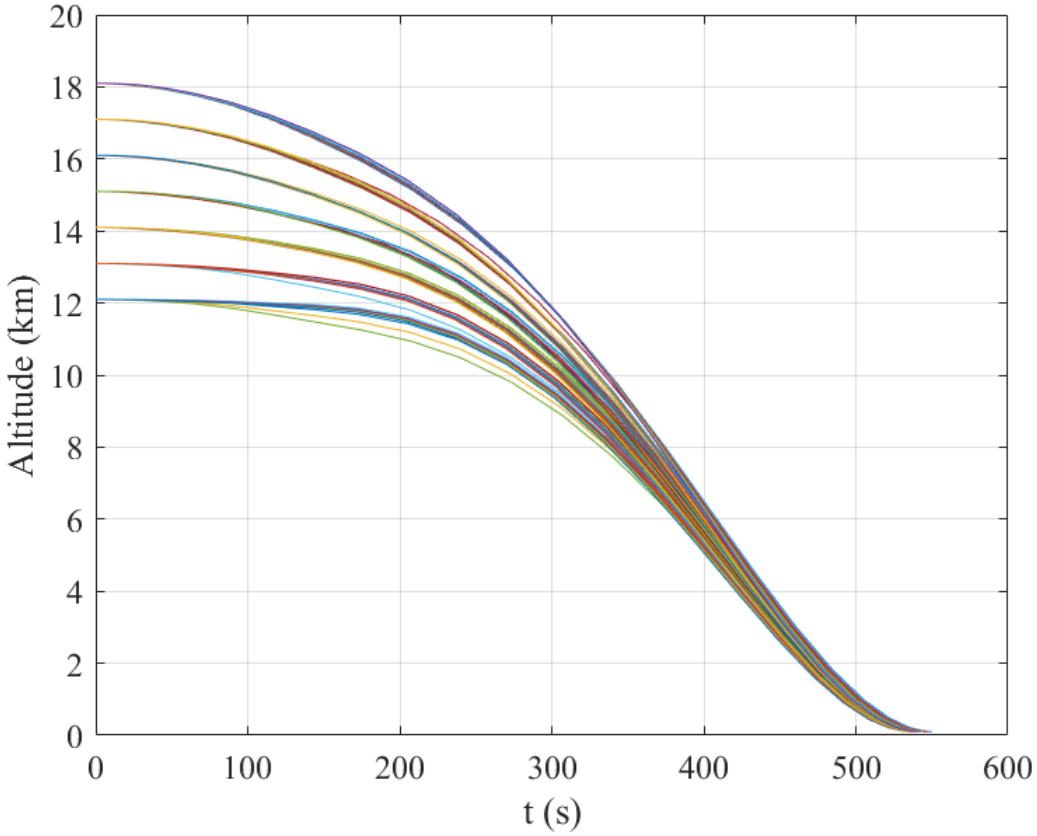


Figure 5.1. Time histories of powered descent altitude with different initial conditions

the problem settings discussed above, each TOAG case is solved via the NLP solver offline. A set of optimal abort trajectories starting at different points on one of the powered descent trajectories is shown in Fig. 5.2. The mean computation time to solve one TOAG problem using the NLP solver is 5.3 seconds.

After generating sufficient optimal TOAG solutions offline, the features of the optimal solution needs to be extracted for every case with a corresponding initial abort condition. An example of extraction of the features of the optimal solution is shown in Fig. 5.3. The features of the optimal solution are only extracted for the abort trajectories with the abort duration greater than 150 seconds. The selected trajectories involve the part of powered descent where the abort becomes more crucial as the vehicle is approaching the ground. Overall, 1583 sets of cases have their features extracted from the offline generated optimal solution.

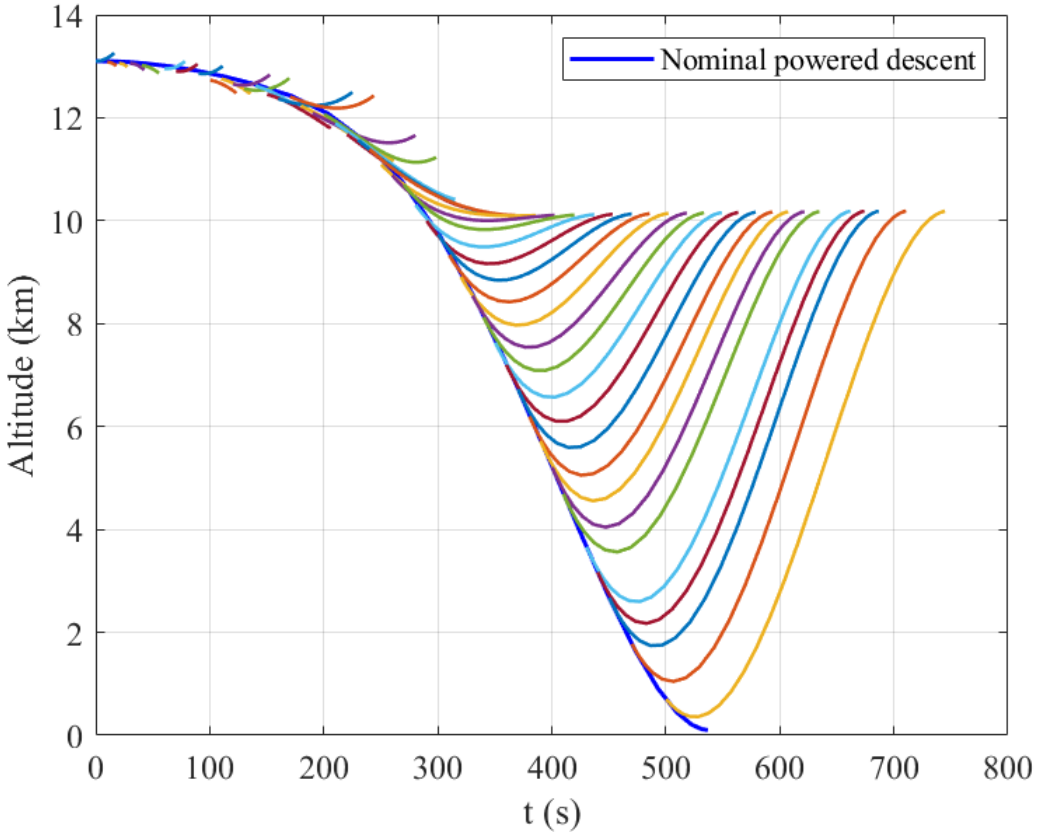
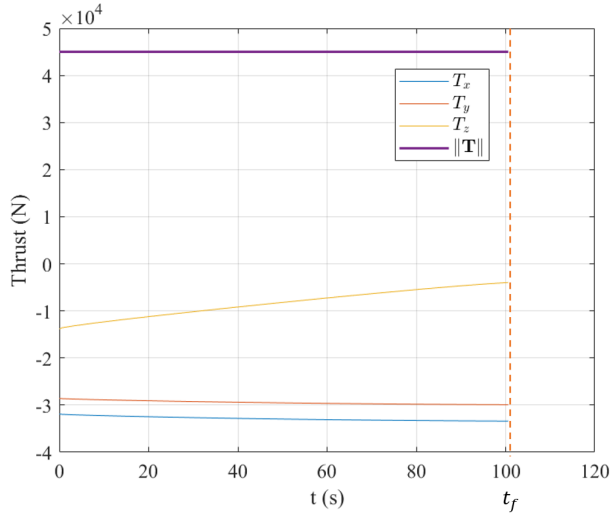
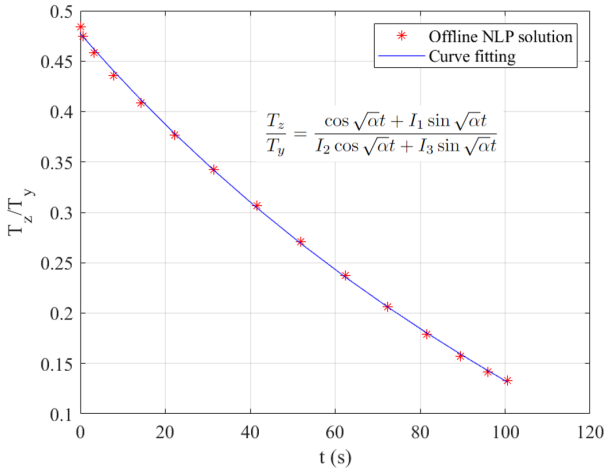


Figure 5.2. Time histories of TOAG altitude with different starting points on a powered decent phase

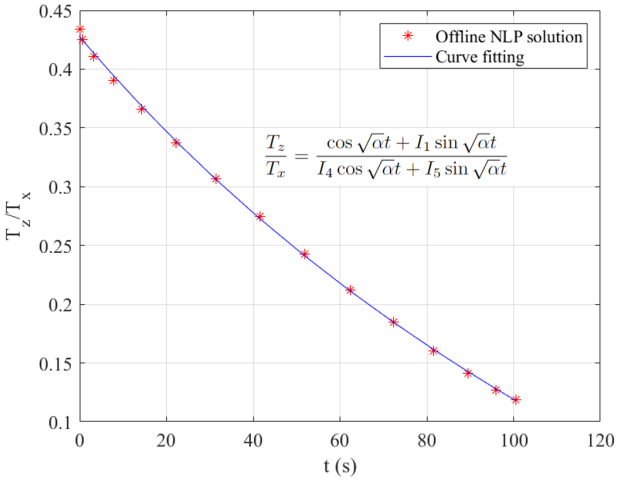
After successful training of the neural networks, the cases within the training range were tested first, where 27527 sets of TOAG cases with different initial conditions were simulated. The absolute mean percentage errors for the terminal orbital parameters for specific angular momentum, specific energy, and inclination, are 0.567, 0.676 and 0.009, respectively, which verifies the high precision of the optimal solutions from the proposed method. The absolute mean difference between the optimal time t_f obtained from NLP and the corresponding one from the learnt solution is 0.003 seconds. Therefore, the solution from learning is near optimal. The mean computation time using the learning-based method is 0.029 seconds. This verifies the performance of the proposed method when solving TOAG problems with initial conditions set within the range of the training dataset.



(a) Optimal control profile and final time t_f



(b) Extracting I_1, I_2, I_3



(c) Extracting I_4, I_5

Figure 5.3. An illustrative example of feature value extraction from offline optimal solution

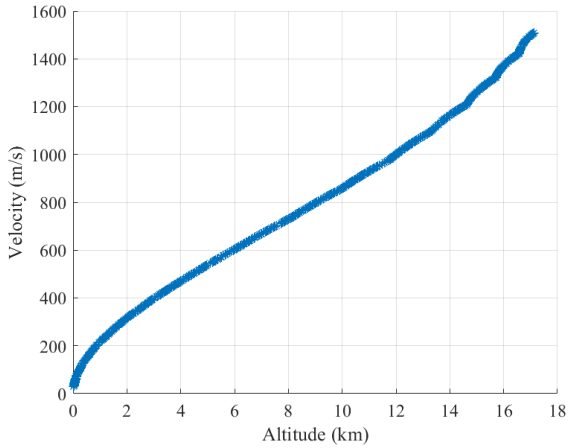
To further evaluate the effectiveness and robustness of the proposed method, the cases with initial conditions outside the training dataset are also adopted for evaluation. A separate powered descent trajectory is generated, where the initial altitude is 19 km that is outside the range of $[12, 18]$ km and the initial velocity is 1600 m/s that is outside the range of [1650,1750] m/s. Then we assume that the powered descent trajectory is aborted at random timestamps ranging from $150 - 550$ s along the powered descent phase. 1000 random TOAG cases are computed using this initial condition setting. The mean percentage errors

in these cases for the terminal orbital parameters, specifically angular momentum, specific energy, and inclination, are 0.64, 0.75 and 0.01, respectively. In contrast, the mean percentage errors for the terminal orbit parameters obtained from the NLP solution, specifically angular momentum, specific energy, and inclination, are 0.27, 0.29 and 0.01, respectively. The absolute mean difference between the optimal time t_f obtained from NLP and the corresponding one from the learnt solution is 0.006 seconds. The mean computation time for the proposed method is 0.035 sec, while the mean computation time using for NLP is 6.2 sec. These results show that the terminal conditions were closely satisfied and the solution from the learning-based method is near optimal for cases with initial conditions outside the training dataset.

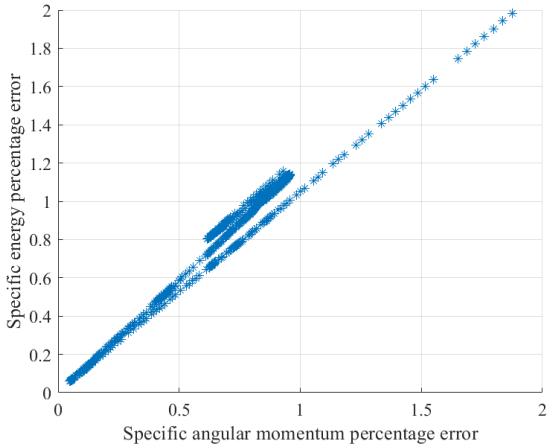
The plots of initial velocities versus altitude for the abort cases with different timestamps along the powered descent phase, percentage errors for the learnt solution, and the terminal mass for all cases are shown in Fig. 5.4. In addition, a set of comparative plots between the NLP and the proposed method with the abort time of 250 sec during the powered descent phase are shown in Fig. 5.5. As the terminal orbital conditions obtained from the proposed method for most of the cases leads to errors within 2 percent, these results verify the robustness of the proposed method for initial conditions outside the training dataset. Also, as the mean computation time of the proposed method is less than 0.04 seconds, it further validates the real time computational performance of the proposed method.

5.2 Experimental Test Results

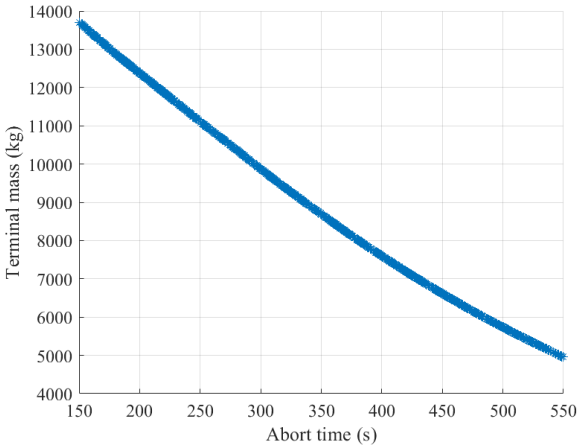
For the experimental tests, the first step is to generate the offline dataset for the quadcopter testbed using the following settings. Mass of the quadcopter a constant value, set as 1.545 kg. The maximum and minimum thrust was assigned to be 15.5 N and 5 N, respectively. For simplicity, the initial altitude where the abort starts is assumed to be a fixed value, i.e., $r(t_0) = [0, 0, 10]$. The initial velocity of abort has v_z pointing toward the ground and random nonzero velocity components in x and y directions, the initial velocities used for generating the offline data were in the range of 1 to 2 m/s for v_x and v_y, and -1 to -0.5 m/s for v_z with 21 evenly spaced points inserted for each variable. Overall, 9261 cases with



(a) Velocity vs. altitude



(b) Specific energy vs. angular momentum percentage errors

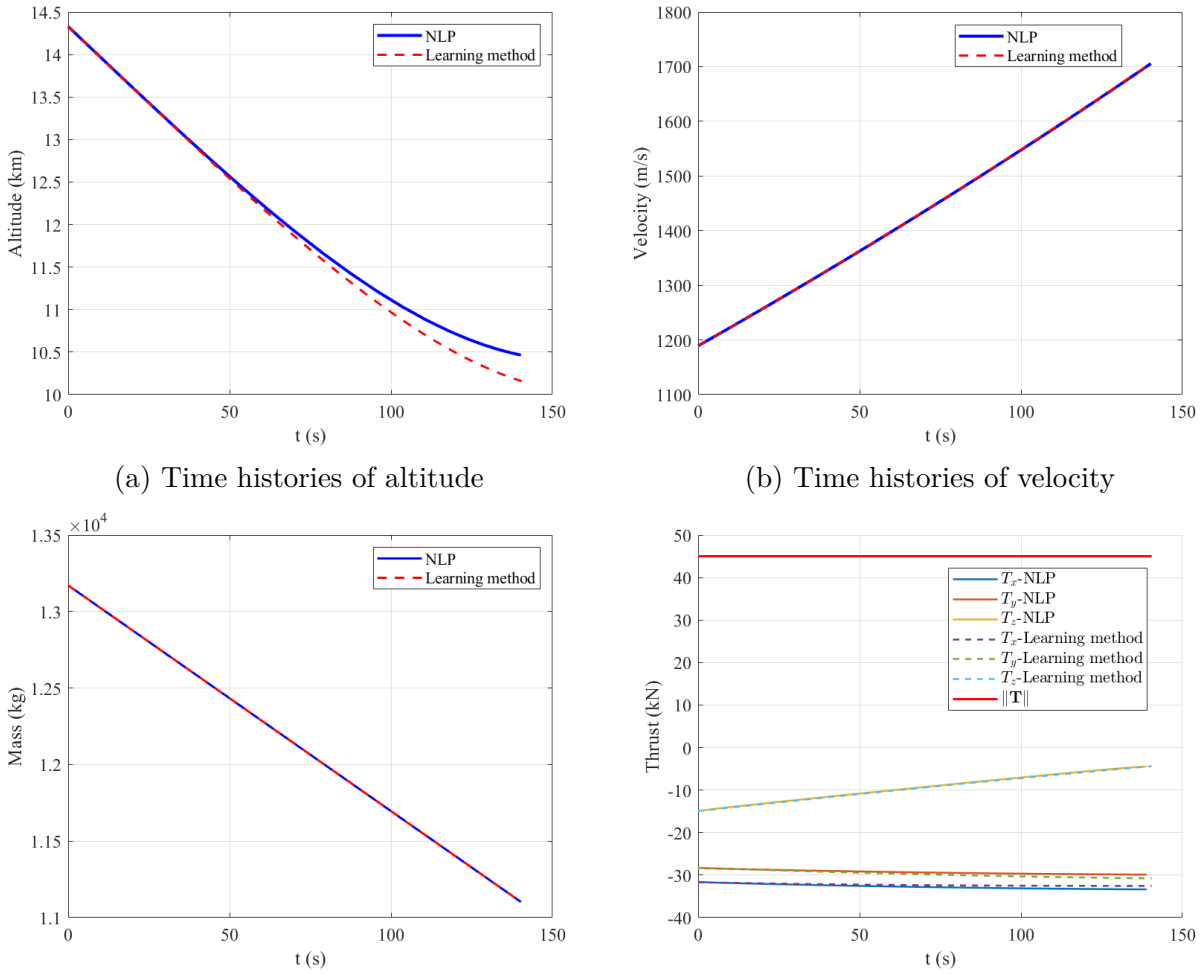


(c) Terminal mass vs. abort time

Figure 5.4. Learning results of the 1000 random cases with initial conditions outside the training dataset.

different combinations of initial conditions are generated. The terminal conditions assume an final abort altitude of 11 m, circular velocities v_x and v_y to be 10 m/s, and v_z to be 0 m/s. By assuming the constant gravity and constant mass, a dataset for the TOAG problem formulated in (4.1) with boundary conditions specified above is generated by solving each case via NLP offline. The histories of altitude and optimal trajectories for the generated dataset are shown in Fig. 5.6.

After extracting the feature values from the generated dataset, the mean terminal position in z is 10.99 m and the mean terminal velocity components of v_x, v_y and v_z are 9.89 m/s, 9.90



(a) Time histories of altitude

(b) Time histories of velocity

(c) Time histories of mass

(d) Thrust control profiles

Figure 5.5. Comparison of state and control histories solved via NLP and the proposed method for an abort mission starting at 250 sec during a powered descent phase.

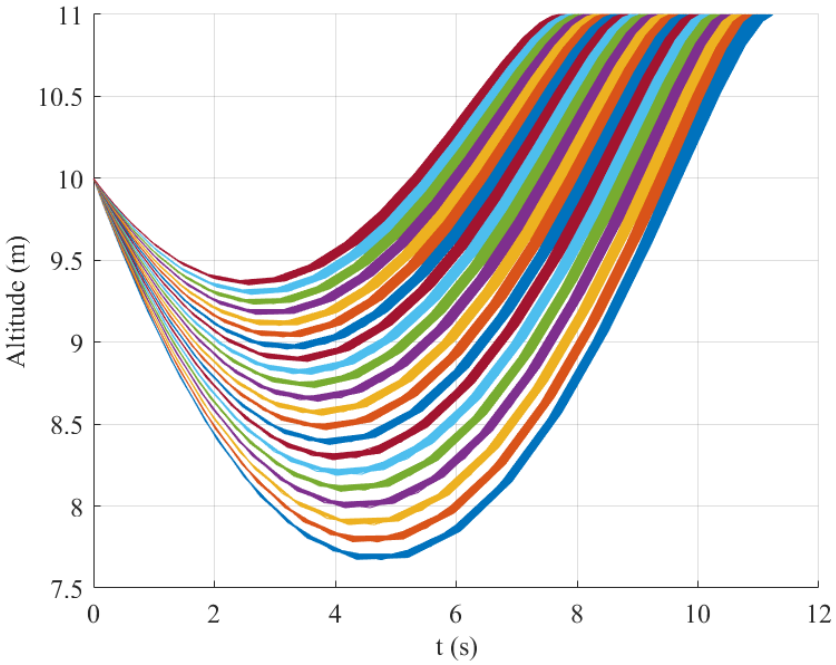
m/s, and 0.01 m/s, respectively. Once the values of identified features are determined for all cases in the generated dataset, the neural network is trained using the Bayesian regularization back-propagation algorithm. The neural network used for training has 4 hidden-layers with 256 neurons per layer, where the rectified linear unit (ReLU) activation functions are assigned to the first 3 hidden layers and a hyperbolic tangent activation function is applied to the final hidden-layer. 80% of the data generated is used to train the neural network which consists of 20% data used to validate the training. The remaining 20% data is considered to be data outside the training set. After constructing the neural network, the tested cases

consider initial conditions both inside and outside the dataset. The optimal control profile is reconstructed using the obtained feature values. The mean terminal position and velocity obtained for the test cases inside the dataset are 10.98 m, 9.85, 9.92, 0.01 m/s in x, y, z directions, respectively. And the mean terminal position and velocity obtained for cases outside the dataset are 10.98 m, 9.95, 9.86, 0.01 m/s in x, y, z directions, respectively. The mean difference of the optimal time t_f between the learning solution and NLP solution for cases inside and outside the dataset is 0.005 and 0.007 seconds, respectively. The average computation time for the proposed method is 0.003 seconds, while the NLP solver takes about 4.1 seconds on average to generate a solution. Comparative results for the state and control time history for one selected case are shown in Fig. 5.7.

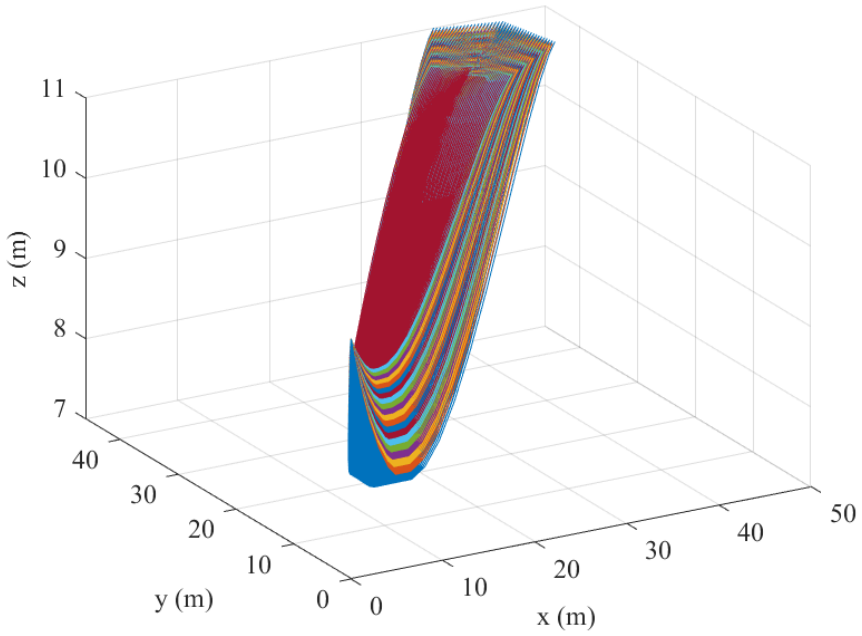
Finally, to validate the real time computational performance of the proposed method, the experimental test was conducted with the control commands generated onboard from the learning-based method. A random initial condition $[1.367, 1.642, -0.596]$ m/s was generated during the calibration process in flight and was fed to the trained neural network on the Raspberry Pi module. The module was able to generate the feature values and construct the control profile in 0.0092 seconds. Once the drone reached the randomly generated initial conditions, the computed control commands were executed. The reconstructed state and control history using the obtained features are shown in Fig. 5.8. The reconstructed states and control profile from the flight logs is shown in Fig. 5.9.

It can be observed from the flight logs that the throttle is almost constant for the abort duration, which is consistent with the theoretical analysis. In addition, the roll, pitch and yaw commands were successfully executed. The initial position has an offset of 12.54, 17.31 and 4.14 m in x, y and z direction respectively. The offset is given for the quadrotor to attain the initial velocity after the hover. The position and velocity in z direction fairly meet the desired values and is off by half a meter in position and 0.8 m/s in velocity. The position errors are calculated by subtracting the offset for the initial position. The position and velocity errors in x and y directions were relatively larger. These errors may be generated due to GPS resolution, atmospheric interactions, or the throttle scaling approximations. Since these mismatches are within reasonable ranges, the real time control solution generated onboard are executable and leads to a successful flight test that mimics the abort mission. Therefore,

the experimental results using the customized drone demonstrate the effectiveness of the proposed TOAG method for onboard implementation.

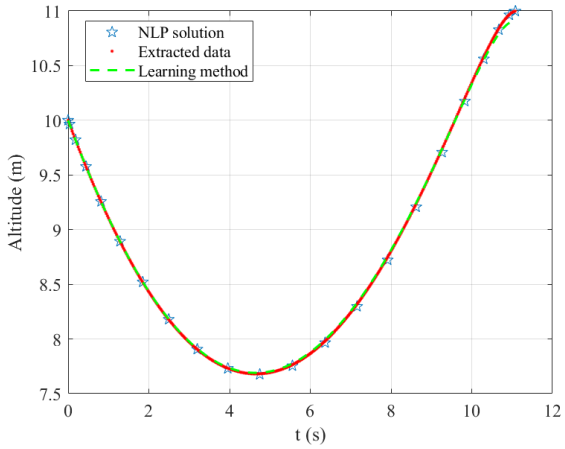


(a) Time histories of altitude

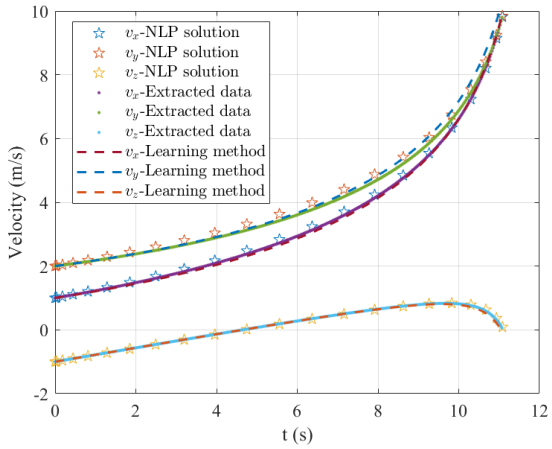


(b) 3D TOAG trajectories

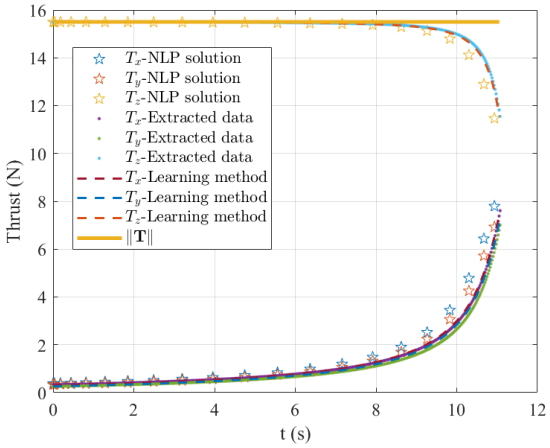
Figure 5.6. Time histories of altitude and optimal trajectories for the generated dataset using the drone model.



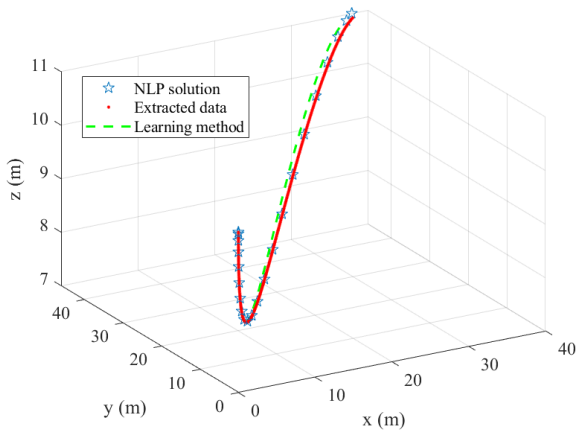
(a) Time history of altitude



(b) Time history of velocity

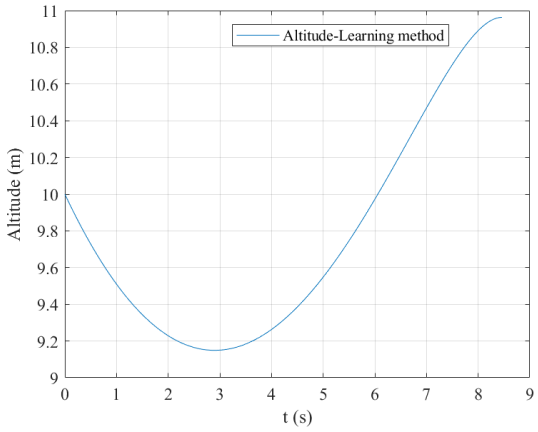


(c) Time history of thrust

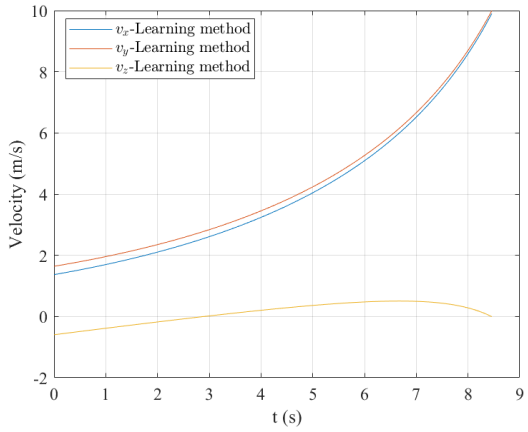


(d) Trajectory

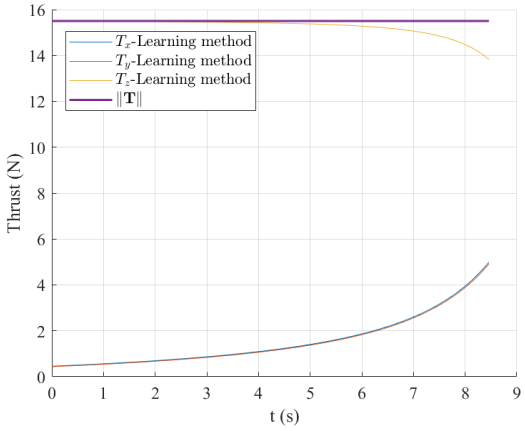
Figure 5.7. State and control time histories and optimal trajectory from NLP, extracted data, and learning results for a case with initial $\mathbf{v_0}$=[1,2,-1] m/s



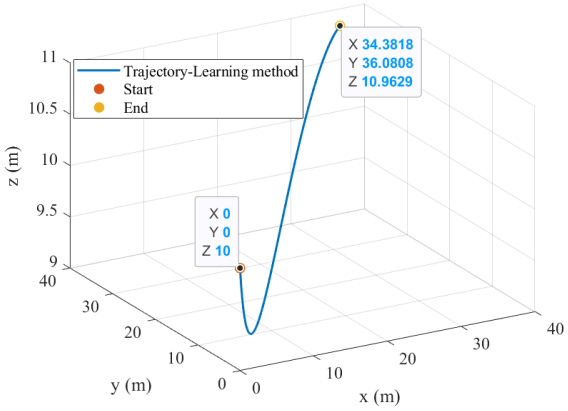
(a) Time history of altitude



(b) Time history of velocity

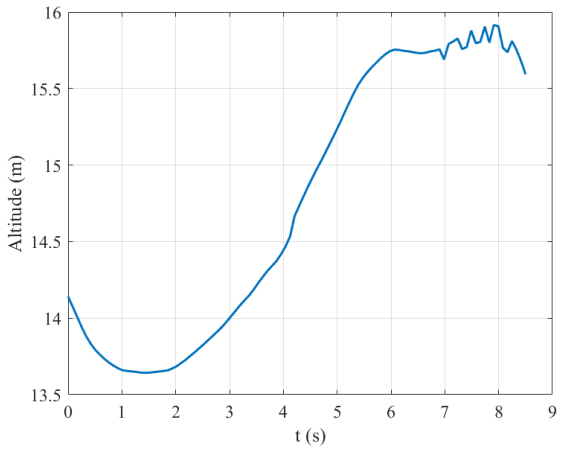


(c) Time history of thrust

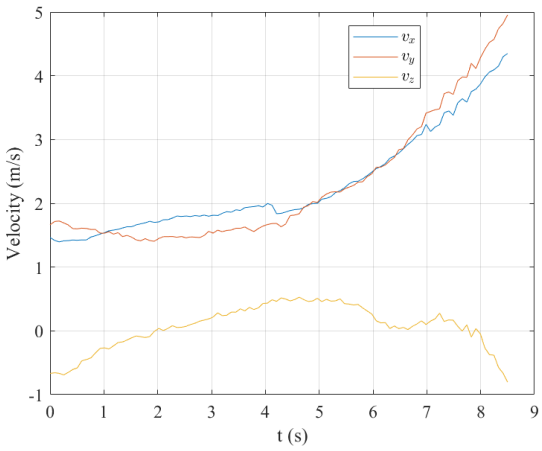


(d) Trajectory

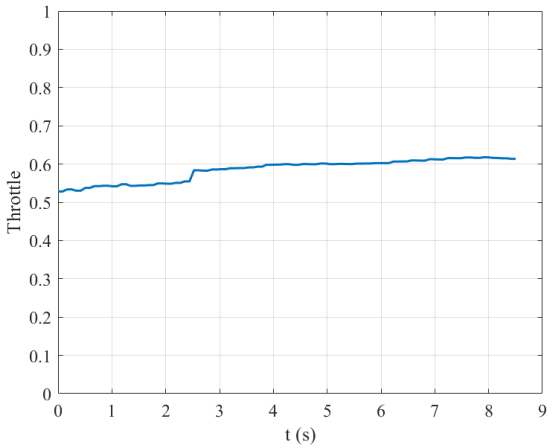
Figure 5.8. State and control time histories and optimal trajectory for the flight test case with randomly generated initial conditions.



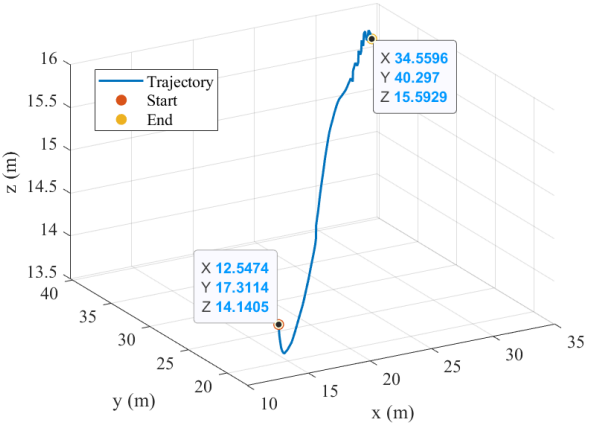
(a) Time history of altitude



(b) Time history of velocity



(c) Time history of thrust



(d) Trajectory

Figure 5.9. State and control time history along with trajectory for the random initial condition generated onboard: regenerated using flight logs

6. CONCLUSION

A feature-based learning method for solving the time-optimal abort problem (TOAG) is developed in this thesis. Combined with optimal control theory and feature learning, a limited number of parameters are identified to represent the optimal control profile, which leads to significantly reduced learning space and training time. From the extensive simulation results, it can be concluded that the proposed method is effective to generate TOAG solutions in real time for the initial abort conditions within or outside the range of training dataset. The verification in experimental tests using a low-cost testbed equipped with a credit-card sized computing unit verified the high computational efficiency and robustness of the proposed method for onboard application.

REFERENCES

[1] S. Creech, J. Guidi, and D. Elburn, "Artemis: An overview of nasa's activities to return humans to the moon," in *IEEE Aerospace Conference*, 2022.

[2] E. Suhir, "Landing on mars: Probabilistic modeling enables quantifying the last "six minutes of terror"," *Acta Astronautica*, vol. 179, pp. 680–684, 2021.

[3] F. Bennett, "Apollo experience report: Mission planning for lunar module descent and ascent," in *NASA Technical Notes: NASA TN D-6846*, 1972.

[4] A. E. Bryson and Y.-C. Ho, *Applied optimal control: optimization, estimation, and control*. Routledge, 2018.

[5] A. V. Rao, "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, vol. 135, no. 1, pp. 497–528, 2009.

[6] O. Von Stryk and R. Bulirsch, "Direct and indirect methods for trajectory optimization," *Annals of operations research*, vol. 37, no. 1, pp. 357–373, 1992.

[7] J. M. Longuski, J. J. Guzmán, and J. E. Prussing, *Optimal control with aerospace applications*. Springer, 2014.

[8] P. Lu and S. A. Sandoval, "Abort guidance during powered descent for crewed lunar missions," in *AIAA Scitech 2021 Forum*, 2021, p. 0505.

[9] P. Lu, B. J. Griffin, G. A. Dukeman, and F. R. Chavez, "Rapid optimal multiburn ascent planning and guidance," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 6, pp. 1656–1664, 2008.

[10] S. A. Sandoval, P. Lu, J. T. Hwang, J. R. Rea, and R. R. Sostaric, "Multiple optima in abort ascent during lunar powered descent," in *AIAA SCITECH 2022 Forum*, 2022, p. 0949.

[11] V. M. Becerra, "Practical direct collocation methods for computational optimal control," in *Modeling and Optimization in Space Engineering*, Springer, 2012, pp. 33–60.

[12] D. G. Hull, "Conversion of optimal control problems into parameter optimization problems," *Journal of guidance, control, and dynamics*, vol. 20, no. 1, pp. 57–60, 1997.

[13] F. Fahroo and I. M. Ross, "Direct trajectory optimization by a chebyshev pseudospectral method," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, pp. 160–166, 2002.

[14] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.

[15] S. Josselyn and I. M. Ross, "Rapid verification method for the trajectory optimization of reentry vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 3, pp. 505–508, 2003.

[16] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Machine learning: A review of classification and combining techniques," *Artificial Intelligence Review*, vol. 26, no. 3, pp. 159–190, 2006.

[17] J. Kim, C. Justin, D. Mavris, and S. Briceno, "Data-driven approach using machine learning for real-time flight path optimization," *Journal of Aerospace Information Systems*, vol. 19, no. 1, pp. 3–21, 2022.

[18] Y. Shi and Z. Wang, "A deep learning-based approach to real-time trajectory optimization for hypersonic vehicles," in *AIAA Scitech 2020 Forum*, 2020, p. 0023.

[19] E. G. Learned-Miller, "Introduction to supervised learning," *I: Department of Computer Science, University of Massachusetts*, p. 3, 2014.

[20] J. W. Kim, B. J. Park, H. Yoo, T. H. Oh, J. H. Lee, and J. M. Lee, "A model-based deep reinforcement learning method applied to finite-horizon optimal control of nonlinear control-affine system," *Journal of Process Control*, vol. 87, pp. 166–178, 2020.

[21] C. Sánchez-Sánchez and D. Izzo, "Real-time optimal control via deep neural networks: Study on landing problems," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 5, pp. 1122–1135, 2018.

[22] S. G. Hendrix, V. Kenny, S. You, A. Khilnani, R. Dai, and J. R. Rea, "Experimental testing for a learning-based powered-descent guidance algorithm," in *AIAA SCITECH 2022 Forum*, 2022, p. 0952.

[23] A. Scorsoglio, R. Furfaro, R. Linares, and B. Gaudet, "Image-based deep reinforcement learning for autonomous lunar landing," in *AIAA Scitech 2020 Forum*, 2020, p. 1910.

[24] S. You, C. Wan, R. Dai, and J. R. Rea, "Learning-based onboard guidance for fuel-optimal powered descent," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 3, pp. 601–613, 2021.

[25] V. Kenny, S. G. Hendrix, S. You, R. Dai, and J. R. Rea, "Feature-based learning for optimal abort guidance," in *AIAA SCITECH 2023 Forum*.

[26] L. S. Pontryagin, *Mathematical theory of optimal processes*. CRC press, 1987.

[27] K. Holmström, "Tomlab–an environment for solving optimization problems in matlab," in *Proceedings for the Nordic Matlab Conference'97*, Citeseer, 1997.

[28] D. Rutishauser and T. Tse, "Hardware-in-the-loop testing for suborbital flights of the safe and precise landing integrated capabilities evolution (splice) project technologies," in *AIAA Scitech 2020 Forum*, AIAA, 2020. DOI: 10.2514/6.2020-0367.

[29] J. N. Estes, S. Pedrotty, and R. R. Sostaric, "Splice flight hardware environmental test qualification program," in *AIAA Scitech 2021 Forum*, AIAA, 2021. DOI: 10.2514/6.2021-0257.