Complex Dynamical Orbit Re-entry Navigation and Control

David Cole and Timothy Sands *

Posted Date: 12 December 2023

doi: 10.20944/preprints202312.0787.v1

Keywords: convex optimization; trajectory optimization; path planning; kinematics

Article

Complex Dynamical Orbit Re-Entry Navigation and Control

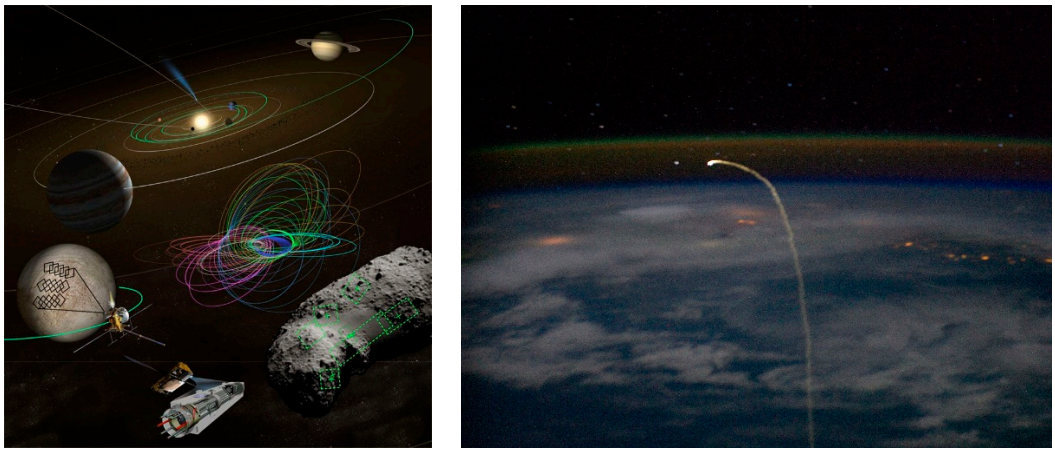
David Cole [1] and Timothy Sands [2,*]

[1] Department of Mechanical & Aerospace Engineering, Naval Postgraduate School, Monterey, CA 93940 USA
[2] Department of Mechanical Engineering (SCPD), Stanford University, Stanford, CA 94305, USA
* Correspondence: dr.timsands@alumni.stanford.edu

Abstract: Calculating a minimum-time trajectory requires the solving of a boundary value problem resulting from invocation of necessary conditions of optimality to set up a set of ordinary differential equations to solve the trajectory between the initial position and desired ending position with set constraints on the path between the two. This manuscript expresses the minimum time optimum trajectory between a satellite in a geosynchronous orbit and a target set on earth's surface utilizing a non-rotating earth centric coordinate system. Expressing motion in coordinates of rotating reference frames necessitates transformation between reference frames, and one such transformation is embodied in the direction cosine matrices formed by a sequence of three successive frame rotations. Rotation about the local wing of an aerospace vehicle is almost never the pitch angle, yet modern application of kinematics often assumes such (with accompanying angular error). The same assertion is usually true about the nature of roll and yaw angles. This manuscript evaluates which sequence is the most advantageous for an object starting in space and then traveling through the atmosphere to a targeted spot on the Earth's surface. Simulation precision (validated using a quaternion normalization condition) reveals the most accurate sequence with an average error of 0.14° and a computational time of 0.013 seconds: resulting in a 97.95% increase in accuracy over the ubiquitous benchmark rotation sequence and a 99.84% increase over the well-known "orbital rotation sequence". Subsequently, an optimal trajectory candidate is proposed and illustrated to yield a total flight time of 2 hours, 34 minutes and 46 seconds, or an average velocity of 3.85 kilometers per second, reaching a targeted spot with a velocity of 11.54 kilometers per second.

Keywords: convex optimization; trajectory optimization; path planning; kinematics

1. Introduction



(**a**) (**b**)

Figure 1. (**a**) Guidance, Navigation, and Control Technology Assessment image. [1] Credit: NASA, Patricia M. Beauchamp. (b) Shuttle STS-135 mission reentry streak from orbit. [2] Credit: NASA, International space station Expedition 28 crew, usage compliant with image use policy [3].

 doi:10.20944/preprints202312.0787.v1

The study presented in this manuscript seek to minimize errors to increase safe deliveries, and the focus is twofold: 1) errors due the choice of kinematic representation (i.e., "representation errors"), where there is a difference between the mathematical representation and the beliefs about the representation by the users; and 2) reentry trajectory optimization to minimize the delivery time.

1.1. Errors Induced by Choice of Kinematic Representation

The study of forces in action, known as kinetics, combined with the properties of motion without reference to forces, or kinematics, forms the study of dynamic motion. In 1775 Leonhard Euler, a Swiss mathematician, began the formal study of kinematics [1]. Following this Baron William Thomson, better known as Lord Kelvin, and a physicist named Guthrie Tait [5] showed that system of dynamic motion maintains energy conservation. A decade later in 1876, Theoretische Kinematik by Franz Reuleaux was republished in English [6] and became a staple in engineering courses from Italy to Russia and the United States by taking practices of engineers and compiling them into a single source document. Prior to the end of the century Thomas Wallace Wright compiled necessary topics for kinematics into a single disquisition [7] that was specifically designed to allow mathematics as simple as basic geometry and trigonometry while still allowing calculus for more compact forms. Five years later in 1903 John Theodore Merz, a German British chemist attempted to unify the kinematic field of thought by publishing a history of European thought on the subject [8].

In a philosophical effort to give credit to the works of Chasle, Hamilton and Rodrigues, Sir Edmund Whittaker published four editions in 1904 [9-12]. In contrast to the approach of Whittaker, Irving Porter Church focused on the engineering approach and published his own guide in 1908 [13]. In an effort to build on Wrights work [14] and reduce the mathematical requirements further Christian Hugo Eduard Study, another German mathematician published reference [15] in 1913. Five years later Andrew Gray argued that the key facet to understand dynamics is kinematics in reference [16]. Gray's assertion was reemphasized in 1954 when Clifford Truesdell stated, "I cannot too strongly urge that a kinematical result is a result valid forever, no matter how time and fashion may change the "laws" of physics" [17] and then again in 1967 when G.C. Fox stated "It is my belief that students have difficulty with mechanics because of an inadequate knowledge of kinematics" [18].

In the second half of 1950's Kane focused on kinetics and expounded on D'Alembert's version of Lagrange's equation [20,21]. Computational capabilities drastically approved over the following decade and were utilized by Fang in order to develop a way to mathematically simulate rigid body rotational kinematics [22]. Henderson, in the late 1970's, further elaborated on the kinematic relationships between rotational sequences as well as the four-dimensional quaternion and direction cosine matrices [23,24]. From this history, the sequential rotation sequences were formed with the two specific sequences being termed the aerospace sequence about repeating axes, sometimes referred to as the "Tait-Bryan angles", and the orbital sequence using axis-type repetitions, or the "proper Euler angles".

In the 1980's, ubiquitous acceptance of two rotation sequences (the so-called "orbital" sequence, and the so–called "aerospace" sequence) appeared in the literature and mature textbooks. [25] In the following decade of the 1990's renewed interest in improvements was evidenced amidst the ubiquitously accepted paradigms elaborated in a comprehensive survey presented in [26] and evaluation of linearization schemes in reference [27]. Renewed interest in the errors resulting from a chosen kinematic selection was sparked at conference in the United Kingdom at the turn of the century [28] and elaborated for attitude estimation the same year in the *Journal of Guidance, Control, and Dynamics*. [29] The differences in choices for parameterizing attitude was presented geometrically in 2013. [30] From the available rotation choices, the question may be posed: of the twelve possible sequences, which is the best to convert inertial coordinates to body coordinates? In 2018, Smeresky and Rizzo [31] studied the sequence known as the so–called "aerospace sequence" (also referred to as "3-2-1") and concluded that the yaw and pitch angles do not accurately depict a rotation about the body's z–axis and y–axis respectively. Conversely, only the roll angle, or last angle in the rotation sequence, is accurate in the rotation about the body's x-axis. These facts drive some of the

 doi:10.20944/preprints202312.0787.v1

innovations in this manuscript. In 2019, [32] presented the full analytical derivations of the attitude error kinematics equations.

Two figures of merit are used to evaluate the twelve rotation sequences: mean and standard deviation of the representation error to indicate how the tested sequence compares to true roll, pitch and yaw angles and computational burden as shown by total calculation time which shows relative numerical superiority. As an aside, this manuscript separates itself from the work of Smeresky, et. Al, [31] by calculating absolute representation error and precise execution time of the rotation sequence as opposed to tracking errors and end-to-end simulation run-time. Instead, the kinematics portions of this manuscript repeat and validate just–published work. [33]. Meanwhile, the application of Pontryagin's minimization methods [34] follow the procedures laid out by Sandberg [35] as elaborated in [36] are utilized for the trajectory optimization portions of this manuscript's proposals, importantly notice time minimization is proposed here, where Sandberg [35] and Raigoza [37] optimized for minimum fuel usage and Raigoza proposed autonomous collision avoidance during reentry.

The results from kinematic analysis show that the so-called aerospace sequence and the reverse (123 sequence) are contrasting in error and computational time with the aerospace sequence being superior between the two. Additionally, the 123 sequence results in a significantly slower computational time than all other rotation sequences. Symmetric rotation sequences, or those that rotate about the same axis first and third in order, average slower computational times than non-symmetric, or rotations involving all three axes, despite requiring the mathematical process and steps to solve for Euler angles. Of all the non-symmetric rotation sequences, the "aerospace" rotation sequence (also called the "321") was the fastest, while the fastest symmetrical sequence was the "232" demonstrating lower computational requirements.

1.2. Reentry Trajectory Optimization to Minimize the Delivery Time

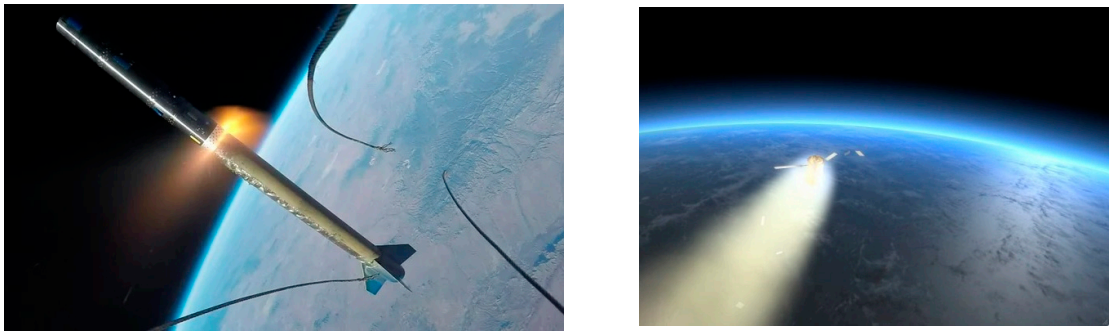
There are many terms that can be associated with an optimum trajectory; whether the problem is looking for the path of least resistance, the path that provides the best fuel burn, or the path that takes the least amount of time to get from point a to point b. Most of the time, these paths are not direct lines connecting the two points. Knowing the starting and end points turn the selection of the flight path into a boundary value problem solved using ordinary differential equations that can be solved in a multitude of ways. The first, and possibly least complex, method is known as the shooting method where the problem is turned into an initial value problem. A major downfall to the shooting method is initial guesses are required for any unknown initial values. If the provided initial guesses are even minutely off, the entire problem will "implode" and fail to provide a viable solution as the problem is extremely sensitive to initial guess errors [NEW 1]. A second method is known as the collocation method where the boundary value problem is provided a set number of points to form a mesh between the start and end points and this mesh is then continuously analyzed to form an optimal path between the two [NEW2]. As the dynamics and variables used are all known positions, the collocation method eliminates the errors induced by guessing errors but does typically require more computing power than the shooting method. The MATLAB function bvp4c is a built-in program that uses the collocation method. Lastly, multiple companies have developed their own proprietary programs that will solve boundary value problems to a set accuracy. This manuscript uses the program "DIDO" to solve the dynamic optimal problem for a missile set in geosynchronous orbit to a target on the earth's surface. An advantage to using DIDO is its ability to solve complex dynamic problems without requiring the user to generate the Hamiltonian or any costates or endpoint vectors [NEW3]. By not having to code the Hamiltonian or costate equations, the opportunities for human error in coding is limited to the initial complex dynamic functions.

In the case of a missile delivered from orbit, the amount of fuel used, while a consideration, does not outweigh the necessity to strike the target as fast as possible. A solid rocket motor will burn at a set rate and does not require the oxygen of the atmosphere to burn, and as such can be planned for as a set constant depending on flight time. To validate the results achieved, the Hamiltonian and all costates are developed to concur with the products from DIDO's program. By manually computing

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

the Hamiltonian and associated equations, natural points are produced that can be analyzed to ensure the validity of the simulation. The results from this manuscript produce a minimum time trajectory from the launch position to chosen target site with a flight time of two hours, thirty-four minutes and forty-six seconds, a fraction of the time required to launch in the opposite direction.

B. Broad context and proposed innovations



(**a**) (**b**)

Figure 2. (**a**) Re-entry mission by UP Aerospace [38] Credit:, usage compliant with image use policy [39] (**b**) European ATV-2 re-entry and breakup because of the atmospheric forces of Earth. [40] Illustration credit: ESA/D. Ducros, usage compliant with image use policy [41].

Just last year, [42] proposed and investigated the potential of using suborbital space vehicles for the transportation of cargo to precise delivery locations. Towards achievement of the necessary high–precision of delivery to ensure safety, this manuscript describes selection of kinematics representations to minimize errors and also includes a formal time–minimizing optimization problem seeking rapid delivery capabilities. Three-dimensional (rotational and translational) motion with six degrees of freedom when applied to an object delivered from geostationary orbit. Due to size constraints in modern systems and to minimize necessary computing power, efficient schemes are desired to provide vehicle guidance. Determining the most accurate and computationally efficient kinematic representation is, therefore, essential to producing an effective space delivery capability.

1. This research simulates an object launched from geosynchronous orbit with limited computing power. Proposed innovations include:
2. Identification of an optimal (minimum time) trajectory from a set launch point to any target on the surface of the earth as measured by minimal time of flight.
3. Identification of most accurate rotation sequence, where accuracy is evaluated based on ability to correctly represent rotations about body axes (the paradigm expected by a pilot).
4. Identification of the options with advantageous computational requirements measured in minimal computational time.

2. Materials and Methods

Due to the design of the project, the optimal trajectory is one that takes the least amount of time for the object to arrive at the targeted destination within a set number of constraints. Pontryagin's Principle was used to accomplish this minimum time trajectory by solving a boundary value problem with ordinary differential equations.

2.1. Dynamic Problem Framing

A state variable is one that describes the corresponding state of a dynamic system and can be written as a function of a control variable and the dynamic function of that system. When the state dynamic functions are known, the performance function can be written as Equation 1, where J is the standard performance function, x is a state vector, u is the control variable, E is the endpoint performance (also known as the cost function) and F is the running performance (also known as the running cost).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

$$J[x(\cdot), u(\cdot)] = E(x_f) + \int_{t_0}^{t_f} F(x(t), u(t))\, dt \tag{1}$$

To minimize the performance function, the state variables are normalized using a common unit of measurement, CU, to ensure that the units make sense. The common unit is used to create a costate identified by λ as shown in Equation 2.

$$\lambda = \frac{CU}{x_{unit}} \tag{2}$$

Each state and costate variable do not just consist of a single value but is instead a vector and combines to create the value ω as shown in Equation 3.

$$\omega(x) = \lambda \cdot x = \lambda_1 x_1 + \lambda_2 x_2 + \cdots \lambda_{N_x} x_{N_x} \tag{3}$$

Knowing this combination, the units of Equation 1 are the common unit established to form the costate vector since the endpoint performance is composed of the common unit and the running performance is the common unit per time unit integrated.

Table 1. Table of proximal variable definitions.

Variable/concept	Definition	Variable/concept	Definition
J	Performance Function	x	General State Vector
F	Running Performance Function	E	Endpoint Performance Function
u	Control Variable	λ	Costate Variable
CU	Common Unit		

2.2. Pontryagin's Principles

Pontryagin used a series of equations to determine a candidate optimal solution written as "HAMVET" [36]. These letters represent the sequence: Hamiltonian, Adjoint Equation, Minimization of the **Ham**iltonian, **V**alue Condition, **E**volution Condition, and **T**ransversality. The process is called HAzMAT in [34].

2.2.1. Hamiltonian construction

The Hamiltonian function is defined by the running cost and costate transposed as a function of the state and control variables as shown in Equation 4.

$$H(\lambda, x, u) = F(x, u) + \lambda^T f(x, u) \tag{4}$$

2.2.2. Adjoint Equations

The adjoint equations describe the rate of change of the costate vector and is defined as the partial derivative of the Hamiltonian with respect to the corresponding state vector as shown in Equation 5.

$$\dot{\lambda} = -\frac{\partial H}{\partial x} \tag{5}$$

By utilizing the adjoint equation, a boundary value problem is created. Knowing the endpoints, or the beginning and ending conditions, the optimal trajectory connecting the endpoints can be solved for.

2.2.3. Minimization of the Hamiltonian

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

The Hamiltonian is minimized with respect to the control variable only as shown in Equation 6 in effort to provide additional variables to set as boundary points for the overall solution. Additionally, minimizing with respect to the control variables provides a verification and validation point to check the solution.

$$\frac{\partial H}{\partial u} = 0 \tag{6}$$

2.2.4 Final Value Condition

In certain problems when the total time is unknown or complicated, the Value Condition becomes necessary to provide an optimal stopping condition and is defined as $\bar{E}$ as shown in Equations 7 and 8.

$$H\left|@t_f\right| - \frac{\partial \bar{E}}{\partial t_f} \tag{7}$$

$$\bar{E} = t_f + v_1\left(x_f - x^f\right) + v_2\left(x_f - x^f\right) + \cdots \tag{8}$$

Table 2. Table of proximal variable definitions.

Variable/concept	Definition	Variable/concept	Definition
H	Hamiltonian Function	$\bar{E}$	Final Value Condition
F	Running Performance	λ	Costate Variable
x	State Variable	u	Control Variable
t	Time		

2.2.5 Hamiltonian Evolution Condition

In some situations, a control input will not provide a constant value Hamiltonian, but instead require control constraints. An example of a situation requiring a control constraint is an airplane with a maximum deflection angle of a control surface. At some point, the control surface will reach its maximum deflection and require a different control input to maintain trajectory. These control constraints are placed into the problem as shown in Equation 9.

$$u^L \leq u(t) \leq u^U \tag{9}$$

Karush observed that using this constraint, if the control variable is equal to the lower constraint, then the partial of the Hamiltonian with respect to that variable must be greater than zero. Similarly, if the control variable equals the maximum constraint, then the partial is less than zero. Finally, if the control variable is between the two constraints, then the partial equals zero. The problem with this solution is that it relies on the assumption that the Hamiltonian has a constant slope with respect to the control variable instead of realizing the possibility that the relationship is not constant in nature. To account for the possibility of a nonlinear relationship, the Lagrangian of the Hamiltonian, $\bar{H}$, is created as shown in Equation 10 where μ is another control variable such that the partial of the Hamiltonian plus this variable is equal to zero.

$$\bar{H}(\mu, \lambda, x, u, t) \coloneqq H(\lambda, x, u, t) + \mu^T \mu \tag{10}$$

Setting the partial derivative of the Lagrangian of the Hamiltonian with respect to the control variable equal to zero allows the use of what is known as the Karush-Kuhn-Tucker, or KKT, conditions. The two conditions are the stationarity condition and the complementarity condition shown in Equations 11 and 12.

$$\frac{\partial H}{\partial u} = 0 \tag{11}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

$$\mu_i \begin{cases} \leq 0 \\ = 0 \\ \geq 0 \end{cases} \quad \begin{matrix} u_i = u_i^{lower} \\ u_i^{lower} < u_i < u_i^{upper} \\ u_i = u_i^{upper} \end{matrix} \tag{12}$$

This process changes the basic boundary value problem established with the formation of the adjoint equation into a differential algebraic non-smooth boundary problem but allows for the possibility of a nonlinear control.

Table 3. Table of proximal variable definitions.

Variable/concept	Definition	Variable/concept	Definition
u	Control Variable	H	Hamiltonian Function
μ	Stationarity Control Variable	λ	Costate Variable
x	State Variable	t	Time
E	Endpoint Performance		

2.2.6. Transversality

Transversality is a tool that enables the generation of additional boundaries necessary to solve the boundary value problem by checking the ending value of the costate vector by setting it equal to the change in the Endpoint Lagrangian developed by the Endpoint Value Condition, defined by Equation 8, with respect to the change in the final condition of each respective state as shown in Equation 13.

$$\lambda_x(t_f) = \frac{\partial \bar{E}}{\partial x_f} \tag{13}$$

2.3. Solving the Boundary Value Problem

There are a few different methods in which to attempt to determine an optimal solution to the boundary value problem. The first, known as the shooting method, is a simple iterative approach in which the boundary value problem is transformed into an initial value problem using the known initial values and subsequently guessing on the remaining unknown initial values. These values are then iterated forward through the state dynamic equations to see if the final conditions set forth are met. If the required conditions are not met, the initial conditions that were guessed are changed and equations are run again. This process repeats until the conditions do meet at which point the solution becomes a candidate for the optimal solution pending verification and validation. The shooting method is extremely sensitive to small errors in the initial values used for the problem as these errors are then magnified through the dynamic state equations during the iteration.

The second method of solving the boundary value problem is known as the collocation method. The collocation method creates a mesh of a set number of points between the initial condition to the final condition and then adjusts each of these points until all conditions of the BVP are satisfied through the dynamic equations given. The collocation method requires more guesses than the shooting method as any unknown final conditions will need to be guessed in addition to the initial conditions. Additionally, this method requires a great deal more computing power.

Lastly, some companies or corporations create their own software designed to solve the boundary value problem. These programs are typically not either shooting or collocation methods, but instead generate their own algorithm to produce the solution. For this project, a program called "DIDO" [43] is used that takes the problem with the dynamic equations and formats Pontryagin's Principle into a boundary value problem and then produces a candidate solution. To understand the solution that DIDO gives, it is important that the use works through Pontryagin's Principle and conduct verification or validation in order to determine if the candidate solution is truly optimal.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

For kinetics, this manuscript will discuss the use of Newton's and Euler's principles and specifically how they combine to form Chasle's theorem to form a direct line to explaining the six degrees of freedom in kinetics. Direction cosines and the matrices they form, or quaternions, describe the body of an object in terms of a predefined axis.

2.4. Kinetics of Rotational Motion

In accordance with Euler's equations of motion, the sums of externally applied torques will angularly accelerate a proportional mass moment of inertia as depicted in Equation 14, where τ is torque, J is the rotation tensor, and ω is the angular velocity.

$$\tau|_n = J\alpha_n \rightarrow \tau = J\dot{\omega} + \omega \times J\omega \tag{14}$$

2.5. Kinetics of Translational Motion

Similarly, an object of a defined mass will accelerate proportionally to the sum of forces acting on that mass in accordance with Newton's second law as depicted in Equation 15 where F is the force, m is the defined mass, a is the acceleration, ω is angular velocity, r′ is the position vector, and v′ is velocity vector.

$$F = ma + m\frac{d\omega}{dt} \times r' + 2m\omega \times v' + m\omega \times (\omega \times r') \tag{15}$$

Newton's law applies in the inertial frame and allows us to express our results in a set of coordinates in another defined frame of reference.

Table 4. Table of proximal variable definitions.

Variable/concept	Definition	Variable/concept	Definition
τ	Torque	J	Rotation Tensor
ω	Angular Velocity	m	Mass
a	Acceleration	v	Velocity
r′	Position Vector	t	Time

2.6. Kinetics of Disturbance Forces and Torques

Disturbance forces and torques act on a body in accordance with equations (14) and (15) and are accounted for within the Simulink model for simulation found in Appendix (A).

2.7. Kinetics of Motion in Three Degrees

By utilizing direction cosines, we are able to take a set of coordinates relative to the non-rotating inertial frame and express them in terms of the body frame to account for motion in all three degrees as depicted in Figure 1, where all lines are unit vectors and the thin solid line represents the non-rotating inertial frame and each dash line increasing in weight indicates the first prime, second prime and finally the body frame respectively.

Table 5. Table of proximal variable definitions.

Variable/concept	Definition	Variable/concept	Definition
θ	Pitch Euler Angle	ϕ	Roll Euler Angle
ψ	Yaw Euler Angle	B	Body Axis Designation
i	Inertial Axis	‘ and ‘’	Intermediate Axes

The DCM is labeled in the order that the mathematical calculations are solved, right to left in accordance with algebraic multiplication rules to pre-multiply the total transformation matrix. Equation (16) represents the three mathematical calculations for a "3-2-1" rotation and is then multiplied to generate the total DCM in Equation (17).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

$$\begin{Bmatrix} X_B \\ Y_B \\ Z_B \end{Bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\Phi) & sin(\Phi) \\ 0 & -sin(\Phi) & cos(\Phi) \end{bmatrix}}_{1-rotation\ about\ Z^{''}=Z_B} \underbrace{\begin{bmatrix} cos(\theta) & 0 & -sin(\theta) \\ 0 & 1 & 0 \\ sin(\theta) & 0 & cos(\theta) \end{bmatrix}}_{2-rotation\ about\ Y'} \underbrace{\begin{bmatrix} cos(\psi) & sin(\psi) & 0 \\ -sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{3-rotation\ about\ Z_i} \begin{Bmatrix} X_i \\ Y_i \\ Z_i \end{Bmatrix} \tag{16}$$

$$\begin{Bmatrix} X_B \\ Y_B \\ Z_B \end{Bmatrix} = \begin{bmatrix} cos(\theta)cos(\psi) & cos(\theta)sin(\psi) & -sin(\theta) \\ sin(\Phi)sin(\theta)cos(\psi) - cos(\Phi)sin(\psi) & sin(\Phi)sin(\theta)sin(\psi) + cos(\Phi)cos(\psi) & sin(\Phi)cos(\theta) \\ cos(\Phi)sin(\theta)cos(\psi) + sin(\theta)sin(\psi) & cos(\Phi)sin(\theta)sin(\psi) - sin(\Phi)cos(\psi) & cos(\Phi)cos(\theta) \end{bmatrix} \begin{Bmatrix} X_i \\ Y_i \\ Z_i \end{Bmatrix} \tag{16}$$

A mathematical problem arising from the DCM is the inability to pull values directly from the matrix due to quadrant ambiguity presented by the law of sines and cosines. To solve the quadrant ambiguity problem portions of the DCM have to be individually selected that allow use of the atan2 function to solve for a specific angle by cancelling out other portions, as shown in Equation 18 in which the atan2 function distinguishes between the four standard reference quadrants.

$$\tan^{-1}\left(\frac{[DCM](1,2)}{[DCM](1,1)}\right) \equiv \text{atan2}\left(\frac{cos(\theta)sin(\psi)}{cos(\theta)cos(\psi)}\right) = \psi \tag{17}$$

An additional mathematical error from the law of sines and cosines is singularities in the DCM from solutions equaling 1/0. Lastly, the DCM, based on the chosen order of rotation, results in errors in the final result with only the "last" rotation having zero error due to being the only rotation about the body axis. To account for zero error in all three Euler angles, an additional rotation needs to be completed for each angle in the "last" position.

One way to remove the singularity problem presented by DCMs is to use the quaternion where the three-dimensional rotation is executed around a 4th axis; a scaled, stretched and normalized eigenvector, instead of the three standard axes. This requires induction of three imaginary axes of motion as opposed to the standard inertial frame and is shown in Figure (1c) and Equation (19) where q represents the different values of the quaternion and η is the angle that allows a single rotation [1,2]. The quaternion works so long as we enforce the quaternion normalization condition shown in Equation (19). Specifically, Equation 19 shows a direct comparison to the DCM shown in Equation (17).

Table 6. Table of proximal variable definitions.

Variable/concept	Definition	Variable/concept	Definition
DCM	Direction Cosine Matrix	q	Quaternion
B	Body Reference Frame	i	Inertial Reference Frame
u	Controller	ω	Angular Velocity
J	Rotational Tensor	ω_d	Desired ω

$$\begin{Bmatrix} X_B \\ Y_B \\ Z_B \end{Bmatrix} = [DCM] \begin{Bmatrix} X_i \\ Y_i \\ Z_i \end{Bmatrix} = \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2+q_3q_4) & 2(q_1q_3+q_2q_4) \\ 2(q_1q_2+q_3q_4) & 1-2(q_1^2+q_3^2) & 2(q_2q_3+q_1q_4) \\ 2(q_1q_3+q_2q_4) & 2(q_2q_3+q_1q_4) & 1-2(q_1^2+q_2^2) \end{bmatrix} \begin{Bmatrix} X_i \\ Y_i \\ Z_i \end{Bmatrix} \forall \underbrace{q^Tq + q_4^2 = q_1^2+q_2^2+q_3^2+q_4^2 = 1}_{Quaternion\ normalization\ condition} \tag{18}$$

2.8. Guidance of Motion in Three Degrees

The guidance of objects is known as the outer loop function and generates commands for the inner loop discussed in section 2.5 by accounting for three degrees of freedom of translational motion governed by Newton's 2nd Law expressed in Equation 15.

2.9. Decoupling Nonlinear Coupled Motion

Due to the transport theorem, ω couples the three degrees of freedom of translational and rotational motion. Additionally, equations 14 and 15 are internally coupled within the inertia matrix with ω, causing motion in all three degrees with inputs to only one. A non-linear decoupling controller is placed in a feedback control, or a desired output is placed in a feed forward control, to

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

decouple the nonlinear couple motion as displayed in Equation 20 where u is the controller and ω_d is the desired output angular velocity.

$$T = u - \underbrace{(\omega \times J\omega)}_{\substack{non-linear \\ decoupling \\ controller}} = u + \underbrace{(\omega_d \times J\omega_d)}_{\substack{Desired\ output \\ for \\ feedforward}} \tag{19}$$

2.10. Interim Summary

By using Newton's and Euler's equations and invoking Chasle's method, we are able to express six degrees of freedom for a three-dimensional movement of an object. DCMs are then used to describe the position of a body in terms of a defined axes, with quaternions used to eliminate singularities and validate the DCM. From these equations we pull out the true Euler angles.

Table 7. Table of proximal variable definitions.

Variable/concept	Definition	Variable/concept	Definition
r	Distance (km)	v	Planet Relative Velocity
θ	Longitude Angle	φ	Latitude Angle
γ	Flight Path Angle	ψ	Heading Azimuth
L	Lift	D	Drag
m	Mass	T	Thrust
ρ	Air Density	A	Surface Area
C_D	Coefficient of Drag	ρ_P	Propellant Density
A_B	Burn Area	P_C	Chamber Pressure
μ	Gravitational Parameter		

2.11. Dynamic Modeling

The dynamic equations of motion describing the vehicle in flight are defined as shown in Equation 21, where r is the distance from the center of the planet in kilometers, θ is the longitude angle, φ is the latitude angle, v is the planet relative velocity, γ is the planet relative flight path angle (or pitch), and ψ is the planet relative heading azimuth (or yaw).

$$\begin{aligned} \dot{r} &= vsin(\gamma) \\ \dot{\theta} &= \frac{vcos(\gamma)sin(\psi)}{rcos(\varphi)} \\ \dot{\varphi} &= \frac{vcos(\gamma)cos(\psi)}{r} \\ \dot{v} &= -\frac{D}{m} - \frac{\mu sin(\gamma)}{r^2} + \frac{T}{m} \\ \dot{\gamma} &= \frac{L_\gamma}{mv} - \frac{\mu cos(\gamma)}{r^2 v} + \frac{v}{r}cos(\gamma) \\ \dot{\psi} &= \frac{L_\psi}{mvcos(\gamma)} + \frac{v}{r}cos(\gamma)sin(\psi)tan(\varphi) \end{aligned} \tag{20}$$

2.11.1. Truth Model Development

The truth model, or high-fidelity model, for this application would include a changing drag force, D, with altitude as is defined by the drag equation shown in Equation 22 due to changing air density and velocity.

$$F_D = \frac{1}{2}\rho v^2 C_D A \tag{21}$$

For any motor other than a nuclear reactor, the mass of the vehicle will change during flight as fuel burns where the mass flow rate of a solid rocket motor propellant grain is defined by Equation

 doi:10.20944/preprints202312.0787.v1

23 as a function of the density of the propellant, burn area, chamber pressure and empirical constants that describe the pressure dependence and temperature effects of the propellant. A solid rocket motor is required due to operating in a vacuum for most of the flight path as the solid rocket motor's oxidizer is mixed into the propellant grain, enabling its use in space.

$$\dot{m} = \rho_P A_B a P_C^n \tag{22}$$

Additionally, the thrust of any rocket motor is not going to be constant due to ramp up and burn off periods, but the motor can be designed so that these periods are extremely short and negligible during the duration of the burn. The solar pressure disturbance acting on the vehicle while in orbit needs to be considered as well.

2.11.2. Medium Fidelity Optimization Model

Taking the high-fidelity model described by the truth model, a medium fidelity model can be derived from these equations. A way to do this is by assuming a constant set of forces acting on the vehicle during the duration of the flight. For example, instead of modeling atmospheric pressure at varying altitude, the median or average altitude could be picked and use that pressure to calculate the drag on a set vehicle size or design. As the mass flow rate will depend on the specific impulse and thrust requirements with the burn parameter (neutral, regressive, or progressive), it is difficult to accurately input this data before the propellant composition and grain design is chosen. These constants, along with the Earth's gravitational parameter, would then be placed into the dynamic equations and run to verify the project design and ensure useable trajectory outputs are given. Even with these modifications, the problem set would be true in nature so long as the scaling is consistent throughout the process.

In this model, additional constraints could be placed to prevent flight maneuvering through certain areas where there is a higher density of satellites or a final desired impact angle on the target. Both constraints would be beneficial if applied. The first so that the vehicle is not knocked off trajectory or causing catastrophic damage to another satellite during its flight path. The second so that maximum penetration is achieved, and blast area is near a true circle instead of an ellipse that extends beyond the point of impact. A secondary benefit from a vertical impact is the near impossibility of intercepting from modern defense systems due to the speed of the vehicle flight at endgame. The outputs from this model would give trajectories of the vehicle but would not take the additional step of converting those trajectory and state inputs into true control input commands.

2.11.3. Low Fidelity Model

As the medium-fidelity model is already assuming a constant thrust, drag and mass, the only way to simplify the problem further is to assume a constant gravitational force acting on the vehicle during the flight instead of as a function of altitude and force vehicle to operate in one plane at a time (i.e., zero out the flight path angle or heading azimuth). In the case of making gravity a constant, less acceleration over time is expected compared with the truth model due to losing the increased gravitational pull near end game, but the overall direction and orientation of the flight path should look similar.

If one of the angles was made constant, the model would be limited to a two-dimensional system, making the ability to turn towards the target off the north-south or east-west axis impossible. Additionally, by zeroing one of the control angles and making the other a constant input, the longitude and latitude states would be shown as constant. In these flight paths, the vehicle is expected to look like the path of a ball thrown off the side of a building, arcing down toward the earth until it impacts after its initial turn in the appropriate direction. Table 1 below shows a comparison of the variables and concepts considered for each fidelity level of the model.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

Table 8. High, medium, and low fidelity model comparison of incorporating variables/concepts.

Variable/concept	Low fidelity	Medium fidelity	High fidelity
Control actuator input	No	No	Yes
Coupled motion	No	No	Yes
Reference frame conversion	No	No	Yes
Variable mass	No	No	Yes
Variable drag	No	No	Yes
Variable gravity	No	Yes	Yes
Path constraints	No	Yes	Yes
Simultaneous state calculations	No	Yes	Yes
Constant gravity	Yes	No	No
Isolate states	Yes	No	No
Zero thrust	Yes	No	No

2.12. Trajectory Optimization Boundary Value Problem Development

2.12.1. Constraints

Due to real world considerations, an object can only take a certain amount of force on its control fins before they break. Because of this, the controls identified by the lift force in flight path angle and heading angle are limited to $1\frac{kgkm}{s^2}$ or 1000 Newtons of force. With the extremely high velocities the vehicle will encounter, the heating rate of the nose, $\dot{Q}$, must also be limited. A value of $4.9 \times 10^9 \frac{W}{m^2}$ was arbitrarily chosen based on density change rate.

2.12.2. Canonical Scaling

To validate the solving of the program in incremental steps, this milestone is run initially with constant drag instead of fluctuating as a function of altitude. Additionally, the model is run isolating the flight pitch angle and planet relative heading angle separately. To solve the model mathematically, the equations were canonically scaled with the scaling terms displayed in the unnumbered equation immediately below, where $R = 6378$km, $M = 10$, $AU = 1$ and $g = 0.00981$ km/sec^2:

$$\tilde{r} = \frac{r}{R};\ \tau = \frac{t}{TU};\ \tilde{v} = \frac{v}{V};\ \tilde{m} = \frac{m}{M};\ \tilde{D} = \frac{D}{FU};\ \tilde{T} = \frac{T}{FU};\ V = \frac{R}{TU};\ TU = \sqrt{\frac{R}{g}};\ FU = MA;\ A = \frac{R}{TU^2};\ aU = 1 \quad (23)$$

For units to cancel appropriately, θ and φ are scaled with the relation of $\left(\frac{aU}{TU}\right) * \dot{\tilde{\theta}} = \dot{\theta}$, the gravitational parameter was scaled by $\tilde{\mu} = \left(\frac{TU^2}{R^3}\right)\mu$, and the controls were scaled by $\tilde{L} = \left(\frac{TU^2}{RM}\right)L$. Full canonical scaling of each equation is found in Appendix B.

Table 9. Table of proximal variable definitions.

Variable/concept	Definition	Variable/concept	Definition
r	Distance	R	Scaling Term
τ	Time	TU	Scaling Term
v	velocity	V	Scaling Term
m	Mass	M	Scaling Term
D	Drag	FU	Scaling Term
T	Thrust	g	Gravitational Constant

2.12.3. Development of Necessary Conditions

To find the minimum time trajectory, the problem is viewed through the steps of Pontryagin's Principle by completing the HAMVET process. A full walkthrough of Pontryagin's Principle is found in Appendix B.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

2.12.3.1. Hamiltonian Construction

As there must be one costate for every state, the Hamiltonian can be written as seen in Equation 24 with path constraint controls identified as ε.

$$\bar{H}(x,\lambda,u) = \lambda_r\left(\tilde{v}sin(\gamma)\right) + \lambda_\theta\left(\frac{\tilde{v}cos(\gamma)sin(\psi)}{\tilde{r}cos(\varphi)}\right) + \lambda_\varphi\left(\frac{\tilde{v}cos(\gamma)cos(\psi)}{\tilde{r}}\right) + \lambda_v\left(-\frac{\tilde{D}}{\tilde{m}} - \frac{\tilde{\mu}sin(\gamma)}{\tilde{r}^2} + \frac{\tilde{T}}{\tilde{m}}\right)\cdots$$
$$\cdots + \lambda_\gamma\left(\frac{\tilde{L}_\gamma}{\tilde{m}\tilde{v}} - \frac{\tilde{\mu}cos(\gamma)}{\tilde{r}^2\tilde{v}} + \frac{\tilde{v}}{\tilde{r}}cos(\gamma)\right) + \lambda_\psi\left(\frac{\tilde{L}_\gamma}{\tilde{m}\tilde{v}cos(\gamma)} + \frac{\tilde{v}}{\tilde{r}}cos(\gamma)sin(\psi)cos(\varphi)\right) + \varepsilon_1\left(\tilde{L}_\gamma\right) + \varepsilon_2\left(\tilde{L}_\psi\right) + \varepsilon_3\left(\tilde{k}_Q\sqrt{\tilde{\rho}}\tilde{v}^3\right) \quad (24)$$

2.12.3.2 Adjoint Equation Development

The rate of change of each costate is defined as the negative of the partial derivative of the Hamiltonian with respect to the corresponding state. These Equations are shown in Equation 25.

$$-\dot{\lambda}_r = \left(\frac{\partial H}{\partial r}\right) = -\lambda_\theta\frac{\tilde{v}cos\gamma sin\psi}{\tilde{r}^2cos\varphi} - \lambda_\varphi\frac{\tilde{v}cos\gamma cos\psi}{\tilde{r}^2} + \lambda_v\frac{2\tilde{u}sin\gamma}{\tilde{r}^3} - \lambda_\gamma\left(\frac{\tilde{v}cos\gamma}{\tilde{r}^2} - \frac{2\tilde{\mu}cos\gamma}{\tilde{v}\tilde{r}^3}\right) - \lambda_\psi\frac{\tilde{v}}{\tilde{r}^2}cos\gamma sin\psi tan\varphi$$
$$-\dot{\lambda}_\theta = \left(\frac{\partial H}{\partial \theta}\right) = 0$$
$$-\dot{\lambda}_\varphi = \left(\frac{\partial H}{\partial \varphi}\right) = \lambda_\theta\frac{\tilde{v}cos\gamma sin\psi sin\varphi}{\tilde{r}^2cos\varphi} + \lambda_\psi\frac{\tilde{v}cos\gamma sin\psi(tan^2\varphi + 1)}{\tilde{r}}$$
$$-\dot{\lambda}_v = \left(\frac{\partial H}{\partial v}\right) = \lambda_r sin\gamma + \lambda_\theta\frac{cos\gamma sin\psi}{\tilde{r}cos\varphi} + \lambda_\varphi\frac{cos\gamma cos\psi}{\tilde{r}} + \lambda_\gamma\left(\frac{cos\gamma}{\tilde{r}} - \frac{\tilde{L}_\gamma}{\tilde{m}\tilde{v}^2} + \frac{\tilde{\mu}cos\gamma}{\tilde{v}^2\tilde{r}^2}\right) - \lambda_\psi\left(\frac{\tilde{L}_\psi}{\tilde{m}\tilde{v}^2cos\gamma} - \frac{cos\gamma sin\psi tan\varphi}{\tilde{r}}\right) + 3\varepsilon_3\tilde{k}_Q \quad (25)$$
$$-\dot{\lambda}_\gamma = \left(\frac{\partial H}{\partial \gamma}\right) = \lambda_r\tilde{v}cos\gamma - \lambda_\theta\frac{\tilde{v}sin\gamma sin\psi}{\tilde{r}cos\varphi} - \lambda_\varphi\frac{\tilde{v}sin\gamma cos\psi}{\tilde{r}} - \lambda_\gamma\left(\frac{vsin\gamma}{\tilde{r}} - \frac{\tilde{\mu}sin\gamma}{\tilde{v}\tilde{r}^2}\right) - \lambda_v\frac{\tilde{u}cos\gamma}{\tilde{r}^2} - \lambda_\psi\left(\frac{\tilde{L}_\psi sin\gamma}{\tilde{m}\tilde{v}cos^2\gamma} - \frac{\tilde{v}sin\gamma sin\psi tan}{\tilde{r}}\right.$$
$$-\dot{\lambda}_\psi = \left(\frac{\partial H}{\partial \psi}\right) = \lambda_\theta\frac{\tilde{v}cos\gamma cos\psi}{\tilde{r}cos\varphi} - \lambda_\varphi\frac{\tilde{v}cos\gamma sin\psi}{\tilde{r}} + \lambda_\psi\frac{\tilde{v}cos\gamma cos\psi tan\varphi}{\tilde{r}}$$

2.12.3.1 Hamiltonian Minimization

Minimize the Hamiltonian with respect to each control to find possible verification factors available for the solution. These two equations, or the stationarity conditions, are shown in Equation 27 and 28 with the complementarity conditions listed in Equation 29.

Table 10. Table of proximal variable definitions.

Variable/concept	Definition	Variable/concept	Definition
H	Hamiltonian Function	λ	Costate Variable
r	Distance	v	Velocity
γ	Flight Path Angle	ψ	Heading Azimuth
φ	Latitude Angle	θ	Longitude Angle
L	Lift Control Value	m	Mass
D	Drag	T	Thrust
μ	Gravitational Parameter	ε	Path Constraint Control

$$\frac{\partial H}{\partial u} = \frac{\partial H}{\partial \tilde{L}_\gamma} = 0 = \frac{\lambda_\gamma}{\tilde{m}\tilde{v}} + \varepsilon_1 \quad (26)$$

$$\frac{\partial H}{\partial u} = \frac{\partial H}{\partial \tilde{L}_\psi} = 0 = \frac{\lambda_\psi}{\tilde{m}\tilde{v}cos\gamma} + \varepsilon_2 \quad (27)$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

$$\varepsilon_1 \begin{cases} \leq 0 \ if & \tilde{L}_\gamma = -10 \\ = 0 \ if & -10 \leq \tilde{L}_\gamma \leq 10 \\ \geq 0 \ if & \tilde{L}_\gamma = 10 \end{cases}$$
$$\varepsilon_2 \begin{cases} \leq 0 \ if & \tilde{L}_\psi = -10 \\ = 0 \ if & -10 \leq \tilde{L}_\psi \leq 10 \\ \geq 0 \ if & \tilde{L}_\psi = 10 \end{cases} \quad (28)$$
$$\varepsilon_3 \begin{cases} \leq 0 \ if & \dot{Q} = 0 \\ = 0 \ if & 0 \leq \dot{Q} \leq 4.9e^9 \\ \geq 0 \ if & \dot{Q} = 4.9e^9 \end{cases}$$

Table 11. Table of proximal variable definitions.

Variable/concept	Definition	Variable/concept	Definition
H	Hamiltonian Function	L	Lift Control Unit
u	Control Variable	m	Mass
v	Velocity	λ	Costate Variable
γ	Flight Path Angle	ψ	Heading Azimuth
ε	Path Constraint Variable	$\dot{Q}$	Heating Rate
E	Endpoint Value Function	t	Time
r	Distance	θ	Longitude Angle
φ	Latitude Angle		

2.12.3.1 Endpoint Value Condition

The value of the Hamiltonian at the final time can be found by taking the partial derivative of the endpoint cost with respect to time and is shown in Equation 29.

$$\vec{E} = t_f + \nu_1(r_f - r^f) + \nu_2(\theta_f - \theta^f) + \nu_3(\varphi_f - \varphi^f); \ -\frac{\partial E}{\partial t_f} = 1 = \bar{H}(t_f) \quad (29)$$

2.12.3.1 Hamiltonian Evolution Condition

The Hamiltonian Evolution Condition shows how the Hamiltonian behaves with respect to time and is found by taking the partial derivative of the Hamiltonian with respect to time as shown in Equation 30.

$$\frac{\partial H}{\partial t} = \frac{\partial \mathcal{H}}{\partial t} = 0 \quad (30)$$

2.12.3.1 Transversality Condition

Transversality is used to determine more points for verification and validation and is done by taking the partial derivative of the endpoint cost function with respect to each state individually as shown in Equation 31.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

$$\vec{E} = t_f + \nu_1(r_f - r^f) + \nu_2(\theta_f - \theta^f) + \nu_3(\varphi_f - \varphi^f)$$
$$\lambda_r(t_f) = \frac{\partial \vec{E}}{\partial r_f} = \nu_1$$
$$\lambda_\theta(t_f) = \frac{\partial \vec{E}}{\partial \theta_f} = \nu_2$$
$$\lambda_\varphi(t_f) = \frac{\partial \vec{E}}{\partial \varphi_f} = \nu_3 \quad (31)$$
$$\lambda_v(t_f) = \frac{\partial \vec{E}}{\partial v} = 0$$
$$\lambda_\gamma(t_f) = \frac{\partial \vec{E}}{\partial \gamma_f} = 0$$
$$\lambda_\psi(t_f) = \frac{\partial \vec{E}}{\partial \psi_f} = 0$$

Table 12. Table of proximal variable definitions.

Variable/concept	Definition	Variable/concept	Definition
E	Endpoint Value Function	t	Time
r	Distance	θ	Longitude Angle
φ	Latitude Angle	λ	Costate Variable
t	Time	v	Velocity
γ	Flight Path Angle	ψ	Heading Azimuth

2.12.4. Development of Necessary Conditions

By completing the HAMVET process, it is shown that the theta costate is constant throughout the duration of flight, and the Hamiltonian is constant with an endpoint value of -1; or in other words, constant at -1. Velocity, flight pitch angle and planet relative heading co-states are all equal to zero at final time. With the canonical scaling used, the dynamic equations of motion can be written the same way with new boundary conditions as defined in equation (32).

$$(r_0, \theta_0, \varphi_0, v_0, \gamma_0, \psi_0) = (42164,0,0,0,0,0)$$
$$r_f = 6378$$
$$(\theta_f, \varphi_f) = (36.6002, -121.8947)\ (degrees)$$
$$\tilde{\lambda}_v(t_f) = 0 \quad (32)$$
$$\tilde{\lambda}_\gamma(t_f) = 0$$
$$\tilde{\lambda}_\psi(t_f) = 0$$
$$\bar{H}(t_f) = -1 = constant$$

3. Results

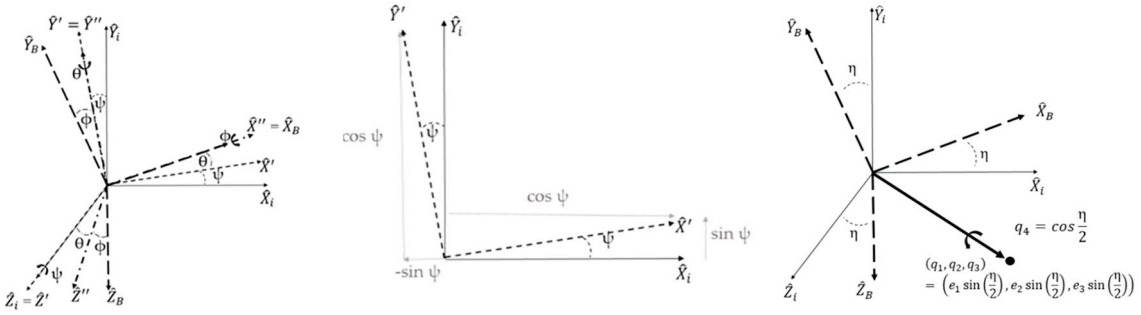
The trajectory problem was solved in MATLAB using the DIDO program. The complete DIDO program files can be found in Appendix C.

3.1. Trajectory Optimization Solution

3.1.1. Low Fidelity Dynamic Verification

The low fidelity numerical simulation was run twice, once for each isolated control state of γ and ψ and produced the altitude and state magnitudes shown in Figures 2 and 3.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1



(**a**) (**b**) (**c**)

Figure 3. (**a**) A depiction of three-dimensional comparison of the body frame to the inertial frame through prime axes by the Euler angles pitch (θ), roll (ϕ) and yaw (ψ). (**b**) depicts determining the sign value of each sine or cosine for direction cosine matrix (DCM) of rotating about the Zi axis. 1c depicts a rotation using a quaternion to directly compare the inertial frame to the body frame by mathematically rotating around a 4th axis, defined in relation to the eigenvector, by the angle η.

Table 13. Table of proximal variable definitions.

Variable/concept	Definition	Variable/concept	Definition
φ	Latitude Angle	v	Velocity
γ	Flight Path Angle	ψ	Heading Azimuth
θ	Longitude Angle	r	Distance

With the flight path isolated by zeroing out azimuth, the altitude and states behaved exactly as expected. Altitude dropped at a constant parabolic rate as expected with the constant forces applied and gravity and thrust overriding the drag. Velocity increased at a constant rate due to these forces while latitude and longitude remained constant due to no turn inputs and isolated controls. With $\dot{\psi}$ set to equal 0, the initial input angle of 0 remains constant with a constant gravitational force, γ slowly decreases in angle magnitude as it is a function of the radius and velocity but on much smaller scale. The end portion of the plot where pitch and azimuth look to start spiking is where if the simulation was run for a longer time, the vehicle bounces off the earth and proceeds back up.

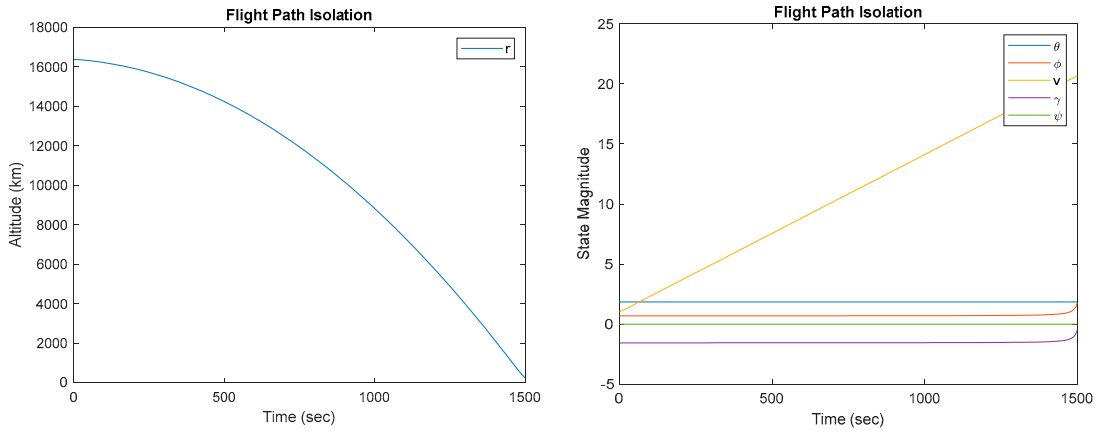
When the azimuth angle is isolated and pitch is set to an initial angle with no change, the altitude behaves as expected. Velocity increases at a constant rate as expected with the constant forces acting on the vehicle while latitude and longitude remain constant as the cosine of -90 degrees is near zero, causing little to no change in the overall state. γ remains constant as we have set the rate change to zero. Initial simulation runs resulted in azimuth angles on the order of 1014 due to the angle not being limited so radians continued to climb to numbers that are not within the range of angles. To fix this problem, the "wrapToPi" function was placed in front of the $\dot{\psi}$ equation to convert the output of the angle to a value between -π and π (-180 to 180 degrees).

Additionally, as expected, r, v, latitude, and longitude remained constant from one simulation to the other. Based on these observations, the base dynamics of the model have been validated and the simulation can proceed to a medium fidelity, or optimization model.

3.1.2. Medium fidelity model validation

The dynamic model was simulated another time with the vehicle at the same longitude as the target and making the rate of change of the flight pitching angle to zero in the dynamics equations while accounting for a changing gravitational force due to altitude. The scaled outputs from this simulation are shown in Figure 4.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1



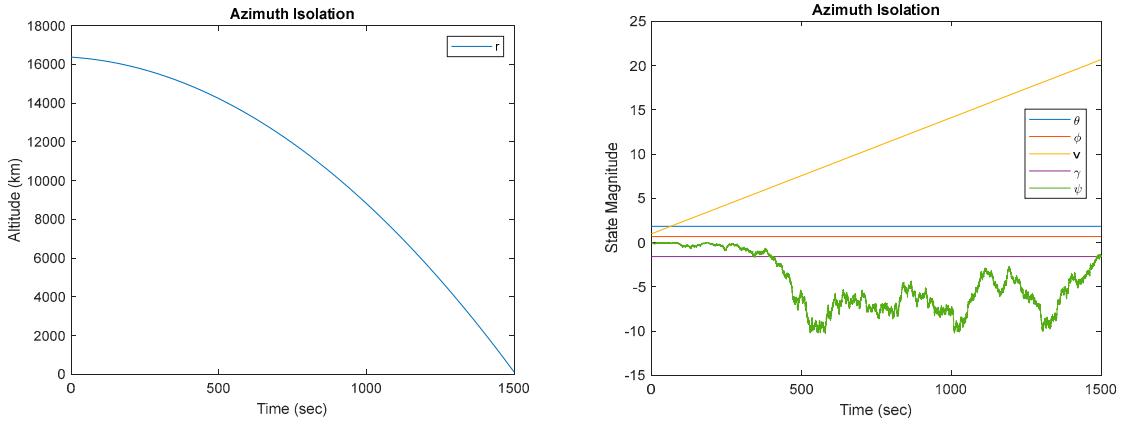
(**a**) (**b**)

Figure 4. Subfigure (**a**) is altitude flight descent and (**b**) shows the state magnitudes with isolating the flight path angle and constant control input.

Table 14. Table of proximal variable definitions.

Variable/concept	Definition	Variable/concept	Definition
φ	Latitude Angle	v	Velocity
γ	Flight Path Angle	ψ	Heading Azimuth
θ	Longitude Angle	r	Distance
L	Lift Control Variable		

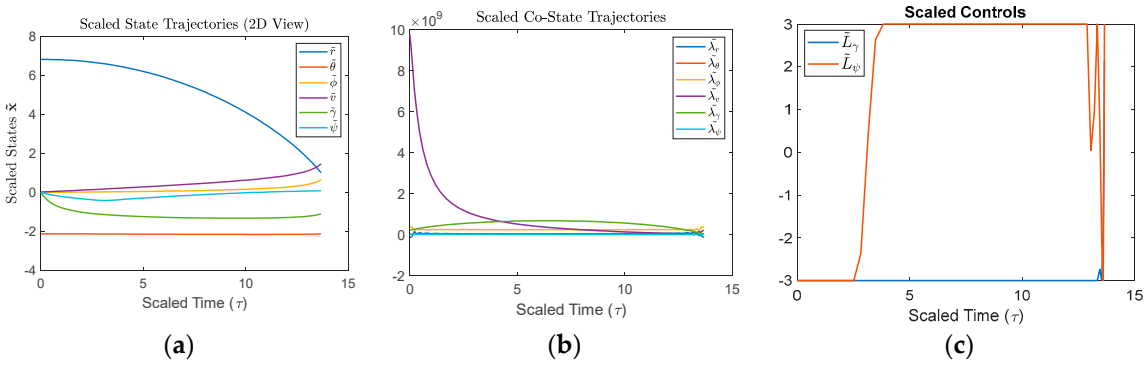
As expected, the altitude falls in a parabolic fashion towards the earth with a corresponding increase in velocity throughout the simulation. To verify that we have isolated the azimuth heading control, the controls page shows a nearly constant flight path angle control. For analyzing purposes, the propagated states and Hamiltonian are shown in Figure 5 and a three-dimensional depiction of the flight path is shown in Figure 6.



(**a**) (**b**)

Figure 5. Subfigure (**a**) is altitude flight descent and (**b**) shows the state magnitudes with isolating the azimuth angle and constant control input.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1



(**a**) (**b**) (**c**)

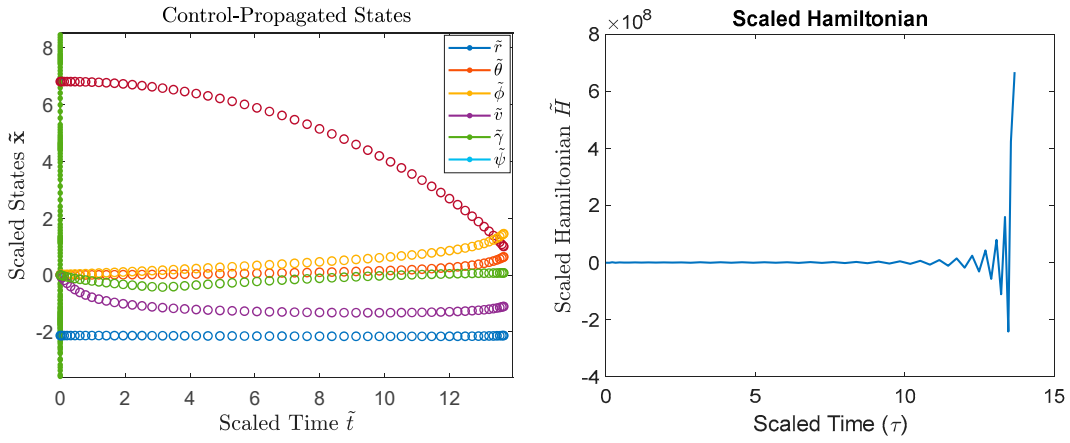
Figure 6. Subfigure (**a**) is the simulation output of the scaled states, and (**b**) is the scaled co-states, and 4c is the scaled controls.

Table 15. Table of proximal variable definitions.

Variable/concept	Definition	Variable/concept	Definition
φ	Latitude Angle	v	Velocity
γ	Flight Path Angle	ψ	Heading Azimuth
θ	Longitude Angle	r	Distance

As seen in Figure 5a, the state trajectories follow that of the propagated states to satisfy the feasibility check. The Hamiltonian remains constant for the first 90% of the simulation before it destabilizes and diverges at the end of the simulation leading to some doubt as to the optimality of the simulation. Conversely, the theta co-state does stay constant for the duration of the simulation as expected. This leads to the belief that the Hamiltonian is spiking due to a lack of determinacy on the time value in the problem which could possibly be resolved by adding a running cost in the final problem. Additionally, all co-states ended as expected with velocity, flight path angle and heading azimuth all ending near zero.

The next test had both controls activated to test the overall validity of the dynamic model while maintaining constant forces throughout the simulation. No constraints were placed on the model other than limiting the controls to approximately 1000N as an arbitrary limit on the forces allowed on the control fins. Due to singularity issues that arose when -90 degrees was chosen as the initial flight path angle, an initial angle of -89 was used. Figure 7 shows the scaled outputs of this simulation.



(**a**) (**b**)

Figure 7. Subfigure (**a**) is altitude flight descent and (**b**) shows the state magnitudes with isolating the azimuth angle and constant control input.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

Table 16. Table of proximal variable definitions.

Variable/concept	Definition	Variable/concept	Definition
φ	Latitude Angle	v	Velocity
γ	Flight Path Angle	ψ	Heading Azimuth
θ	Longitude Angle	r	Distance
L	Lift Control Variable	λ	Costate Variable

The trajectories act as expected with the altitude lowering from its initial altitude to final in a parabolic fashion with a corresponding increase in velocity. For analysis, the simulation was again propagated through the ode45 program. Figure 8 shows the control propagated states and Hamiltonian.

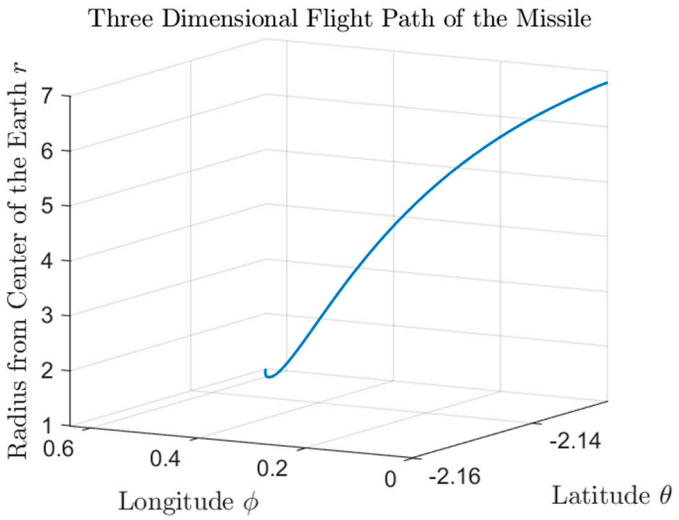
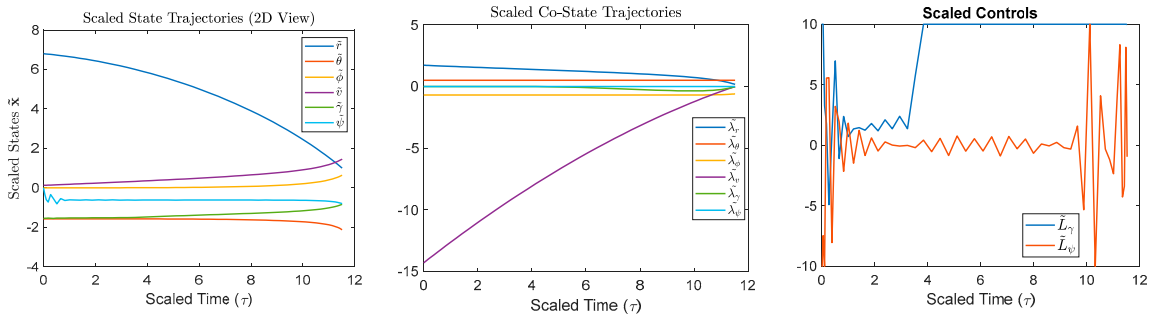


Figure 8. Three-Dimensional plot of the vehicle flight path.

As shown in the propagated states, the trajectories of the states perfectly match those of the DIDO solution. Additionally, the Hamiltonian is nearly constant at 1 for the duration of the simulation with a maximum fluctuation of two ten-thousandths from a steady -1. For the purposes of this simulation, this will be labeled as constant at -1. Lastly, the co-states converge to zero as expected from the HAMVET process. Between these points, the feasibility of the model is verified.

This flight path appears closer to what is expected with a large downward trajectory while building speed prior to turning in a smooth manner towards the target instead of arcing around the target which was shown in the single control plots. This smooth flight profile is shown in Figure 9.



(**a**) (**b**) (**c**)

Figure 9. Subfigure (**a**) is the simulation output of the scaled states, (**b**) is the scaled co-states, and (**c**) is the scaled controls.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

These tests verify the general build of the model using propagated states while isolating each individual control and using the DIDO program to solve for an extremal solution and repeating the process with both controls activated without path constraints or varying thrust or drag.

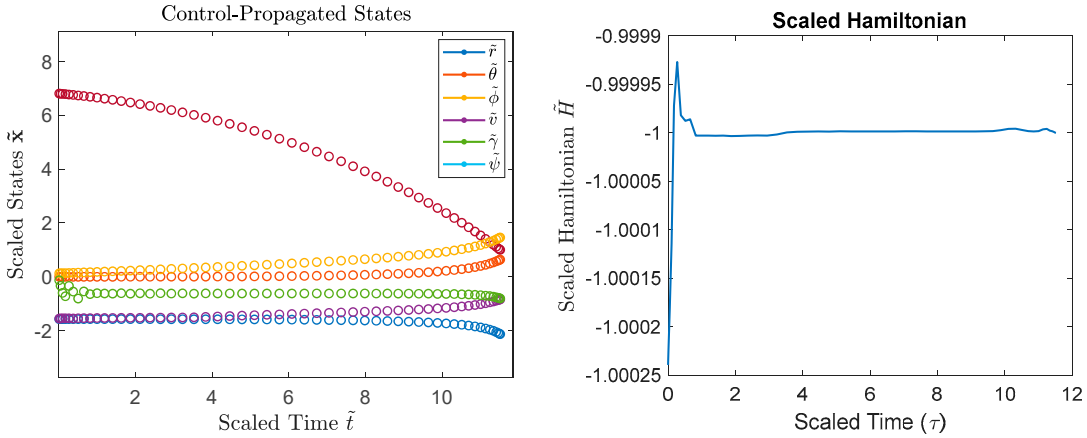
3.1.3. Optimization Model

The model was updated to account for the constrain the heating rate of the vehicle during flight through varying density based off of altitude in accordance with Equation 33 where k_Q is the heating constant, ρ is the air density at altitude and v is the velocity of the vehicle and air density was calculated using Equation 34 where ρ_o is the air density at sea level, h is the altitude, r_p is the radius of the earth, and H_s is a height scaling factor equal to 7500.

$$\dot{Q} = k_Q \sqrt{\rho} v^3 < 4.9e^9 \frac{W}{m^2} \tag{33}$$

$$\rho = \rho_0 \exp\left(-\frac{h - r_p}{H_S}\right) \tag{34}$$

The remaining initial conditions remained constant from the previous test. Figure 10 shows the scaled outputs of this simulation.



(a) (b)

Figure 10. Subfigure (**a**) is the propagated states of the simulation and (**b**) is the Hamiltonian vs time.

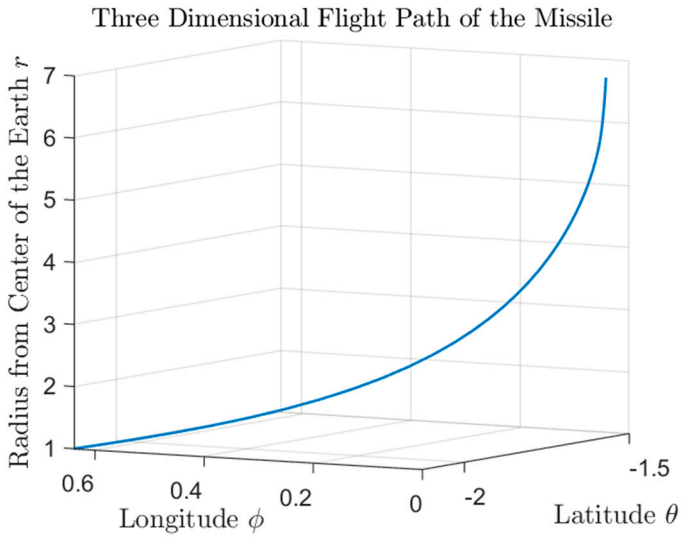
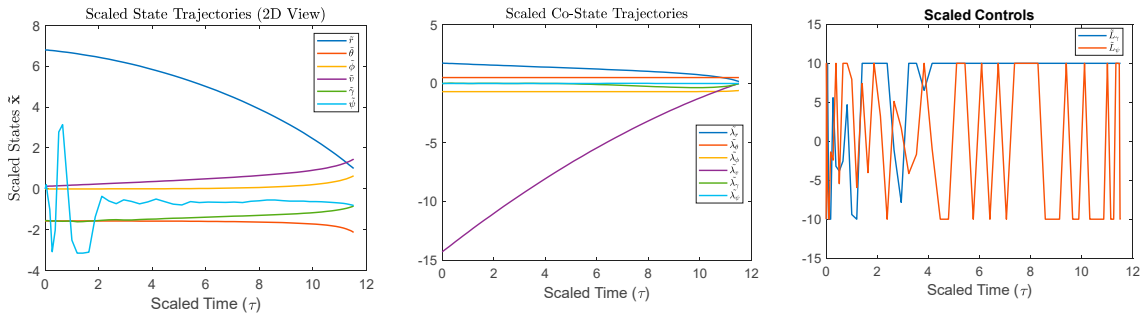


Figure 11. Three-Dimensional plot of the Vehicle Flight Path.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1



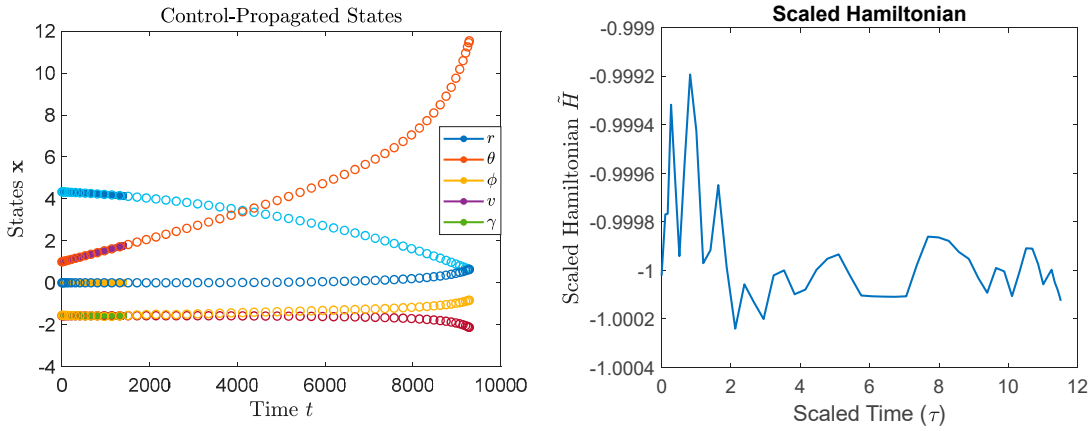
(**a**) (**b**) (**c**)

Figure 12. Subfigure (**a**) is the simulation output of the scaled states, (**b**) is the scaled co-states and (**c**) is the scaled controls.

Table 17. Table of proximal variable definitions.

Variable/concept	Definition	Variable/concept	Definition
φ	Latitude Angle	v	Velocity
γ	Flight Path Angle	ψ	Heading Azimuth
θ	Longitude Angle	r	Distance
L	Lift Control Variable	λ	Costate Variable

As seen in **Error! Reference source not found.**, the trajectories act as expected with the altitude lowering from its initial altitude to final in a parabolic fashion with a corresponding increase in velocity. Of note, the heading azimuth state fluctuates much more drastically than seen in any other tests to this point. For verification and validation, Figure 11 shows the control propagated states and Hamiltonian.



(**a**) (**b**)

Figure 13. Subfigure (**a**) is the propagated states of the simulation with r divided by 1e4 to balance the chart and (**b**) is the Hamiltonian vs time.

The Hamiltonian is nearly constant at 1 for the duration of the simulation with a maximum fluctuation of 8 ten-thousandths from a steady -1. For the purposes of this simulation, this will be labeled as constant at -1. The feasibility check follows the dynamic path for all except for φ with iteration unable to maintain due to small step sizes, most likely due to singularities in the denominator of the equations due to the denominator of the azimuth state dynamic containing the cosine of the flight path angle. At any point the flight path angle crosses -90 degrees, or straight down, this value turns to zero and is mathematically impossible to calculate. An additional cause for the failure to continue solving is the drastic changes in the azimuth state as the ode45 solver is unable to follow the sudden change in direction while propagating states forward. Attempts at solving this

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

included lessening the initial downward angle of the vehicle, but this is undesirable as it presents a bias towards a certain hemisphere instead of allowing global targeting. Additionally, when this initial angle was placed in for troubleshooting, the feasibility check would pass as the singularities were no longer present, but the overall solution was deemed as a suboptimal solution instead of the extremal solution that is achieved with a near 90-degree initial angle. The stationarity condition checks, and complementarity condition checks are shown in Figure 12.

Table 18. Table of proximal variable definitions.

Variable/concept	Definition	Variable/concept	Definition
ε	Path Constraint Variable	L	Lift Control Variable

Based on the stationarity condition in equation (26), the path constraint control, ε_1 should be equal and opposite of the function identified as Stat1 in Figure 12a. Similarly, the path constraint control ε_2 should be equal and opposite of the curve labeled as Stat2 in accordance with equation (27). While not perfectly matching in the slope of the curve, the magnitudes for the first path constraint control match with ε_1 maxing out at 8.57e-4 and Stat1 with maximum negative of -8.57e-4. The second stationarity check is a little off in that the constraint control remains constant at zero for the duration of the simulation, but Stat2 fluctuates to a maximum of 2.6e-4. While this number is small, a closer relationship in slope in the first stationarity check and magnitude in the second would be preferred.

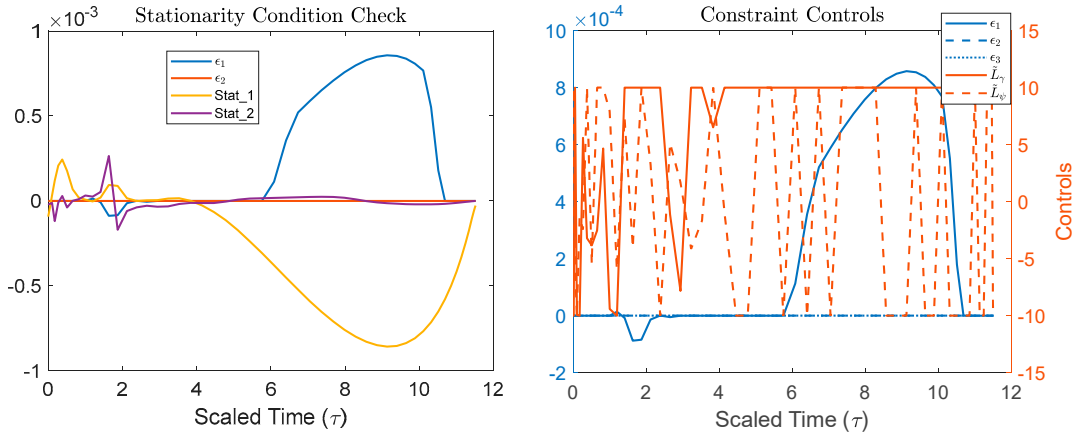
According to the complementarity conditions in Equation 28 and that the scaled controls max out at 10 and -10, the constraint controls should remain constant at zero for the duration of the simulation with the possibility of the control fluctuating either positively or negatively when the control is touching the maximum deflection of 10. As shown in Figure 12b, the only time that ε_1 fluctuates is when $\tilde{L}_\gamma$ is holding at 10.0, allowing a value greater than or equal to zero during this time. From the scaled time of 5 to 11, this is shown in the maximum fluctuation of the control to a value of 8.57e-4. Of concern is that while $\tilde{L}_\gamma$ is at a positive 10 from scaled time 1 to 2.5, there is a negative ε_1 which should not occur. While the values remain small in the range of 1e-4, according to the complementarity conditions there should not be a negative value at all during this time, but instead should be greater than or equal to zero.

The second complementarity condition follows the same rules as the first regarding when a positive and negative value are allowed. With ε_2, however, the value remains constant at zero for the duration of the simulation. This path constraint control passes the check as even when the controls are maxed, the constraint control is allowed to maintain zero. Due to being unable to constrain the heating rate of the vehicle due to constant velocity increases with a maximum velocity at the final time, the heating rate constraint is never activated and ε_3 remains constant at zero.

Lastly, the co-states converge to zero as expected from the HAMVET process. The $\tilde{\lambda}_v$ costate ends at 1.2928e-4, $\tilde{\lambda}_\gamma$ ends at -0.0198, and $\tilde{\lambda}_\psi$ ends at -1.4421e-4. With $\tilde{\lambda}_v$ having a maximum magnitude of -14.2817, the end point is less than one ten-thousandth of a percent of the maximum magnitude. Doing the same comparison with $\tilde{\lambda}_\gamma$ and $\tilde{\lambda}_\psi$, the end points result in 5.5% and 3.3% of the maximum magnitude respectively. Between these points, the feasibility of the model is verified. The states, co-states and controls were descaled and are shown in Figure 13, but due to the large magnitude differences and scale of the plot all the states appear to be at zero.

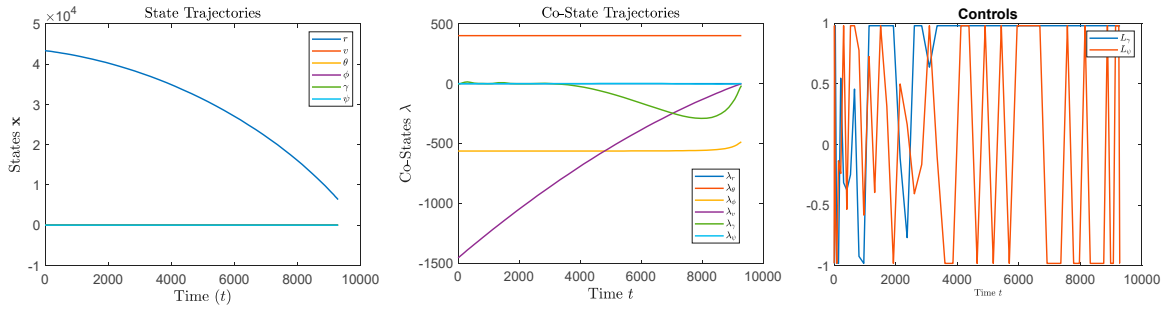
To observe the behavior of each state, they are plotted individually with the state and co-state simultaneously as shown in Figures 14 and 15.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1



(**a**) (**b**)

Figure 14. Subfigure (**a**) the stationarity condition check and (**b**) is the complementarity condition check.



(**a**) (**b**) (**c**)

Figure 15. Subfigure (**a**) is the simulation output of the scaled states, (**b**) is the scaled co-states and (**c**) is the scaled controls.

Table 19. Table of proximal variable definitions.

Variable/concept	Definition	Variable/concept	Definition
φ	Latitude Angle	v	Velocity
γ	Flight Path Angle	ψ	Heading Azimuth
θ	Longitude Angle	r	Distance
λ	Costate Variable	L	Lift Control Variable

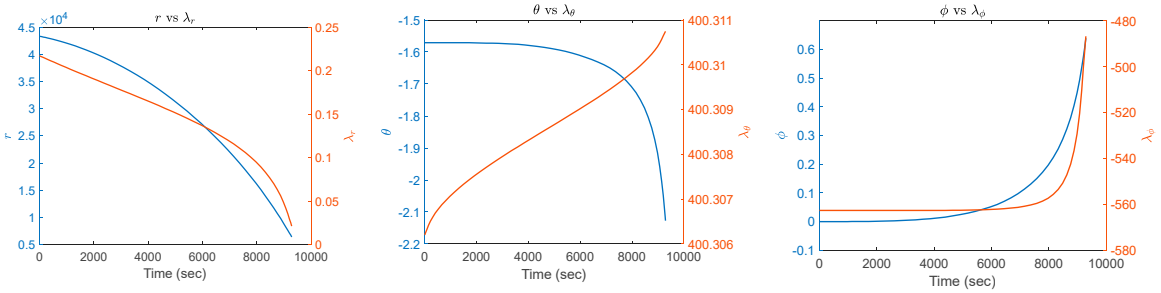
Figures 14 and 15 are beneficial in observing the direct behavior of each individual state. During the simulations and testing, it was observed that the heating rate could not be constrained to any value other than the maximum value expected at sea level with the maximum velocity. By looking at Figure 14a, this becomes apparent as the velocity is constantly increasing due to the constant thrust on the vehicle, and therefore the maximum heating rate will occur at the exact point that it was calculated and cannot be lowered. To constrain the heating rate, the velocity will need to be lowered by either a variable thrust motor or utilizing a non-extremal solution. Figure 14 a and b indicate that the vehicle falls almost vertically from the release point for the first third of flight before making a smooth turn towards the target. This behavior is expected due to the limited control authority from control fins in orbit.

Comparing the behavior of the flight path angle and heading azimuth illustrates a difference in the method used for adjustments between the two. Figure 15b shows that the flight path angle gradually decreases towards zero as the vehicle flies and impacts with a flight path angle of -47.86 degrees, or nearly 48 degrees nose down below the planet relative horizon. In contrast to this method, the heading azimuth makes large and rapid adjustments at the beginning of the trajectory and then

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

smooths out at the end, impacting with a heading azimuth of -46.93 degrees, or 47 degrees to the left of planet relative 0 (north). Due to the search window of -180 degrees to 180 degrees, or a full 360-degree circle, the heading azimuth bounces from -180 to 180 degrees and continues but should be viewed as continuing through planet relative south heading as the vehicle maneuvers. This portion of the flight is the same portion that causes the validation check to fail with the sharp changes in direction of the state vector.

Even with these concerns, an extremal solution was found with the vehicle directly hitting the intended target. The optimal flight time of the trajectory for this specific target case is two hours, thirty-four minutes, and forty-six seconds. The vehicle impacts with a final velocity of 11.54 km/sec, equaling a kinetic energy of 0.333 terajoules. This kinetic energy is roughly equivalent to 300,000 lbs. of TNT. For perspective, this is 150 times the strength of a standard 2000lb bomb carried by fighter aircraft without any explosives in the vehicle itself. The three-dimensional trajectory of the vehicle flight path is shown in Figure 16.



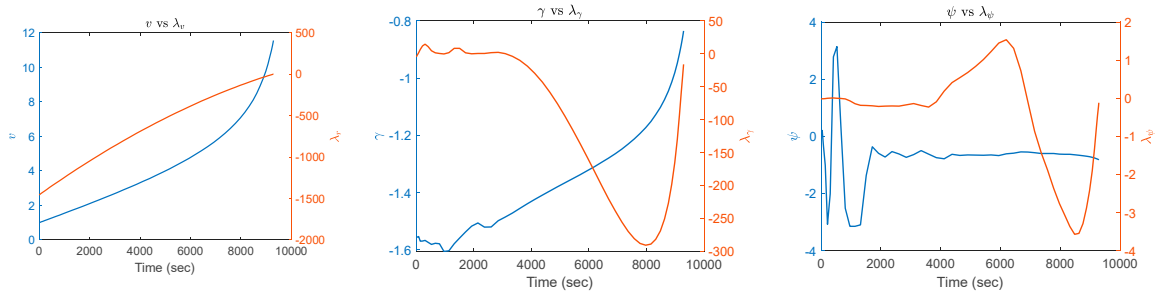
(**a**) (**b**) (**c**)

Figure 16. Subfigure (**a**) is the r state and co-vector, (**b**) is the θ state and co-vector, and (**c**) is the φ state and co-vector.

3.2. Coordinate Transformation Results

3.2.1. Simulation Configuration and Accuracy

The simulation was run using multiple Matlab Simulink solvers in order to determine the accuracy of the simulation as demonstrated in Figure 17 and Table X.



(a) **(b)** **(c)**

Figure 17. Subfigure (**a**) is the v state and co-vector, (**b**) is the γ state and co-vector, and (**c**) is the ψ state and co-vector.

Table 20. High, medium, and low fidelity model comparison of incorporating variables/concepts.

Solver	Solver Type	Frequency (Hz)	Max Error
ode4	Runge-Kutta	34.6	$1.6\text{x}10^{-10}$
ode5	Dormand-Prince	34.6	$1.1\text{x}10^{-12}$
ode5	Dormand-Prince	40	$1.7\text{x}10^{-9}$
ode4	Dormand-Prince	40	$3.1\text{x}10^{-13}$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

Auto	Auto Select	FAILED	FAILED
ode4	Runge-Kutta	90	$2.1x10^{-15}$
ode4	Runge-Kutta	110	$1.4x10^{-15}$
ode4	Runge-Kutta	130	$1.4x10^{-11}$
ode4	Runge-Kutta	100	$0.9\ x10^{-15}$

[1] Runge-Kutta (ode4) with 100Hz was selected

Using the quaternion normalization condition, the simulation was verified to be accurate to near machine precision in the case of input to a single Euler angle while accounting for six degrees of freedom. For commanded Pitch, Roll and Yaw simultaneously, best precision was found to be $4.8x10^{-11}$ using the ode4 solver with a frequency of 117Hz.

3.2.2. Dynamics of Motion in Six Degrees

The only change between tests was the order of the mathematical rotation conducted in the DCM in order to validate the accuracy and computation required for each and allow a direct comparison.

3.2.2.1. Kinetics of Rotational Motion

The simulation ran in accordance with Euler's equations of motion accounting for the changing angular velocities and accelerations found with a falling and rotating object, while accounting for the rotation of the earth under the falling missile.

3.2.2.2. Kinetics of Translational Motion

With mass remaining constant, the simulation runs in accordance with Newton's second law. The simulation starts in a geosynchronous orbit with less velocity than required to maintain orbit. This initial condition, coupled with the commanded inputs, immediately initiates a dive towards the earth with a constant thrust along the XBody axis.

3.2.2.3. Kinetics of Disturbance Forces and Torques

The simulation accounts for magnetic disturbances and the gravity gradient as the missile falls to the earth but does not account for drag disturbances. While drag would slow the missile, because the forces used remain constant, the DCM comparison remains valid.

3.2.2.4. Kinetics of Motion

As discussed in Section III, the only "true" Euler angle output from a DCM is the last mathematical rotation. For the purposes of analysis, the true Roll angle, Φ, is pulled from executing a 321 rotation; the true pitch angle, Θ, is from a 132 rotation; and the true yaw angle, ψ, is from a 213 rotation. The true Euler angles from these rotations is given as $\Phi = -0.5825°$, $\Theta = 10.2404°$, and $\psi = 0.0863°$. With the defined maneuver of $\Phi = 10°$, $\Theta = -30°$, and $\psi = 30°$, the desired angles, rates and acceleration trajectories remain constant.

Changing the mathematical rotation order changes the flightpath and distance traveled of the missile as displayed in Figure 18. Derived Euler angles with associated error and computational burden expressed in run time for each rotation are found in Table 3.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

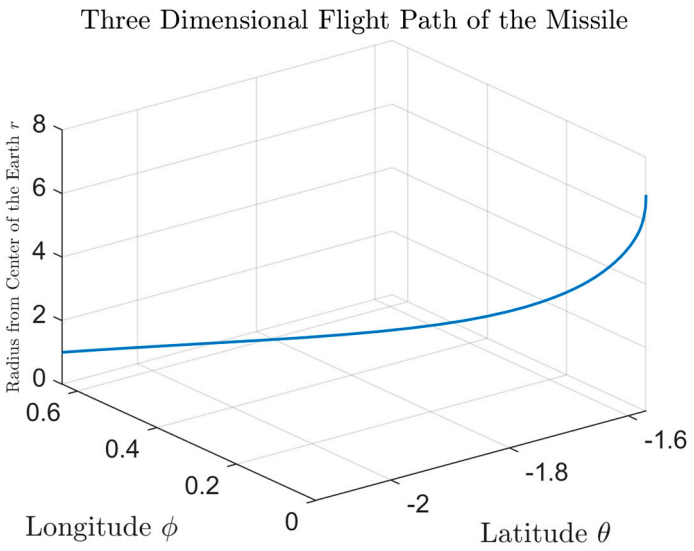


Figure 18. Three-Dimensional plot of the Vehicle Flight Path.

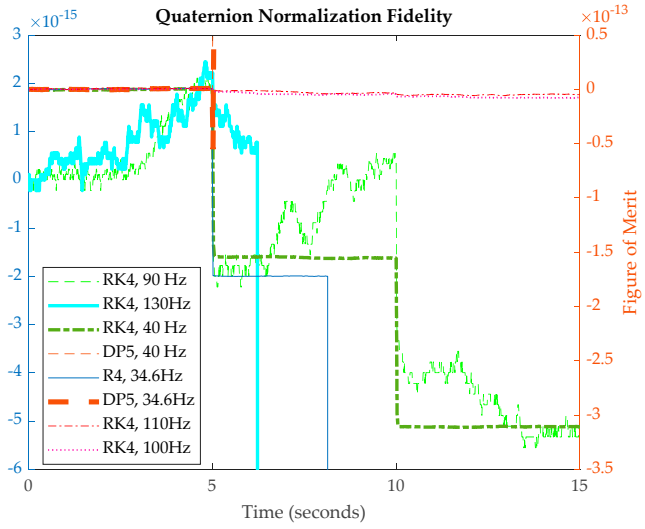
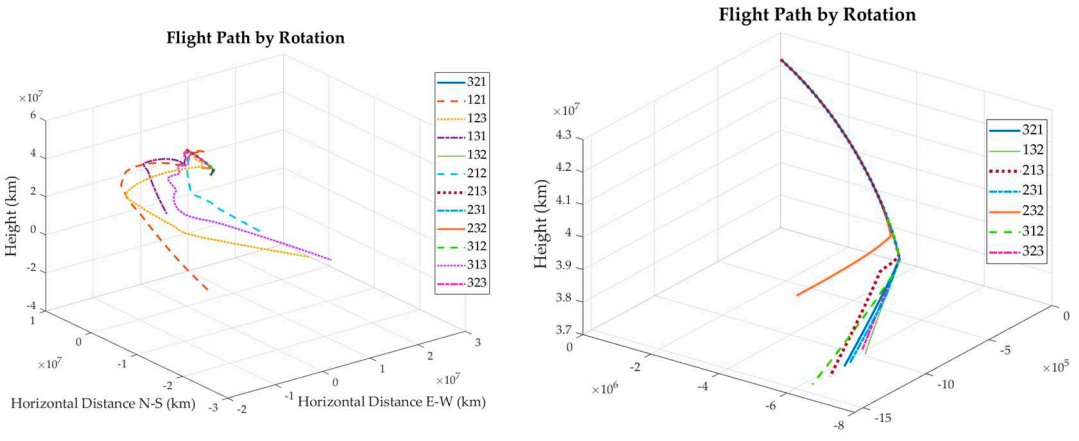


Figure 19. Comparison of Quaternion Normalizations (on the ordinate) for different step sizes and solver schemes.



(**a**) (**b**)

Figure 20. Three-dimensional flight path of the missile based on the order of mathematical rotation chosen in the DCM and a zoomed in portion of the rotations traveling closest to each other.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

Table 21. Derived Euler angles, associated error and simulation run time.

Rotation	Angle	Derived Value (Deg)	Error (Deg)	Average Error (Deg)	Run Time (Sec)	Rotation	Angle	Derived Value (Deg)	Error (Deg)	Average Error (Deg)	Run Time (Sec)
	Roll (Φ)	130.7395	131.322				Roll (Φ)	0.4906	1.0731		
121	Pitch (Θ)	45.0	34.7596	55.3893	0.005	231	Pitch (Θ)	-10.2447	20.4851	7.312	0.014
	Yaw (ψ)	0.0	0.0863				Yaw (ψ)	-0.2915	0.3778		
	Roll (Φ)	144.7919	145.3744				Roll (Φ)	108.6987	109.2812		
123	Pitch (Θ)	-6.3093	16.5497	58.7043	0.014	232	Pitch (Θ)	0.0	10.2404	97.58703	0.013
	Yaw (ψ)	-14.1025	14.1888				Yaw (ψ)	-173.153	173.2395		
	Roll (Φ)	0.0	0.5825				Roll (Φ)	0.9839	1.5664		
131	Pitch (Θ)	-164.374	174.6144	108.4614	0.013	312	Pitch (Θ)	10.2398	0.0006	0.5805	0.014
	Yaw (ψ)	-150.101	150.1873				Yaw (ψ)	0.2608	0.1745		
	Roll (Φ)	-0.2011	0.3814				Roll (Φ)	0.0	0.5825		
132	Pitch (Θ)	10.2404	0.0	0.143967	0.013	313	Pitch (Θ)	-156.248	166.4886	89.83103	0.012
	Yaw (ψ)	0.1368	0.0505				Yaw (ψ)	-102.336	102.422		
	Roll (Φ)	76.5964	77.1789				Roll (Φ)	-0.5825	0.0		
212	Pitch (Θ)	-16.0803	26.3207	34.52863	0.012	321	Pitch (Θ)	-10.2373	20.4777	6.8473	0.015
	Yaw (ψ)	0.0	0.0863				Yaw (ψ)	0.1505	0.0642		
	Roll (Φ)	-0.8559	0.2734				Roll (Φ)	9.9889	10.5714		
213	Pitch (Θ)	-10.2349	20.4753	6.916233	0.013	323	Pitch (Θ)	0.0	10.2404	37.2335	0.011
	Yaw (ψ)	0.0863	0.0				Yaw (ψ)	90.975	90.8887		

4. Discussion

Based on this analysis, it can be shown that the most accurate single DCM rotation method for a missile delivered from orbit is the 132 sequences. Of all the rotation schemes, the 132 is the only rotation resulting in less than 1 degree of error in all three Euler angle calculations from the true Euler angles derived. The next closest solution is the 312 sequences with the most accurate pitch value of any rotation not used to calculate the true value. In all simulations when a rotation is repeated (i.e., 131 or 313), the average error is drastically larger than the remaining rotation schemes. All simulations had similar computational burdens, with eleven of the twelve rotation schemes separated by three one-thousandths of a second. Between the two most accurate rotation schemes, the 132 rotation was one one-thousandth of a second quicker than the 312 (0.013 vs 0.014 seconds). Therefore, in order to maximize accuracy for this type of delivery without increasing computing power required, a 132-rotation scheme will provide the most accurate solution, while any solution not rotating by one of the Euler angles will lead to the greatest miss.

Due to the construct of the simulation, the DCM was continuously updated as the missile fell to the ground, causing an updated Euler angle. While the method of solving for each angle resulted in no quadrant ambiguity, the angles values were dependent on other "non-true" values from the DCM sequence selected. To find a constant true Euler angle, further research can be conducted to resolve this dependency.

The optimal trajectory solution is validated through a series of tests with error in the level of tens of thousandths of a point through the duration of flight as determined during the development of the Hamiltonian and using Pontryagin's Principle. The program is written in a manner that is easily replicated for any target by merely changing the desired latitude and longitude of impact while still accounting for variable density and gravitational forces at altitude.

Combining the means to calculate a desired flight path with the ability to convert it into useable information and commands for the missile control system provides a means for defense agencies to fill a gap within the hypersonic weapons systems capabilities that will defeat current defense systems available. The optimal trajectories calculated are being tracked in open loop, amplifying the importance of eliminating kinematically induced errors as described in this manuscript. With these conclusions in mind, one can say that an orbitally delivered kinetic missile that accurately impacts

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

only the desired target at over 33 times the speed of sound and comparable to an atomic weapon with zero fallout is within our reach.

Author Contributions: Conceptualization, D.C.; methodology, D.C. and T.S; software, D.C. and T.S.; validation, D.C and T.S.; formal analysis, D.C.; investigation, D.C.; resources, D.C and T.S.; data curation, D.C.; writing—original draft preparation, D.C.; writing—review and editing, T.S. and J.H.; supervision, T.S.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data supporting reported results can be obtained by contacting the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A SIMULINK® models

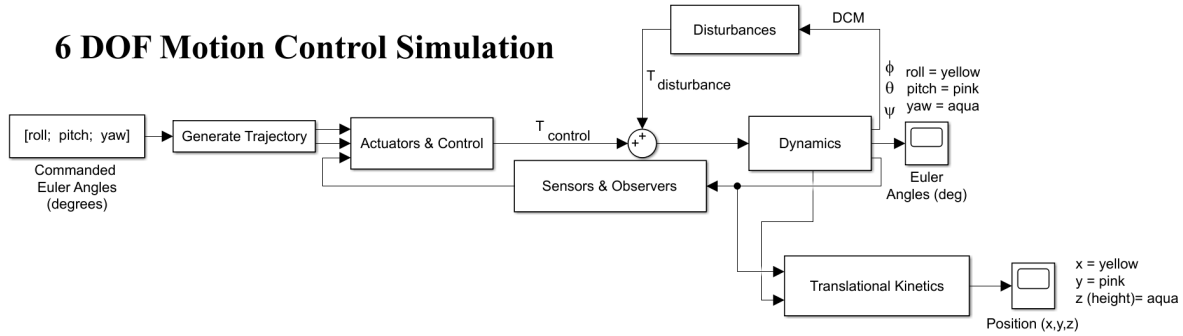


Figure A1. Simulink System Page.

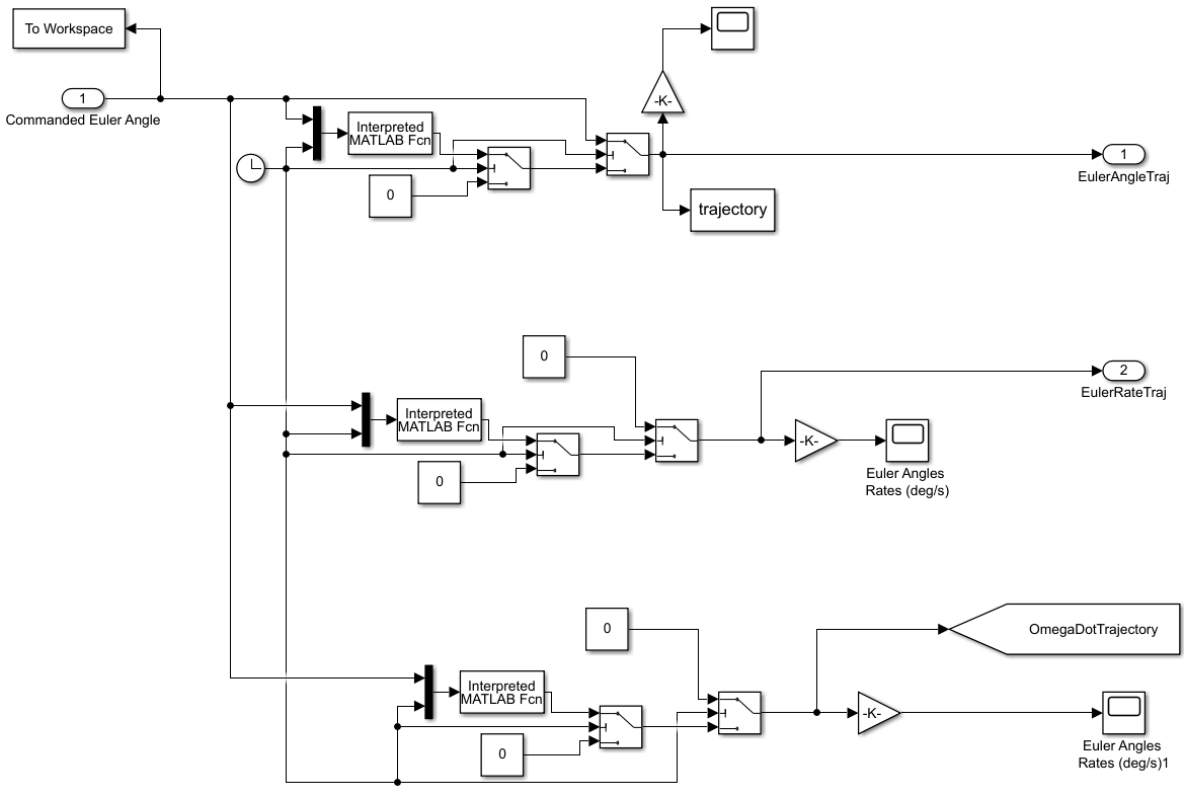


Figure A2. Trajectory Generation Subsystem.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

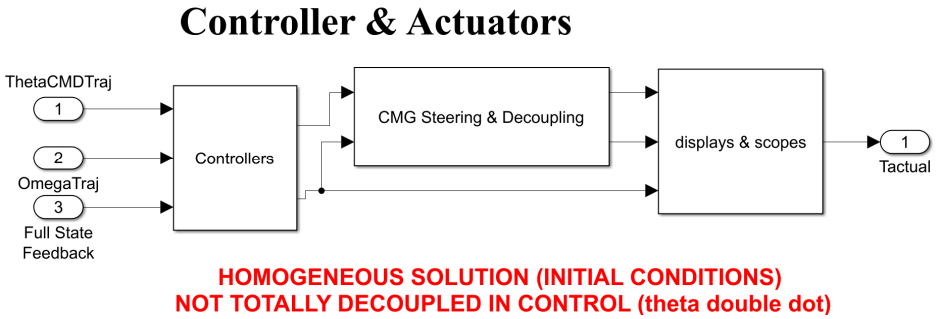


Figure A3. Actuators and controls subsystem.

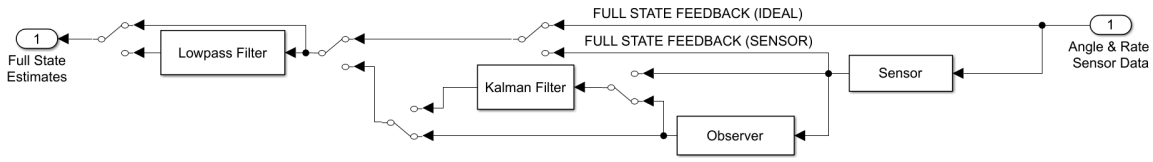


Figure A4. Sensors and observer's subsystem.

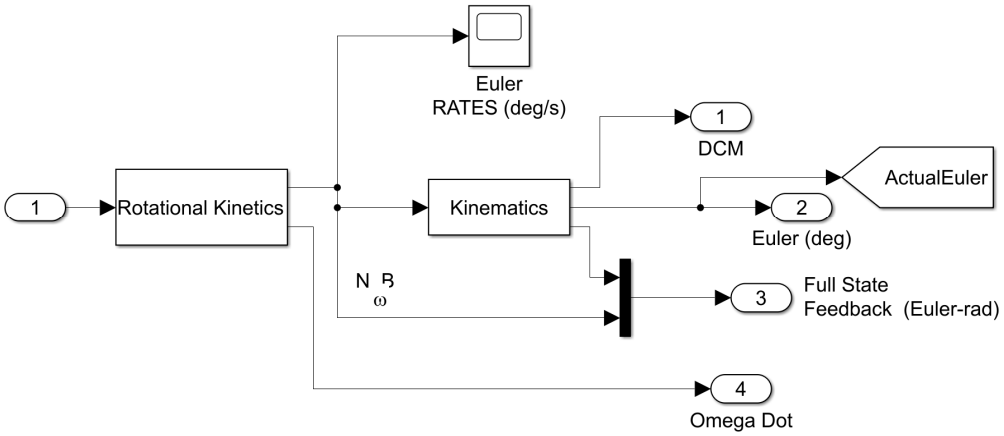


Figure A5. Dynamics Subsystem.

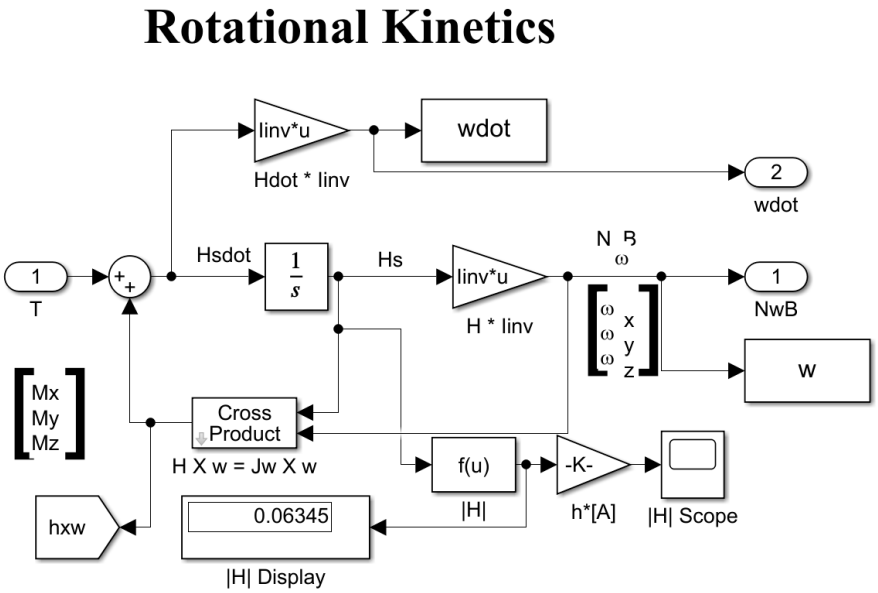


Figure A6. Rotational Kinetics Subsystem.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

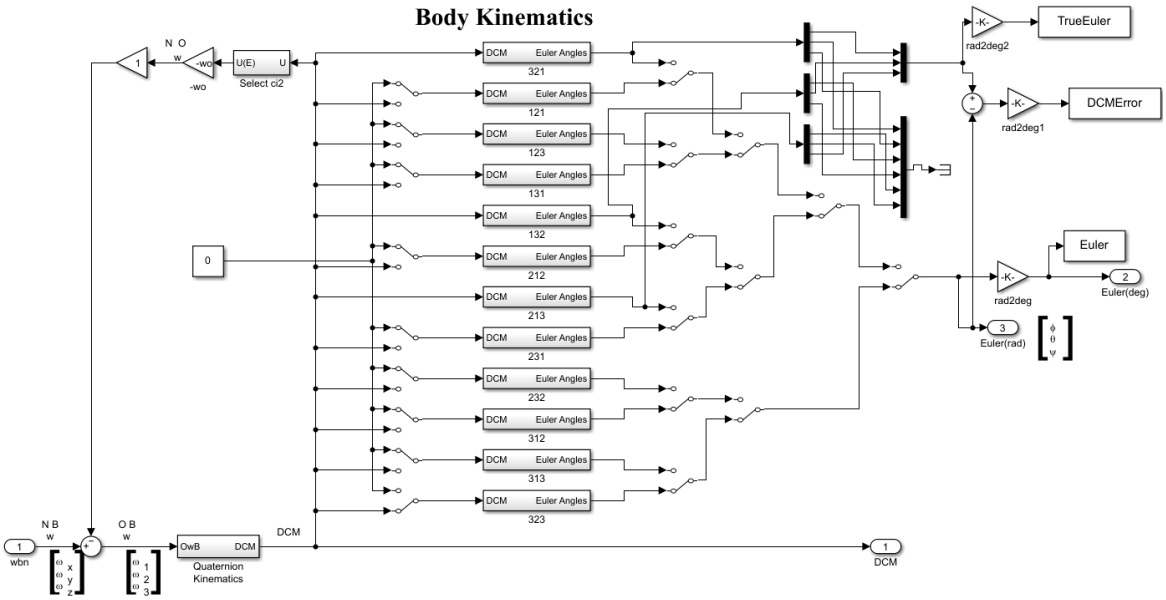


Figure A7. Kinematics Subsystem.

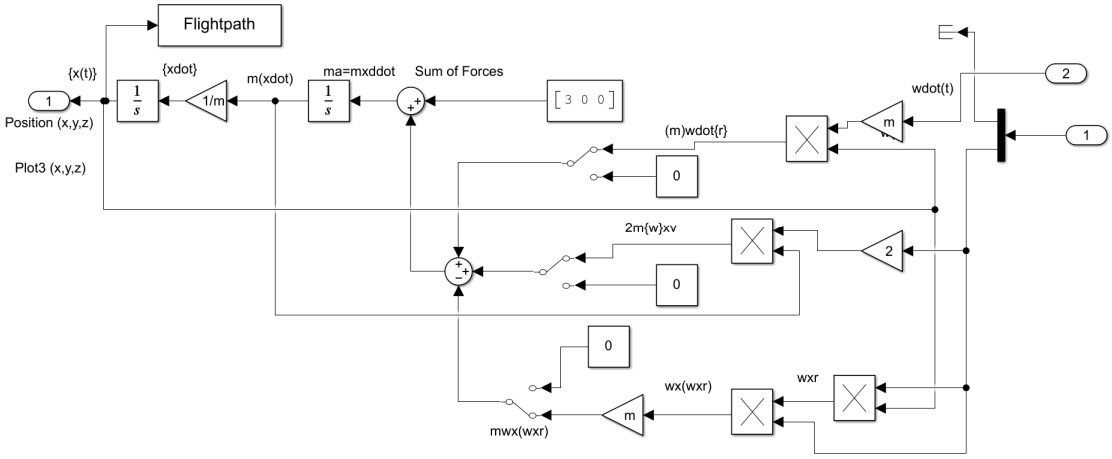


Figure A8. Translational Kinetics Subsystem.

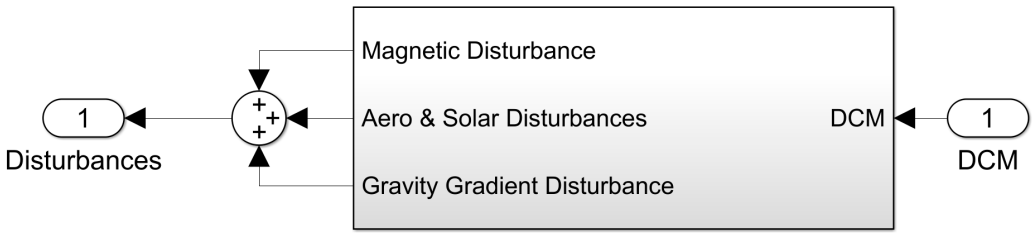


Figure A9. Disturbances Summing Subsystem.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

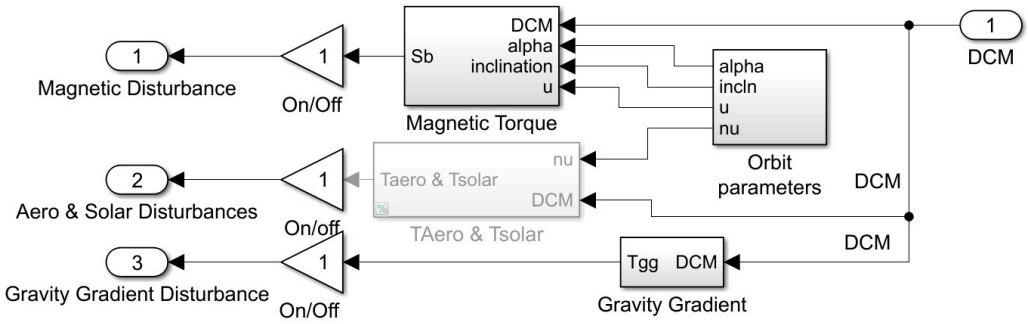


Figure A10. Disturbances Subsystem.

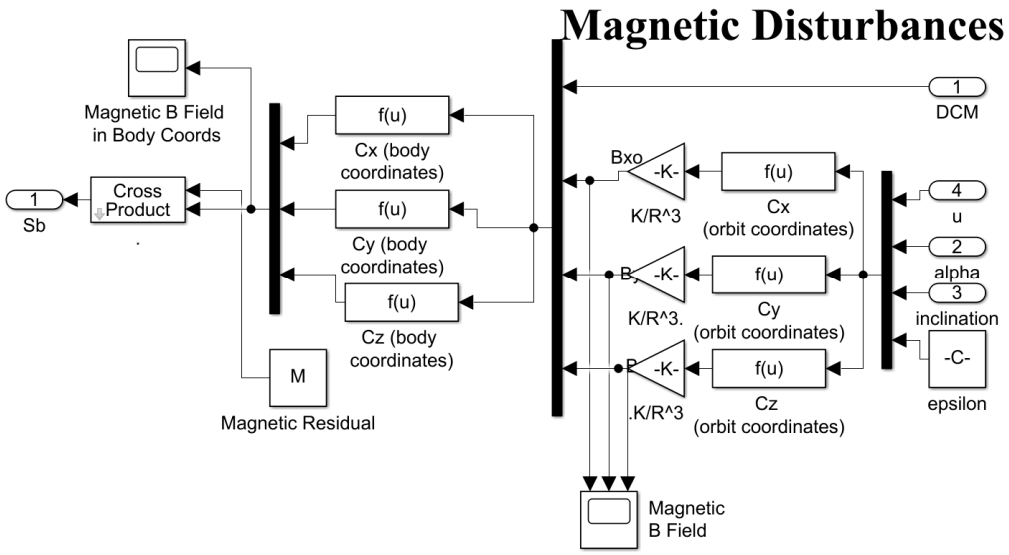


Figure A11. Magnetic Disturbances Subsystem.

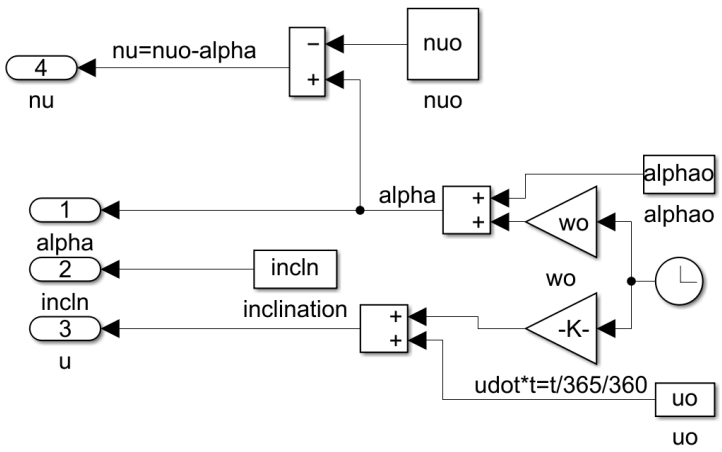


Figure A12. Orbit Parameters Subsystem.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

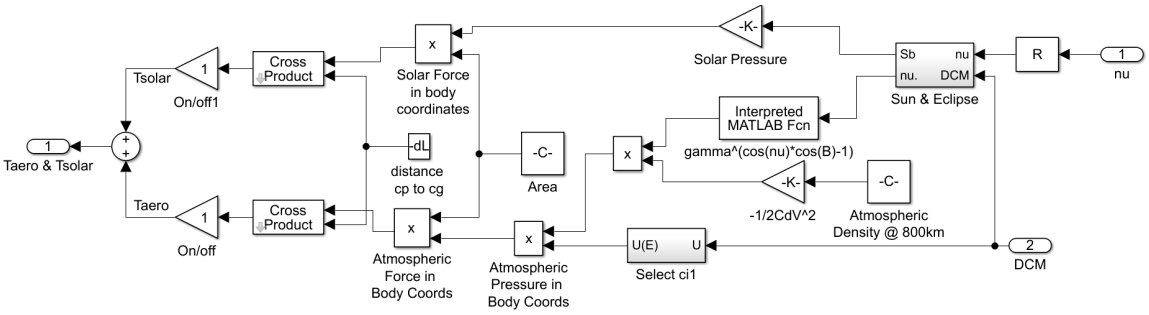


Figure A13. Solar and Aero Disturbances Subsystem.

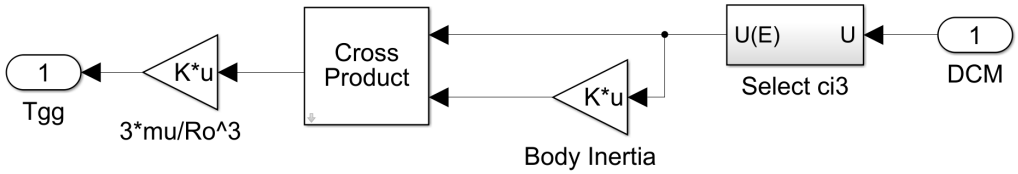


Figure A14. Orbit Parameters Subsystem.

Appendix B Complete canonical scaling and Pontryagin's principle

B.1. Canonical scaling

$$\tilde{r}=\frac{r}{R};\ \tau=\frac{t}{TU};\tilde{v}=\frac{v}{V};\tilde{m}=\frac{m}{M};\tilde{D}=\frac{D}{FU};\tilde{T}=\frac{T}{FU};V=\frac{R}{TU};TU=\sqrt{\frac{R}{g}};FU=MA;A=\frac{R}{TU^2};aU=1 \quad (35)$$

$$x=[r,\theta,\varphi,v,\gamma,\psi]^T\in\mathbb{R},\ \ u=[L_\gamma,L_\psi]\in\mathbb{R}$$

Minimize $\quad J[x(\cdot),u(\cdot),t_f]=\ t_f$

Subject To

$$\dot{r}=\ v\sin\gamma$$

$$\dot{\theta}=\frac{v\cos\gamma\sin\psi}{r\cos\varphi}$$

$$\dot{\varphi}=\frac{v\cos\gamma\cos\psi}{r}$$

$$\dot{v}=-\frac{D}{m}-\frac{\mu\sin\gamma}{r^2}+\frac{T}{m}$$

$$\dot{\gamma}=\frac{L_\gamma}{mv}-\frac{\mu\cos\gamma}{r^2v}+\frac{v}{r}\cos\gamma$$

$$\dot{\psi}=\frac{L_\psi}{mv\cos\gamma}+\frac{v}{r}\cos\gamma\sin\psi\tan\varphi$$

$$t_0=0$$

$$(r_0,\theta_0,\varphi_0,v_0,\gamma_0,\psi_0)=(42164,0,0,0,0,0) \quad (36)$$

$$r_f=6378$$

$$(\theta_f,\varphi_f)=(36.6002,-121.8947)\ (degrees)$$

$$-10\le\tilde{L}_\gamma\le10$$

$$-10\le\tilde{L}_\psi\le10$$

$$\mu=3.986004418\times10^5\left(\frac{km^3}{s^2}\right)$$

$$m=5000\ (kg)$$

$$T=4000\ lbf=17793\ N\left(\frac{kgm}{s^2}\right)=17.793\left(\frac{kg*km}{s^2}\right)$$

$$k_Q=164{,}000\frac{W}{km\cdot K}(Therman\ Conductivity\ of\ Tungsten)$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

B.1.1. r

$$\dot{\tilde{r}} = \frac{d\left({}^{r}/_{R}\right)}{d\left({}^{t}/_{TU}\right)} = \frac{dr\left(\frac{1}{R}\right)}{dt\left(\frac{1}{TU}\right)} = \frac{TU}{R}\frac{dr}{dt} \tag{37}$$

$$\dot{\tilde{r}} = \frac{TU}{R}v\sin\gamma = \frac{TU}{R}(V\tilde{v})\sin\gamma = \frac{VTU}{R}\tilde{v}\sin\gamma = \frac{RTU}{TUR}\tilde{v}\sin\gamma \tag{38}$$

$$\dot{\tilde{r}} = \tilde{v}\sin\gamma \quad => \left(\frac{Distance\ Unit}{Time\ Unit} \times unitless\right) \tag{39}$$

B.1.2. Θ

$$\left(\frac{aU}{TU}\right) * \dot{\tilde{\theta}} = \dot{\theta} \tag{40}$$

$$\dot{\tilde{\theta}} = \frac{v\cos\gamma\sin\psi}{r\cos\varphi} / \left(\frac{au}{TU}\right) = \frac{(\tilde{v}V)\cos\gamma\sin\psi}{(\tilde{r}R)\cos\varphi} / \left(\frac{au}{TU}\right) = \frac{\left(\tilde{v}\frac{R}{TU}\right)\cos\gamma\sin\psi}{(\tilde{r}R)\cos\varphi} / \left(\frac{au}{TU}\right) \tag{41}$$

$$\dot{\tilde{\theta}} = \frac{\tilde{v}\cos\gamma\sin\psi}{\tilde{r}TU\cos\varphi} / \left(\frac{au}{TU}\right) \tag{42}$$

$$\dot{\tilde{\theta}} = \frac{\tilde{v}\cos\gamma\sin\psi}{\tilde{r}\cos\varphi} \quad => \left(\frac{unitless}{time}\right) \tag{43}$$

B.1.3. φ

$$\left(\frac{aU}{TU}\right) * \dot{\tilde{\phi}} = \dot{\phi} \tag{44}$$

$$\dot{\tilde{\phi}} = \frac{\frac{v\cos\gamma\cos\psi}{r}}{\left(\frac{au}{TU}\right)} = \frac{\frac{(\tilde{v}V)\cos\gamma\cos\psi}{(\tilde{r}R)}}{\left(\frac{au}{TU}\right)} = \frac{\frac{\left(\tilde{v}\frac{R}{TU}\right)\cos\gamma\cos\psi}{(\tilde{r}R)}}{\left(\frac{au}{TU}\right)} \tag{45}$$

$$\dot{\tilde{\phi}} = \frac{\tilde{v}\cos\gamma\cos\psi}{\tilde{r}} \quad => \left(\frac{unitless}{time}\right) \tag{46}$$

B.1.4. v

$$\dot{\tilde{v}} = \frac{d\left({}^{v}/_{V}\right)}{d\left({}^{t}/_{TU}\right)} = \frac{dv\left(\frac{1}{V}\right)}{dt\left(\frac{1}{TU}\right)} = \frac{TU}{V}\frac{dv}{dt} \tag{47}$$

$$\dot{\tilde{v}} = \frac{TU}{\frac{R}{TU}}\left(-\frac{D}{m} - \frac{\mu\sin\gamma}{r^2} + \frac{T}{m}\right) = \frac{TU}{\frac{R}{TU}}\left(-\frac{\tilde{D}FU}{\tilde{m}M} - \frac{\mu\sin\gamma}{(\tilde{r}R)^2} + \frac{\tilde{T}FU}{\tilde{m}M}\right) = \frac{TU}{\frac{R}{TU}}\left(-\frac{\tilde{D}MA}{\tilde{m}M} - \frac{\mu\sin\gamma}{(\tilde{r}R)^2} + \frac{\tilde{T}MA}{\tilde{m}M}\right) \tag{48}$$

$$\dot{\tilde{v}} = \frac{TU}{\frac{R}{TU}}\left(-\frac{\tilde{D}M\frac{R}{TU^2}}{\tilde{m}M} - \frac{\mu\sin\gamma}{(\tilde{r}R)^2} + \frac{\tilde{T}M\frac{R}{TU^2}}{\tilde{m}M}\right) = \frac{TU^2}{R}\left(-\frac{\tilde{D}R}{\tilde{m}TU^2} - \frac{\mu\sin\gamma}{(\tilde{r}R)^2} + \frac{\tilde{T}R}{\tilde{m}TU^2}\right) \tag{49}$$

$$\dot{\tilde{v}} = -\frac{\tilde{D}}{\tilde{m}} - \frac{\mu\sin\gamma\,TU^2}{\tilde{r}R^3} + \frac{\tilde{T}}{\tilde{m}} \qquad set\ \tilde{\mu} = \left(\frac{TU^2}{R^3}\right)\mu \tag{50}$$

$$\dot{\tilde{v}} = -\frac{\tilde{D}}{\tilde{m}} - \frac{\tilde{\mu}\sin\gamma}{\tilde{r}^2} + \frac{\tilde{T}}{\tilde{m}} \qquad => \left(\frac{Dist}{Time^2}\right) \tag{51}$$

B.1.5. γ

$$\left(\frac{aU}{TU}\right) * \dot{\tilde{\gamma}} = \dot{\gamma} \tag{52}$$

$$\dot{\tilde{\gamma}} = \frac{\left(\frac{L_\gamma}{mv} - \frac{\mu\cos\gamma}{r^2 v} + \frac{v}{r}\cos\gamma\right)}{\frac{au}{TU}} = \frac{\left(\frac{L_\gamma}{\tilde{m}M\tilde{v}V} - \frac{\mu\cos\gamma}{(\tilde{r}R)^2\tilde{v}V} + \frac{\tilde{v}V}{\tilde{r}R}\cos\gamma\right)}{\frac{au}{TU}} \tag{53}$$

$$\dot{\tilde{\gamma}} = \frac{\left(\frac{L_\gamma}{\tilde{m}M\tilde{v}\frac{R}{TU}} - \frac{\mu\cos\gamma}{(\tilde{r}R)^2\tilde{v}\frac{R}{TU}} + \frac{\tilde{v}\frac{R}{TU}}{\tilde{r}R}\cos\gamma\right)}{\frac{au}{TU}} = \frac{L_\gamma TU^2}{\tilde{m}\tilde{v}RM} - \frac{\mu\cos\gamma\,TU^2}{\tilde{r}^2\tilde{v}R^3} + \frac{\tilde{v}}{\tilde{r}}\cos\gamma \tag{54}$$

$$\dot{\tilde{\gamma}} = \frac{L_\gamma TU^2}{\tilde{m}\tilde{v}RM} - \frac{\tilde{\mu}\cos\gamma}{\tilde{r}^2\tilde{v}} + \frac{\tilde{v}}{\tilde{r}}\cos\gamma \qquad set\ \tilde{L} = \left(\frac{TU^2}{RM}\right)L \tag{55}$$

$$\dot{\tilde{\gamma}} = \frac{\tilde{L}_\gamma}{\tilde{m}\tilde{v}} - \frac{\tilde{\mu}\cos\gamma}{\tilde{r}^2\tilde{v}} + \frac{\tilde{v}}{\tilde{r}}\cos\gamma \quad => \left(\frac{unitless}{time}\right) \tag{56}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

B.1.6. ψ

$$\dot{\tilde{\psi}} = \frac{\left(\frac{L_\psi}{mv\cos\gamma} + \frac{v}{r}\cos\gamma\sin\psi\tan\varphi\right)}{\frac{au}{TU}} = \frac{\left(\frac{L_\psi}{\tilde{m}M\tilde{v}V\cos\gamma} + \frac{\tilde{v}V}{\tilde{r}R}\cos\gamma\sin\psi\tan\varphi\right)}{\frac{au}{TU}} \tag{57}$$

$$\dot{\tilde{\psi}} = \frac{\left(\frac{L_\psi}{\tilde{m}M\tilde{v}\frac{R}{TU}\cos\gamma} + \frac{\tilde{v}\frac{R}{TU}}{\tilde{r}R}\cos\gamma\sin\psi\tan\varphi\right)}{\frac{au}{TU}} = \frac{L_\psi TU^2}{\tilde{m}\tilde{v}RM\cos\gamma} + \frac{\tilde{v}}{\tilde{r}}\cos\gamma\sin\psi\tan\varphi \tag{58}$$

$$\dot{\tilde{\psi}} = \frac{\tilde{L}_\psi}{\tilde{m}\tilde{v}\cos\gamma} + \frac{\tilde{v}}{\tilde{r}}\cos\gamma\sin\psi\tan\varphi \quad => \left(\frac{unitless}{Time}\right) \tag{59}$$

B.1.7. Q

$$\dot{Q} = k_Q\sqrt{\rho}v^3 \rightarrow \left(\frac{kg^{.5}}{m^{.5}}\right)\left(\frac{kg^{.5}}{m^{1.5}}\right)\left(\frac{m^3}{s^3}\right) \rightarrow \frac{kmg^2}{m^2s^2} \rightarrow \frac{W}{m^2} \tag{60}$$

$$\dot{\tilde{Q}} = \frac{\dot{Q}}{QU} \text{ and set } KU = \frac{M^{.5}}{R^{.5}}, DU = \frac{M}{R^3} \tag{61}$$

$$\dot{\tilde{Q}} = \frac{TU}{QU}\left(KU\tilde{k}_Q\right)\sqrt{DU\tilde{\rho}}(V\tilde{v})^3 = \frac{TU}{QU}\left(\frac{M^{.5}}{R^{.5}}\right)\tilde{k}_Q\left(\frac{M^{.5}}{R^{1.5}}\right)\sqrt{\tilde{\rho}}\left(\frac{R^3}{TU^3}\right)\tilde{v}^3 = \frac{TU}{QU}\left(\frac{MR^2}{R^2TU^3}\right)\tilde{k}_Q\sqrt{\tilde{\rho}}\tilde{v}^3 \tag{62}$$

$$set\ QU = \frac{R^2TU^2}{MR^2} \rightarrow \dot{\tilde{Q}} = \tilde{k}_Q\sqrt{\tilde{\rho}}\tilde{v}^3 \rightarrow \frac{mass}{time^2} \tag{63}$$

B.2. HAMVET

B.2.1. Hamiltonian construction

$$\bar{H}(x,\lambda,u) = F(x,u) + \lambda^T f(x,u) + \varepsilon^T h(x,u) \tag{64}$$

$$\bar{H}(x,\lambda,\varepsilon,u) = 0 + \lambda\begin{bmatrix} r \\ \theta \\ \varphi \\ v \\ \gamma \\ \psi \end{bmatrix} \text{ since we need to have one costate for every state} \tag{65}$$

$$\begin{aligned}\bar{H}(x,\lambda,\varepsilon,u) = {} & \lambda_r(\tilde{v}\sin\gamma) + \lambda_\theta\left(\frac{\tilde{v}\cos\gamma\sin\psi}{\tilde{r}\cos\varphi}\right) + \lambda_\varphi\left(\frac{\tilde{v}\cos\gamma\cos\psi}{\tilde{r}}\right) + \lambda_v\left(-\frac{\tilde{D}}{\tilde{m}} - \frac{\tilde{\mu}\sin\gamma}{r^2} + \frac{\tilde{T}}{\tilde{m}}\right) \\ & + \lambda_\gamma\left(\frac{\tilde{L}_\gamma}{\tilde{m}\tilde{v}} - \frac{\tilde{\mu}\cos\gamma}{\tilde{r}^2\tilde{v}} + \frac{\tilde{v}}{\tilde{r}}\cos\gamma\right) + \lambda_\psi\left(\frac{\tilde{L}_\psi}{\tilde{m}\tilde{v}\cos\gamma} + \frac{\tilde{v}}{\tilde{r}}\cos\gamma\sin\psi\tan\varphi\right) + \varepsilon_1(\tilde{L}_\gamma) \\ & + \varepsilon_2(\tilde{L}_\psi) +\end{aligned} \tag{66}$$

B.2.2. Adjoint equations

$$\dot{\lambda} = -(\partial H)/\partial x \tag{67}$$

So,

$$\begin{aligned}-\dot{\lambda}_r = \left(\frac{\partial\bar{H}}{\partial r}\right) = {} & -\lambda_\theta\frac{\tilde{v}\cos\gamma\sin\psi}{\tilde{r}^2\cos\varphi} - \lambda_\varphi\frac{\tilde{v}\cos\gamma\cos\psi}{\tilde{r}^2} + \lambda_v\frac{2\tilde{\mu}\sin\gamma}{\tilde{r}^3} - \lambda_\gamma\left(\frac{\tilde{v}}{\tilde{r}^2}\cos\gamma - \frac{2\tilde{\mu}\cos\gamma}{\tilde{v}\tilde{r}^3}\right) \\ & - \lambda_\psi\left(\frac{\tilde{v}}{\tilde{r}^2}\cos\gamma\sin\psi\tan\varphi\right)\end{aligned} \tag{68}$$

$$-\dot{\lambda}_\theta = 0 \left(\frac{\partial\bar{H}}{\partial\theta}\right) \tag{69}$$

$$-\dot{\lambda}_\varphi = \left(\frac{\partial\bar{H}}{\partial\varphi}\right) = \lambda_\theta\frac{\tilde{v}\cos\gamma\sin\psi\sin\varphi}{\tilde{r}\cos^2\varphi} + \lambda_\psi\left(\frac{\tilde{v}\cos\gamma\sin\psi\,(\tan^2\varphi + 1)}{\tilde{r}}\right) \tag{70}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

$$-\dot{\lambda}_v = \left(\frac{\partial \bar{H}}{\partial v}\right) = \lambda_r(\sin\gamma) + \lambda_\theta \frac{\cos\gamma \sin\psi}{\tilde{r}\cos\varphi} + \lambda_\varphi \frac{\cos\gamma\cos\psi}{\tilde{r}} + \lambda_\gamma\left(\frac{\cos\gamma}{\tilde{r}} - \frac{\tilde{L}_\gamma}{\tilde{m}\tilde{v}^2} + \frac{\tilde{\mu}\cos\gamma}{\tilde{v}^2\tilde{r}^2}\right) - \lambda_\psi\left(\frac{\tilde{L}_\psi}{\tilde{m}\tilde{v}^2\cos\gamma} - \frac{\cos\gamma\sin\psi\tan\varphi}{\tilde{r}}\right) + 3\varepsilon_3\tilde{k}_Q\tilde{v}^2\sqrt{\tilde{\rho}} \quad (71)$$

$$-\dot{\lambda}_\gamma = \left(\frac{\partial \bar{H}}{\partial \gamma}\right) = \lambda_r(\tilde{v}\cos\gamma) - \lambda_\theta\frac{\tilde{v}\sin\gamma\sin\psi}{\tilde{r}\cos\varphi} - \lambda_\varphi\frac{\tilde{v}\sin\gamma\cos\psi}{\tilde{r}} - \lambda_\gamma\left(\frac{\tilde{v}\sin\gamma}{\tilde{r}} - \frac{\tilde{\mu}\sin\gamma}{\tilde{v}\tilde{r}^2}\right) - \lambda_v\frac{\tilde{\mu}\cos\gamma}{\tilde{r}^2} - \lambda_\psi\left(\frac{\tilde{v}}{\tilde{r}}\sin\gamma\sin\psi\tan\varphi - \frac{\tilde{L}_\psi\sin\gamma}{\tilde{m}\tilde{v}\cos^2\gamma}\right) \quad (72)$$

$$-\dot{\lambda}_\psi = \left(\frac{\partial \bar{H}}{\partial \psi}\right) = \lambda_\theta\frac{\tilde{v}\cos\gamma\cos\psi}{\tilde{r}\cos\varphi} - \lambda_\varphi\frac{\tilde{v}\cos\gamma\sin\psi}{\tilde{r}} + \lambda_\psi\left(\frac{\tilde{v}}{\tilde{r}}\cos\gamma\cos\psi\tan\varphi\right) \quad (73)$$

B.2.3. Minimizing the Hamiltonian with respect to u

$$\frac{\partial \bar{H}}{\partial u} = \frac{\partial \bar{H}}{\partial \tilde{L}_\gamma} = 0 = \frac{\lambda_\gamma}{\tilde{m}\tilde{v}} + \varepsilon_1 \quad (74)$$

$$\frac{\partial \bar{H}}{\partial u} = \frac{\partial \bar{H}}{\partial \tilde{L}_\psi} = 0 = \frac{\lambda_\psi}{\tilde{m}\tilde{v}\cos\gamma} + \varepsilon_2 \quad (75)$$

$$\varepsilon_1\begin{cases} \leq 0 \ if & \tilde{L}_\gamma = -10 \\ = 0 \ if -10 \leq & \tilde{L}_\gamma \leq 10 \\ \geq 0 \ if & \tilde{L}_\gamma = 10 \end{cases} \quad (76)$$

$$\varepsilon_2\begin{cases} \leq 0 \ if & \tilde{L}_\psi = -10 \\ = 0 \ if -10 \leq & \tilde{L}_\psi \leq 10 \\ \geq 0 \ if & \tilde{L}_\psi = 10 \end{cases} \quad (77)$$

$$\varepsilon_3\begin{cases} \leq 0 \ if & \dot{Q} = 0 \\ = 0 \ if & 0 \leq \dot{Q} \leq 120 \\ \geq 0 \ if & \dot{Q} = 120 \end{cases} \quad (78)$$

B.2.4. Value condition

$$H\left|@t_f\right| = -\frac{\partial E}{\partial t_f} \quad (79)$$

$$H\left|@t_f\right| = H(\lambda(t_f), x(t_f), \varepsilon(t_f), u(t_f), t_f) \quad (80)$$

$$\vec{E} = t_f + \nu_1(r_f - r^f) + \nu_2(\theta_f - \theta^f) + \nu_3(\varphi_f - \varphi^f) \quad (81)$$

$$-\frac{\partial E}{\partial t_f} = 1 \quad (82)$$

$$\bar{H}(t_f) = -1 \quad (83)$$

B.2.5. Hamiltonian evolution condition

$$\frac{\partial H}{\partial t} = \frac{\partial \mathcal{H}}{\partial t} = 0 \quad (84)$$

B.2.6. Transversality

$$\vec{E} = E(x_f) + \nu^t e(x_f) \ with \ \lambda(t_f) = \frac{\partial \vec{E}}{\partial x_f} \quad (85)$$

$$\vec{E} = t_f + \nu_1(r_f - r^f) + \nu_2(\theta_f - \theta^f) + \nu_3(\varphi_f - \varphi^f) \quad (86)$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

$$\lambda_r(t_f) = \frac{\partial \vec{E}}{\partial r_f} = \nu_1 \tag{87}$$

$$\lambda_\theta(t_f) = \frac{\partial \vec{E}}{\partial \theta_f} = \nu_2 \tag{88}$$

$$\lambda_\varphi(t_f) = \frac{\partial \vec{E}}{\partial \varphi_f} = \nu_3 \tag{89}$$

$$\lambda_v(t_f) = \frac{\partial \vec{E}}{\partial v} = 0 \tag{90}$$

$$\lambda_\gamma(t_f) = \frac{\partial \vec{E}}{\partial \gamma_f} = 0 \tag{91}$$

$$\lambda_\psi(t_f) = \frac{\partial \vec{E}}{\partial \psi_f} = 0 \tag{92}$$

So, the boundary value problem becomes:

$$\dot{\tilde{r}} = \tilde{v}\sin\gamma \tag{93}$$

$$\dot{\tilde{\theta}} = \frac{\tilde{v}\cos\gamma\sin\psi}{\tilde{r}\cos\varphi} \tag{94}$$

$$\dot{\tilde{\varphi}} = \frac{\tilde{v}\cos\gamma\cos\psi}{\tilde{r}} \tag{95}$$

$$\dot{\tilde{v}} = -\frac{\tilde{D}}{\tilde{m}} - \frac{\tilde{\mu}\sin\gamma}{r^2} + \frac{\tilde{T}}{\tilde{m}} \tag{96}$$

$$\dot{\tilde{\gamma}} = \frac{\tilde{L}_\gamma}{\tilde{m}\tilde{v}} - \frac{\tilde{\mu}\cos\gamma}{\tilde{r}^2\tilde{v}} + \frac{\tilde{v}}{\tilde{r}}\cos\gamma \tag{97}$$

$$\dot{\tilde{\psi}} = \frac{\tilde{L}_\psi}{\tilde{m}\tilde{v}\cos\gamma} + \frac{\tilde{v}}{\tilde{r}}\cos\gamma\sin\psi\tan\varphi \tag{98}$$

$$\dot{\tilde{Q}} = \tilde{k}_Q\sqrt{\tilde{\rho}}\tilde{v}^3 \le 120 \tag{99}$$

$$-\dot{\tilde{\lambda}}_r = -\lambda_\theta\frac{\tilde{v}\cos\gamma\sin\psi}{\tilde{r}^2\cos\varphi} - \lambda_\varphi\frac{\tilde{v}\cos\gamma\cos\psi}{\tilde{r}^2} + \lambda_v\frac{2\tilde{\mu}\sin\gamma}{\tilde{r}^3} - \lambda_\gamma\left(\frac{\tilde{v}}{\tilde{r}^2}\cos\gamma - \frac{2\tilde{\mu}\cos\gamma}{\tilde{v}\tilde{r}^3}\right) - \lambda_\psi\left(\frac{\tilde{v}}{\tilde{r}^2}\cos\gamma\sin\psi\tan\varphi\right) \tag{100}$$

$$-\dot{\tilde{\lambda}}_\theta = 0 \tag{101}$$

$$-\dot{\tilde{\lambda}}_\varphi = \lambda_\theta\frac{\tilde{v}\cos\gamma\sin\psi\sin\varphi}{\tilde{r}\cos^2\varphi} + \lambda_\psi\left(\frac{\tilde{v}\cos\gamma\sin\psi\,(\tan^2\varphi+1)}{\tilde{r}}\right) \tag{102}$$

$$-\dot{\tilde{\lambda}}_v = \lambda_r(\sin\gamma) + \lambda_\theta\frac{\cos\gamma\sin\psi}{\tilde{r}\cos\varphi} + \lambda_\varphi\frac{\cos\gamma\cos\psi}{\tilde{r}} + \lambda_\gamma\left(\frac{\cos\gamma}{\tilde{r}} - \frac{\tilde{L}_\gamma}{\tilde{m}\tilde{v}^2} + \frac{\tilde{\mu}\cos\gamma}{\tilde{v}^2\tilde{r}^2}\right) - \lambda_\psi\left(\frac{\tilde{L}_\psi}{\tilde{m}\tilde{v}^2\cos\gamma} - \frac{\cos\gamma\sin\psi\tan\varphi}{\tilde{r}}\right) + 3\varepsilon_3\tilde{k}_Q\tilde{v}^2\sqrt{\tilde{\rho}} \tag{103}$$

$$-\dot{\tilde{\lambda}}_\psi = \lambda_\theta\frac{\tilde{v}\cos\gamma\cos\psi}{\tilde{r}\cos\varphi} - \lambda_\varphi\frac{\tilde{v}\cos\gamma\sin\psi}{\tilde{r}} + \lambda_\psi\left(\frac{\tilde{v}}{\tilde{r}}\cos\gamma\cos\psi\tan\varphi\right) \tag{104}$$

With Boundary Conditions:

$$\left(\tilde{r}_0, \tilde{\theta}_0, \tilde{\varphi}_0, \tilde{v}_0, \tilde{\gamma}_0, \tilde{\psi}_0\right) = (6.61,0,0,0,0,0) \tag{105}$$

$$\tilde{r}_f = 1 \tag{106}$$

$$\left(\tilde{\theta}_f, \tilde{\varphi}_f\right) = (36.6002, -121.8947)\ (degrees) \tag{107}$$

$$\tilde{\lambda}_v(t_f) = 0 \tag{108}$$

$$\tilde{\lambda}_\gamma(t_f) = 0 \tag{109}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1

$$\tilde{\lambda}_{\psi}(t_f) = 0 \tag{110}$$

$$\bar{H}(t_f) = -1 = constant \tag{111}$$

With Complementarity Conditions:

$$\varepsilon_1 \begin{cases} \leq 0 \ \ if & \tilde{L}_{\gamma} = -10 \\ = 0 \ if & -10 \leq \tilde{L}_{\gamma} \leq 10 \\ \geq 0 \ if & \tilde{L}_{\gamma} = 10 \end{cases} \tag{111}$$

$$\varepsilon_2 \begin{cases} \leq 0 \ \ if & \tilde{L}_{\psi} = -10 \\ = 0 \ if & -10 \leq \tilde{L}_{\psi} \leq 10 \\ \geq 0 \ if & \tilde{L}_{\psi} = 10 \end{cases} \tag{111}$$

$$\varepsilon_3 \begin{cases} \leq 0 \ \ if & \dot{Q} = 0 \\ = 0 \ if & 0 \leq \dot{Q} \leq 4.9 \\ \geq 0 \ if & \dot{Q} = 4.9 \end{cases} \tag{111}$$

Appendix C DIDO Program Code

C.1. Preamble file

```
function [rBar, thetaBar, phiBar, vBar, gammaBar, psiBar, LgammaBar, LpsiBar, tBar, ...

r0Bar, theta0Bar, phi0Bar, v0Bar, gamma0Bar, psi0Bar, t0Bar, ...

rfBar, thetafBar, phifBar, vfBar, gammafBar, psifBar, tfBar, ...

uBar, TBar, mBar, DBar, Cd, S, R, V, kq, FU, DU, KU] ...

= MS4Preamble(primal)
%==========================================================================
% Preamble for Missile Launched from Geostationary Orbit to Specific Target min time
%==========================================================================
%Scaled Values
rBar = primal.states(1,:); %States
thetaBar = primal.states(2,:);
phiBar = primal.states(3,:);
vBar = primal.states(4,:);
gammaBar = primal.states(5,:);
psiBar = primal.states(6,:);

LgammaBar = primal.controls(1,:); %Controls
LpsiBar = primal.controls(2,:);

tBar = primal.time; %Time

r0Bar = primal.initial.states(1); %Initial States
theta0Bar = primal.initial.states(2);
phi0Bar = primal.initial.states(3);
v0Bar = primal.initial.states(4);
gamma0Bar = primal.initial.states(5);
psi0Bar = primal.initial.states(6);

t0Bar = primal.initial.time; %Initial Time
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1


```
rfBar = primal.final.states(1); %Final States
thetafBar = primal.final.states(2);
phifBar = primal.final.states(3);
vfBar = primal.final.states(4);
gammafBar = primal.final.states(5);
psifBar = primal.final.states(6);

tfBar = primal.final.time; %Final Time

uBar = primal.constants.uBar; %Constants
TBar = primal.constants.TBar;
mBar = primal.constants.mBar;
DBar = primal.constants.DBar;
g = primal.constants.g;
Cd = primal.constants.Cd;
S = primal.constants.S;
FU = primal.constants.FU;
R = primal.constants.R;
V = primal.constants.V;
TU = primal.constants.TU;
kq = primal.constants.kq;
DU = primal.constants.DU;
KU = primal.constants.KU;
```

```
%eof
C.1. Pz %Search Space for rBar

-pi, pi; %Search Space for thetaBar (Longitude)

-pi/2, pi/2; %Search Space for phiBar (Latitude)

0, 5; %Search Space for vBar

-pi, pi; %Search Space for gammaBar

-pi, pi]; %Search Space for psiBar
search.controls = [-11, 11; %Search Space for Lgamma

-11, 11]; %Search Space for Lpsi
%=========================================================================
%Defining Problem Specific Constants
%Unscaled Constants
mu = 3.986004418e5; %Earth Gravitational Parameter km/sec2
Thrust = 17.799; %4000lbf converted to kgkm/s^2
mass = 5000; %Initial Mass

MS4.constants.g = 0.00981; %Gravitational Constant of Earth in km/s^2
MS4.constants.Cd = 0.8; %Drag Coefficient
MS4.constants.S = 3.44e-6; %Drag Reference Area (km^2)
MS4.constants.kq = sqrt(1000)*9e-5; %kg^.5/km^.5

%Scaling Factors
```

 doi:10.20944/preprints202312.0787.v1


```
MS4.constants.R = 6378; %Distance Scaling
MS4.constants.TU = sqrt(MS4.constants.R/MS4.constants.g);%Time Scaling
MS4.constants.AU = 1; %Angle Scaling => Radians remain unscaled
MS4.constants.V = MS4.constants.R/MS4.constants.TU; %Velocity Scaling
MS4.constants.M = 10; %Mass Scaling
MS4.constants.A = MS4.constants.R/(MS4.constants.TU^2);%Acceleration Scaling
MS4.constants.FU = MS4.constants.M*MS4.constants.A; %Force Scaling
MS4.constants.DU = MS4.constants.M/(MS4.constants.R^3); %Density Scaling
MS4.constants.KU = (MS4.constants.M^.5)/(MS4.constants.R^.5);

%Scaled Constants
MS4.constants.uBar = (MS4.constants.TU^2/(MS4.constants.R^3))*mu; %Scaled Earth Gravitational Parameter
%MS4.constants.TBar = Thrust/MS4.constants.FU; %Scaled Thrust
MS4.constants.TBar = Thrust;
MS4.constants.mBar = mass/MS4.constants.M; %Scaled Mass in kg
MS4.constants.DBar = 1.2344;
%======================================================================
%Boundary Conditions
rInitial = 43378; %kilometers - Does not change
rFinal = 6378; %kilometers - Changes with tgt location
r0Bar = rInitial/MS4.constants.R; %Scale initial condition
rfBar = rFinal/MS4.constants.R; %Scale final condition
phi0Bar = deg2rad(0); %Latitude is unscaled in radians
theta0Bar = deg2rad(-90); %Longitude is unscaled in radians
v0Bar = 1/MS4.constants.V; %Initial velocity is near zero
gamma0Bar = deg2rad(-89); %Flight Path angle is unscaled in radians
psi0Bar = deg2rad(0); %Flight Path Azimuth is unscaled in radians
phifBar = deg2rad(36.6002); %Latitude of target to radians
thetafBar = deg2rad(-121.8947); %Longitude of target to radians

bounds.events = [r0Bar, r0Bar; %Radius of geosynchronous orbit

theta0Bar, theta0Bar;%Set initial latitude of orbit

phi0Bar, phi0Bar; %Set initial longitude of orbit

v0Bar, v0Bar; %v0 is 0

gamma0Bar, gamma0Bar;%Set initial flightpath angle

psi0Bar, psi0Bar; %Set initial flightpath azimuth

rfBar, rfBar; %Target radius

thetafBar, thetafBar;%Target latitude

phifBar, phifBar]; %Target longitude
%======================================================================
%Initial and Final Time Constraints
bounds.initial.time = [0, 0]; %Clock starts at beginning
bounds.final.time = [0, 15]; %Guess on upper bound
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1


```
%==========================================================================
%Path Constraints
bounds.path = [-10, 10; %Control Constraint, -10<=LBar<=10

-10, 10; %Control Constraint, -10<=LBar<=10

0, 4.9]; %Heating Rate Constraint
%==========================================================================
%Problem Definition
MS4.cost = 'MS4Cost';
MS4.dynamics = 'MS4Dynamics';
MS4.events = 'MS4Events';
MS4.path = 'MS4Path';
MS4.search = search;
MS4.bounds = bounds;

%==========================================================================
%DIDO Formulation check

check(MS4);

%==========================================================================
%Node selection
algorithm.nodes = 55;

%==========================================================================
%Run DIDO and time the program
tic
[cost, primal, dual] = dido(MS4, algorithm);
toc

%==========================================================================
%Output Analysis

[rBar, thetaBar, phiBar, vBar, gammaBar, psiBar, LgammaBar, LpsiBar, tBar, ...

r0Bar, theta0Bar, phi0Bar, v0Bar, gamma0Bar, psi0Bar, t0Bar, ...

rfBar, thetafBar, phifBar, vfBar, gammafBar, psifBar, tfBar, ...

uBar, TBar, mBar, DBar, Cd, S, R, V, kq, FU, DU, KU] ...

= MS4Preamble(primal);


MinTime = cost*MS4.constants.TU

lam_rBar = dual.dynamics(1,:);
lam_thetaBar = dual.dynamics(2,:);
lam_phiBar = dual.dynamics(3,:);
lam_vBar = dual.dynamics(4,:);
lam_gammaBar = dual.dynamics(5,:);
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1


```
lam_psiBar = dual.dynamics(6,:);

eps1 = dual.path(1,:);
eps2 = dual.path(2,:);
eps3 = dual.path(3,:);

stat1 = lam_gammaBar./(mBar.*vBar);
stat2 = lam_psiBar./(mBar.*vBar.*cos(gammaBar));
%=========================================================================
%PLOTS
figure;
plot(tBar, [rBar; thetaBar; phiBar; vBar; gammaBar; psiBar]); grid on;
title('Scaled State Trajectories (2D View)','Interpreter','latex');
xlabel('Scaled Time (\tau)','FontSize',16);
ylabel('Scaled States $\mathbf{\tilde{x}}$','Interpreter','latex','FontSize',16);
legend('$\tilde{r}$','$\tilde{\theta}$','$\tilde{\phi}$','$\tilde{v}$','$\tilde{\gamma}$','$\tilde{\
psi}$','Interpreter','latex','FontSize', 10);

figure;
plot(tBar, [LgammaBar; LpsiBar]); grid on;
title('Scaled Controls')
legend('$\tilde{L}_\gamma$','$\tilde{L}_\psi$','Interpreter','latex','FontSize',10)
xlabel('Scaled Time (\tau)','FontSize',16)

figure;
plot(tBar, [lam_rBar; lam_thetaBar; lam_phiBar; lam_vBar; lam_gammaBar; lam_psiBar]); grid
on;
title('Scaled Co-State Trajectories','Interpreter','latex');
legend('$\tilde{\lambda_r}$','$\tilde{\lambda_\theta}$','$\tilde{\lambda_\phi}$','$\tilde{\la
mbda_v}$','$\tilde{\lambda_\gamma}$','$\tilde{\lambda_\psi}$','Interpreter','latex','FontSize', 10);
xlabel('Scaled Time (\tau)','FontSize',16);

figure;
plot(tBar, dual.Hamiltonian); grid on;
title('Scaled Hamiltonian');
xlabel('Scaled Time (\tau)','FontSize',16);
ylabel('Scaled Hamiltonian $\tilde{H}$','Interpreter','latex','FontSize',16);

figure;
plot3(thetaBar,phiBar,rBar); grid on;
title('Three Dimensional Flight Path of the Missile','Interpreter','latex');
xlabel('Latitude ${\theta}$','Interpreter','latex','FontSize',16);
ylabel('Longitude ${\phi}$','Interpreter','latex','FontSize',16);
zlabel('Radius from Center of the Earth ${r}$','Interpreter','latex','FontSize',10);

figure;
yyaxis left
plot(tBar, [eps1; eps2; eps3]); grid on;
title('Constraint Controls','Interpreter','latex');
xlabel('Scaled Time (\tau)','FontSize',16);
ylabel('${\epsilon}$','Interpreter','latex','FontSize',16);
yyaxis right
```

 doi:10.20944/preprints202312.0787.v1


```
plot(tBar, [LgammaBar; LpsiBar]);
legend('${\epsilon_1}$','${\epsilon_2}$','${\epsilon_3}$','$\tilde{L}_\gamma$','$\tilde{L}_\psi$','Interpreter','latex','FontSize', 10);
ylabel('Controls');

figure;
plot(tBar, [eps1; eps2; stat1; stat2]); grid on;
title('Stationarity Condition Check','Interpreter','latex');
xlabel('Scaled Time (\tau)','FontSize',16);
legend('${\epsilon_1}$','${\epsilon_2}$','Stat_1','Stat_2','Interpreter','latex','FontSize',10);
%==========================================================================
%Unscale States, Co-States, Controls
r = rBar.*MS4.constants.R;
theta = thetaBar;
phi = phiBar;
v = MS4.constants.V.*vBar;
gamma = gammaBar;
psi = psiBar;
T = tBar.*MS4.constants.TU;

lam_r = (MS4.constants.TU/MS4.constants.R) .* lam_rBar;
lam_theta = (MS4.constants.TU/MS4.constants.AU) .* lam_thetaBar;
lam_phi = (MS4.constants.TU/MS4.constants.AU) .* lam_phiBar;
lam_v = (MS4.constants.TU/MS4.constants.V) .* lam_vBar;
lam_gamma = (MS4.constants.TU/MS4.constants.AU) .* lam_gammaBar;
lam_psi = (MS4.constants.TU/MS4.constants.AU) .* lam_psiBar;

Lgamma = LgammaBar./(MS4.constants.TU^2/(MS4.constants.R*MS4.constants.M));
Lpsi = LpsiBar./(MS4.constants.TU^2/(MS4.constants.R*MS4.constants.M));

% figure;
% plot(T, [(r.*.0003); v]); grid on;
% title('State Trajectories','Interpreter','latex');
% xlabel('Time ${t}$','Interpreter','latex','FontSize',16);
% ylabel('States $\mathbf{{x}}$','Interpreter','latex','Fontsize',16);
% legend('${r}$','${v}$','Interpreter','latex','FontSize',16);

figure;
plot(T, [r; v; theta; phi; gamma; psi]); grid on;
title('State Trajectories','Interpreter','latex');
xlabel('Time $(t)$','Interpreter','latex','FontSize',16);
ylabel('States $\mathbf{{x}}$','Interpreter','latex','FontSize',16);
legend('${r}$','${v}$','${\theta}$','${\phi}$','${\gamma}$','${\psi}$','Interpreter','latex','FontSize',10);

figure;
plot(T, [lam_r; lam_theta; lam_phi; lam_v; lam_gamma; lam_psi]); grid on;
title('Co-State Trajectories','Interpreter','latex');
xlabel('Time ${t}$','Interpreter','latex','FontSize',16);
ylabel('Co-States $\mathbf{{\lambda}}$','Interpreter','latex','FontSize',16);
legend('${\lambda_r}$','${\lambda_\theta}$','${\lambda_\phi}$','${\lambda_v}$','${\lambda_\gamma}$','${\lambda_\psi}$','Interpreter','latex','FontSize',10);
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1


```
figure;
plot(T, [Lgamma; Lpsi]); grid on;
title('Controls');
legend('${L_\gamma}$','${L_\psi}$','Interpreter','latex','FontSize',10);
xlabel('Time ${t}$','Interpreter','latex','FontSize',10);
%=========================================================================
%Individual State Comparisons
figure;
yyaxis left
plot(T,r);
xlabel('Time (sec)'),ylabel('${r}$','Interpreter','latex');
yyaxis right
plot(T,lam_r);
ylabel('${\lambda_r}$','Interpreter','latex')
title('${r}$ vs ${\lambda_r}$','Interpreter','latex')

figure;
yyaxis left
plot(T,theta);
xlabel('Time (sec)'),ylabel('${\theta}$','Interpreter','latex');
yyaxis right
plot(T,lam_theta);
ylabel('${\lambda_\theta}$','Interpreter','latex')
title('${\theta}$ vs ${\lambda_\theta}$','Interpreter','latex')

figure;
yyaxis left
plot(T,phi);
xlabel('Time (sec)'),ylabel('${\phi}$','Interpreter','latex');
yyaxis right
plot(T,lam_phi);
ylabel('${\lambda_\phi}$','Interpreter','latex')
title('${\phi}$ vs ${\lambda_\phi}$','Interpreter','latex')

figure;
yyaxis left
plot(T,v);
xlabel('Time (sec)'),ylabel('${v}$','Interpreter','latex');
yyaxis right
plot(T,lam_v);
ylabel('${\lambda_r}$','Interpreter','latex')
title('${v}$ vs ${\lambda_v}$','Interpreter','latex')

figure;
yyaxis left
plot(T,gamma);
xlabel('Time (sec)'),ylabel('${\gamma}$','Interpreter','latex');
yyaxis right
plot(T,lam_gamma);
ylabel('${\lambda_\gamma}$','Interpreter','latex')
title('${\gamma}$ vs ${\lambda_\gamma}$','Interpreter','latex')
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1


```
figure;
yyaxis left
plot(T,psi);
xlabel('Time (sec)'),ylabel('${\psi}$','Interpreter','latex');
yyaxis right
plot(T,lam_psi);
ylabel('${\lambda_\psi}$','Interpreter','latex')
title('${\psi}$ vs ${\lambda_\psi}$','Interpreter','latex')

% %=========================================================================
% Feasibility Check
t_vec = T(:);
[T_feas, X_feas] = ode45(@(t,y)stateDynamics(t,y,t_vec,[Lgamma; Lpsi]),[t_vec(1) t_vec(end)],[r(1);theta(1);phi(1);v(1);gamma(1);psi(1)]);
r_feas = X_feas(:,1);
theta_feas = X_feas(:,2);
phi_feas = X_feas(:,3);
v_feas = X_feas(:,4);
gam_feas = X_feas(:,5);
psi_feas = X_feas(:,6);

figure;
hLines = plot(T_feas,(r_feas./1e4),'.-',T_feas,theta_feas,'.-',T_feas,phi_feas,'.-',T_feas,v_feas,'.-',T_feas,gam_feas,'.-',T_feas,psi_feas,'.-'); hold on; grid on;
hMarkers4 = plot(t_vec,[r./1e4; theta; phi; v; gamma; psi],'o');
set(hLines,'MarkerSize',12);
set(hMarkers4,'LineWidth',1);
title('Control-Propagated States','Interpreter','latex');
xlabel('Time ${t}$','Interpreter','latex','FontSize',16);
ylabel('States $\mathbf{{x}}$','Interpreter','latex','FontSize',16);
legend('${r}$','${\theta}$','${\phi}$','${v}$','${\gamma}$','Interpreter','latex');


%=========================================================================
%% FEASIBILITY CHECK FUNCTION
function dXdt = stateDynamics(t,X,tData,uData,MS4)

Lg_t = interp1(tData,uData(1,:),t,'pchip');

Lp_t = interp1(tData,uData(2,:),t,'pchip');



r = X(1);

theta = X(2);

phi = X(3);

v = X(4);
```

 doi:10.20944/preprints202312.0787.v1


```
gamma = X(5);

psi = X(6);



drdt = v .* sin(gamma);

dtdt = (v.*cos(gamma).*sin(psi))./(r.*cos(phi));

dpdt = (v.*cos(gamma).*cos(psi))./r;

dvdt = -(.1211/5000)-((3.986004418e5.*sin(gamma))./(r.^2))+(1.7461/5000);

dgdt = (Lg_t./(5000.*v))-((3.9860044815e5.*cos(gamma))./((r.^2).*v))+((v./r).*cos(gamma));

didt = (Lp_t./(5000.*v.*cos(gamma)))+((v./r).*cos(gamma).*sin(psi).*tan(phi));



dXdt = [drdt; dtdt; dpdt; dvdt; dgdt; didt];
end
```

C.1. Dynamics file

All appendix sections must be cited in the main text. In the appendices, Figures, Tables, etc. should be labeled starting with "A"—e.g., Figure A1, Figure A2, etc.

```
function dxdtBar = MS4Dynamics(primal)
%Missile from Geosynchronous Orbit to target on Earth in minimum time

%=========================================================================

[rBar, thetaBar, phiBar, vBar, gammaBar, psiBar, LgammaBar, LpsiBar, tBar, ...

r0Bar, theta0Bar, phi0Bar, v0Bar, gamma0Bar, psi0Bar, t0Bar, ...

rfBar, thetafBar, phifBar, vfBar, gammafBar, psifBar, tfBar, ...

uBar, TBar, mBar, DBar, Cd, S, R, V, kq, FU, DU, KU] ...

= MS4Preamble(primal);


%Equations of Motion

rdotBar = vBar.*sin(gammaBar);
thetadotBar = (vBar.*cos(gammaBar).*sin(psiBar))./(rBar.*cos(phiBar));
phidotBar = (vBar.*cos(gammaBar).*cos(psiBar))./rBar;
[T, a, P, rho] = atmosisa((rBar.*R.*1000)-(R*1000));%Unscale altitude and change to meters
rhoBar = rho./DU; %Scale Density by Density Unit Scale Factor
SBar = S/(R^2); %Scale ReferenceArea
Dscale = .5.*rhoBar.*vBar.^2*Cd*SBar; %Scaled Drag
vdotBar = -(DBar/mBar)-((uBar.*sin(gammaBar))./(rBar.^2))+(TBar./mBar);
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 December 2023 doi:10.20944/preprints202312.0787.v1


```
    gammadotBar = (LgammaBar./(mBar.*vBar))-
((uBar*cos(gammaBar))./(rBar.^2.*vBar))+((vBar./rBar).*cos(gammaBar));
    psidotBar =
(LpsiBar./(mBar.*vBar.*cos(gammaBar)))+((vBar./rBar).*cos(gammaBar).*sin(psiBar).*tan(phiBar));


    dxdtBar = [rdotBar;

    thetadotBar;

    phidotBar;

    vdotBar;

    gammadotBar;

    psidotBar];


    %eof
    C.1. Cost file
    function [EndpointCost, RunningCost] = MS4Cost(primal)
    %Missile from Geosynchronous Orbit to target on Earth in minimum time
    %=====================================================================
    [rBar, thetaBar, phiBar, vBar, gammaBar, psiBar, LgammaBar, LpsiBar, tBar, ...

    r0Bar, theta0Bar, phi0Bar, v0Bar, gamma0Bar, psi0Bar, t0Bar, ...

    rfBar, thetafBar, phifBar, vfBar, gammafBar, psifBar, tfBar, ...

    uBar, TBar, mBar, DBar, Cd, S, R, V, kq, FU, DU, KU] ...

    = MS4Preamble(primal);
    EndpointCost = tfBar;
    RunningCost = 0;
    %eof
    C.1. Events file
    function endpointFunction = MS4Events(primal)
    %Missile from Geosynchronous Orbit to target on Earth in minimum time
    %Endpoint File
    %=====================================================================

    [rBar, thetaBar, phiBar, vBar, gammaBar, psiBar, LgammaBar, LpsiBar, tBar, ...

    r0Bar, theta0Bar, phi0Bar, v0Bar, gamma0Bar, psi0Bar, t0Bar, ...

    rfBar, thetafBar, phifBar, vfBar, gammafBar, psifBar, tfBar, ...

    uBar, TBar, mBar, Cd, S, R, V, kq, FU, DU, KU] ...

    = MS4Preamble(primal);
```

 doi:10.20944/preprints202312.0787.v1


```
endpointFunction = zeros(9,1);

%Beginnning Enpoints
endpointFunction(1) = r0Bar;
endpointFunction(2) = theta0Bar;
endpointFunction(3) = phi0Bar;
endpointFunction(4) = v0Bar;
endpointFunction(5) = gamma0Bar;
endpointFunction(6) = psi0Bar;

%Final Endpoints
endpointFunction(7) = rfBar;
endpointFunction(8) = thetafBar;
endpointFunction(9) = phifBar;
```

C.1. Path file

```
function H = MS4Path(primal)
%Missile from Geosynchronous Orbit to target on Earth in minimum time
%---------------------------------------
% Call preamble and load primal variables:
%---------------------------------------
[rBar, thetaBar, phiBar, vBar, gammaBar, psiBar, LgammaBar, LpsiBar, tBar, ...

r0Bar, theta0Bar, phi0Bar, v0Bar, gamma0Bar, psi0Bar, t0Bar, ...

rfBar, thetafBar, phifBar, vfBar, gammafBar, psifBar, tfBar, ...

uBar, TBar, mBar, DBar, Cd, S, R, V, kq, FU, DU, KU] ...

= MS4Preamble(primal);
%=========================================================
% path constraint function:
H(1,:) = LgammaBar;
H(2,:) = LpsiBar;

h = (rBar.*R.*1000);
rho0 = 1.225; %Sea level density in kg/m^3
Hs = 7500; %Height scale
rp = R*1000;
rho = rho0*exp(-(h-rp)/Hs);
vd = vBar.*V.*1000;
qdot = kq.*sqrt(rho).*(vd.^3);
H(3,:) = qdot/1e9; %Scale in Engineering Units
```

References

1. Beauchamp, P. Guidance, Navigation, and Control Technology Assessment – Onboard and Ground Navigation and Mission Design. Available online: https://science.nasa.gov/resource/guidance-navigation-and-control-technology-assessment-onboard-and-ground-navigation-and-mission-des/ (accessed 30 November 30, 2023)
2. Shuttle Reentry Streak from Orbit. Available online: https://apod.nasa.gov/apod/ap110801.html (accessed 30 November 2023)
3. NASA Image Use Policy For educational or informational purposes. Available online: https://www.nasa.gov/nasa-brand-center/images-and-media/ (accessed 30 November 30, 2023)

4. Euler, L. (Euler) Formulae Generales pro Translatione Quacunque Corporum Rigidorum (General Formulas for the Translation of Arbitrary Rigid Bodies), Presented to the St. Petersburg Academy on 9 October 1775. and First Published in *Novi Commentarii Academiae Scientiarum Petropolitanae* **1776**, *20*, pp. 189–207 (E478) and was reprinted in *Theoria motus corporum rigidorum*, ed. nova, **1790**, pp. 449–460 (E478a) and later in his collected works *Opera Omnia*, Series 2, Volume 9, pp. 84–98. Available online: https://math.dartmouth.edu/~euler/docs/originals/E478.pdf (accessed on 24 October 2021).
5. Thompson, W.; Tait, P.G. *Elements of Natural Philosophy*; Cambridge University Press: Cambridge, UK, 1872.
6. Reuleaux, F.; Kennedy Alex, B.W. *The Kinematics of Machinery: Outlines of a Theory of Machines*; Macmillan: London, UK, 1876; Available online: https://archive.org/details/kinematicsofmach00reuluoft (accessed on 24 October 2021).
7. Wright, T.W. *Elements of Mechanics Including Kinematics, Kinetics and Statics*; D. Van Nostrand Company: New York, NY, USA; Harvard University: Cambridge, MA, USA, 1896.
8. Merz, J.T. *A History of European Thought in the Nineteenth Century*; Blackwood: London, UK, 1903; p. 5.
9. Whittaker, E.T. *A Treatise on the Analytical Dynamics of Particles and Rigid Bodies*; Cambridge University Press: Cambridge, UK, 1904.
10. Whittaker, E.T. *A Treatise on the Analytical Dynamics of Particles and Rigid Bodies*; Cambridge University Press: Cambridge, UK, 1917.
11. Whittaker, E.T. *A Treatise on the Analytical Dynamics of Particles and Rigid Bodies*; Cambridge University Press: Cambridge, UK, 1927.
12. Whittaker, E.T. *A Treatise on the Analytical Dynamics of Particles and Rigid Bodies*; Cambridge University Press: Cambridge, UK, 1937.
13. Church, I.P. *Mechanics of Engineering*; Wiley: New York, NY, USA, 1908; p. 111. [Google Scholar]
14. Wright, T.W. *Elements of Mechanics Including Kinematics, Kinetics, and Statics, with Applications*; Nostrand: New York, NY, USA, 1909.
15. Study, E. ; Delphenich, D.H., Translator; Foundations and goals of analytical kinematics. *Sitzber. d. Berl. Math. Ges.* **1913**, *13*, 36–60. Available online: http://neo-classical-physics.info/uploads/3/4/3/6/34363841/study-analytical_kinematics.pdf (accessed on 14 April 2017).
16. Gray, A. A *Treatise on Gyrostatics and Rotational Motion*; MacMillan: London, UK, 1918; ISBN 978-1-4212-5592-7. (Published 2007).
17. Truesdell C. The present status of the controversy regarding the bulk viscosity of fluids. *Proc. R. Soc.* **1954**, *226*(1164). http://doi.org/10.1098/rspa.1954.0237
18. Fox, G. Methods for Constructing Invariant Amplitudes Free from Kinematic Singularities and Zeros. *Phys. Rev.* **1967**, *157*, 1493.
19. Rose, M.E. *Elementary Theory of Angular Momentum*; John Wiley & Sons: New York, NY, USA, 1957; ISBN 978-0-486-68480-2. (Published 1995).
20. Kane, T.R. *Analytical Elements of Mechanics Volume 1*; Academic Press: New York, NY, USA; London, UK, 1959.
21. Kane, T.R. *Analytical Elements of Mechanics Volume 2 Dynamics*; Academic Press: New York, NY, USA; London, UK, 1961.
22. Fang, A.C.; Zimmerman, B.G. Digital Simulation of Rotational Kinematics; NASA Technical Report NASA TN D-5302; NASA: Washington, DC, USA, October 1969. Available online: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19690029793.pdf (accessed on 24 October 2021).
23. Henderson, D.M. Euler Angles, Quaternions, and Transformation Matrices—Working Relationships; As *NASA Technical Report NASA-TM-74839*; July 1977. Available online: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19770024290.pdf (accessed on 24 October 2021).
24. Henderson, D.M. Euler Angles, Quaternions, and Transformation Matrices for Space Shuttle Analysis; Houston Astronautics Division as *NASA Design Note 1.4-8-020*; 9 June 1977. Available online: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19770019231.pdf (accessed on 24 October 2021).
25. Hughes, P.C. *Spacecraft Attitude Dynamics*; J. Wiley: Hobken, NJ, USA, 1986; Chapter 1.
26. Shuster, M.D. A Survey of Attitude Representations. *J. Astronaut. Sci.* **1993**, *41*(4), 439–517.
27. Bach, R.; Paielli, R. Linearization of Attitude-Control Error Dynamics. *IEEE Trans. Autom. Control* **1993**, *38*(10), 1521–1525.
28. Mortari, D. The Attitude Error Estimator. In Proceedings of the International Conference on Dynamics and Control of Systems and Structures in Space, Cambridge, UK, 14–18 July 2002.
29. Markley, F.L. Attitude Error Representations for Kalman Filtering. *J. Guid. Control Dyn.* **2003**, 26(2), 311–317.
30. Tanygin, S. Projective Geometry of Attitude Parameterizations with Applications to Control. *J. Guid. Control Dyn.* **2013**, *36*(5), 656–666.
31. Smeresky, B.; Rizzo, A.; Sands, T. Kinematics in the Information Age. *Mathematics* **2018**, *6*(9), 148.
32. Bani Younes, A.; Mortari, D. Derivation of All Attitude Error Governing Equations for Attitude Filtering and Control. *Sensors* **2019**, *19*(21), 4682.

33. Hall, D.; Sands, T. Vehicle Directional Cosine Calculation Method. *Vehicles* **2023**, *5*(1), 114-132.
34. Pontryagin, L.S.; Boltyanski, V.G.; Gamkrelidze, R.S.; Mishchenko, E.F. *The Mathematical Theory of Optimal Processes*; CRC Press: Boca Raton, FL, USA, 1962.
35. Sandberg, A.; Sands, T. Autonomous Trajectory Generation Algorithms for Spacecraft Slew Maneuvers. *Aerospace* **2022**, *9*(3), 135.
36. Ross, M. *A Primer on Pontryagin's Principle in Optimal Control*, Second Edition. Collegiate Publishers: San Francisco, USA, 2015.
37. Raigoza, K.; Sands, T. Autonomous Trajectory Generation Comparison for De-Orbiting with Multiple Collision Avoidance. *Sensors* **2022**, *22*(18), 7066.
38. GoPro Awards: On a Rocket Launch to Space. Available online: https://www.youtube.com/watch?v=bDoh8zQDT38 (accessed 30 November 2023)
39. Youtube image use policy, "In the United States, works of commentary, criticism, research, teaching, or news reporting may be considered fair use." Available online: https://support.google.com/youtube/answer/9783148?hl=en (accessed 30 November 30, 2023)
40. An illustration of the European ATV-2 re-entry and breakup because of the atmospheric forces of Earth. Available online: https://www.nbcnews.com/id/wbna43496909 (accessed 30 November 30, 2023)
41. ESA Image Use Policy, "You may use ESA images or videos for educational or informational purposes". Available online: https://www.esa.int/ESA_Multimedia/Copyright_Notice_Images (accessed 30 November 2023).
42. Walton, R.; Goehlich, R. SPACE CARGO: Ultra-fast Delivery on Earth –Potential of Using Suborbital Space Vehicles for the Transportation of Cargo. *International Journal of Aviation, Aeronautics, and Aerospace* **2022**, *9*(1).
43. DIDO Optimal control software. Available online: (https://www.mathworks.com/products/connections/product_detail/dido.html) accessed 16 February 2023
44. Morrison, D.D.; Riley, J.D.; Zancanaro, J.F. Multiple shooting method for two-point boundary value problems. *Commun. ACM* **1962**, *5*, 613–614.
45. Bialecki, Bernard. "Sinc-Collection Methods for Two-Point Boundary Value Problems." Ima Journal of Numerical Analysis 11 (1991): 357-375.
46. The MathWorks, Inc. (2022). "DIDO Optimal Control Software". MATLAB version: 9.13.0 (R2022b). Available online: https://www.mathworks.com/products/connections/product_detail/dido.html.