Pages: 61-73 / Vol. 16/ Issue 2 / Summer- Fall 2023

Journal of Aerospace Science and Technology

www:jast.ias.ir

Scientific-Research Article

Guidance Law design for the Final Stage of the Orbit Injection Problem using the High-Order Vector Expansion Method

Morteza Sharafi [1] , Mahdi Jafari[2]*, Mojtaba Alavipour[3]

1-Faculty of Electrical & Computer Engineering, Malek Ashtar University of Technology, Iran
2- Faculty of Aerospace, Malek Ashtar University of Technology, Iran
3- Faculty of Aerospace K.N.Toosi University of Technology, Iran

ABSTRACT

In this article, the optimal guidance law design for the injection problem with a 3D model and initial deviations under final constrained conditions and time has been studied. The main goal of the research is to investigate the effectiveness of the Vectorized high-order expansion (VHOE) method for extracting the near-optimal path and minimizing the final condition errors. After reviewing the high-order extrapolation methods, the VHOE method is briefly reviewed and then used to solve the injection problem up to the third order. This guidance law is minimally sensitive to large initial state deviations and can be implemented online with minimal computations in practice. The effectiveness of the high-order extrapolation method is examined through Monte Carlo simulations, and it is shown that the guidance law based on the VHOE method not only has adequate accuracy but can also be a suitable alternative for the path-following problem.

Keywords: *Vectorized high order expansion, non-linear optimal control, booster landing, optimal guidance.*

Introduction

Finding the optimal path to reach the desired orbit with path constraints is a classical optimal control problem [1-6]. Many theoretical efforts have been made to analyze this problem. Accurate or approximate analytical solutions are obtained based on the dynamics of the problem. To solve the problem practically, various methods have been developed, which are categorized as direct, indirect, and hybrid methods.

Direct methods [6-10] transform the optimal control problem into a nonlinear programming (NLP) problem in which many discrete approaches can be envisioned. Then the optimization problem at a large scale is solved by nonlinear programming software. However, solving the problem at a large scale makes the solution computationally expensive.

Indirect methods [11-13] transform the optimal control problem into a boundary value problem. In optimal control, the maximum Pontryagin's principle is explicitly or implicitly used. The uncertainties of the problem are initial costs that must be considered to estimate the desired conditions at the final time. Hybrid approaches attempt to combine direct and indirect methods to take advantage of their respective qualities [15-14].

1 Assistant Professor
2 Assistant Professor (Corresponding Author) Email: *m.jafari.h@mut.ac.ir
3 PhD

10.22034/jast.2023.388139.1145
Submit: 03.03.2023 / Accepted: 05.05.2023
Print ISSN:2345-3648 Online ISSN: 1735-2134

A direct method is first used to create a good initial guess, and an indirect method is used to achieve accurate convergence. Among the numerous hybrid techniques applied to optimize the path, particle swarm optimization [16], genetic algorithms [17], and Hamilton-Jacobi-Bellman-based dynamic programming [18] can be mentioned.

In 2008, the high-order expansion method was studied using differential algebra to solve space dynamics problems in [19].

Differential algebra is a tool used to perform algebraic operations not only on the value of a function at a particular independent variable but also on its higher-order derivatives involved in equations. Therefore, more information can be extracted from solving a problem. This method was developed by Martin Berz [20] in 1999 for solving problems in particle physics.

Another method for robust guidance in space guidance problems is also investigated and proposed based on the high-order differential algebra method in [21]. In this method, the general concepts of the reference [19] are preserved, and the optimal control problem for generating guidance commands is solved.

The difference is that this time the researchers have aimed to calculate the correction of the guidance command due to the deviation in the state variable.

This article examines two examples: the first example is the problem of spacecraft guidance with continuous propulsion from Earth to Mars, and the second example is similar to the problem of using drag to reduce speed during the trip to Mars, which was also studied in the previous reference.

In [22], a similar method to previous references has been implemented for several different examples, and the governing equations of the desired guidance problem dynamics have been extracted. In another study, the problem of interest is modeled with limited values of propulsion force using the differential algebra method in [23].

The reason for considering this change in modeling the problem is that if the constraint on the size of the propulsion force in guidance is not taken into account, the size of the guidance command is likely to be large at the beginning of the problem, and in real-world problems, it is necessary to limit this value because it is beyond the physical capabilities of an implementation.

One of the other interesting applications of high-order extrapolation and the use of differential algebra is presented in [24].

Another application of high-order extrapolation and the use of differential algebra in its computation is studied in [25], where the problem of transferring a spacecraft from the Lagrange point towards the moon and its collision with the moon to destroy the spacecraft is considered.

Since this maneuver is the last possible maneuver for the spacecraft, which may be due to its being out of order or similar reasons, fuel is naturally limited for this maneuver, and multiple orbital corrections are not possible. Therefore, the designated path for this mission must be precise. Sometimes, a change or deviation in the final time of the problem is also likely to have a non-negligible effect on solving the dynamic equations of the problem.

In [26], by normalizing the time variable, a state variable is added to the sum of the problem variables, with the initial value of this variable representing the nominal time. Then, the deviation from this value is considered a parameter, and the deviation from the final time in the problem is easily modeled. A study in [27] has shown the advantage of using high-order differential algebraic extrapolation for this purpose.

In this study, the deviation from the reference path due to the mentioned disturbances has been predicted for six different bodies using fourth-order differential algebraic extrapolation. These bodies are in various orbits, and their deviation due to initial disturbances has been studied.

Then, this study was completed in [28-29], and in this study, the probability of collision was estimated using Monte Carlo analysis. Recently, in [30], this issue has also been studied in the software package (MISS) under the COMPASS project of the European Union, in which methods for analyzing the probability of collision between spacecraft or satellites with each other or with space debris have been studied.

A different approach to high-order extrapolation, called Vector High-Order Expansion (VHOE), is introduced in [31] by considering the optimal control problem for guiding an unmanned aerial vehicle during the landing phase. In this method, differential algebra is not used, and the necessary mathematical structure for performing calculations has been implemented by introducing sensitivity variables.

In [32], the line-of-sight equations are used to extract a guidance law for an unmanned aerial vehicle that performs consecutive maneuvers to pass through predetermined points. Finally, high-order vector extrapolation in [33] is used to design an

almost optimal guidance law. In [35, 34], the authors used high-order extrapolation for the vertical landing problem of a satellite on Falcon 9 and compared it with various methods, including SDRE. Also, in [36], a method for online updating of VHOE is introduced. In this study, the desired problem is modeled in optimal control, and its mission is to guide the spacecraft to enter the atmosphere at high speed to reduce its speed and then exit it at a specific final time and conditions.

This article investigates the design of an optimal guidance law for the satellite problem, considering constrained final conditions and time. This guidance law not only establishes a measure of optimality but also has minimal sensitivity to large initial state deviations because the effect of nonlinear terms in the mathematical model of the guidance problem is considered using high-order extrapolation. In section 2, the problem formulation is presented, and a mathematical model for solving the satellite problem in three dimensions is provided. After obtaining the nominal solution to the problem using the GPOPS extension in section 3, the problem is solved considering deviations in the initial conditions. In section 4, a simulation of the nominal path with a path-tracking regulator is examined, and in section 5, after obtaining the guidance law using high-order vector extrapolation with the help of the Taylor series, a Monte Carlo simulation is performed to examine this guidance law with 2,000 runs. Finally, it is shown that the guidance law based on high-order vector extrapolation using the Taylor series is not only sufficiently accurate but can also be a good alternative to the nominal path-tracking problem.

Problem Formulation

To solve the satellite guidance problem and meet its related requirements using orthogonal functions, two different mathematical models will be introduced. The first mathematical model will be used to implement a 3-degree-of-freedom simulation and linearization for use in the implementation of a regulator. However, the second mathematical model will be used to solve the guidance problem, namely to extract the optimal guidance path and commands and to extract the sensitivities (using both Taylor series expansion and orthogonal functions).

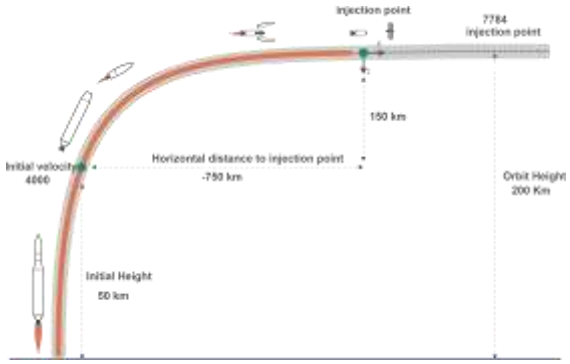

Figure 1: Schematic diagram of the circuit injection problem [19].

Mathematical model for simulation

Assuming a particle in this case, Newton's second law has been applied in the Cartesian inertial system at the injection point, and the motion equations for 3D are presented in the form of equation (1),

$$\begin{gathered}\dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \\ \dot{v_x} = \frac{L\cos\gamma\cos\chi - Y\sin\chi - D\sin\gamma\cos\chi}{m} + \frac{T_x}{m} \\ \dot{v_y} = \frac{L\cos\gamma\sin\chi + Y\cos\chi - D\sin\gamma\sin\chi}{m} + \frac{T_y}{m} \\ \dot{v_z} = \frac{-L\sin\gamma - D\cos\gamma}{m} + g + \frac{T_z}{m} \\ \dot{m} = -\frac{\sqrt{T_x^2 + T_y^2 + T_z^2}}{I_{sp}g0} \\ \dot{\alpha} = \alpha_{rate} \\ \dot{\beta} = \beta_{rate}\end{gathered} \quad (1)$$

In these equations, x , y , z , v_x, v_y , v_z , m , α , β are the state variables of the problem, and α_{rate}, β_{rate}, T_x, T_y, and T_z are the control variables that will be used for booster guidance. The selected control variables are the rates of change of the attack angle, the side-slip angle, and the thrust force in the horizontal, out-of-plane, and vertical directions, respectively. A liquid-fueled satellite booster engine is considered.

In this set of equations, density and gravity are related by the following equations:

$$g = g_0\left(\frac{R_E}{R_E - z}\right)^2, \quad \rho = \rho_0 e^{-\frac{h}{h_0}} \quad (2)$$

In this equation, h is equal to -z, where $h_0 = 7.5km$ km and $\rho_0 = 1.225\frac{kg}{m^3}$ are respectively the reference altitude and sea level density, and g_0 is the

gravitational acceleration at zero altitudes, which is equivalent to $9.81\frac{m}{s^2}$. $R_E = 6378\ km$ is the radius of the Earth, which is equal to $R_E = 6378\ km$. L, D, and Y represent lift, drag, and side force, respectively.

Mathematical model for solving the 3D satellite guidance problem

Now in this section, a mathematical model is presented for the satellite guidance problem, which is slightly different from the simulation model. This model is based on the implementation of Newton's second law in the implementation path system, while the previous model was extracted in the inertial system connected to the Earth. According to [19], the mathematical model is in the form of the following.

$$\begin{aligned}
\dot{x} &= v\sin\gamma\cos\chi \\
\dot{y} &= v\sin\gamma\sin\chi \\
\dot{z} &= v\cos\gamma \\
\dot{v} &= -\frac{D}{m} + g\cos\gamma + \frac{T}{m}\cos\alpha\cos\beta \\
\dot{\chi} &= \frac{Y}{mv\sin\gamma} + \frac{T\cos\alpha\sin\beta}{mv\sin\gamma} \\
\dot{\gamma} &= \frac{L}{mv} + \frac{T\sin\alpha}{mv} - \frac{g}{v}\sin\gamma \\
\dot{\alpha} &= \alpha_{rate} \\
\dot{\beta} &= \beta_{rate} \\
\dot{m} &= -\frac{T}{I_{sp}g0} \\
T &= constant
\end{aligned} \tag{3}$$

In this equation, the state variables x, y, z, v, χ, γ, α, β, and m represent the range, lateral deviation, altitude, speed magnitude, heading angle, flight path angle, attack angle, side-slip angle, and mass, respectively. On the other hand, α_{rate} and β_{rate} are the control variables of the problem, which represent the attack angle rate and the side-slip angle rate, respectively. Note that T is the thrust, the value of which should be selected. It is also assumed that the thrust vector is aligned with the satellite's longitudinal axis and its magnitude is constant. Finally, the equations related to the calculation of aerodynamic forces are considered as follows:

$$\begin{aligned}
L &= \frac{1}{2}\rho v^2 s C_{L_\alpha}\sin 2\alpha \\
D &= \frac{1}{2}\rho v^2 s C_{D_0} \\
Y &= \frac{1}{2}\rho v^2 s C_{L_\alpha}\sin 2\beta
\end{aligned} \tag{4}$$

Nominal solution of the guidance problem

The first step in solving the high-order guidance problem is to extract the nominal solution of the problem. GPOPS will be used for this purpose. It should be noted that, like variable mass simulation, the problem is solved in GPOPS. In the specific problem, the goal is to satisfy the injection conditions in the orbit, taking into account the initial conditions with a range of 900 kilometers, a cross-range of 30 kilometers, and a height of 50 kilometers, while the initial mass is 25000 kilograms and the target orbit height is 200 kilometers. In addition, the initial values of velocity, angle of attack, side slip angle, flight path angle, heading angle, mission time, and thrust size are determined by GPOPS to optimally calculate these values during the optimization process. All values, parameters, and aerodynamic coefficients are adapted from reference [19]. To obtain the solution using GPOPS, cost function (5) is defined.

$$J = \int_0^{t_f}\left\{\frac{1}{2}\alpha^2 + \frac{1}{2}\beta^2 + \frac{1}{2}100\alpha_{rate}^2 + \frac{1}{2}100\beta_{rate}^2\right\}dt \tag{5}$$

In this cost function, the angle of attack, side slip angle, and all control variables are weighted. In addition, all control variables are considered to perform the orbital injection operation with minimum angle of attack and side slip and minimum rotation rate. The optimization results are shown in the following graphs. To extract this nominal trajectory, the complete form of the problem equations is considered in terms of density, gravity, and variable mass, and the modeling is also three-dimensional. The final conditions are as follows:

$$
\begin{aligned}
x(t_f) &= 0 \\
y(t_f) &= 0 \\
z(t_f) &= 0 \\
v(t_f) &= \sqrt{\frac{\mu}{R + h_{orbit}}} \\
\gamma(t_f) &= \frac{\pi}{2} \\
\chi(t_f) &= 0 \\
\alpha(t_f) &= 0 \\
\beta(t_f) &= 0
\end{aligned}
\tag{6}
$$

It is defined that according to this definition, the final value of the range and the cross-range are zero, while the final height is considered to be 200 kilometers (i.e., z is equal to zero). The final velocity is also considered to be equal to the velocity required for a circular orbit at an altitude of 200 kilometers. Finally, after solving GPOPS, in addition to obtaining the initial and final values of the state variables assigned to GPOPS, the total mission time, the time history of all state variables, pseudo-state, and control variables are determined by GPOPS. Figure (2) shows the 3D trajectory and its time history.

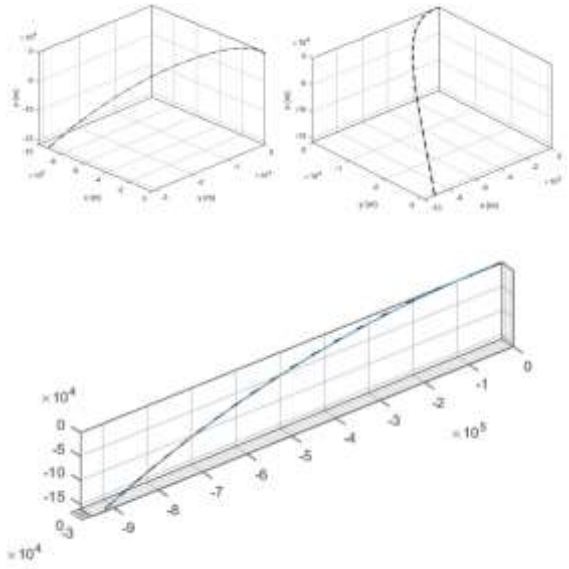

Figure 2: shows the nominal 3D trajectory from three views.

Solving the optimal control problem with deviation in initial conditions

Now, deviations in the initial conditions of the problem are to be considered. Real-world problems are always prone to disturbances and uncertainties. Therefore, if the initial conditions at the time of implementing the guidance law, for example, in the implementation of a real-world problem or simulation, are different from the nominal value, the application of the optimal control command obtained in the previous step will lead to significant deviations in the final conditions. The second objective of this problem is to provide a method for compensating for these deviations. A common solution is to use famous methods such as LQ, SDRE, MPC, and other similar methods for tracking the nominal trajectory.

These methods are widely used in practice, but the main problem is that returning to the nominal trajectory, especially when the initial deviation is large, is not optimal. However, assuming that the problem with the deviated initial conditions is itself a new optimal control problem, a better solution is to find a new optimal path and replace it with the previous optimal path. However, it should be noted that solving the optimal control problem again should be done online, and the use of numerical methods such as spectral methods is not cost-effective in terms of time and computational cost. Additionally, due to the iterative nature of numerical solutions, the possibility of solution divergence exists. However, a suitable method to solve this problem is to replace the nominal trajectory using the high-order expansion method, which has an offline computational burden. But with one calculation of the coefficients, it can be used online, and depending on the size of the initial deviation, the optimal control command and optimal trajectory can be replaced. The implementation of this method will be discussed in the next section.

High-order vector expansion for multi-variable problems

This section introduces an effective mathematical tool for modeling multi-variable problems in the form of high-order vector expansion, and the necessary algorithms for implementing this method are presented. Therefore, with the definition of operators, the appearance of the implementation of multi-dimensional problems will be similar to one-dimensional problems because their calculations are implemented in the form of matrix-vector equations. With this method, even the Taylor series for a function with multiple independent variables can be represented as a simple equation. Let us assume that the problem under consideration for Taylor series expansion is a dynamic problem (e.g., optimal

control) with two variables (e.g., a state variable and a pseudo-state variable):

$$\begin{aligned}\dot{x} &= f(x,\lambda)\\ \dot{\lambda} &= g(x,\lambda)\end{aligned} \tag{7}$$

Now, with the Taylor series expansion

$$\begin{aligned}\dot{x} = f(x_n,\lambda_n) &+ \frac{\partial f}{\partial x}\delta x + \frac{\partial f}{\partial \lambda}\delta\lambda + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}\delta x^2\\ &+ \frac{1}{2}\frac{\partial^2 f}{\partial x \partial \lambda}2\delta x\delta\lambda\\ &+ \frac{1}{2}\frac{\partial^2 f}{\partial \lambda^2}\delta\lambda^2 + \cdots\end{aligned} \tag{8}$$

$$\begin{aligned}\dot{\lambda} = g(x_n,\lambda_n) &+ \frac{\partial g}{\partial x}\delta x + \frac{\partial g}{\partial \lambda}\delta\lambda\\ &+ \frac{1}{2}\frac{\partial^2 g}{\partial x^2}\delta x^2\\ &+ \frac{1}{2}\frac{\partial^2 g}{\partial x \partial \lambda}2\delta x\delta\lambda\\ &+ \frac{1}{2}\frac{\partial^2 g}{\partial \lambda^2}\delta\lambda^2 + \cdots\end{aligned}$$

$$\begin{aligned}&\dot{x}\\ &= f(x_n,\lambda_n) + \begin{bmatrix}\frac{\partial f}{\partial x} & \frac{\partial f}{\partial \lambda}\end{bmatrix}\begin{bmatrix}\delta x\\ \delta\lambda\end{bmatrix}\\ &+ \begin{bmatrix}\frac{1}{2}\frac{\partial^2 f}{\partial x^2} & \frac{1}{2}\frac{\partial^2 f}{\partial x\partial\lambda} & \frac{1}{2}\frac{\partial^2 f}{\partial \lambda^2}\end{bmatrix}\begin{bmatrix}\delta x^2\\ 2\delta x\delta\lambda\\ \delta\lambda^2\end{bmatrix} + \cdots\\ &\dot{\lambda}\\ &= g(x_n,\lambda_n) + \begin{bmatrix}\frac{\partial g}{\partial x} & \frac{\partial g}{\partial \lambda}\end{bmatrix}\begin{bmatrix}\delta x\\ \delta\lambda\end{bmatrix}\\ &+ \begin{bmatrix}\frac{1}{2}\frac{\partial^2 g}{\partial x^2} & \frac{1}{2}\frac{\partial^2 g}{\partial x\partial\lambda} & \frac{1}{2}\frac{\partial^2 g}{\partial \lambda^2}\end{bmatrix}\begin{bmatrix}\delta x^2\\ 2\delta x\delta\lambda\\ \delta\lambda^2\end{bmatrix} + \cdots\end{aligned} \tag{9}$$

These equations can still be written in a more compact form as below,

$$\begin{aligned}&\begin{bmatrix}\dot{x}\\ \dot{\lambda}\end{bmatrix}\\ &= \begin{bmatrix}f(x_n,\lambda_n)\\ g(x_n,\lambda_n)\end{bmatrix} + \begin{bmatrix}\frac{\partial f}{\partial x} & \frac{\partial f}{\partial \lambda}\\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial \lambda}\end{bmatrix}\begin{bmatrix}\delta x\\ \delta\lambda\end{bmatrix}\\ &+ \begin{bmatrix}\frac{1}{2}\frac{\partial^2 f}{\partial x^2} & \frac{1}{2}\frac{\partial^2 f}{\partial x\partial\lambda} & \frac{1}{2}\frac{\partial^2 f}{\partial \lambda^2}\\ \frac{1}{2}\frac{\partial^2 g}{\partial x^2} & \frac{1}{2}\frac{\partial^2 g}{\partial x\partial\lambda} & \frac{1}{2}\frac{\partial^2 g}{\partial \lambda^2}\end{bmatrix}\begin{bmatrix}\delta x^2\\ 2\delta x\delta\lambda\\ \delta\lambda^2\end{bmatrix} + \cdots\end{aligned} \tag{10}$$

And finally, assuming that $\boldsymbol{\delta z} = [\delta x \quad \delta\lambda]^T$, equation (10) can be written in a very concise form using complete vector expansion [31-33]:

$$\begin{aligned}\begin{bmatrix}\dot{x}\\ \dot{\lambda}\end{bmatrix} &= \begin{bmatrix}f(x_n,\lambda_n)\\ g(x_n,\lambda_n)\end{bmatrix} + \boldsymbol{\Delta}_1\langle\boldsymbol{\delta z}\rangle^1\\ &+ \boldsymbol{\Delta}_2\langle\boldsymbol{\delta z}\rangle^2\end{aligned} \tag{11}$$

$$\boldsymbol{\Delta}_1 = \begin{bmatrix}\frac{\partial f}{\partial x} & \frac{\partial f}{\partial \lambda}\\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial \lambda}\end{bmatrix} \tag{12}$$

$$\boldsymbol{\Delta}_2 = \begin{bmatrix}\frac{1}{2}\frac{\partial^2 f}{\partial x^2} & \frac{1}{2}\frac{\partial^2 f}{\partial x\partial\lambda} & \frac{1}{2}\frac{\partial^2 f}{\partial \lambda^2}\\ \frac{1}{2}\frac{\partial^2 g}{\partial x^2} & \frac{1}{2}\frac{\partial^2 g}{\partial x\partial\lambda} & \frac{1}{2}\frac{\partial^2 g}{\partial \lambda^2}\end{bmatrix}$$

If we move a nominal part to the left-hand side of the equation and simplify it, it yields:

$$\begin{bmatrix}\dot{\delta x}\\ \dot{\delta\lambda}\end{bmatrix} = \boldsymbol{\Delta}_1\langle\boldsymbol{\delta z}\rangle^1 + \boldsymbol{\Delta}_2\langle\boldsymbol{\delta z}\rangle^2 + \cdots \tag{13}$$

Here, $\langle\boldsymbol{\delta z}\rangle^1 open-angle$ refers to the vector expansion of order 1 and higher. Since what comes next holds true,

$$\begin{bmatrix}\dot{\delta x}\\ \dot{\delta\lambda}\end{bmatrix} = \frac{d}{dt}\boldsymbol{\delta z} = \frac{d}{dt}\langle\boldsymbol{\delta z}\rangle^1 \tag{14}$$

Finally the following is obtained,

$$\begin{aligned}\langle\dot{\boldsymbol{\delta}}z\rangle^1 &= \boldsymbol{\Delta}_1\langle\boldsymbol{\delta z}\rangle^1 + \boldsymbol{\Delta}_2\langle\boldsymbol{\delta z}\rangle^2 + \cdots\\ \langle\dot{\boldsymbol{\delta}}z\rangle^1 &= \sum_{i=1}^{m}\boldsymbol{\Delta}_i\langle\boldsymbol{\delta z}\rangle^i\end{aligned} \tag{15}$$

General Solution of High-Order Optimal Control Problems

Assuming that the optimal control problem is in the following form:

$$\begin{aligned}\dot{\boldsymbol{s}} &= f(\boldsymbol{s},\boldsymbol{\lambda})\\ \dot{\boldsymbol{\lambda}} &= g(\boldsymbol{s},\boldsymbol{\lambda})\end{aligned} \tag{16}$$

In this model, it is assumed that the optimal control commands are explicitly calculable functions of the state and pseudo-state variables and can be substituted into the problem to write the dynamic equations in the above form. In equation (16), s and λ are vectors of state and pseudo-state variables, respectively, each with n elements. Also, the functions f and g are differentiable up to any desired

order m. Now, assuming that $\boldsymbol{z} = [\boldsymbol{s} \quad \boldsymbol{\lambda}]^T$, the above set of equations can be summarized in the following equation:

$$\dot{\boldsymbol{z}} = h(\boldsymbol{z}) \tag{17}$$

In the next step, assuming that the nominal values for s and λ are known, the right-hand side of the above equation is expanded in a Taylor series around the nominal solution up to any desired order m, and then the nominal solution values are eliminated from both sides of the equation. Therefore:

$$\dot{\boldsymbol{\delta}}z = \sum_{i=1}^{m} \boldsymbol{\Delta}_i \langle \boldsymbol{\delta} z \rangle^i \tag{18}$$

In the above equation, the Taylor series expansion is written using vector expansion as described, so $\boldsymbol{\Delta}_i$ in this equation is a matrix with time-varying elements (since the Taylor series expansion is around the nominal solution) representing the coefficients of the Taylor series corresponding to the dimensions of the $\langle \boldsymbol{\delta} z \rangle^i$ vector. Furthermore, $\boldsymbol{\delta} z = [\boldsymbol{\delta} s \quad \boldsymbol{\delta} \lambda]^T$ represents the deviations of the state and pseudo-state variables from their nominal values, as defined earlier.

It is assumed here that in the defined optimal control problem, the initial and final conditions of the state variables are constrained. The goal is to extract δλ and δs for a general value of $\boldsymbol{\delta} s(0) = \boldsymbol{\delta} s_0$, subject to the condition that $\boldsymbol{\delta} s(t_f) = \boldsymbol{\delta} s_f = 0$.

For a specific value of $\boldsymbol{\delta} s_0$, equation (18), which is a boundary value problem with separate values, can be solved using numerical methods. However, if $\boldsymbol{\delta} s_0$ changes, the problem must be solved again for this new value. To address this issue, it is assumed that δλ and δs are functions of an indeterminate form (which must be determined) of $\boldsymbol{\delta} s_0$ (which is the initial deviation):

$$\begin{aligned} \boldsymbol{\delta} s &= \boldsymbol{\delta} s(\boldsymbol{\delta} s_0) \\ \boldsymbol{\delta} \lambda &= \boldsymbol{\delta} \lambda(\boldsymbol{\delta} s_0) \end{aligned} \tag{19}$$

As a result, it can be written for $\boldsymbol{\delta} z$:

$$\boldsymbol{\delta} z = \boldsymbol{\delta} z(\boldsymbol{\delta} s_0) \tag{20}$$

Now, similar to the assumption made for the expansion of equation (17), here equation (20) is also rewritten in terms of a Taylor series using vector expansion:

$$\boldsymbol{\delta} z = \sum_{i=1}^{m} \boldsymbol{\Psi}_i \langle \boldsymbol{\delta} s_0 \rangle^i \tag{21}$$

In this equation, $\boldsymbol{\Psi}_i$ is the matrix of Taylor series coefficients that are indeterminate and time-varying. These coefficients are known as sensitivity coefficients. In other words, by obtaining these coefficients, $\boldsymbol{\delta} z$ can be calculated for different values of $\boldsymbol{\delta} s_0$, and these coefficients indicate the sensitivity of the state and pseudo-state variable deviations to the initial state deviation. To obtain the sensitivity coefficients, equation (21) is substituted into equation (17). After performing the calculations and equating terms with equal powers, it can be seen that $\boldsymbol{\delta} s_0$ is eliminated from the calculations, and the following set of equations can be presented to calculate the sensitivity of order i:

$$\dot{\boldsymbol{\Psi}}_i = \boldsymbol{\Delta}_1 \boldsymbol{\Psi}_i + \boldsymbol{\Omega}_i \tag{22}$$

Where $\boldsymbol{\Omega}_i$ is the nonhomogeneous matrix of linear differential equations provided for the sensitivities. For $i = 1$, the nonhomogeneous part is equal to zero, and for $i > 1$, this value is calculated in terms of lower-order sensitivities. Further details on the calculation method can be found in references [31,32].

Therefore, to calculate the sensitivity matrices, starting from $i = 1$, $\boldsymbol{\Psi}_1$ can be calculated, and then in the next step, after calculating $\boldsymbol{\Omega}_2$ in terms of $\boldsymbol{\Psi}_1$, $\boldsymbol{\Psi}_2$ can be calculated. Consequently, these steps can be continued until $\boldsymbol{\Psi}_m$ is calculated. Finally, with the sensitivities available, $\boldsymbol{\delta} \lambda$ and $\boldsymbol{\delta} s$ can be calculated for each $\boldsymbol{\delta} s_0$ based on equation (21).

Suppose that by solving the corresponding differential equations, the sensitivity matrices are extracted. Now these matrices are separated into upper and lower blocks (from the middle) as follows,

$$\boldsymbol{\delta} z = \sum_{i=1}^{m} \begin{bmatrix} \boldsymbol{\Psi}_i^U \\ \boldsymbol{\Psi}_i^L \end{bmatrix} \langle \boldsymbol{\delta} s_0 \rangle^i \tag{23}$$

Where the upper block $\boldsymbol{\Psi}_i^U$ and the lower block $\boldsymbol{\Psi}_i^L$ are separated from the matrix, since:

$$\boldsymbol{\delta} z = \begin{bmatrix} \boldsymbol{\delta} s \\ \boldsymbol{\delta} \lambda \end{bmatrix} \tag{24}$$

As a result,

$$\begin{bmatrix} \boldsymbol{\delta} s \\ \boldsymbol{\delta} \lambda \end{bmatrix} = \sum_{i=1}^{m} \begin{bmatrix} \boldsymbol{\Psi}_i^U \\ \boldsymbol{\Psi}_i^L \end{bmatrix} \langle \boldsymbol{\delta} s_0 \rangle^i \tag{25}$$

By separating the terms, it can be written as follows,

$$\begin{aligned}\boldsymbol{\delta s} &= \sum_{i=1}^{m} \boldsymbol{\Psi}_i^U \langle \boldsymbol{\delta s}_0 \rangle^i \\ \boldsymbol{\delta \lambda} &= \sum_{i=1}^{m} \boldsymbol{\Psi}_i^L \langle \boldsymbol{\delta s}_0 \rangle^i\end{aligned} \tag{26}$$

Where the first equation is for the state variables and the second equation is for the pseudo-state variables. Now, only the equation related to the state variable is considered (since the description of the sensitivities for the pseudo-state variables is similar). By expanding this equation the following is achieved,

$$\boldsymbol{\delta s} = \boldsymbol{\Psi}_1^U \langle \boldsymbol{\delta s}_0 \rangle^1 + \boldsymbol{\Psi}_2^U \langle \boldsymbol{\delta s}_0 \rangle^2 + \cdots \tag{27}$$

If we assume that the problem has two state variables, then $\boldsymbol{\delta s} = [\delta x \quad \delta y]^T$, and the sensitivity matrix $\boldsymbol{\Psi}_i$will be a 2x2 matrix with indeterminate and time-varying coefficients.

$$\begin{bmatrix}\delta x \\ \delta y\end{bmatrix} = \boldsymbol{\Psi}_1^U \langle \boldsymbol{\delta s}_0 \rangle^1 + \boldsymbol{\Psi}_2^U \langle \boldsymbol{\delta s}_0 \rangle^2 + \cdots \tag{28}$$

Therefore, based on that,

$$\begin{bmatrix}\delta x \\ \delta y\end{bmatrix} = \boldsymbol{\Psi}_1^U \begin{bmatrix}\delta x_0 \\ \delta y_0\end{bmatrix} + \boldsymbol{\Psi}_2^U \begin{bmatrix}\delta x_0^2 \\ 2\delta x_0 \delta y_0 \\ \delta y_0^2\end{bmatrix} + \cdots \tag{29}$$

By writing the sensitivity matrices in terms of their components:

$$\begin{aligned}&\begin{bmatrix}\delta x \\ \delta y\end{bmatrix} \\ &= \begin{bmatrix}S^x_{\delta x_0} & S^x_{\delta y_0} \\ S^y_{\delta x_0} & S^y_{\delta y_0}\end{bmatrix} \begin{bmatrix}\delta x_0 \\ \delta y_0\end{bmatrix} \\ &+ \begin{bmatrix}S^x_{\delta x_0^2} & S^x_{\delta x_0 \delta y_0} & S^x_{\delta y_0^2} \\ S^y_{\delta x_0^2} & S^y_{\delta x_0 \delta y_0} & S^y_{\delta y_0^2}\end{bmatrix} \begin{bmatrix}\delta x_0^2 \\ 2\delta x_0 \delta y_0 \\ \delta y_0^2\end{bmatrix} \\ &+ \cdots\end{aligned} \tag{30}$$

Now, by selecting each row, the key equation for calculating the deviations (in terms of initial deviations and sensitivities) for each state variable can be obtained. For example, for δx, it yields:

$$\begin{aligned}\delta x = S^x_{\delta x_0} \delta x_0 + S^x_{\delta y_0} \delta y_0 \\ + \left\{ S^x_{\delta x_0^2} \delta x_0^2 \right. \\ + S^y_{\delta x_0 \delta y_0} 2\delta x_0 \delta y_0 \\ \left. + S^y_{\delta y_0^2} \delta y_0^2 + \cdots \right\}\end{aligned} \tag{31}$$

This process can be carried out for all the state and pseudo-state variables, and the deviation values can be calculated at all times based on the initial deviations and sensitivities.

Now that the sensitivity coefficients have been extracted, they can be used to obtain the following,

$$\begin{cases} s_n^{new}(t) = s_n^{old}(t) + \delta s(t) \\ \lambda_n^{new}(t) = \lambda_n^{old}(t) + \delta \lambda(t) \end{cases} \tag{32}$$

In simple terms, this means that by adding the computed deviation values to the nominal solution, a new nominal solution can be obtained. Therefore, since λ_n^{new} has been extracted, the new nominal control command can also be obtained using $u_n^{new} = -\frac{B}{r}\lambda_n^{new}$.

If the cost function is quadratic and the control is affine, this equation can be used, where B is the control variable coefficient in the differential equations and R is the control variable weighting matrix in the cost function. For more details on the implementation of the high-order vector extrapolation method for multi-variable systems, readers can refer to [31-33].

Simulation of Nominal Path with Path Following Regulator

In the previous section, the nominal solution for the injection problem was obtained using GPOPS software, and its path was presented. Here, this nominal solution is used in the simulation and a linear LQ regulator is implemented to execute the commands of this solution. Note that in the GPOPS simulation and solution, the satellite mass is variable based on the engine thrust, and its changes are fully considered. However, numerical solutions of the open-loop optimal control problem always have numerical errors, albeit small ones. These errors will become significant if a regulator is not used. Therefore, an LQ regulator is used here to follow the nominal path and compensate for computational errors. In total, 8 state variables and 5 control variables are defined for the tracking problem. The

block diagram for implementing path guidance and tracking is presented in Figure (3).

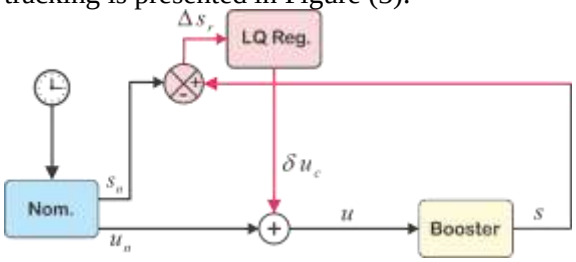

Figure 3: Block diagram of path guidance and pursuit implementation.

Moreover, 8 separate simulations were performed. 4 simulations were conducted with initial deviation only in position, and 4 simulations were conducted with initial deviation only in speed. The 3D path of motion is shown in Figure (4). Deviations of 1 and 1- kilometers were considered in the x-direction, and deviations of 2 and 2- kilometers were considered in the transverse direction. For deviations in speed, values of 10 and 10 meters per second were considered in the z-channel, and 15 and 15 meters per second were considered in the y-channel. The satellite, with the help of an LQ regulator, attempted to compensate for these deviations and converge to the injection point in orbit. Table 1 shows the initial deviations.

Table 1: Initial deviations for single-run scenario simulations

S.N.	$\delta v_{z_0}[\frac{m}{s}]$	$\delta v_{y_0}[\frac{m}{s}]$	$\delta y_0[m]$	$\delta x_0[m]$
1	0	0	0	1000
2	0	0	0	-1000
3	0	0	2000	0
4	0	0	-2000	0
5	0	15	0	0
6		-15	0	0
7	10	0	0	0
8	-10	0	0	0

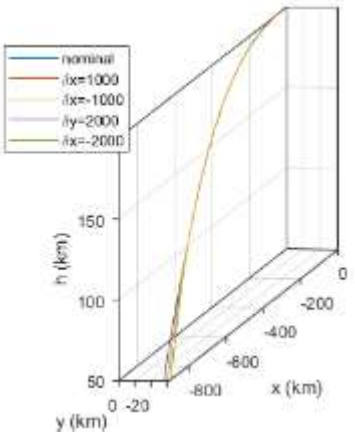

Figure 4: 3D path for initial position deviations.

In the above figures, only nominal path tracking has been performed, and no new optimal path has been generated. In the following, using the high-order vector expansion method based on the Taylor series, the optimal path without pursuit has been extracted.

Extraction of guidance law using high-order expansion method with the help of the Taylor series

Now that the nominal solution has been extracted and its quality has been examined, in this section, the third-order solution of the 3D injection problem is extracted and several simulation scenarios are then conducted. For the first step, the nonlinear problem needs to be approximated with the Taylor series, and the equations in their standard form should be presented. Given the cost function in equation (5), if the Hamiltonian is in the form shown below,

$$
\begin{aligned}
H = {} & \frac{1}{2}\alpha^2 + \frac{1}{2}\beta^2 + \frac{1}{2}100\alpha_{rate}^2 \\
& + \frac{1}{2}100\beta_{rate}^2 + \\
& \lambda_x\{v\sin\gamma\cos\chi\} + \lambda_y\{v\sin\gamma\sin\chi\} + \\
& \lambda_z\{v\cos\gamma\} + \lambda_v\left\{-\frac{D}{m} + g\cos\gamma \right. \\
& \left. - T'^{\cos\alpha\cos\beta}\right\} + \\
& \lambda_\chi\left\{\frac{Y}{mv\sin\gamma} - \frac{T'\cos\alpha\sin\beta}{v\sin\gamma}\right\} + \\
& \lambda_\gamma\left\{\frac{L}{mv} - \frac{T'^{\sin\alpha}}{v} - \frac{g}{v}\sin\gamma\right\} + \\
& \lambda_\alpha\{\alpha_{rate}\} + \lambda_\beta\{\beta_{rate}\}
\end{aligned}
\tag{33}
$$

Now, the standard form of the high-order problem for solution is as follows,

$$\begin{cases} \dot{x} = h_x = v\sin\gamma\cos\chi \\ \dot{y} = h_y = v\sin\gamma\sin\chi \\ \dot{z} = h_z = v\cos\gamma \\ \dot{v} = h_v = -\dfrac{D}{m} + g\cos\gamma - \dfrac{T}{m}\cos\alpha\cos\beta \\ \dot{\chi} = h_\chi = \dfrac{Y}{mv\sin\gamma} - \dfrac{T\cos\alpha\sin\beta}{mv\sin\gamma} \\ \dot{\gamma} = h_\gamma = \dfrac{L}{mv} - \dfrac{T\,sin\,\alpha}{mv} - \dfrac{g}{v}\cos\gamma \\ \dot{\alpha} = h_\alpha = \alpha_{rate} \\ \dot{\beta} = h_\beta = \beta_{rate} \end{cases}$$

$$\begin{cases} \dot{\lambda_x} = h_{\lambda_x} = -\dfrac{\partial H}{\partial x} \\ \dot{\lambda_y} = h_{\lambda_y} = -\dfrac{\partial H}{\partial y} \\ \dot{\lambda_z} = h_{\lambda_z} = -\dfrac{\partial H}{\partial z} \\ \dot{\lambda_v} = h_{\lambda_v} = -\dfrac{\partial H}{\partial v} \\ \dot{\lambda_\chi} = h_{\lambda_\chi} = -\dfrac{\partial H}{\partial \chi} \\ \dot{\lambda_\gamma} = h_{\lambda_\gamma} = -\dfrac{\partial H}{\partial \gamma} \\ \dot{\lambda_\alpha} = h_{\lambda_\alpha} = -\dfrac{\partial H}{\partial \alpha} \\ \dot{\lambda_\beta} = h_{\lambda_\beta} = -\dfrac{\partial H}{\partial \beta} \end{cases}, \quad \begin{cases} \dfrac{\partial H}{\partial \alpha_{rate}} = 0 \rightarrow \alpha_{rate} \\ \dfrac{\partial H}{\partial \beta_{rate}} = 0 \rightarrow \beta_{rate} \end{cases} \tag{34}$$

is presented. The first part of this equation provides the differential equations for the state variables and defines h_x to h_β, in which the control variables must be substituted into both sets of differential equations for the state and the pseudo-state variables based on the optimality principle. The differential equations for the pseudo-state variables are provided above, and the definitions of h_{λ_x} to h_{λ_β} are also presented. Now, we can write:

$$\begin{cases} s_n^{new}(t) = s_n^{old}(t) + \delta s(t) \\ \lambda_n^{new}(t) = \lambda_n^{old}(t) + \delta\lambda(t) \end{cases} \tag{35}$$

This includes a total of 16 nonlinear equations, which need to be Taylor-expanded up to the third order concerning the 16 state and pseudo-state variables. Therefore, Δ_i consists of 16 rows, where the first row corresponds to the derivatives of h_x, the second row corresponds to the derivatives of h_y, and so on up to h_{λ_β}. Next, the sensitivity equations should be solved after the model is extracted using the corresponding equation.

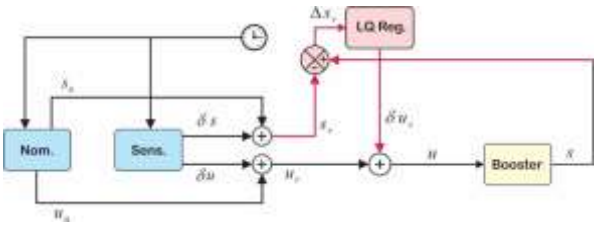

Figure 5: Block diagram of implementing the 3-DOF guidance law using the high-order expansion method.

Now, the simulation results are presented here. According to the table of single-run scenarios presented earlier, 8 different scenarios with different initial deviations, considering the calculated sensitivities using the Taylor series expansion, have

been simulated. It can be seen that the results are qualitatively different from similar simulations that only converged to the nominal path, where here the reference path is generated, and the satellite follows the reference path with the help of a tracker.

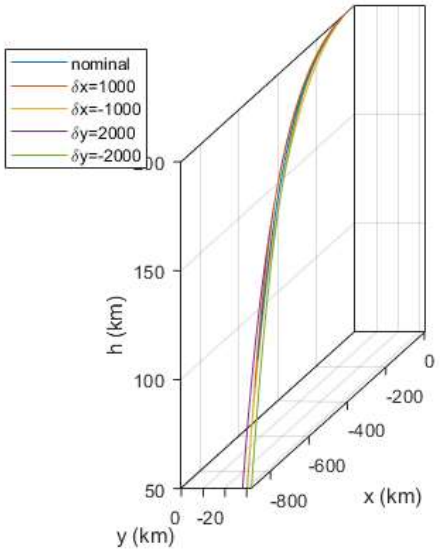

Figure 6: 3D path using high-order Taylor series expansion for initial position deviations.

Monte Carlo simulation

To evaluate the guidance law, Monte Carlo simulations are necessary as relying on a few simulations is not sufficient for a comprehensive evaluation. Therefore, Monte Carlo simulation is performed for the guidance law. For the Monte Carlo simulation, it is assumed that the deviation in x ranges from -1000 to 1000 meters, for y from -2000 to 2000 meters, and for the velocity components v_y and v_z from -15 to 15 meters per second and -10 to 10 meters per second, respectively. In these Monte Carlo simulations, it is assumed that the simulation starts from an altitude of 50 kilometers, and as a result, the initial deviation in altitude is not considered. Moreover, since the mission's duration and the trust size are fixed, the guidance law has little flexibility concerning velocity deviations, and thus only brief deviations in the y and z directions are considered for velocity. As a result, the satellite's end-phase error should be budgeted according to the mission requirements, which limits the errors.

This Monte Carlo simulation includes 2000 runs, where the initial deviation values are selected from a uniform distribution.

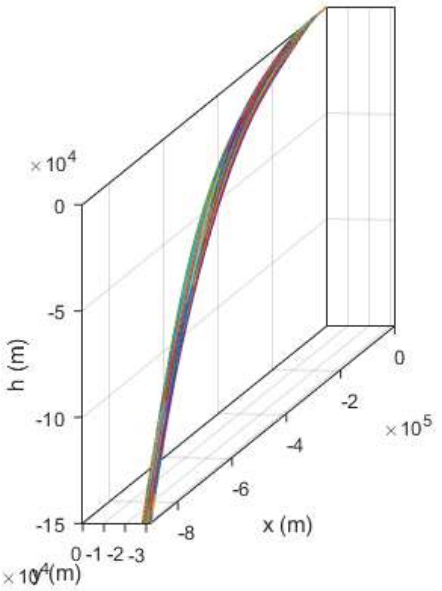

Figure 7: Path curves for Monte Carlo simulation with high-order Taylor series guidance law.

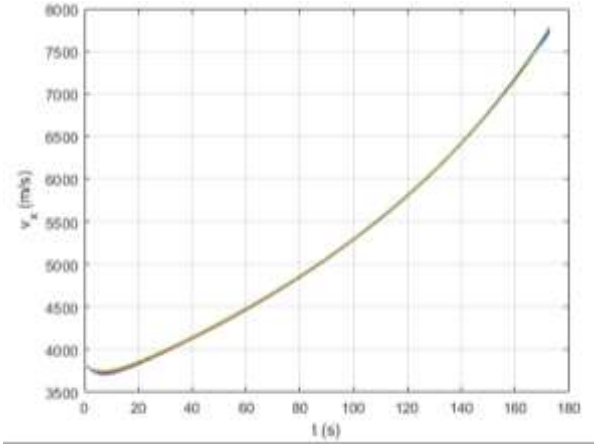

Figure 8: Velocity x-component curves for Monte Carlo simulation with high-order Taylor series guidance law.

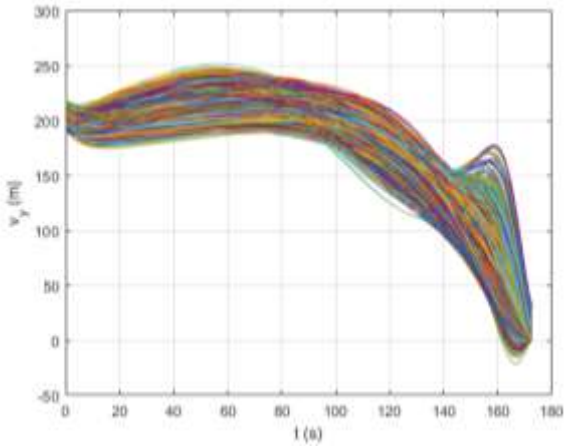

Figure 9: Velocity y-component curves for Monte Carlo simulation with high-order Taylor series guidance law.

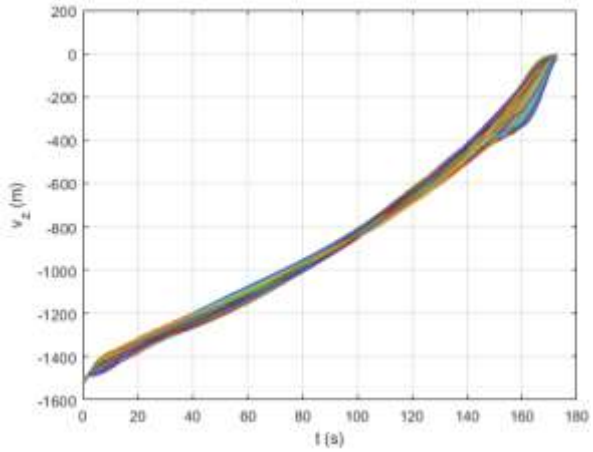

Figure 10: Velocity z-component curves for Monte Carlo simulation with high-order Taylor series guidance law.

Table 2: Final position and velocity errors for the guidance law.

Taylor series-based guidance law						
error	$v_z\ (m/s)$	$v_y\ (m/s)$	$v_x\ (m/s)$	$z\ (m)$	$y\ (m)$	$x\ (m)$
mean	-0.3	0.62	-2.51	-2.09	11.4	-106.3
standard deviation	1.9	2.7	6.57	10.07	25.98	257.5

The results presented in Table 2 demonstrate the high accuracy of the high-order vector extrapolation method. In the case of the optimal guidance law for this mission, two important features are of interest: the first is the satisfaction of the position and velocity accuracy at the final point, and the second is the size of the cost function. So far, it has been shown that the position and velocity errors for the Taylor series-based guidance law have high quality based on statistical data. Now, the cost function is examined. The table below shows the statistical results for the total cost function, cost function for the tracker, final mass, and consumed mass (fuel mass). The Taylor series-based guidance law has shown high quality.

Table 3: Cost function error

mean	Total Cost	Regulator Cost
standard deviation	9189	353.91
mean	9508	1186.3

Conclusion

The main objective of this research is to investigate the development and enhancement of the high-order expansion method for optimal guidance law design, and various implementations of this method have been examined using different approaches. After studying the implementation method and its concepts, the effectiveness of the high-order expansion method and its implementation has been investigated. In this study, satellite guidance has been considered, and the main goal is to achieve the minimum possible error at the final point. In this article, the problem of three-dimensional satellite guidance has been addressed, and after modeling and obtaining the optimal nominal solution, sensitivity variables have been extracted using the high-order expansion method up to the third order. To examine theoretical errors, 8 simulations with different initial deviations were performed. Subsequently, more comprehensive simulations were conducted to evaluate the quality of the high-order expansion method in the presence of uncertainties and to assess the impact of higher-order terms. The simulations were implemented by taking into account the variable's gravity, density, and mass. Finally, Monte Carlo simulations were used to compare the quality of the high-order expansion-based guidance law, and it was shown that the guidance law based on high-order expansion using the Taylor series is not only accurate but can also be a good substitute for the nominal trajectory tracking problem.

References

[1] Leitmann, G., On a class of variational problems in rocket flight, Journal of Aerospace Sci. 26, no. 9, (Sep.1959), 586-591.
[2] Leitmann, G., Optimization techniques with applications to Aerospace System, Academic Press New-York,1962.
[3] Lawden, D.F. Optimal trajectories for space navigation, Butterworths Publishing Corporation, 1963.
[4] Neustadt, L.W., A general theory of minimum-fuel space trajectories, SIAM Journal on Control 3, no. 2 (1965), 317--356.
[5] Conway, B.A., Spacecraft Trajectory Optimization, Cambridge University Press,2010.
[6] M. Alavipour, A. A. Nikkhah, J. Roshanian, Minimum time multiple-burn optimization of an upper stage with a finite thrust for satellite injection into geostationary orbit, Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 2017.
[7] Hargraves, C. R., Paris, S. W., Direct trajectory optimization using nonlinear programming and collocation, Journal of Guidance, Control, and Dynamics, vol. 0, no. 4, pp. 338–342, 1987.
[8] Enright, P. J., Conway, B. A., Discrete approximations to optimal trajectories using direct transcription and nonlinear programming, Journal of Guidance, Control, and Dynamics, vol. 15, no. 4, pp. 994–1002, 1992.
[9] Betts, J. T., Very low thrust trajectory optimization using a direct SQP method, Journal of Computational and applied Mathematics, vol. 120, issues1-2, pp. 27–40, Aug. 2000.

[10] Ross, I. M., Fahroo, F., Pseudospectral knotting methods for solving optimal control problems, Journal of Guidance, Control, and Dynamics, vol. 27, no. 3, pp. 397–405, 2004.
[11] Pontryagin, L., Boltyanskii, V., Gramkrelidze, R., Mischenko, E., The mathematical theory of optimal processes, Wiley Interscience, 1962.
[12] Hull, D.G., Optimal control theory for applications, Springer, 2003.
[13] Trélat, E., Contrôle optimal – Théorie et Applications, Vuibert, 2005.
[14] Boltyanski, V.G., Poznyak, A.S., The robust maximum principle, Birkhäuser, 2012.
[15] Betts, J. T., Survey of numerical methods for trajectory optimization, Journal of Guidance, Control, and Dynamics, vol. 21, no. 2, pp. 193–207, 1998.
[16] Rao, A., A survey of numerical methods for optimal control, Advances in the Astronautical Sciences, vol. 135, no. 1, pp. 497–528, 2009.
[17] Pontani, M., Conway, B.A., Particle Swarm Optimization Applied to Space Trajectories, Journal of Guidance, Control, and Dynamics, Vol. 33, No. 5 (2010), pp. 1429-1441, DOI: 10.2514/1.48475.
[18] Steffens, M.J., A combined global and local methodology for launch vehicle trajectory design-space exploration and optimization, Thesis Georgia Institute of technology, April 2014.
[19] Li, Y., Chen, W., Zhou, H., and Yang, L., "Conjugate Gradient Method with Pseudospectral Collocation Scheme for Optimal Rocket Landing Guidance", Aerospace Science and Technology, doi: 10.1016/j.ast.2020.105999, Vol. 104, Article Number. 105999, (2017).
[20] Di Lizia, P., Armellin, R., and Lavagna, M., "Application of High Order Expansions of Two-point Boundary Value Problems to Astrodynamics", Celestial Mechanics and Dynamical Astronomy, doi: 10.1007/s10569-008-9170-5, Vol. 102, No. 4, pp. 355–375, (2008).
[21] Berz, M., "*Advances in Imaging and Electron Physics Modern Map Methods in Particle Beam Physics",* 1st Edition, Academic Press: San Diego, Vol.108, No.1, (1999). pp.210-305.
[22] Di Lizia, P., Armellin, A., Ercoli-Finzi, A., and Berz, M., "High-order Robust Guidance of Interplanetary Trajectories Based on Differential Algebra", Journal of Aerospace Engineering, doi: 10.7446/jaesa.0101.05, Vol. 1, No. 1, pp. 43–57, (2008).
[23] Di Lizia, P., Armellin, A., Bernelli-Zazzera, F., and Berz, M., "High Order Optimal Control of Space Trajectories with Uncertain Boundary Conditions", Acta Astronaut, doi: 10.1016/j.actaastro.2013.07.007, Vol. 93, pp. 217–229, (2014).
[24] Di Lizia, P., Armellin, R., Morselli, A., and Bernelli-Zazzera, F., "High Order Optimal Feedback Control of Space Trajectories with Bounded Control", Acta Astronaut, doi: 10.1016/j.actaastro.2013.02.011, Vol. 94, No. 1, pp. 383–394, (2014).
[25] Wittig, A., and Armellin, R., "High Order Transfer Maps for Perturbed Keplerian Motion", Celestial Mechanics and Dynamical Astronomy, doi: 10.1007/s10569-015-9621-8, Vol. 122, No. 4, pp. 333–358, (2015).
[26] Vetrisano, M., and Vasile, M., "Analysis of Spacecraft Disposal Solutions from LPO to the Moon with High Order Polynomial Expansions", Advances in Space Research, doi: 10.1016/j.asr.2017.04.005, Vol. 60, No. 1, pp. 38–56, (2017).
[27] Sun, Z. J., Di Lizia, P., Bernelli-Zazzera, F., Luo, Y.Z., and Lin, K.P., "High-order State Transition Polynomial with Time Expansion Based on Differential Algebra", Acta (Astronautica) , doi: 10.1016/j.actaastro.2019.03.068, Vol. 163, Part B, No.7, pp. 45–55, (2019).
[28] Morselli, A., Armellin, R., Di Lizia, P., and Bernelli Zazzera, F., "A High Order Method for Orbital Conjunctions Analysis: Sensitivity to Initial Uncertainties", Advances in Space Research, doi: 10.1016/j.asr.2013.11.038, Vol. 53, No. 3, pp. 490–508, (2014).
[29] Morselli, A., Armellin, R., Di Lizia, P., and Bernelli Zazzera, F., "A High Order Method for Orbital Conjunctions Analysis: Monte Carlo Collision Probability Computation", Advances in Space Research, doi: 10.1016/j.asr.2014.09.003, Vol. 55, No. 1, pp. 311–333, (2015).
[30] Gonzalo, J.L., Colombo, C., and Di Lizia, P., "Introducing MISS, a New Tool for Collision Avoidance Analysis and Design", Journal of Space Safety Engineering, doi: 10.1016/j.jsse.2020.07.010, Vol. 7, No. 3, pp. 282–289, (2020).
[31] Moghadasian, M., and Roshanian, J., "Optimal Landing of Unmanned Aerial Vehicle using Vectorised High Order Expansions Method", Modares Mechanical Engineering, Vol. 19, No. 11, pp. 2761–2769, (2019).
[32] Moghadasian, M., and Roshanian, J., "Continuous Maneuver of Unmanned Aerial Vehicle using High Order Expansions Method for Optimal Control Problem", Modares Mechanical Engineering, Vol. 17, No. 12, pp. 382–390, (2018).
[33] Moghadasian, M., and Roshanian, J., "Approximately Optimal Manoeuvre Strategy for Aero-assisted Space Mission", Advances in Space Research, doi: 10.1016/j.asr.2019.04.003, Vol. 64, No. 2, pp. 436–450, (2019).
[34] M.Sharafi, N.Rahbar, A.Moharampour, A.R.Kashaninia, "Comparing Performance of Vectorized High Order Expansions and SDRE Method for Vertical Landing Mission of Booster" *Journal of Aerospace Mechanics*, Vol.18, No.3, pp 69-85 , 2022.
[35] M.Sharafi, N.Rahbar, A.Moharampour, A.R.Kashaninia, "Designing the Nonlinear Guidance Law Adaptable to Initial Deviations for the Vertical Landing of the Booster" *JAST*, Vol 15, No 1,pp 57-65 ,2022.
[36] M.Sharafi, N.Rahbar, A.Moharampour, A.R.Kashaninia "Performance Analysis of the Vectorized High Order Expansions Method in the Accurate Landing Problem of Reusable Boosters" *Advances in Space Research*, Vol 71, Issue 5, pp 2155-2174, 2023.
[37] RP Optimization Research LLC, "GPOPS-II: Next-Generation Optimal Control Software," 2016. [Online]. Available: https://www.gpops2.com/.