

# Optimal Ascent/Reentry/Landing Trajectory Scheme for an Atmospheric Measuring UAV\*

Jeffrey A. Mays

*Department of Mechanical and Aerospace Engineering  
Naval Postgraduate School*

*1 University Circle, Monterey, CA 93943, USA*

mays.jeff20@gmail.com

**Abstract** – The selected problem for this paper focuses on the optimal ascent/reentry/landing trajectory for a 6 degree-of-freedom vertically integrated vertical take-off and landing unmanned aerial vehicle. The vehicle’s purpose is to optimally enter an upper atmospheric corridor at a predefined desired state for applications spanning from atmospheric measurement sweeping, surveillance, mapping, defense, etc. The primary use-case noted in this paper will be to measure the atmosphere with a suite of sensors by simply flying through the corridor of interest. Once the atmospheric corridor has been successfully flown through, the vehicle must optimally guide itself back to the landing pad, where it will softly land, allowing fast-paced reusability of the system. This paper will define the system, break up the full trajectory into individual optimal control problems, solve the problems while also checking for feasibility/optimality, and conclude with the resulting set of trajectory solutions that meets the requirements outlined in the paper.

**Index Terms** – UAV, Trajectory Optimization, Ascent, Reentry, Landing.

## I. INTRODUCTION

For many decades, radiosondes have been a widely used tool for scientists, researchers, engineers, and even hobbyists to measure characteristics of the atmosphere. These radiosondes are battery-powered telemetry recording devices with a suite of sensors that are passed through the atmosphere, typically by a weather balloon. They measure altitude, pressure, temperature, humidity, and wind velocity and are typically tracked as they ascend. In most cases, measurements during ascent are telemetered to a ground station, but can also be stored onboard for recovery by a recovery crew. Once the radiosonde reaches a certain altitude, due to various atmospheric conditions, the weather balloon either pops or loses pressure, causing the radiosonde to begin its descent back to the Earth. Radiosondes are very powerful tools, not only because they are able to measure the atmosphere (as compared to a theoretical mathematical model) but they are also usually inexpensive and cheap. However, there are some downsides to radiosondes that make them not the most optimal tool to measure the atmosphere.

Radiosondes are open-loop systems, and are characterized by the environment they are released in. Normally, a radiosonde is released on the ground while attached to a helium filled weather balloon. While simple, the weather balloon will ascend

at a rate relative to its buoyancy, meaning radiosondes can be slow to ascend, and their rates and lateral displacement can vary widely throughout flight. Furthermore, a radiosonde will only measure the atmosphere the system itself passes through. In many cases, scientists and researchers desire an atmospheric measurement that fits within a linear set of nodes or a predefined space, but this is not an achievable measurement from a radiosonde. Over the course of its flight, it may laterally travel 50+ miles before reaching the desired altitude. Because of this lack of control, typically it is common to interpolate the radiosonde measurements with those of weather models to help quantify error bounds [1], which is not optimal in data assimilation.

As previously noted, radiosondes’ ascent can vary. Typically, their ascent rates are between 250-350 meters per minute. At face value, this might seem relatively fast. However, when attempting to measure a large sweep of an atmosphere spanning thousands of meters (tens of miles), it can take a significantly long time to measure, meaning an actively changing atmosphere may change state by the time the radiosonde flies through the point of interest, assuming it is even able to achieve that.

Furthermore, radiosondes, while cheap, are rarely recoverable and usually fly so far away (tens of miles) from the release point that telemetered signals can be lost, and onboard equipment unreturned. This leads to pollution of plastics, electronics, and rubber materials, with the radiosonde and its balloon ending up anywhere the wind may have taken it. The radiosonde itself is also descending from its respective apogee, which could land in a populated area.

It is possible for aerial drones to complete a similar profile as a radiosonde, but with a controllability advantage. A drone can be controlled and theoretically told to fly a predetermined ascending profile through a region. The drones could even fly through specific node points used in a weather forecast model to help validate the forecast and its current/future predictions. Furthermore, while an aerial drone is not as cheap as the radiosonde, a drone would be recoverable, meaning each radiosonde isn’t a blank wasted cost for a singular (slow and inaccurate) measurement of the atmosphere.

There are already companies working on using drones to measure weather. A company called Meteomatics creates quadrotors used to regularly measure low to mid atmospheric

---

\* This work is in support of the 299 Applied Trajectory Optimization Certificate as required for the AE/ME4881 Trajectory Planning & Guidance course.

profiles [2]. Their business model centers around measuring the atmosphere wherever needed, and at a higher fidelity than most weather models. Another company, Area-I, created a fixed-wing style drone to record atmospheric information in areas not easily accessible or unsafe. A recent use of one of the drones was sent into hurricane Ian to capture atmospheric data [3]. These two examples are just a few of the practical use cases for drones to carry out atmospheric monitoring that would give the user an advantage over a conventional radiosonde. Likewise, to the two examples, this paper will focus on developing a drone that will be able to achieve a similar result as the examples provided, but while also applicable to the course this paper is constructed within.

Given the use case and examples of other companies that measure the atmosphere with drones, why is trajectory optimization required? An optimal trajectory is required in this project due to the flight envelope and vehicle design. Manually designing an ascent and reentry guidance trajectory under the necessary constraints by hand is highly infeasible, especially when nonlinear dynamics come into play. Optimization allows for aspects to be solved governed by the dynamics such as: what is the optimal angle to fly through the desired atmospheric region? How does the vehicle, after flying through said region, reorient itself above a launch-landing pad (LLP)? How will the vehicle nullify its dynamic rates prior to landing? How can the drone be sure to meet specific performance requirements, such as flight path angles or desired endpoint conditions? If all requirements are to be met, the only way to solve this problem is to have a well optimized trajectory that not only accomplishes the mission, but also is able to constrain the vehicle during various phases of the mission with the best chance for success. Without optimization, the problem would likely either be infeasible, or need to be drastically simplified.

The purpose of this paper is to apply Trajectory Optimization to a practical and probable real-world application in support of the ME/AE4881 Trajectory Planning & Guidance course. The selected problem for this paper focuses on the development and guidance & control of a multi-phase optimal outer-loop guidance trajectory for a 6 degree of freedom (DOF) vertically integrated vertical take-off and landing (VTOL) unmanned aerial vehicle (UAV). For this paper, the vehicle's purpose is to optimally enter a predefined upper atmospheric corridor for the purpose of atmospheric measurement sweeping. Once the atmospheric corridor has been successfully flown through, the vehicle must optimally guide itself back to the landing pad, where it will softly land. Since most other applications of this vehicle would minimally require optimally found ascent/reentry/landing trajectories, the specific application of this problem in this paper can be easily expanded to many other applications. This project will focus on the vehicle definition, dynamic modeling, inner/outer loop control architecture, and formal definitions and solutions to the proposed optimal ascent/reentry/landing trajectories.

This project will be solved using a common engineering simulation and development software, MATLAB, as well as an optimal trajectory indirect solver named "DIDO." DIDO, along with various mathematical models of the drone of interest, will

be used to make an *a priori* trajectory that the drone's inner loop controllers will be meant to track based on various control and state constraints defined by the trajectory designer.

## II. SYSTEM OVERVIEW

This section describes a high-level overview of the system and attempts to identify and clarify to the reader how various aspects of the project will be solved.

### A. Vehicle Description

The vehicle in this project is a 6 DOF vertically integrated VTOL UAV and closely mimics a common "launch vehicle." This vehicle does not yet exist, but theoretically could with specific drone related technologies. This paper is not meant to describe in detail the lower-level hardware of the drone, but rather enough context for the reader to sufficiently justify the choices made in the later developed optimal trajectories.

Fig. 1 illustrates the system along with its primary body axes: Center of Mass (COM), and Fabrication (FAB). These two frames are aligned with one another but are in different locations along their  $x$ -axes. The COM frame is centered at the center of mass, whereas the FAB frame is located aft of the vehicle. The  $X$  axis is considered the roll axis, the  $Y$  axis is the pitching axis, and the  $Z$  axis is the yawing axis, respectively.

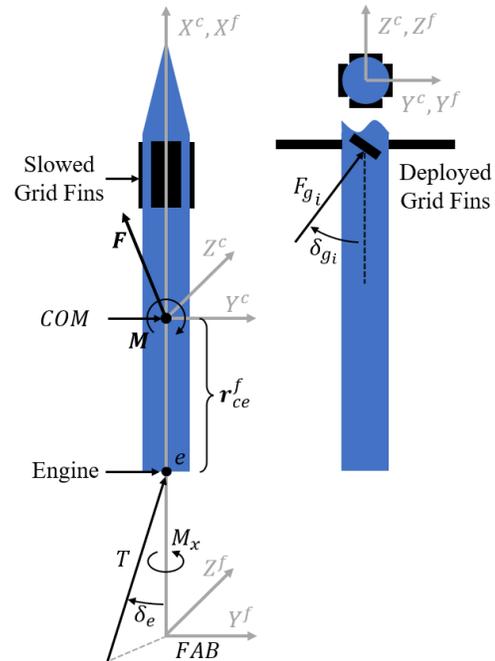


Fig. 1 General vehicle overview

The engine onboard this vehicle is theoretical but could plausibly mimic an electric ducted fan (EDF). This type of engine could produce relative propulsive forces and torques that allows the vehicle to accomplish the mission. With the engine being electrically driven, no propellant mass is consumed, meaning the overall vehicle mass and center of mass are approximately constant throughout flight. Furthermore, the thrust vector produced by this vehicle can be vectored from an

attached gimbal joint, similarly to how a traditional vertically integrated launch vehicles vector thrust. This forms the bases for one of the two major effectors that will allow the vehicle to accomplish the desired mission.

In the descent phase of flight, rather than using the engine to manipulate the trajectory, grid fins are deployed as the secondary effectors on this vehicle. These grid fins not only act as air brakes, which help reduce the descent rate of the vehicle, but also give the vehicle aerodynamic maneuverability during unpowered engine flight phases while the dynamic pressure is not appreciable. The deployed grid fins also act to help shift the aerodynamic center of the vehicle forward/aft of the center of mass, depending on the incoming freestream air. This is an important aspect in ensuring the system remains aerodynamically stable under different flight envelopes.

Table I lists out a general overview of physical characteristics for this hypothetical vehicle. Note that all the following parameters are not necessarily applicable to the later developed trajectories.

TABLE I  
VEHICLE PARAMETERS

Parameter	Rep.	Value	Unit
Mass	$m$	2	kg
Total Length	$l_{ref}$	1.5	m
Diameter	$d_{ref}$	0.1	m
Ixx (roll) Inertia	$I_{xx}^c$	0.01	kg-m <sup>2</sup>
Iyy (pitch) Inertia	$I_{yy}^c$	0.15	kg-m <sup>2</sup>
Izz (yaw) Inertia	$I_{zz}^c$	0.15	kg-m <sup>2</sup>
Eng. Moment Arm	$r_{ce}^c$	$[0.75 \ 0 \ 0]^T$	m
Grid Fin Moment Arm	$r_{cg}^c$	$[0.75 \ 0 \ 0]^T$	m
Engine Max Thrust	$T_{max}$	39.24	N
Engine Min Control Thrust	$T_{min,c}$	5	N
Engine Min OFF Thrust	$T_{min,off}$	0	N
Gimbal Max Tilt Angle	$\delta_e$	6	deg
Roll fractional	$\delta_{rf}$	-0.025	--

**B. Ascent Trajectory Overview**

The ascent phase of this UAV can be broken up into four sub-trajectories: Liftoff, Ascent, Corridor, and Coast. Fig. 2 illustrates these four trajectories, also referred to as segments. Note that these segments will be described in greater detail later in the paper as optimal control problems.

The **Liftoff** segment is an initial segment to allow the system to achieve some non-zero airspeed prior to commanding transient behavior. During this time, the guidance and control architecture is attempting to maintain a perfectly vertical vector that is normal to the center of the LLP. This segment is hardcoded to end at a set time of 1 second. Given the system properties, the Liftoff trajectory will end at  $\sim 5 \text{ m}$  above the ground while traveling  $\sim 10 \text{ m/s}$  vertically.

The **Ascent** trajectory is an *a priori* guidance trajectory that is optimized to guide the vehicle, based on the dynamics of the system, from the end of the Liftoff trajectory to the beginning of the Corridor trajectory. Based on the desired Corridor entrance criteria defined by the user, this optimal trajectory will manipulate the vehicle’s state and engine control vector such that all criteria are met.

The **Corridor** trajectory is a vertically oriented linear trajectory meant to span a segment of atmosphere. The corridor

has a particular set of path requirements that are defined via the spanned altitude. The Ascent trajectory is expected to feed the Corridor segment with the desired position and velocity vector. Once in the Corridor segment, guidance & control will attempt to stay within a desired region during the vehicle’s ascent, eventually ending with a vehicle exit of the pre-defined corridor.

The **Coast** trajectory is the last phase of ascent. Its purpose is to simply allow the vehicle to reach its ballistic apogee. No control effectors are used in this segment. The Coast trajectory ends at apogee where vertical velocity is zero.

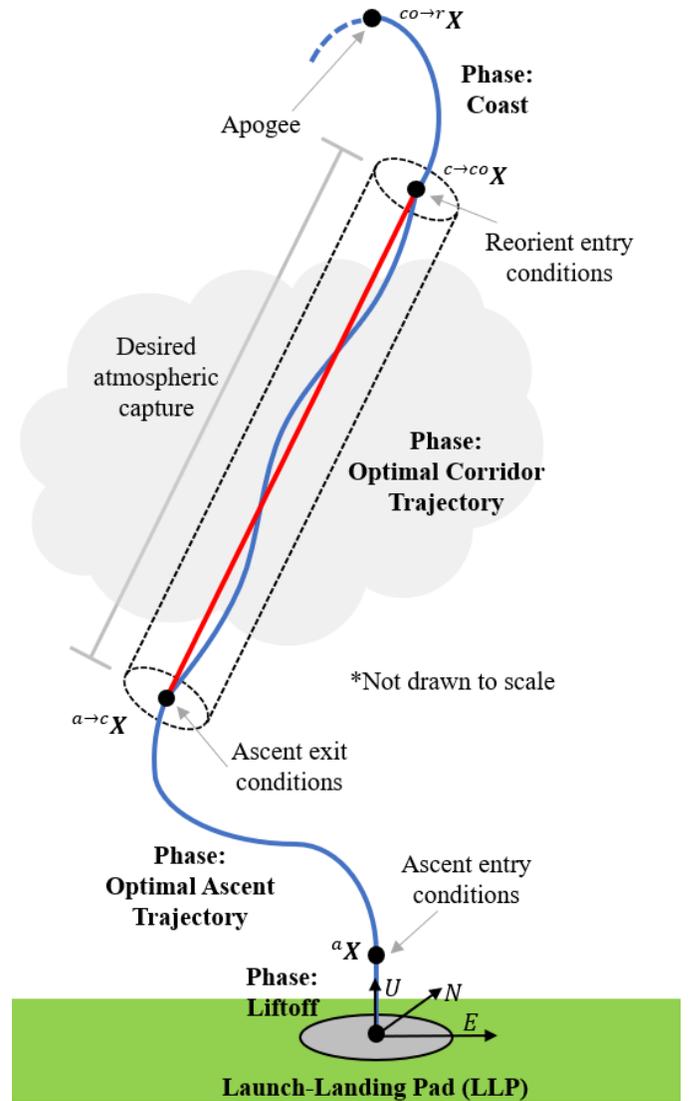


Fig. 2 Ascent trajectory overview

**C. Descent Trajectory Overview**

The descent phase of this UAV can be broken up into three trajectories: Reentry, Landing, and T=W. Fig. 3 illustrates these three segments.

The **Reentry** trajectory is first trajectory that contains a negative vertical velocity with respect to (w.r.t.) the UEN

frame. Its sole purpose is to reorient the vehicle above the LLP for handoff to the Landing trajectory. It accomplishes this by deploying the four forward section grid fins and using them to produce aerodynamic forces that augment the outcome of the vehicle dynamics such that the trajectory’s endpoint constraints are met. The extension of the grid fins at the of the vehicle act as both aerodynamically stabilizing wings (pull the aerodynamic center aft of the center of mass w.r.t. the freestream) and as control effectors to allow the vehicle to have full 6DOF control during Reentry. During this trajectory, the engine is disabled and not used.

The **Landing** segment is the segment that looks to nullify the vertical and lateral dynamics and allow the vehicle to enter the final segment at a pre-defined state. During this trajectory, the engine is reenabled and grid fin control is disabled. However, the grid fins are still deployed to aid in the slowing down of the vehicle. This segment also has a defined glide slope constraint, which forces the vehicle to remain in a conical space prior during the entire trajectory. The apex of the cone coincides with the start of the final segment explained in the next paragraph.

The **T=W**, or *thrust equals weight*, segment is the final segment of descent. It is a predefined segment meant to provide additional margin for the full integrated system prior to touchdown. The desire is for the vehicle to land perfectly vertical without any lateral velocity and a “slow” vertical velocity such that the landing gear will not exceed their structural tolerances at touchdown. The T=W trajectory should aid in the reliability of this occurring.

D. System Architecture Overview

Based on the prior sections, there needs to be a system in place to make the true vehicle applicable to fly the developed outer loop trajectories. Fig. 4 pictorially illustrates the system overview, spanning from how the user interacts with the vehicle, to how the avionics and physical vehicle interacts with the real world. Note that  $X$  corresponds to the true system state,  $\hat{X}$  are the estimated sensor states based on the true state, and  $\delta$  are low-level actuator control commands.

Using the Integrated Ground System (IGS), the user can manipulate the vehicle while powered on. At power up, the system is in an idle state and sitting on the LLP, where the vehicle is passively waiting for commands. The user is then able to upload a guidance trajectory to the vehicle’s internal memory, accessible from the vehicle’s Main Flight Computer (MFC). This uploaded guidance trajectory is created from an optimization routine (either before or on launch day) that considers constraints and desired performance of the vehicle set by the user. Once this trajectory is uploaded to the vehicle, the user can designate when the launch should occur.

The launch occurs through a user command to the vehicle’s MFC Sequencer software capability (SWC). The SWC aids not only in sequencing for a launch attempt, but also in helping the vehicle know which phase of flight it is in such that it can enable/disable other SWC. The sequencer is actively used while the vehicle is powered on, and aids in operation from *prior to liftoff* to *post touchdown*.

Once launched, the guidance and control SWC within the

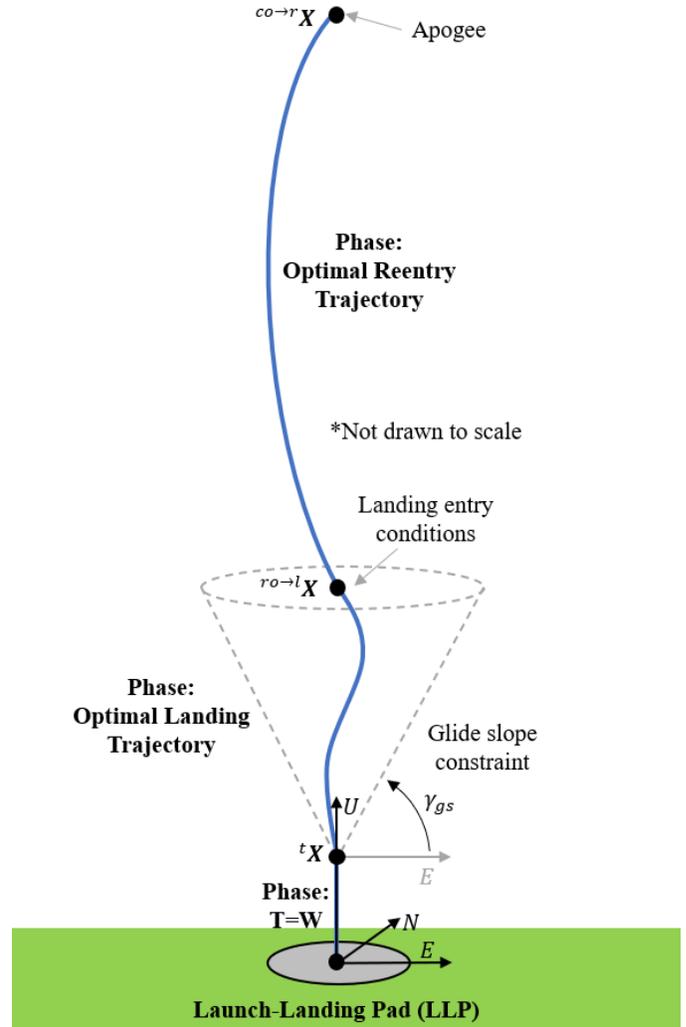


Fig. 3 Descent trajectory overview

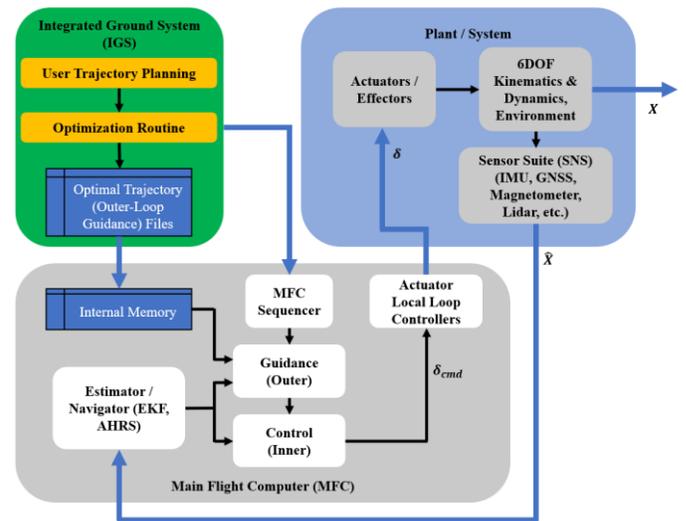


Fig. 4 System architecture overview

MFC reads both the desired optimal trajectory as well as the Estimator / Navigator SWC, which is derived from sensed states driven by the vehicle's true state. The guidance and control act as outer and inner cascaded control loops which augment the actuators to cause the vehicle to follow the desired state planned out by the uploaded trajectory. Ultimately, the inner loop sends commands to the actuator local-loop controllers, which are then sent to the actual actuators on the vehicle. The guidance and control architectures are comprised of proportional-derivative controllers and are gain scheduled depending on the phase of flight.

The plant is the true system, and this contains the plant actuators, sensor models, and 6DOF dynamics of the vehicle. The only interaction between the MFC and the true system (plant) is the commands sent to the actuators, and the sensed sensor states from the true sensors. By augmenting the actuators via the prior cascaded control loops, the dynamics are manipulated and thus cause a difference in the system's propagation through time, in turn changing the required guidance and control signals. This architecture mimics a classic control loop used to manifest a dynamical system to a desired setpoint.

It should be noted that while the real system would have sensors paired with an optimal estimator, this project's focus is not concerned with the estimation of the vehicle states (as depicted in the Estimator / Navigator block) but rather the trajectory optimization and outer loop guidance of the system. Therefore, for the present study, all sensed information required by the guidance and flight control systems is assumed to be perfectly known. No sensor dynamics, deterministic errors or noises are modelled, although the full vehicle integrated simulation would contain provisions for their inclusion.

### III. DYNAMIC MODELING

This section details the dynamics and subsystem models that will be used to characterize an optimal trajectory for the specified mission. First the optimization equations of motion will be discussed. Then, the equations that characterize the aerodynamics, engine, and grid fins will be presented.

#### A. Optimization Dynamics

In many modern applications of generating optimal control trajectories, it is common practice to take higher order "high-fidelity" (HiFi) modeled dynamics and break them down into a simplified "low-fidelity" (LoFi) model. This renders an easier and computationally cheaper solution while still retaining enough fidelity to mimic the HiFi system within the respective flight envelop. This is especially crucial when the optimal control problem is to be solved in real time, in which case likely drastic measures are taken to simplify the system (double integrator assumption, flattened dynamics, ignored aerodynamics, etc.). For this project, the optimization routine is performed a priori, and therefore does not have any explicit requirements limiting its fidelity from a solution time perspective. However, simplifications will still be made to render a tangible solution without overcomplicating the problem. The intent of this optimal trajectory solution is to

provide the inner loop control systems with a trackable setpoint, rendering explicit attitude and inner loop control commands redundant. Furthermore, higher order dynamics, such as Coriolis and centripetal effects [4], can be safely ignored for this project due to the later defined flight envelop. These elements could be added in the future should they ever produce non-negligible accelerations.

For completeness, before developing any equations or discussing trajectory problems, Table II describes some useful nomenclature for the reader.

TABLE II  
NOMENCLATURE

Variable	Description
$\mathbf{x}$	A bold variable denotes a vector
$x$	A non-bold variable denotes a scalar
ECEF, $e$	Earth-Centered Earth-Fixed
UEN, $u$	Up-East-North
COM, $c$	Center of Mass
$\mathbf{x}_0$	Initial vector
$\mathbf{x}_f$	Final vector
$\mathbf{r}_{ab}^c$	Position vector from $a$ to $b$ expressed in frame $c$ . $\mathbf{r}_{uc}^u$ would mean a position vector in cartesian coordinates from the UEN to the COM, expressed in UEN.
$\mathbf{v}_{ab}^c$	Velocity vector from $a$ to $b$ expressed in frame $c$ .
$\mathbf{v}_{f,ab}^c$	Final state velocity vector from $a$ to $b$ expressed in frame $c$ .
$r_{f,y}^u$	Final east position in the UEN frame
${}^a\mathbf{x}$	A vector that comes from source " $a$ "

As previously stated, the attitude and explicit control trajectories for the inner loop systems are not required for this system. Therefore, rather than a full 6DOF set of dynamical equations that govern the motion of the vehicle though its outlined mission, the problem can be reduced to positional 3DOF, where the vehicle itself is considered a point mass. Assuming we can operate in a quasi-inertial Up-East-North (UEN) frame, acceleration of the vehicle then follows Newton's second law.

$$\mathbf{a}_{ec}^u = \frac{\mathbf{F}^u}{m} + \mathbf{g}^u \quad (1)$$

$\mathbf{a}_{ec}^u$  is the acceleration of COM in the Earth-Centered Earth-Fixed (ECEF) frame expressed in the UEN frame,  $m$  is a scalar mass,  $\mathbf{F}^u$  are all inertial forces expressed in the UEN frame applied to the COM, and  $\mathbf{g}^u$  is the gravitational acceleration of the Earth. Note that this gravitational attraction does not include centripetal effects from the Earth's rotation, and therefore only contains one non-zero component,  $g_x^u$ . The inertial forces in the UEN frame,  $\mathbf{F}^u$ , come from multiple sources and will be described in the next sections. The change in position can be described by

$$\dot{\mathbf{r}}_{uc}^u = \mathbf{v}_{ec}^u - \mathbf{v}_{eu}^u \quad (2)$$

Since the UEN frame is not moving with respect to the rigid Earth,  $\mathbf{v}_{eu}^u$  is assumed zero. The above equations then fully characterize the position, velocity, and acceleration of the

vehicle as described for a point mass. The full state vector is then the following.

$$\mathbf{x} = [\mathbf{v}_{ec}^u \quad \mathbf{r}_{uc}^u]^T$$

These above set of equations are very simple but could be easily expanded into a higher order set of equations by the inclusion of a rotating Earth assumption. However, for the purposes of this project, the above set of dynamics should be more than sufficient to render an applicable optimal trajectory for the system.

For the remainder of this paper, the velocity and position states will mostly be referred to as  $\mathbf{r}^u$  or  $\mathbf{v}^u$  to represent the position or velocity of the COM in the UEN frame. It may also be referred to in its full form as described in Table II. Should any additional super or subscripts be used, they will likely refer to an initial time, such as  $\mathbf{r}_0^u$ , or a final time, such as  $\mathbf{v}_f^u$ . Furthermore, elements of the cartesian state may be directly referred to by their element order of  $(x, y, z)$ , in which case there will be additional subscripts to denote the element within the vector. For example,  $v_{f,z}$  will be assumed to refer to the final COM velocity in the UEN frame, specifically in the North direction since  $z$  is the 3<sup>rd</sup> element of the UEN resolved vector.

### B. Aerodynamics

Aerodynamics are an integral part of any aerial vehicle that must fly through an atmospheric fluid. In many optimal trajectory applications where aerodynamics persists, it is common to simplify the problem down such that the aerodynamics are of lower order, or even ignored. In this project, the aerodynamics will be empirically derived from [5], with the purpose of simply producing a plausible aerodynamic drag force, discussed more later. Should this system ever one day exist, then additional effort can be made to update the model used in this project to depict the system more accurately for the true system.

Based on [5], the aerodynamic tables in Fig. 5 can be derived, where  $C_X$ ,  $C_Y$ , and  $C_Z$  represent aerodynamic coefficients in FAB frame components, with  $C_X$  representing axial drag. Some heuristic work was performed to ensure these values produced feasible aerodynamic forces. A sinusoid was also used to continuously fill gaps between various points of estimation. Aerodynamics are interpolated based on a set of aerodynamic angles along with whether the grid fins are “stowed” or “deployed.”

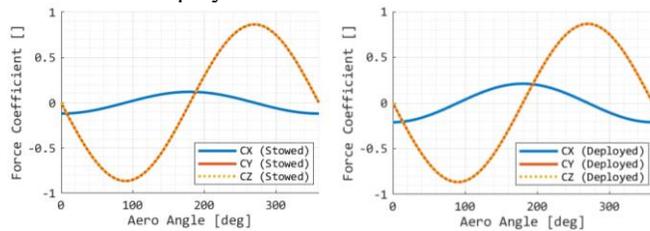


Fig. 5 System architecture overview

As previously discussed, the optimization dynamics are 3DOF in position, and therefore do not possess the ability to apply aerodynamic angles into the system of equations due to a

lack of an attitude state at any given time. Assuming the vehicle never encounters a significant angle of attack for a long period of time, and the nose of the craft, whether it be the nosecone on ascent or the engine exit plane on descent, will always be facing into the freestream vector, lateral body frame aerodynamic forces can be considered zero. Therefore, an aerodynamic angle of zero degrees will be assumed throughout the entirety of the trajectory to drive out the axial aerodynamic force coefficient,  $C_X$ . The only point that  $C_X$  will change value is during the transition from ascent, where the grid fins will be “stowed,” to descent, where the grid fins will be “deployed.” When deployed, the axial drag is increased. Since the aerodynamic drag is a scalar based on total velocity in the UEN frame, its application to the vehicle will be directly opposing to the global UEN velocity vector’s direction.

The force of drag in the UEN frame can be described by

$${}^a\mathbf{F}^u = \frac{[v_x \quad v_y \quad v_z]^T}{\sqrt{v_x^2 + v_y^2 + v_z^2}} * \bar{q} L_{ref} D_{ref} C_X \quad (3)$$

where  $\bar{q}$  is the dynamic pressure, and  $L_{ref}$  and  $D_{ref}$  are the reference length and diameter, which are equivalent to the length and diameter of the vehicle. Expanding, we get

$${}^a\mathbf{F}^u = \frac{[v_x \quad v_y \quad v_z]^T * \left[ \frac{1}{2} \rho L_{ref} D_{ref} C_X (v_x^2 + v_y^2 + v_z^2) \right]}{\sqrt{v_x^2 + v_y^2 + v_z^2}} \quad (4)$$

and then simplifying like terms and separating into their own vector components, we get

$${}^aF_x^u = v_x \left[ \frac{1}{2} \rho L_{ref} D_{ref} C_X \sqrt{v_x^2 + v_y^2 + v_z^2} \right] \quad (5)$$

$${}^aF_y^u = v_y \left[ \frac{1}{2} \rho L_{ref} D_{ref} C_X \sqrt{v_x^2 + v_y^2 + v_z^2} \right] \quad (6)$$

$${}^aF_z^u = v_z \left[ \frac{1}{2} \rho L_{ref} D_{ref} C_X \sqrt{v_x^2 + v_y^2 + v_z^2} \right] \quad (7)$$

Since the  $C_X$  drag coefficient is negative, the above forces will oppose the velocity vector of the vehicle. This completes the equations that define the aerodynamic forces that act on the vehicle for the optimization dynamics.

### C. Engine & TVC

The engine and thrust vector gimbals are both located near the aft portion of the vehicle and are the main source of forces and torques on the vehicle. This theoretical system can produce all axes forces and torques on the system while also not contributing to total mass rate changes to the full vehicle. Fig. 6 describes the sea level thrust lookup table of the engine.

Fig. 6 is used to set the min and max bounds on the thrust control vector path constraint. No dynamical effects of the engine or atmospheric loss is modeled for simplicity. However, this model can be easily expanded with transient dynamics and performance losses of the engine based on the environment in future iterations of this work. This simplicity is acceptable for the optimization dynamics. Fig. 7 illustrates the engine and gimbal system.

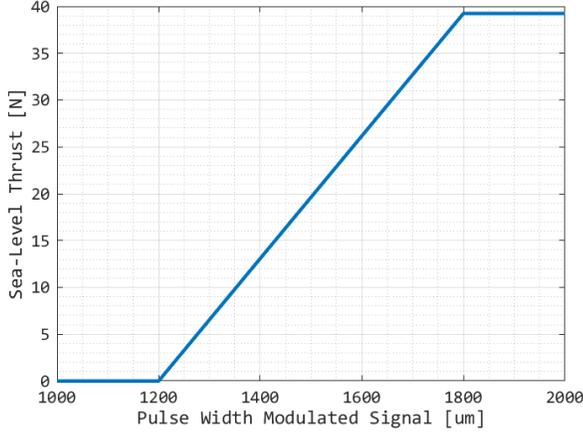


Fig. 6 Engine thrust performance

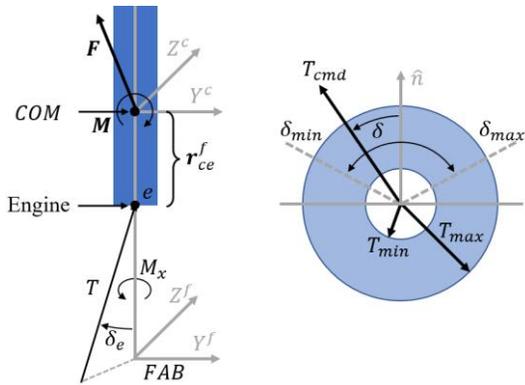


Fig. 7 Engine and gimbal design

For the true system, the engine and gimbal model accept a thrust command,  $T$ , and gimbal attitude commands to provide all axis forces and torques on the vehicle. These commands would be separated out and be heavily coupled with the inner loop controllers of the system. However, for the optimization dynamics of this system, a significant amount of simplification can be made.

Because the optimization dynamics are 3DOF, body moments are no longer relevant. As such, no understanding of attitude can be made either. Therefore, rather than engine and gimbal control commands, the engine model in these optimization dynamics will be resolved to a thrust vector in the UEN frame, with thrust magnitude, tilt, and heading used to characterize the total force vector. This is a drastic simplification but should still be applicable to the full system for the purposes of generating an outer loop optimal trajectory. It can be assumed that whatever constraints posed to the tilt and heading control states, manifest equivalently as similar constraints on the steady state attitude and direction vectors of the true system. The path constraint limit on tilt is more difficult to quantify in this methodology, but it can be assumed through stability, non-linear monte-carlo analysis, and iteration within simulations and analysis that the bounds would be defined

sufficiently small that the full nonlinear system would still be able to track the desired trajectory path. Fig. 8 illustrates the control vector for the engine and gimbal within the UEN frame.

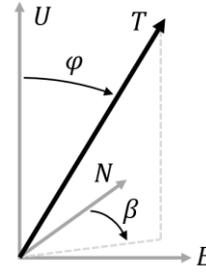


Fig. 8 Thrust, tilt, and heading defined vector

With the engine thrust vector defined via the thrust magnitude, tilt, and heading with respect to the UEN frame, the following equations describe the subsystem's forces.

$${}^e F_x^u = T \cos \varphi \quad (8)$$

$${}^e F_y^u = T \sin \varphi \sin \beta \quad (9)$$

$${}^e F_z^u = T \sin \varphi \cos \beta \quad (10)$$

For this project, it is assumed that a maximum tilt angle of 25 degrees is appropriate for the system of interest. This completes the equations that define the engine and gimbal forces that act on the vehicle for the optimization dynamics.

#### D. Grid Fins

The grid fins are located near the nose of the vehicle and are nominally in a stowed configuration. It is not until the vehicle starts its descent from apogee that the grid fins are deployed. This design choice is to ensure the aerodynamic center is translated aft of the center of mass with respect to the freestream air during the change in direction. This is a mostly passive stabilization system that allows the vehicle to remain upright (zero tilt in the UEN frame) with its engine facing downward during descent. Secondly, the grid fins act as drag brakes that decrease the descent rate due to a large step change in drag.

However, the grid fins on this vehicle are not completely passive, as they are allowed to rotate amongst themselves independently. This is primarily to allow for inner-loop control to make corrections to the descent trajectory, which will be plagued by unmodeled winds and other modeling simplifications. For the optimization dynamics, similarly to how the engine and gimbal were simplified, the grid fin effectors are also drastically simplified. Since the additional axial drag is already modeled within the aerodynamic axial force coefficient, the grid fins can be assumed to only influence the lateral UEN dynamics of the system (east and north). Furthermore, since the grid fins are an aerodynamic effector, their force components are a function of the dynamic pressure experienced at the vehicle. If we assume one control state of the system is resolved in the East-West plane, and another is

resolved in the North-South plane, then the grid fin forces in the UEN frame can be described by

$${}^g\mathbf{F}^u = [0 \quad 1 \quad 1]^T * \bar{q} S_{ref}^g [0 \quad \delta_E \quad \delta_N] \quad (11)$$

where  $\bar{q}$  is the dynamic pressure,  $S_{ref}^g$  is the reference area of the grid fins, and  $[\delta_E \quad \delta_N]$  is the control effect of the system that can accelerate the vehicle in the lateral direction. Note that  $S_{ref}^g$  could be empirically tuned to restrict the setpoint command of the full fidelity grid fins in HiFi simulation. Breaking up the above equation into their respective UEN components yields

$${}^gF_x^u = 0 \quad (12)$$

$${}^gF_y^u = \frac{1}{2} \rho (v_x^2 + v_y^2 + v_z^2) S_{ref}^g \delta_E \quad (13)$$

$${}^gF_z^u = \frac{1}{2} \rho (v_x^2 + v_y^2 + v_z^2) S_{ref}^g \delta_N \quad (14)$$

This completes the equations that define the grid fin forces that act on the vehicle for the optimization dynamics. Note that these effectors are only enabled during the Reentry trajectory and are latched to zero in all other phases of flight.

#### E. Dynamic Modeling Conclusion

The above sections describe the 3DOF dynamics, aerodynamics, engine, and grid fin models as they are applicable to developing a set of optimal trajectories. These models will be combined and used to optimize trajectories with the previously described mission objective. The following is a conclusion of the system state and control vectors used for this project. Note that the full trajectory described in this project is separated into multiple phases, with each phase having different characteristics set by the trajectory designer. Those parameters are left out of the state or control vector seen by the numerical optimizer since they are scheduled based on the trajectory phase and mission timeline but will be described in more detail later.

$$\mathbf{x} = [v_{uc}^u \quad r_{uc}^u]^T \quad (15)$$

$$\mathbf{u} = [T \quad \varphi \quad \beta \quad \delta_E \quad \delta_N]^T \quad (16)$$

$$\mathbf{p} = [\rho \quad C_x \quad L_{ref} \quad D_{ref} \quad S_{ref}^g \quad \mathbf{g}^u]^T \quad (17)$$

The complete equations of motion in the UEN frame, as they are broken up into their subcomponents, are presented below.

$$\dot{v}_x^u = \frac{{}^aF_x^u + {}^eF_x^u}{m} + g_x^u \quad (18)$$

$$\dot{v}_y^u = \frac{{}^aF_y^u + {}^eF_y^u + {}^gF_y^u}{m} \quad (19)$$

$$\dot{v}_z^u = \frac{{}^aF_z^u + {}^eF_z^u + {}^gF_z^u}{m} \quad (20)$$

$$\dot{r}_x^u = v_x^u \quad (21)$$

$$\dot{r}_y^u = v_y^u \quad (22)$$

$$\dot{r}_z^u = v_z^u \quad (23)$$

When developing an optimal control problem, it is common for the optimal control designer to scale the equations of motion such that the numerical solvers are more easily able derive a solution; some problem sets cannot be solved without this step. For the dynamics previously discussed, the states and control parameters were chosen such that they were all similarly sized and not too egregious from one another, which led to future optimal control problem statements that do not require scaling/balancing to be performed. Therefore, scaling in this project can be assumed to be either not performed, or simply scaled by unity.

Now that the system models are described and the optimization dynamics completely formulated, attention can be turned to developing and solving the optimization problems of interest to develop the previously described trajectories.

#### IV. TRAJECTORY FORMULATION AND SOLUTION

This section illustrates and discusses the problem formulations and solutions for this project's various trajectory phases. The trajectories discussed in this section are:

- Liftoff
- Ascent (optimized)
- Corridor (optimized)
- Coast
- Reentry (optimized)
- Landing (optimized)
- T=W

Each of the previously discussed trajectory phases will be discussed as a sub-section. If the trajectory of interest is optimized, it will include the application of Pontryagin's minimization principle, of which nomenclature and convention are heavily derived from [6]. It will also include feasibility checks and Verification & Validation (V&V) of the derived solution. The tool used to computationally solve the optimized problems for each trajectory phase is a computer program called "DIDO." DIDO is a MATLAB based in-direct optimal control collocation solver in which these problem formulations will be implemented. Since the objective of this paper is not to answer *how* to setup these problems in software, information regarding how these problems were formulated in MATLAB is not included.

Note that some trajectories within the entire mission profile are not optimal, but rather acting simply as bridges from one state to the next. For organization and completeness purposes, those trajectories will still be discussed in this section, regardless of whether Pontryagin's minimization principle is required. Note also that many equations will be derived, but then simply re-referenced later to prevent redundant repetitions of the same equation.

##### A. Liftoff Trajectory

The Liftoff trajectory is simply a flight phase that looks to move the vehicle off the LLP; it is not part of the solved optimal control trajectories. For this mission, it was decided that a one second perfectly vertical max thrust control input would

describe the Liftoff trajectory, which finishes at an above ground level altitude of ~5 m, traveling ~10 m/s upward. Fig. 9 illustrates the Liftoff trajectory.

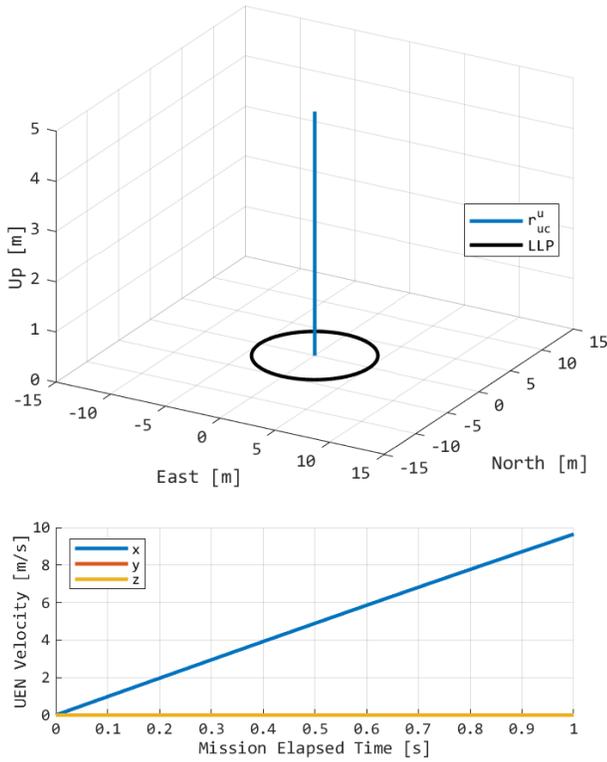


Fig. 9 Liftoff

### B. Ascent Trajectory

The Ascent trajectory is formatted as the following optimal control problem, Optimal Problem 1: Ascent. Note that its initial conditions are taken from the final Liftoff trajectory state. For simplicity, the optimization equations (Eqns. 18-23) will be referred to as

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{p})$$

for the remainder of the document to simplify the reference. For the Ascent trajectory, Table III helps characterize the endpoint constraint on position.

This optimal control problem has a fully defined initial state vector based on the final state of the Liftoff trajectory, along with path constraints on all its control effectors. Furthermore, it has an endpoint constraint inequality stating it must fly through a circular of radius  $R_a$  at the final altitude. This matches the corridor entrance as previously described. The optimal control problem is minimized via time.

The above optimal control problem can be solved through the **HAMVET** method: **H**amiltonian, **A**djoint, **M**inimize, **V**alue, **E**valuation, and **T**ransversality [6].

#### Ascent Hamiltonian

The Hamiltonian is a combination of the running cost, the costate vector, and the dynamics, formulated as

$$H(\boldsymbol{\mu}, \boldsymbol{\lambda}, \mathbf{x}, \mathbf{u}, t) = F(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda}^T f(\mathbf{x}, \mathbf{u}) \quad (24)$$

Optimal Problem 1: Ascent

$$\begin{aligned} & \mathbf{x} \in \mathbb{R}^6, \quad \mathbf{u} \in \mathbb{R}^5 \\ \text{Min.} & \quad J[\mathbf{x}(\cdot), \mathbf{u}(\cdot), t_f] = t_f \\ \text{s.t.} & \quad \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{p}) \\ & \quad t_0 = 0 \text{ s} \\ & \quad \mathbf{v}_0^u = \mathbf{v}_{liftoff}^u \text{ m/s} \\ & \quad \mathbf{r}_0^u = \mathbf{r}_{liftoff}^u \text{ m} \\ & \quad (\mathbf{v}_{f,y}^u, \mathbf{v}_{f,z}^u) = (0, 0) \text{ m/s} \\ & \quad r_{f,x}^u = 50 \text{ m} \\ & \quad 0 \leq (r_{f,y}^u - a_a)^2 + (r_{f,z}^u - b_a)^2 \leq R_a^2 \\ & \quad h_1(\mathbf{u}, t) := 5 \text{ N} \leq T \leq 39.24 \text{ N} \\ & \quad h_2(\mathbf{u}, t) := 0 \text{ deg} \leq \varphi \leq 25 \text{ deg} \\ & \quad h_3(\mathbf{u}, t) := 0 \text{ deg} \leq \beta \leq 360 \text{ deg} \\ & \quad h_4(\mathbf{u}, t) := 0 \leq \delta_E \leq 0 \\ & \quad h_5(\mathbf{u}, t) := 0 \leq \delta_N \leq 0 \end{aligned}$$

TABLE III

PARAMETER DATA FOR THE OPTIMAL ASCENT CONTROL PROBLEM

Parameter	Value
$a_a$	10 m
$b_a$	0 m
$R_a$	2 m
$t^0$	0 s
$C_X$	-0.1202 (stowed)

For this assignment, path constraints exist, which require construction of the Lagrangian of the Hamiltonian for this optimization problem. The additional path constraints are included as part of an  $\mathbf{h}$  function. The Lagrangian of the Hamiltonian is then,

$$\bar{H}(\boldsymbol{\mu}, \boldsymbol{\lambda}, \mathbf{x}, \mathbf{u}, t) = F(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda}^T f(\mathbf{x}, \mathbf{u}) + \boldsymbol{\mu}^T \mathbf{h}(\mathbf{x}, \mathbf{u}) \quad (25)$$

where  $F(\mathbf{x}, \mathbf{u})$  is the Lagrange (running) cost element of the cost function,  $f(\mathbf{x}, \mathbf{u})$  are the system dynamics,  $\boldsymbol{\lambda}^T$  are a vector of co-vectors, and  $\boldsymbol{\mu}^T \mathbf{h}(\mathbf{x})$  is included from the Karush-Kuhn-Tucker (KKT) complementary criterion due to the path constraints. Constructing the Lagrangian of the Hamiltonian based on the previously defined optimal control problem definition, we get the following.

$$F(\mathbf{x}, \mathbf{u}) = 0 \quad (26)$$

$$\boldsymbol{\lambda}^T f(\mathbf{x}, \mathbf{u}, \mathbf{p}) = A + B \quad (27)$$

$$\boldsymbol{\mu}^T \mathbf{h}(\mathbf{u}) = \mu_T T + \mu_\varphi \varphi + \mu_\beta \beta + \mu_{\delta_E} \delta_E + \mu_{\delta_N} \delta_N \quad (28)$$

The  $A$  and  $B$  components from Eqn. 27 are

$$A = \lambda_{v_x} \left( \frac{{}^a F_x^u + {}^e F_x^u}{m} + g_x^u \right) + \lambda_{v_y} \left( \frac{{}^a F_y^u + {}^e F_y^u + {}^g F_y^u}{m} \right) + \lambda_{v_z} \left( \frac{{}^a F_z^u + {}^e F_z^u + {}^g F_z^u}{m} \right)$$

$$B = \lambda_{r_x} v_x + \lambda_{r_y} v_y + \lambda_{r_z} v_z$$

This concludes the construction of our Hamiltonian and Lagrangian of the Hamiltonian.

### Ascent Hamiltonian Minimization

Performing the Hamiltonian Minimization Condition (HMC) involves a box-constrained minimization problem on the Hamiltonian. Taking the partial derivative of the Lagrangian of the Hamiltonian with respect to its control vectors and equating that partial derivative to 0 along with involving the stationery and complementarity conditions gives us the following.

$$\frac{\partial \bar{H}}{\partial T} = 0 = \lambda_{v_x} \left( \frac{\cos \varphi}{m} \right) + \lambda_{v_y} \left( \frac{\sin \varphi \sin \beta}{m} \right) + \lambda_{v_z} \left( \frac{\sin \varphi \cos \beta}{m} \right) + \mu_T$$

$$\frac{\partial \bar{H}}{\partial \varphi} = 0 = \lambda_{v_x} \left( \frac{-T \sin \varphi}{m} \right) + \lambda_{v_y} \left( \frac{T \cos \varphi \sin \beta}{m} \right) + \lambda_{v_z} \left( \frac{T \cos \varphi \cos \beta}{m} \right) + \mu_\varphi$$

$$\frac{\partial \bar{H}}{\partial \beta} = 0 = \lambda_{v_y} \left( \frac{\sin \varphi \cos \beta}{m} \right) + \lambda_{v_z} \left( \frac{-\sin \varphi \sin \beta}{m} \right) + \mu_\beta$$

$$\frac{\partial \bar{H}}{\partial \delta_E} = 0 = \lambda_{v_y} \left( \frac{\frac{1}{2} \rho (v_x^2 + v_y^2 + v_z^2) S_{ref}^{gin}}{m} \right) + \mu_{\delta_E}$$

$$\frac{\partial \bar{H}}{\partial \delta_N} = 0 = \lambda_{v_z} \left( \frac{\frac{1}{2} \rho (v_x^2 + v_y^2 + v_z^2) S_{ref}^{gin}}{m} \right) + \mu_{\delta_N}$$

The above completes the stationary condition minimization. From the KKT complimentary criterion, we can also say

$$\mu_T \begin{cases} \leq 0 & \text{if } T = 5 \text{ N} \\ = 0 & \text{if } 5 \text{ N} < T < 39.24 \text{ N} \\ \geq 0 & \text{if } T = 39.24 \text{ N} \end{cases} \quad (29)$$

$$\mu_\varphi \begin{cases} \leq 0 & \text{if } \varphi = 0 \text{ deg} \\ = 0 & \text{if } 0 \text{ deg} < \varphi < 25 \text{ deg} \\ \geq 0 & \text{if } \varphi = 25 \text{ deg} \end{cases} \quad (30)$$

$$\mu_\beta \begin{cases} \leq 0 & \text{if } \beta = 0 \text{ deg} \\ = 0 & \text{if } 0 \text{ deg} < \beta < 360 \text{ deg} \\ \geq 0 & \text{if } \beta = 360 \text{ deg} \end{cases} \quad (31)$$

$$\mu_{\delta_E} \{ \text{unrestricted since } h_4^i = h_4^u \} \quad (32)$$

$$\mu_{\delta_N} \{ \text{unrestricted since } h_5^i = h_5^u \} \quad (33)$$

where we do not know the specific value of the path co-vector, but we will know how it will react in the face of a path constraint on the control trajectory solution.

### Ascent Adjoint Equations

The Adjoint equations are the equations that describe the time rate of change of the co-vectors as defined by the negative time rate of change of the Lagrangian of the Hamiltonian, or

$$\frac{\partial \bar{H}}{\partial \mathbf{x}} = -\dot{\lambda} \quad (34)$$

Taking the partial derivative of the Lagrangian of the Hamiltonian with respect to every element of the state vector,  $\mathbf{x}$ , we get

$$\begin{aligned} \frac{\partial \bar{H}}{\partial v_x} = -\dot{\lambda}_{v_x} = \lambda_{v_x} & \left( \frac{\left[ \frac{1}{2} \rho L_{ref} D_{ref} C_X (2v_x^2 + v_y^2 + v_z^2) \right]}{m \sqrt{v_x^2 + v_y^2 + v_z^2}} \right) \\ & + \lambda_{v_y} \left( \frac{v_y \left[ \frac{1}{2} \rho L_{ref} D_{ref} C_X (2v_x^2 + v_y^2 + v_z^2) \right]}{m \sqrt{v_x^2 + v_y^2 + v_z^2}} \right) \\ & + \frac{\rho v_x S_{ref}^g \delta_E}{m} \\ & + \lambda_{v_z} \left( \frac{v_z \left[ \frac{1}{2} \rho L_{ref} D_{ref} C_X (2v_x^2 + v_y^2 + v_z^2) \right]}{m \sqrt{v_x^2 + v_y^2 + v_z^2}} \right) \\ & + \frac{\rho v_x S_{ref}^g \delta_N}{m} \Big) + \lambda_{r_x} \end{aligned}$$

$$\begin{aligned} \frac{\partial \bar{H}}{\partial v_y} = -\dot{\lambda}_{v_y} = \lambda_{v_x} & \left( \frac{v_x \left[ \frac{1}{2} \rho L_{ref} D_{ref} C_X (v_x^2 + 2v_y^2 + v_z^2) \right]}{m \sqrt{v_x^2 + v_y^2 + v_z^2}} \right) \\ & + \lambda_{v_y} \left( \frac{\left[ \frac{1}{2} \rho L_{ref} D_{ref} C_X (v_x^2 + 2v_y^2 + v_z^2) \right]}{m \sqrt{v_x^2 + v_y^2 + v_z^2}} \right) \\ & + \frac{\rho v_y S_{ref}^g \delta_E}{m} \\ & + \lambda_{v_z} \left( \frac{v_z \left[ \frac{1}{2} \rho L_{ref} D_{ref} C_X (v_x^2 + 2v_y^2 + v_z^2) \right]}{m \sqrt{v_x^2 + v_y^2 + v_z^2}} \right) \\ & + \frac{\rho v_y S_{ref}^g \delta_N}{m} \Big) + \lambda_{r_y} \end{aligned}$$

$$\begin{aligned} \frac{\partial \bar{H}}{\partial v_z} = -\dot{\lambda}_{v_z} = \lambda_{v_x} & \left( \frac{v_x \left[ \frac{1}{2} \rho L_{ref} D_{ref} C_X (v_x^2 + v_y^2 + 2v_z^2) \right]}{m \sqrt{v_x^2 + v_y^2 + v_z^2}} \right) \\ & + \lambda_{v_y} \left( \frac{v_y \left[ \frac{1}{2} \rho L_{ref} D_{ref} C_X (v_x^2 + v_y^2 + 2v_z^2) \right]}{m \sqrt{v_x^2 + v_y^2 + v_z^2}} \right) \\ & + \frac{\rho v_z S_{ref}^g \delta_E}{m} \\ & + \lambda_{v_z} \left( \frac{\left[ \frac{1}{2} \rho L_{ref} D_{ref} C_X (v_x^2 + v_y^2 + 2v_z^2) \right]}{m \sqrt{v_x^2 + v_y^2 + v_z^2}} \right) \\ & + \frac{\rho v_z S_{ref}^g \delta_N}{m} \Big) + \lambda_{r_z} \end{aligned}$$

$$\frac{\partial \bar{H}}{\partial r_x} = -\dot{\lambda}_{r_x} = 0$$

$$\frac{\partial \bar{H}}{\partial r_y} = -\dot{\lambda}_{r_y} = 0$$

$$\frac{\partial \bar{H}}{\partial r_z} = -\dot{\lambda}_{r_z} = 0$$

The above provide the dynamics of the co-states as a function of time in between the Ascent trajectory endpoints.

### Ascent Transversality Conditions

The Endpoint Lagrangian is defined as

$$\bar{E}(\mathbf{x}_0, \mathbf{x}_f, \mathbf{v}, t_0, t_f) = E(\mathbf{x}_0, \mathbf{x}_f, t_0, t_f) + \mathbf{v}^T \mathbf{e}(\mathbf{x}_0, \mathbf{x}_f, t_0, t_f) \quad (35)$$

where  $E(t_f)$  is the endpoint cost,  $\mathbf{v}$  are the endpoint co-vectors, and  $\mathbf{e}(x_f)$  are the endpoint error equations. The endpoint error equations, which encompass both initial and terminal conditions, can be expressed as the following for the Ascent trajectory. Remember that the initial state vector is completely known, and the desire is to enter the Corridor trajectory within a predefined circular entrance at a particular altitude and at zero lateral velocity.

$$\mathbf{e}(\mathbf{x}_0, \mathbf{x}_f) = \begin{bmatrix} \mathbf{v}_0 - \mathbf{v}^0 \\ \mathbf{r}_0 - \mathbf{r}^0 \\ v_{f,y} - v_y^f \\ v_{f,z} - v_z^f \\ r_{f,x} - r_x^f \\ (r_{f,y}^u - a_a)^2 + (r_{f,z}^u - b_a)^2 \end{bmatrix} \Rightarrow \mathbf{v} \in \mathbb{R}^{10}$$

From the proposed optimal control problem, we know  $E(t_f) = t_f$ , therefore our Endpoint Lagrangian becomes

$$\begin{aligned} \bar{E}(\mathbf{x}_0, \mathbf{x}_f, \mathbf{v}, t_f) = t_f &+ [v_1 \ v_2 \ v_3] \mathbf{v}_0 + [v_4 \ v_5 \ v_6] \mathbf{r}_0 + v_7 (v_{f,y} - v_y^f) \\ &+ v_8 (v_{f,z} - v_z^f) + v_9 (r_{f,x} - r_x^f) \\ &+ v_{10} \left( (r_{f,y}^u - a_a)^2 + (r_{f,z}^u - b_a)^2 \right) \end{aligned}$$

The above Endpoint Lagrangian can be used to find the terminal transversality conditions.

$$\lambda(t_f) = \frac{\partial \bar{E}}{\partial \mathbf{x}_f} \quad (36)$$

The Endpoint Lagrangian at all the initial conditions would be equivalent to their  $\mathbf{v}$  value since the equation is not a function of any other states that would not drop as part of the partial derivative, and therefore provide no new analytical information. The terminal transversality conditions however have some terms that do not cancel, and therefore can be expanded.

$$\begin{aligned} \frac{\partial \bar{E}}{\partial v_{f,y}} &= \lambda_{v_y}(t_f) = v_7 \\ \frac{\partial \bar{E}}{\partial v_{f,z}} &= \lambda_{v_z}(t_f) = v_8 \\ \frac{\partial \bar{E}}{\partial r_{f,x}} &= \lambda_{r_{f,x}}(t_f) = v_9 \\ \frac{\partial \bar{E}}{\partial r_{f,y}} &= \lambda_{r_{f,y}}(t_f) = 2v_{10}(r_{f,y} - a) \\ \frac{\partial \bar{E}}{\partial r_{f,z}} &= \lambda_{r_{f,z}}(t_f) = 2v_{10}(r_{f,z} - b) \end{aligned}$$

With the complementarity condition, we know that all the  $\mathbf{v}$  above are unrestricted in value. However,  $v_{10}$  can be approximated due to the inequality on the final lateral position.

$$v_{10} \begin{cases} \leq 0 \text{ if } (r_{f,y}^u - a_a)^2 + (r_{f,z}^u - b_a)^2 = 0 \\ = 0 \text{ if } 0 < (r_{f,y}^u - a_a)^2 + (r_{f,z}^u - b_a)^2 < R_a^2 \\ \geq 0 \text{ if } (r_{f,y}^u - a_a)^2 + (r_{f,z}^u - b_a)^2 = R_a^2 \end{cases} \quad (37)$$

### Ascent Hamiltonian Value Conditions

The Hamiltonian Value Condition (HVC) uses the endpoint Lagrangian and is given by

$$\mathcal{H}[@t_f] = -\frac{\partial \bar{E}}{\partial t_f} \quad (38)$$

Taking the partial derivative of our endpoint Lagrangian with respect to time yields the following, which is expected given the optimal control problem is minimum time.

$$\mathcal{H}[@t_f] = -\frac{\partial \bar{E}}{\partial t_f} = -1 \quad (39)$$

### Ascent Hamiltonian Evolution Equation

The Hamiltonian Evolution Equation (HEE) helps describe the change to the Hamiltonian as a function of time. Taking the partial derivative of the Hamiltonian for this optimal control problem yields the following result due to the Hamiltonian not being an explicit function of time.

$$\frac{d\mathcal{H}}{dt} = \frac{\partial \mathcal{H}}{\partial t} = 0 \quad (40)$$

### Ascent Trajectory Results

The Ascent trajectory from the ascent trajectory problem formulation was implemented in DIDO and solved for under the previously described initial, endpoint, and path constraints, leading to the following figure, which contains the solved position and velocity states of the vehicle. Note that this figure also contains a post-simulation (sim) fidelity check, which implements the control vector solved by DIDO into an open loop simulation that utilizes MATLAB's `ode45` propagator. This secondary fidelity simulation is performed to determine the feasibility of the provided control and state solution. Fig. 10 illustrates the Ascent Trajectory's solution. Furthermore, a plot of the control vector can be made as well, shown in Fig. 11.

There are a few elements of note about the Ascent Trajectory's solution that prove the solution matches what was desired by the original formulation:

- The initial states are as defined in the optimal control problem, which are equivalently the final states of the Liftoff trajectory. The vehicle starts at a non-zero upward velocity of ~5 m/s while perfectly aligned with the vertical component of the UEN frame and at the specified position of ~10 meters above the ground.
- To ensure minimum time, the thrust profile over the trajectory is, for most of the time, at full throttle. It reaches the desired endpoint conditions at  $t = 2.41$  s after the end of the Liftoff Trajectory.
- The vehicle then translates laterally over the given altitude during its ascent and ultimately finishes the trajectory within the predefined circular corridor entrance while also at the desired zero lateral velocity. The final position at the specified final altitude is

approximately  $\mathbf{r}_{uc}^u = [50 \ 8 \ 0]^T m$ , which is within/at the circular endpoint constraint.

- The vertical velocity vector starts to slow at the end of the Ascent trajectory, indicating that aerodynamic drag is starting to become more significant due to the increase in dynamic pressure in relation to a constant engine thrust.
- The control vector elements never exceed their control vector path constraints.
- The feasibility simulation closely mimics the provided DIDO solution, indicating the solution is applicable to the system of interest. Deviations between the two could reduce by increasing the DIDO node count.

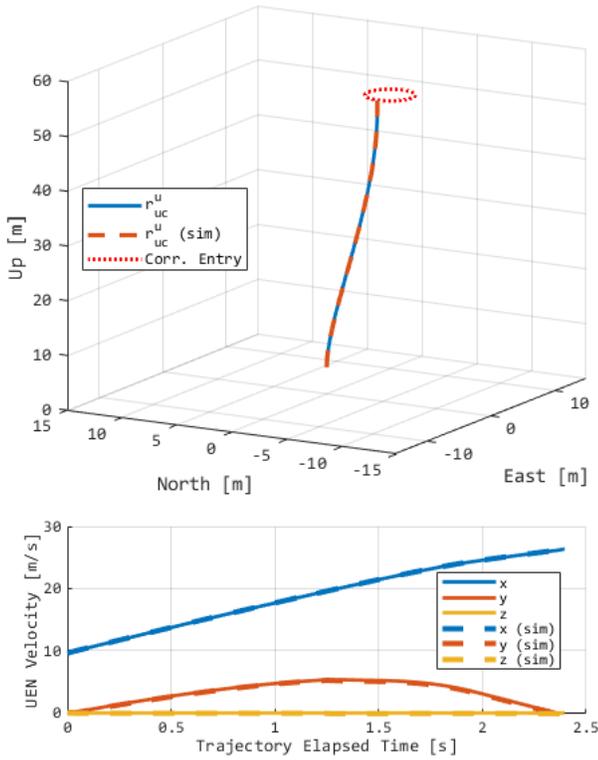


Fig. 10 Ascent state and feasibility simulation states

Analysis will now turn to V&V of optimality. One point of V&V is an investigation of the Hamiltonian evolution from DIDO. The DIDO Hamiltonian is shown in Fig. 12.

The Hamiltonian supports a few conclusions from the previous section. Since the optimal control problem is minimum time and the Hamiltonian evolution is zero in time, the above supports the theory by being very close to a value of -1 for most of the trajectory. However, there are a few deviations from the theoretical value, but what is presented in the above figure should be close enough to verify this solution is indeed minimum time and therefore aligns with the theory. The HEE is also supported given that the Hamiltonian does not evolve through time enough to argue that it is not supported.

Investigation can also be made into the Hamiltonian minimization condition, where the stationary and complimentary conditions are added into Hamiltonian to

minimize path constraints on controls. The complementarity conditions during the optimal control trajectory support the path constraints as defined in the conditionals previously illustrated while minimizing the Hamiltonian. Viewing the control trajectories in Fig. 11, when a particular control path constraint is met, as previously described by the complementarity condition, the respective  $\mu_0$  value in Fig. 13 is non-zero, further supporting the solution aligns with the theory.

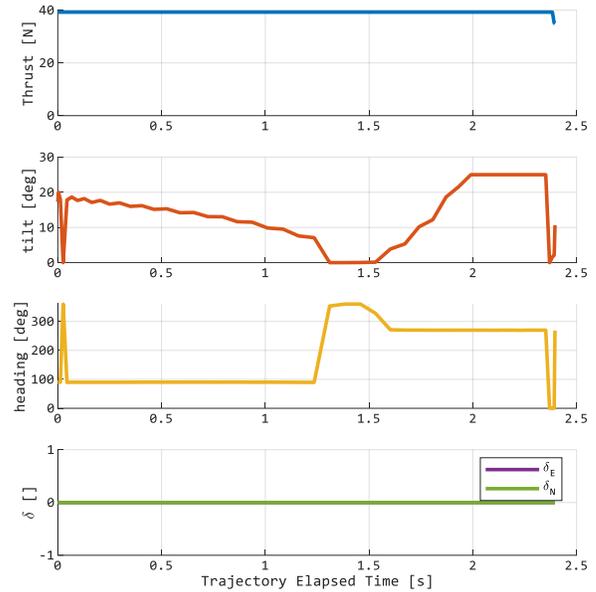


Fig. 11 Ascent trajectory control vectors

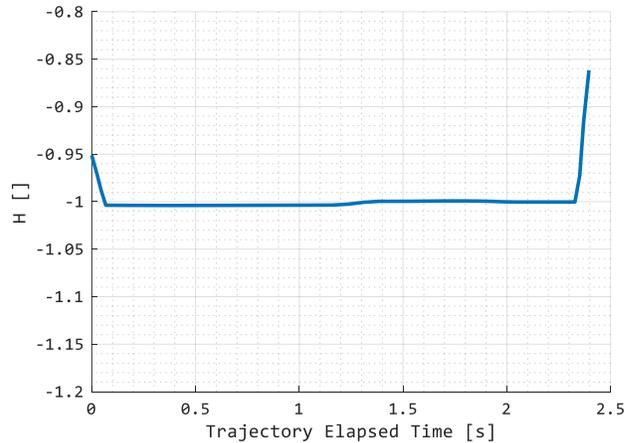


Fig. 12 Ascent Hamiltonian

The co-states can be investigated to help V&V the solution in comparison with the theory. The co-states are shown in Fig. 14. The co-states in this problem are governed by the derived adjoint equations previously derived. As expected, the velocity co-states are changing in time as a function of the change to the vehicle's velocity vector components. Furthermore, the position co-states, based on the theory, were expected to be constant during the trajectory, and this is supported in the adjoint equations, therefore further supporting the solution found was minimum time and supported by the theory.

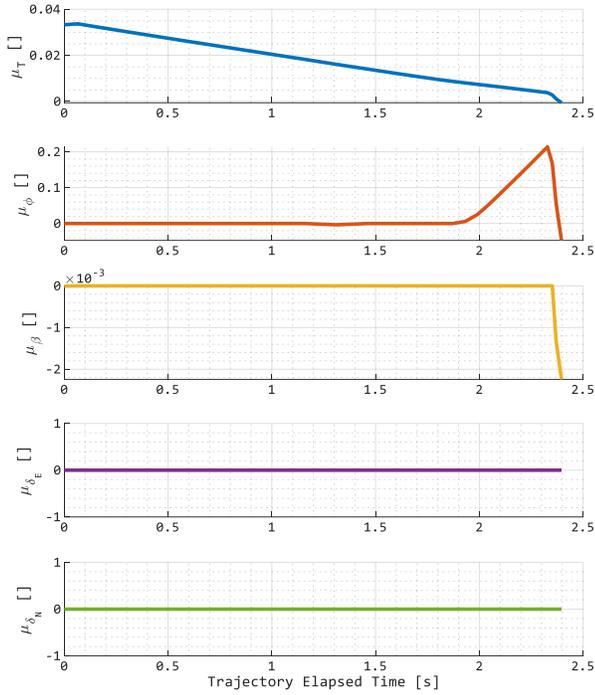


Fig. 13 Ascent control co-vectors

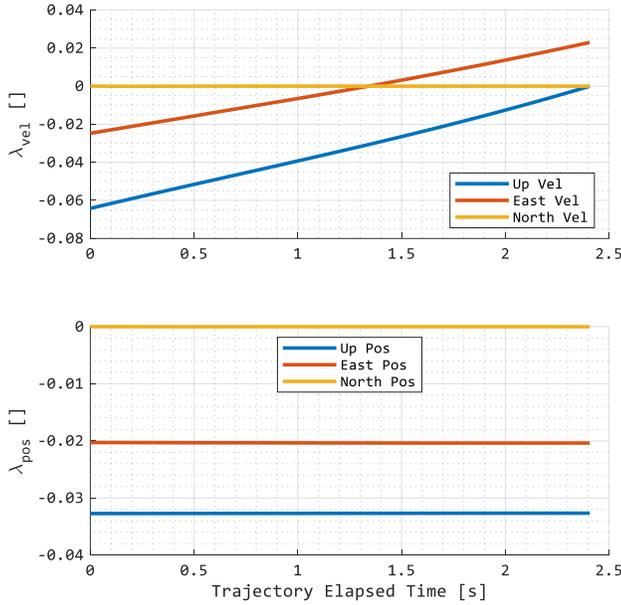


Fig. 14 Ascent co-states

### C. Corridor Trajectory

The Corridor trajectory is formatted as the following optimal control problem, shown in Optimal Problem 2: Corridor. For the Corridor trajectory, Table IV helps characterize variables within the Corridor trajectory problem.

Optimal Problem 2: Corridor

$$\begin{aligned}
 & \mathbf{x} \in \mathbb{R}^6, \mathbf{u} \in \mathbb{R}^5 \\
 \text{Min.} & \quad J[\mathbf{x}(\cdot), \mathbf{u}(\cdot), t_f] = t_f \\
 \text{s.t.} & \quad \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{p}) \\
 & \quad t_0 = 0 \text{ s} \\
 & \quad \mathbf{v}_0^u = \mathbf{v}_{f, \text{ascent}}^u \text{ m/s} \\
 & \quad \mathbf{r}_0^u = \mathbf{r}_{f, \text{ascent}}^u \text{ m} \\
 & \quad (\mathbf{v}_{f, y}^u, \mathbf{v}_{f, z}^u) = (0, 0) \text{ m/s} \\
 & \quad r_{f, x}^u = 250 \text{ m} \\
 & \quad h_1(\mathbf{u}, t) := 5 \text{ N} \leq T \leq 39.24 \text{ N} \\
 & \quad h_2(\mathbf{u}, t) := 0 \text{ deg} \leq \varphi \leq 25 \text{ deg} \\
 & \quad h_3(\mathbf{u}, t) := 0 \text{ deg} \leq \beta \leq 360 \text{ deg} \\
 & \quad h_4(\mathbf{u}, t) := 0 \leq \delta_E \leq 0 \\
 & \quad h_5(\mathbf{u}, t) := 0 \leq \delta_N \leq 0 \\
 & \quad h_6(\mathbf{x}, \mathbf{u}, t) := 0 \leq (r_y^u - a_c)^2 + (r_z^u - b_c)^2 \leq R_c^2
 \end{aligned}$$

TABLE IV  
PARAMETER DATA FOR THE OPTIMAL CORRIDOR CONTROL PROBLEM

Parameter	Value
$a_c$	10 m
$b_c$	0 m
$R_c$	2 m
$t^0$	0 s
$C_x$	-0.1202 (stowed)

This optimal control problem has a fully defined initial state vector based on the final state of the Ascent trajectory, along with path constraints on all its control effectors. Furthermore, it has a path constraint inequality stating it must fly through a cylindrical region of radius  $R_c$  at all altitudes during the Corridor trajectory. At the end of the trajectory, the vehicle is defined to have zero lateral velocity such that the handoff to the ballistic Coast trajectory does not lead to a large displacement prior to apogee. Furthermore, the optimal control problem is minimized in time similar to the Ascent trajectory.

The above defined optimal control problem can be solved through the HAMVET method. Since this optimal trajectory has some solutions that are the same as the Ascent trajectory, some of the following sections will be direct copies.

### Corridor Hamiltonian

The Lagrangian of the Hamiltonian is mostly equivalent to the previously derived Lagrangian of the Hamiltonian as described in. However, Eqn. 28 changes due to the additional cylindrical path constraint.

$$\begin{aligned}
 \boldsymbol{\mu}^T \mathbf{h}(\mathbf{x}, \mathbf{u}) &= \mu_T T + \mu_\varphi \varphi + \mu_\beta \beta + \mu_{\delta_E} \delta_E + \mu_{\delta_N} \delta_N \\
 &+ \mu_{cyl} \left[ (r_y^u - a_c)^2 + (r_z^u - b_c)^2 \right]
 \end{aligned}$$

### Corridor Hamiltonian Minimization

Performing the HMC for the Corridor trajectory is equivalent to the Ascent trajectory and is therefore not repeated here. The KKT conditions for the control  $\boldsymbol{\mu}$  are also equivalent to Eqns. 29-33. The KKT condition for  $\mu_{cyl}$  will come into play during the derivation of the adjoint equations.

### Corridor Adjoint Equations

The Adjoint equations for the Corridor trajectory are not equivalent to those described in the Ascent trajectory due to the addition of a path constraint on position rather than a terminal position endpoint constraint. However, since the path constraint contains only position states, there is no effect to the  $\dot{\lambda}_{v_x}$ ,  $\dot{\lambda}_{v_y}$ , and  $\dot{\lambda}_{v_z}$  co-state differential equations. Taking the partial derivative of the positional states for the adjoint, we get

$$\begin{aligned}\frac{\partial \bar{H}}{\partial r_x} &= -\dot{\lambda}_{r_x} = 0 \\ \frac{\partial \bar{H}}{\partial r_y} &= -\dot{\lambda}_{r_y} = 2\mu_{cyl}(r_y^u - a_c) \\ \frac{\partial \bar{H}}{\partial r_z} &= -\dot{\lambda}_{r_z} = 2\mu_{cyl}(r_z^u - b_c)\end{aligned}$$

The above states that the  $\lambda_{r_x}$  co-state should not change in time, but the others may. Note that the others still may not change if  $\mu_{cyl}$  is never non-zero from a path constraint. The KKT for  $\mu_{cyl}$  is the following.

$$\mu_{gs} \begin{cases} \leq 0 & \text{if } h_6(\mathbf{x}, \mathbf{u}, t) = 0 \\ = 0 & \text{if } 0 < h_6(\mathbf{x}, \mathbf{u}, t) < R_c^2 \\ \geq 0 & \text{if } h_6(\mathbf{x}, \mathbf{u}, t) = R_c^2 \end{cases} \quad (41)$$

Should the path constraint always be met, it is likely  $\dot{\lambda}_{r_y}$  and  $\dot{\lambda}_{r_z}$  may also not change in time due to being multiplied by zero.

### Corridor Transversality Conditions

The Endpoint Lagrangian is for the Corridor trajectory is defined differently than the previous Ascent trajectories due to the removal of the time minimization and the fully defined final position and velocity states.

$$\mathbf{e}(\mathbf{x}_0, \mathbf{x}_f) = \begin{bmatrix} \mathbf{v}_0 - \mathbf{v}^0 \\ \mathbf{r}_0 - \mathbf{r}^0 \\ v_{f,y} - v_y^0 \\ v_{f,z} - v_z^f \\ r_{f,z} - r_z^f \end{bmatrix} \Rightarrow \mathbf{v} \in \mathbb{R}^9$$

From the proposed optimal control problem, we know  $E(t_f) = t_f$ , therefore our Endpoint Lagrangian becomes

$$\begin{aligned}\bar{E}(\mathbf{x}_0, \mathbf{x}_f, \mathbf{v}, t_f) &= t_f + [v_1 \ v_2 \ v_3] \mathbf{v}_0 + [v_4 \ v_5 \ v_6] \mathbf{r}_0 \\ &\quad + v_7(v_{f,y} - v_y^0) + v_8(v_{f,z} - v_z^f) + v_9(r_{f,z} - r_z^f)\end{aligned}$$

The above Endpoint Lagrangian can be used to find the terminal transversality conditions by using Eqn. 36.

$$\frac{\partial \bar{E}}{\partial v_{f,y}} = \lambda_{v_{f,y}}(t_f) = v_7$$

$$\frac{\partial \bar{E}}{\partial v_{f,z}} = \lambda_{v_{f,z}}(t_f) = v_8$$

$$\frac{\partial \bar{E}}{\partial r_{f,x}} = \lambda_{r_{f,x}}(t_f) = v_9$$

With the complementarity condition, we know that all  $\mathbf{v}$  above are unrestricted in value, and therefore the transversality conditions provide no additional mathematical information.

### Corridor Hamiltonian Value Conditions

The HVC for the Corridor trajectory is also minimum time, leading to an expected value of -1 as stated in Eqn. 39.

### Corridor Hamiltonian Evolution Equation

Like the Ascent trajectory, due to the Hamiltonian not being an explicit function of time, the HEE is equivalent to Eqn. 40.

### Corridor Trajectory Results

The Corridor trajectory from the Optimal Problem 2: Corridor trajectory problem formulation was implemented in DIDO and solved for under the previously described initial, endpoint, and path constraints, leading to Fig. 15, which contains the solved position and velocity states of the vehicle. The green cylindrical region represents the path constraint of the Corridor trajectory. Note that this figure also contains a post-simulation fidelity check as previously described in the Ascent trajectory's results section.

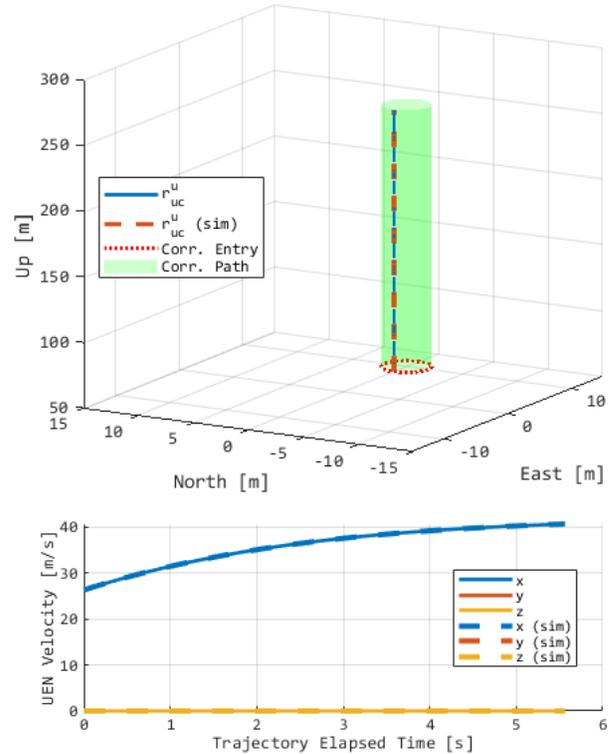


Fig. 15 Corridor state and feasibility simulation states

Furthermore, a plot of the control vector shown in Fig. 16 can be made as well.

There are a few elements of note about the above solution that prove the solution matches what was desired by the original formulation:

- The initial states are as defined in the optimal control problem, which are equivalently the final states of the

Ascent trajectory. The vehicle starts at a non-zero upward velocity of  $\sim 26$  m/s while perfectly aligned with the vertical component of the UEN frame (no lateral velocities) and at the specified position of 50 meters above the ground.

- To ensure minimum time, the thrust profile over the entire trajectory is full throttle. It ascends 200 m and reaches the desired endpoint conditions in 5.56 s.
- The vehicle moves vertically over the given altitude during its ascent all while staying within the predefined cylindrical corridor. The lateral velocity at the end of the trajectory is also zero. The final position at the specified final altitude is approximately  $\mathbf{r}_f^u = [250 \ 8 \ 0]^T$  m, which is within/at the cylindrical region at the final time as defined as part of the optimal control problem's path constraint.
- The vertical velocity vector shows clear indication of aerodynamic drag becoming more significant in relation to the engine thrust. The final vertical velocity of the system was 40.72 m/s.
- The control vector elements never exceed their control vector path constraints.
- The feasibility simulation closely mimics the provided DIDO solution, indicating the solution is applicable to the system of interest. Deviations between the two would reduce by increasing the node count of the DIDO tool.

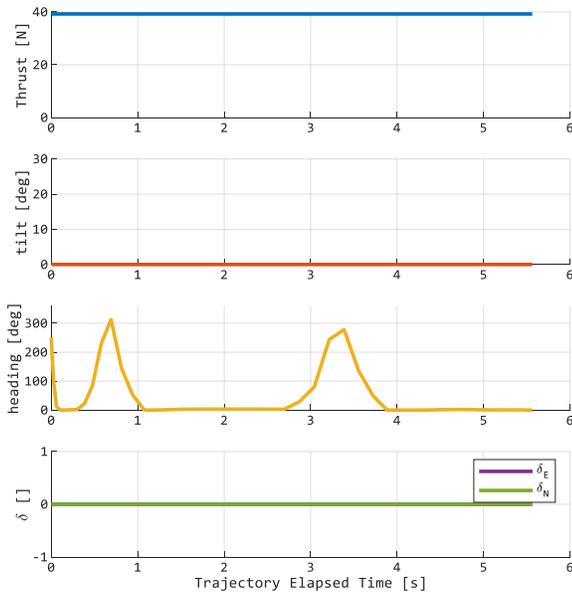


Fig. 16 Corridor trajectory control vectors

Analysis will now turn to V&V of the optimality. One point of V&V is an investigation of the Hamiltonian evolution from DIDO. The DIDO Hamiltonian is shown in Fig. 17. The Hamiltonian supports a few conclusions from the previous section. Since the optimal control problem is minimum time, the theory stated the HVC should yield a Hamiltonian value of -1 at the final time. The Hamiltonian plot supports the theory

by being very close to a value of -1. From the HEE, it can also be seen that the Hamiltonian does not evolve through time.

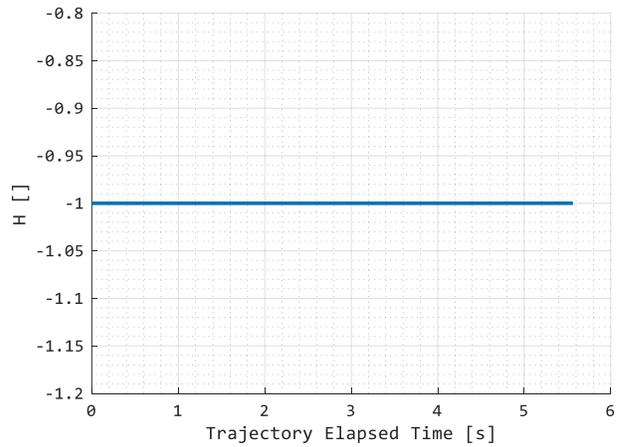


Fig. 17 Corridor Hamiltonian

Investigation can also be made into the HMC, where the stationary and complimentary conditions are added into Hamiltonian to minimize path constraints on controls. Fig. 18 illustrates those KKT co-vectors for the path constraints.

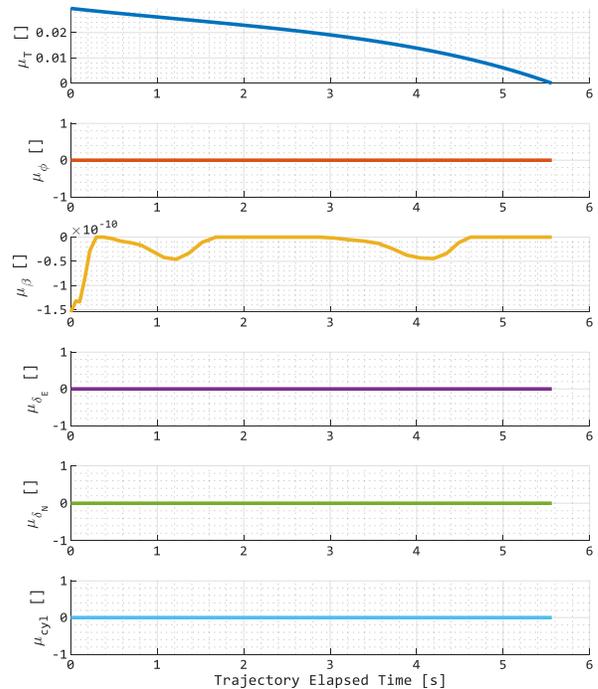


Fig. 18 Corridor control co-vectors

The complementarity conditions during the optimal control trajectory support the path constraints as defined in the conditionals previously illustrated while minimizing the Hamiltonian. Since the Corridor trajectory is simply a straight path and never exceeded its cylindrical path constraint, none of the path constraint limits were exercised with exception to the thrust control, and therefore are approximately zero, supporting the solution aligns with the theory.

The co-states can also be investigated to help V&V the solution in comparison with the theory. The co-states are shown in Fig. 19. The co-states in this problem are governed by the derived adjoint equations. The vertical velocity co-state is non-zero, which is expected given the trajectory is resolved to the vertical channel (lateral velocity is  $\approx 0$ ). Furthermore, the  $y$  and  $z$  position co-states, based on the theory, were expected to be non-constant during the trajectory. However, since the path constraint was not exercised ( $\mu_{cyl} = 0$ ), then it is understood why the lateral position co-states did not change in time. The  $x$  position (up in UEN) co-state was also expected to be constant in time. Based on the theory and the process of V&V, the solution found for the Corridor trajectory was minimum time.

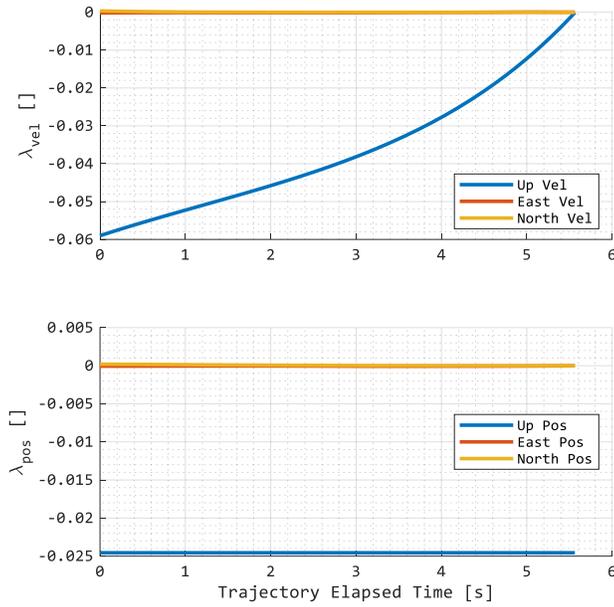


Fig. 19 Corridor co-states

#### D. Coast Trajectory

The Coast trajectory is simply a non-optimized trajectory that ends once the system reaches apogee. All controls are set in their off position, and the ballistics of the system are allowed to propagate. For the trajectory parameters of interest, Fig. 20 illustrates the vehicle as it continues its dynamics from the Corridor trajectory exit. The vertical velocity deceleration is not constant due to the exponential decay of dynamic pressure as the vehicle slows. The system reaches apogee approximately 3.31 s after the Corridor trajectory ends.

#### E. Reentry Trajectory

The Reentry trajectory is formatted as the optimal control problem shown in Optimal Problem 3: Reentry. Note that its initial conditions are taken from the final Coast trajectory state. For the Reentry trajectory, Table V defines parameters that help characterize the endpoint constraint on position.

The endpoint inequality constraint is only for lateral position at the specified final altitude of 100 m while also directly above the LLP. Furthermore, note that the path constraints on control are now different than the Ascent and

Corridor trajectories; the engine control states are set to zero, and the grid fin control effectors are allowed to add forces on the system to guide the vehicle into the desired lateral region. Gravity is the only source of acceleration that moves the vehicle vertically from apogee to the final desired altitude. The optimal control problem is minimized via time similar to the Ascent and Corridor trajectories.

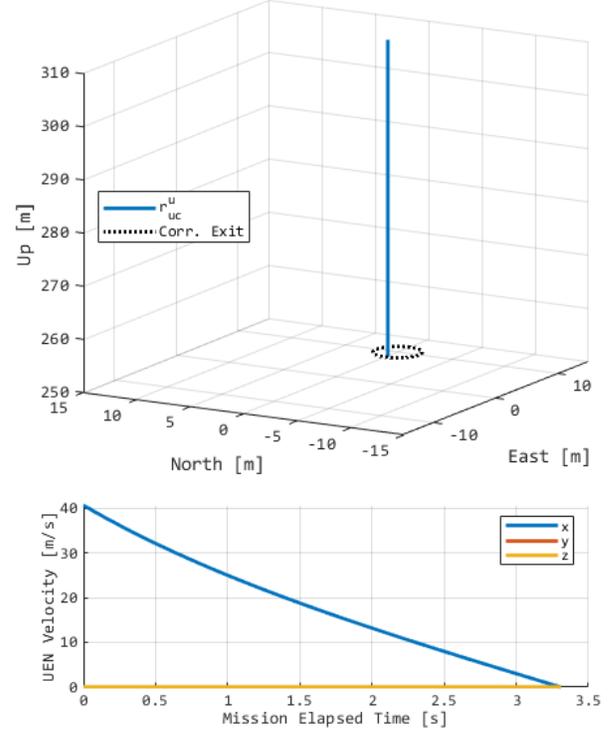


Fig. 20 Coast trajectory states

#### Optimal Problem 3: Reentry

$$\begin{aligned}
 & \mathbf{x} \in \mathbb{R}^6, \quad \mathbf{u} \in \mathbb{R}^5 \\
 \text{Min.} & \quad J[\mathbf{x}(\cdot), \mathbf{u}(\cdot), t_f] = t_f \\
 \text{s.t.} & \quad \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{p}) \\
 & \quad t_0 = 0 \text{ s} \\
 & \quad \mathbf{v}_0^u = \mathbf{v}_{f,coast}^u \text{ m/s} \\
 & \quad \mathbf{r}_0^u = \mathbf{r}_{f,coast}^u \text{ m} \\
 & \quad r_{f,x}^u = 100 \text{ m} \\
 & \quad 0 \leq (r_{f,y}^u - a_r)^2 + (r_{f,z}^u - b_r)^2 \leq R_r^2 \\
 & \quad h_1(\mathbf{u}, t) := 0 \text{ N} \leq T \leq 0 \text{ N} \\
 & \quad h_2(\mathbf{u}, t) := 0 \text{ deg} \leq \varphi \leq 0 \text{ deg} \\
 & \quad h_3(\mathbf{u}, t) := 0 \text{ deg} \leq \beta \leq 0 \text{ deg} \\
 & \quad h_4(\mathbf{u}, t) := -1 \leq \delta_E \leq 1 \\
 & \quad h_5(\mathbf{u}, t) := -1 \leq \delta_N \leq 1
 \end{aligned}$$

 TABLE V  
 PARAMETER DATA FOR THE OPTIMAL REENTRY CONTROL PROBLEM

Parameter	Value
$a_r$	0 m
$b_r$	0 m
$R_r$	4 m
$t^0$	0 s
$C_x$	-0.2104 (deployed)

We can solve the above defined optimal control problem through the HAMVET method, as has been performed in the previous trajectories. Since this optimal trajectory is essentially the same as the Ascent trajectory, only explicit differences will be presented.

### Reentry Hamiltonian

The Lagrangian of the Hamiltonian is equivalent to the previously derived Lagrangian of the Hamiltonian as described in Eqns. 26, 27, and 28 for the Ascent trajectory. Therefore, it will not be repeated here.

### Reentry Hamiltonian Minimization

Performing the HMC for the Reentry trajectory is equivalent to the Ascent trajectory and is therefore not repeated here. However, the KKT conditions are now different given the updated path constraint inequality values.

$$\mu_T \{ \text{unrestricted since } h_1^l = h_1^u \} \quad (42)$$

$$\mu_\phi \{ \text{unrestricted since } h_2^l = h_2^u \} \quad (43)$$

$$\mu_\beta \{ \text{unrestricted since } h_3^l = h_3^u \} \quad (44)$$

$$\mu_{\delta_E} \begin{cases} \leq 0 \text{ if } \delta_E = -1 \\ = 0 \text{ if } -1 < \delta_E < 1 \\ \geq 0 \text{ if } \delta_E = 1 \end{cases} \quad (45)$$

$$\mu_{\delta_N} \begin{cases} \leq 0 \text{ if } \delta_N = -1 \\ = 0 \text{ if } -1 < \delta_N < 1 \\ \geq 0 \text{ if } \delta_N = 1 \end{cases} \quad (46)$$

### Reentry Adjoint Equations

The Adjoint equations for the Reentry trajectory are equivalent to those described in the Ascent adjoint equations in Eqn. 34 and are therefore not repeated here.

### Reentry Transversality Conditions

The Endpoint Lagrangian is defined differently than the previous Ascent trajectory due to the removal of the lateral velocity endpoint constraint.

$$\mathbf{e}(\mathbf{x}_0, \mathbf{x}_f) = \begin{bmatrix} \mathbf{v}_0 - \mathbf{v}^0 \\ \mathbf{r}_0 - \mathbf{r}^0 \\ r_{f,x}^u - r_x^f \\ \left[ (r_{f,y}^u - a_r)^2 + (r_{f,z}^u - b_r)^2 \right] \end{bmatrix} \Rightarrow \mathbf{v} \in \mathbb{R}^3$$

From the proposed optimal control problem, we know  $E(t_f) = t_f$ , and therefore our Endpoint Lagrangian becomes

$$\bar{E}(\mathbf{x}_0, \mathbf{x}_f, \mathbf{v}, t_f) = t_f + [v_1 \ v_2 \ v_3] \mathbf{v}_0 + [v_4 \ v_5 \ v_6] \mathbf{r}_0 + v_7 (r_{f,x}^u - r_x^f) + v_8 \left( (r_{f,y}^u - a_r)^2 + (r_{f,z}^u - b_r)^2 \right)$$

The above Endpoint Lagrangian can be used to find the terminal transversality conditions by using Eqn. 36.

$$\frac{\partial \bar{E}}{\partial r_{f,x}} = \lambda_{r_{f,x}}(t_f) = v_7$$

$$\frac{\partial \bar{E}}{\partial r_{f,y}} = \lambda_{r_{f,y}}(t_f) = 2v_8(r_{f,y} - a_c)$$

$$\frac{\partial \bar{E}}{\partial r_{f,z}} = \lambda_{r_{f,z}}(t_f) = 2v_8(r_{f,z} - b_c)$$

With the complementarity condition, we know that all  $\mathbf{v}$  above are unrestricted in value. However,  $v_8$  can be approximated due to the inequality on the final endpoint lateral position.

$$v_8 \begin{cases} \leq 0 \text{ if } (r_{f,y}^u - a_r)^2 + (r_{f,z}^u - b_r)^2 = 0 \\ = 0 \text{ if } 0 < (r_{f,y}^u - a_r)^2 + (r_{f,z}^u - b_r)^2 < R_r^2 \\ \geq 0 \text{ if } (r_{f,y}^u - a_r)^2 + (r_{f,z}^u - b_r)^2 = R_r^2 \end{cases} \quad (47)$$

where  $a_r$ ,  $b_r$ , and  $R_r$  are defined in Table V.

### Reentry Hamiltonian Value Conditions

The HVC for the Reentry trajectory is also minimum time, similar to the Ascent and Corridor trajectories, leading to an expected value of -1 as stated in Eqn. 39.

### Reentry Hamiltonian Evolution Equation

Like the Ascent and Corridor trajectory, due to the Hamiltonian not being an explicit function of time, the HEE is equivalent to Eqn. 40.

### Reentry Trajectory Results

The Reentry trajectory from the Optimal Problem 3: Reentry trajectory problem formulation was implemented in DIDO and solved for under the previously described initial, endpoint, and path constraints, leading to Fig. 21, which contains the solved position and velocity states of the vehicle. Note that this figure also contains a post-simulation fidelity check as previously described in the Ascent trajectory's results section.

Furthermore, a plot of the control vector, Fig. 22, can be made as well, which show the engine control effector states as zero, and the grid fin control effector states as non-zero.

There are a few elements of note about this solution that prove the solution matches what was desired by the original formulation:

- The initial states are as defined in the optimal control problem, which are equivalently the final states of the Coast trajectory. The vehicle starts at a zero upward and lateral velocity in the UEN frame. The Reentry trajectory starts at an altitude of approximately 309.69 m (end of Coast).
- The minimum time trajectory is mostly vertical. However, there is no force control effectors in the vertical direction other than deployed grid fin aerodynamic drag. The vehicle descends ~210 m and reaches the desired endpoint conditions in 8.81 s from the start of the Reentry trajectory.
- The vehicle translates laterally over the given altitude during its descent and ultimately finishes the trajectory within the predefined circular Reentry exit. The final position at the specified final altitude is approximately

$\mathbf{r}_{uc}^u = [100 \ 2.44 \ -0.22]^T m$ , which is within/near the circular bounding region above the pad, with a radius of  $R_r$ , at  $\mathbf{r}^u = [100 \ 0 \ 0]^T m$ , defined as part of the optimal control problem’s endpoint constraint.

- The vertical velocity vector shows clear indication of aerodynamic drag becoming more significant as the velocity increases, leading to the system almost hitting its terminal velocity. Furthermore, the final vertical velocity of the system was  $-31.56 m/s$ , which is a smaller magnitude than it could be given the additional drag imposed on the system from the deployed grid fins.
- The control vector elements never exceed their control vector path constraints. Furthermore, the grid fin control effectors are the only available source of control. At the start of the trajectory, the control effect is high, which is due to the low dynamic pressure in this region.
- The feasibility simulation closely mimics the provided DIDO solution, indicating the solution is applicable to the system of interest. Deviations between the two would reduce by increasing the node count of the DIDO tool.

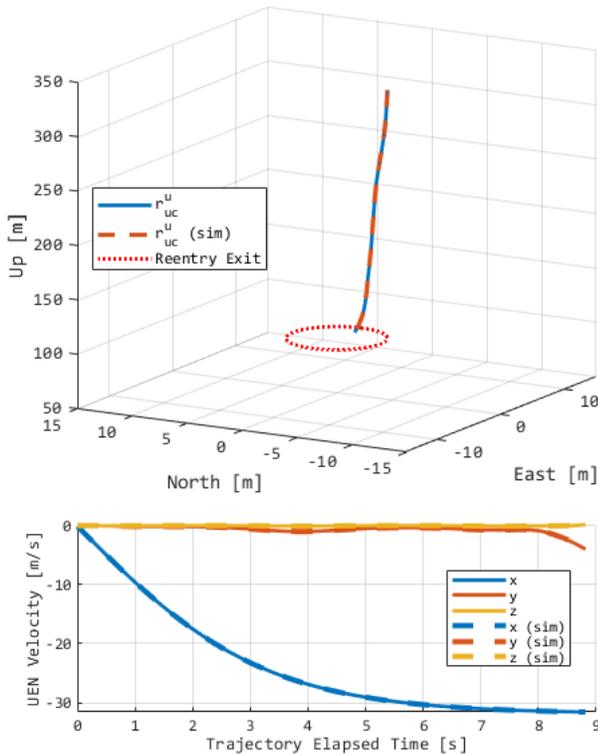


Fig. 21 Reentry state and feasibility simulation states

Analysis will now turn to V&V of the solution for optimality. One point of V&V is an investigation of the Hamiltonian evolution from DIDO. The DIDO Hamiltonian is shown in Fig. 23. The Hamiltonian supports a few conclusions from the previous section. Since the optimal control problem is

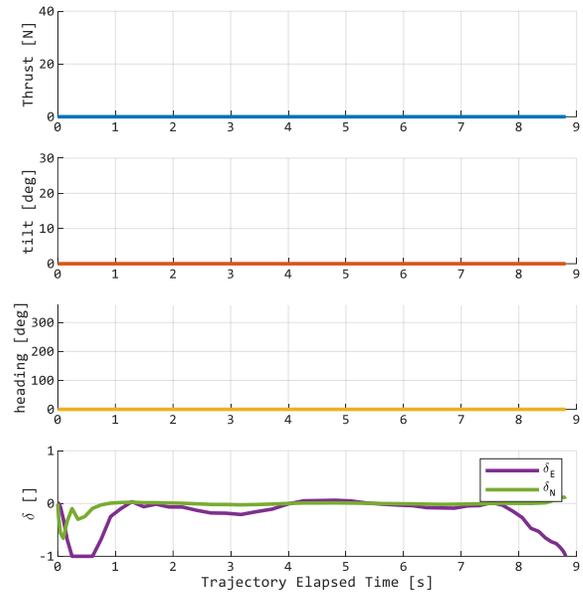


Fig. 22 Reentry trajectory control vectors

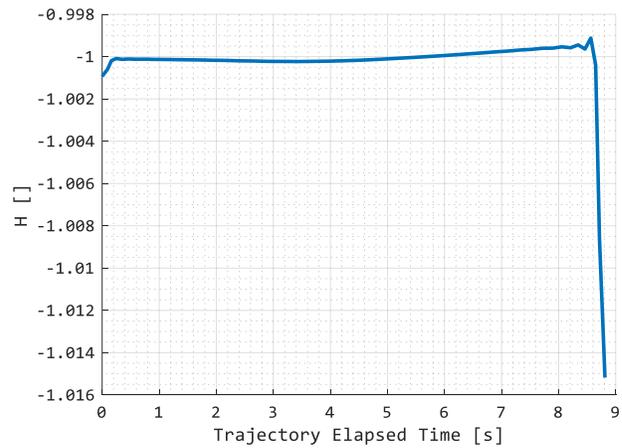


Fig. 23 Reentry Hamiltonian

minimum time and the Hamiltonian evolution is zero in time, Fig. 23 supports the theory by being very close to a value of -1. From the HEE, it can also be seen that the Hamiltonian does not evolve through time (even with the small deviations away from -1). This solution supports the theory.

Investigation can also be made into the HMC, where the stationary and complimentary conditions are added into Hamiltonian to minimize path constraints on controls. Fig. 24 illustrates those KKT co-vectors for the path constraints.

The complementarity conditions during the optimal control trajectory support the path constraints as defined in the conditionals previously illustrated while minimizing the Hamiltonian. Since the Reentry trajectory completely constrains the engine control effectors, it makes sense that their control is latched at zero for all time. Furthermore, looking at the  $\delta_E$  and  $\delta_N$  controls during the Reentry trajectory, it clearly shows saturation of the control limits at the start of the trajectory, verifying why  $\mu_{\delta_E}$  and  $\mu_{\delta_N}$  are non-zero at the start of the trajectory. This is primarily due to the low dynamic

pressure in this region, referenced in Eqns. 12-14. This solution aligns with the theory.

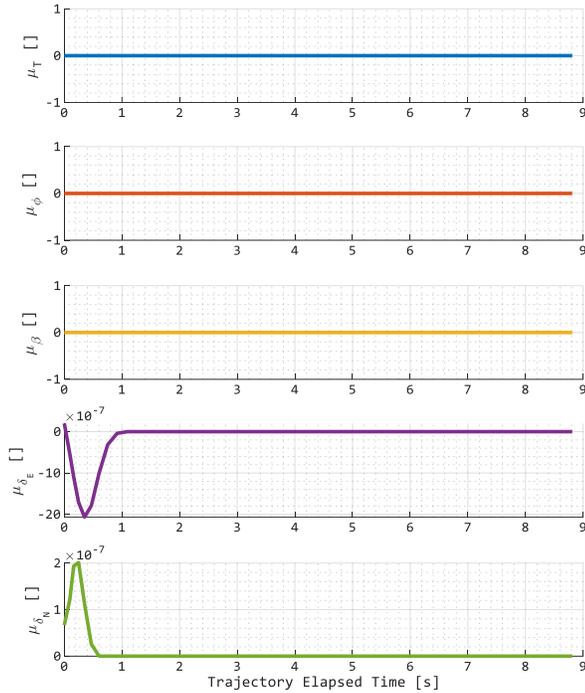


Fig. 24 Reentry control co-vectors

The co-states can also be investigated to help V&V the solution in comparison with the theory. The co-states are shown below in Fig. 25.

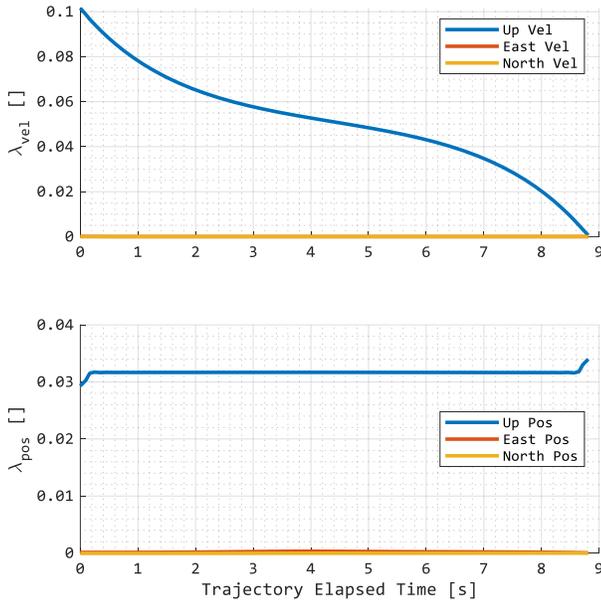


Fig. 25 Reentry co-states

The co-states in this problem are governed by the derived adjoint equations. The vertical velocity co-state is non-zero, which is expected given the trajectory is resolved to the vertical

channel (lateral velocity is  $\approx 0$ ). While difficult to see, the lateral velocity costate values are also non-zero, which is likely partially due to the saturated grid fin control vectors. Furthermore, the position co-states, based on the theory, were expected to be constant during the trajectory, and this is supported in the adjoint equations, therefore further supporting the solution for the Reentry trajectory found was minimum time.

#### F. Landing Trajectory

The Landing trajectory is formatted as the following optimal control problem shown in Optimal Problem 4: Landing. Note that its initial conditions are taken from the final Reentry trajectory state.

Optimal Problem 4: Landing

$$\begin{aligned}
 & \mathbf{x} \in \mathbb{R}^6, \quad \mathbf{u} \in \mathbb{R}^5 \\
 \text{Min.} & \quad J[\mathbf{x}(\cdot), \mathbf{u}(\cdot), t_f] = \int_{t_0}^{t_f} T \, dt \\
 \text{s.t.} & \quad \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{p}) \\
 & \quad t_0 = 0 \, \text{s} \\
 & \quad \mathbf{v}_0^u = \mathbf{v}_{f, \text{reentry}}^u \, \text{m/s} \\
 & \quad \mathbf{r}_0^u = \mathbf{r}_{f, \text{reentry}}^u \, \text{m} \\
 & \quad \mathbf{v}_f^u = (-1, 0, 0) \, \text{m/s} \\
 & \quad \mathbf{r}_f^u = (10, 0, 0) \, \text{m} \\
 & \quad h_1(\mathbf{u}, t) := 0 \, \text{N} \leq T \leq 39.24 \, \text{N} \\
 & \quad h_2(\mathbf{u}, t) := 0 \, \text{deg} \leq \varphi \leq 25 \, \text{deg} \\
 & \quad h_3(\mathbf{u}, t) := 0 \, \text{deg} \leq \beta \leq 360 \, \text{deg} \\
 & \quad h_4(\mathbf{u}, t) := 0 \leq \delta_E \leq 0 \\
 & \quad h_5(\mathbf{u}, t) := 0 \leq \delta_N \leq 0 \\
 & \quad h_6(\mathbf{x}, \mathbf{u}, t) := 85 \, \text{deg} \leq \gamma_{gs}(\mathbf{x}, \mathbf{p}) \leq 90 \, \text{deg}
 \end{aligned}$$

The glide slope,  $\gamma_{gs}$ , is defined by the following equation. This equation constrains the vehicles positional slope as it descends to its desired final position.

$$\gamma_{gs}(\mathbf{x}, \mathbf{p}) = \tan^{-1} \left( \frac{r_x^u - a_l}{\sqrt{(r_y^u - b_l)^2 + (r_z^u - c_l)^2}} \right) \quad (48)$$

For the Landing trajectory, we have the following parameters shown in Table VI that help characterize variables in the optimal control problem.

TABLE VI  
PARAMETER DATA FOR THE OPTIMAL LANDING CONTROL PROBLEM

Parameter	Value
$a_l$	$r_{f,x}^u$
$b_l$	$r_{f,y}^u$
$c_l$	$r_{f,z}^u$
$t^0$	0 s
$C_x$	-0.2104 (deployed)

The Landing trajectory deviates the furthest mathematically from the other trajectories developed in this project thus far. For this trajectory, the engine control state path

constraints are re-enabled, and the engine is now allowed to command zero throttle. The grid fin controls are disabled back to zero, just as they were in the Ascent and Corridor trajectories. However, the grid fins remain deployed, meaning they benefit the system by acting as passive airbrakes which help slow down the vehicle to its fully defined position and velocity endpoint state. This optimal control problem also contains a glide slope path constraint, creating a canonical region of allowable flight profiles for the trajectory, with the vertex of the cone located at the final desired position. This optimal control problem is also no longer a minimum time problem, but rather a minimum integrated thrust.

We can solve the above defined optimal control problem through the HAMVET method, as has been performed in the previous trajectories. Since this optimal trajectory is essentially the same as the prior investigated trajectories, only differences will be presented.

### Landing Hamiltonian

The Lagrangian of the Hamiltonian is mostly equivalent to the previously derived Lagrangian of the Hamiltonian as described in. However, Eqns. 26 and 28 change due to the additional running cost and path constraint.

$$F(\mathbf{x}, \mathbf{u}) = T$$

$$\boldsymbol{\mu}^T \mathbf{h}(\mathbf{x}, \mathbf{u}) = \mu_T T + \mu_\varphi \varphi + \mu_\beta \beta + \mu_{\delta_E} \delta_E + \mu_{\delta_N} \delta_N + \mu_{\gamma_{gs}} \gamma_{gs}(\mathbf{x}, \mathbf{p})$$

The KKT condition for  $\mu_{\gamma_{gs}}$  will come into play during the derivation of the adjoint equations.

### Landing Hamiltonian Minimization

Performing the HMC for the Landing trajectory is equivalent to the Ascent, Corridor, and Reentry trajectories' HMC, except for  $\frac{\partial \bar{H}}{\partial T}$  due to the additional running thrust cost state. The HMC will not, in its entirety, be repeated here, but  $\frac{\partial \bar{H}}{\partial T}$  is rewritten below as it now applies to the Landing trajectory.

$$\frac{\partial \bar{H}}{\partial T} = 0 = 1 + \lambda_{v_x} \left( \frac{\cos \varphi}{m} \right) + \lambda_{v_y} \left( \frac{\sin \varphi \sin \beta}{m} \right) + \lambda_{v_z} \left( \frac{\sin \varphi \cos \beta}{m} \right) + \mu_T$$

The KKT conditions for the control path constraints are also equivalent to Eqns. 30-33, with exception to  $\mu_T$  which is now

$$\mu_T \begin{cases} \leq 0 & \text{if } T = 0 \text{ N} \\ = 0 & \text{if } 0 \text{ N} < T < 39.24 \text{ N} \\ \geq 0 & \text{if } T = 39.24 \text{ N} \end{cases}$$

due to the change to the inequality on the allowable minimum thrust.

### Landing Adjoint Equations

The Adjoint equations for the Landing trajectory are no longer equivalent to those described in the Ascent, and Reentry trajectories due to the addition of the path constraint. However, since the path constraint contains only position states, there is no effect to the  $\dot{\lambda}_{v_x}$ ,  $\dot{\lambda}_{v_y}$ , and  $\dot{\lambda}_{v_z}$  co-state differential equations.

Taking the partial derivative of the remaining  $\mathbf{x}$  position states for the adjoint, we get

$$\frac{\partial \bar{H}}{\partial r_x} = -\dot{\lambda}_{r_x} = \frac{\mu_{\gamma_{gs}} \sec^{-1} \left( \frac{r_x^u - a_l}{\sqrt{(r_y^u - b_l)^2 + (r_z^u - c_l)^2}} \right)^2}{\sqrt{(r_y^u - b_l)^2 + (r_z^u - c_l)^2}}$$

$$\frac{\partial \bar{H}}{\partial r_y} = -\dot{\lambda}_{r_y} = \frac{-\mu_{\gamma_{gs}} \sec^{-1} \left( \frac{r_x^u - a_l}{\sqrt{(r_y^u - b_l)^2 + (r_z^u - c_l)^2}} \right)^2 (r_x^u - a_l)(r_y^u - b_l)}{\left( (r_y^u - b_l)^2 + (r_z^u - c_l)^2 \right)^{\frac{5}{2}}}$$

$$\frac{\partial \bar{H}}{\partial r_z} = -\dot{\lambda}_{r_z} = \frac{-\mu_{\gamma_{gs}} \sec^{-1} \left( \frac{r_x^u - a_l}{\sqrt{(r_y^u - b_l)^2 + (r_z^u - c_l)^2}} \right)^2 (r_x^u - a_l)(r_z^u - c_l)}{\left( (r_y^u - b_l)^2 + (r_z^u - c_l)^2 \right)^{\frac{5}{2}}}$$

Note that in the previous trajectory problems adjoint equations above were constant in time. Now, due to the additional glide slope path constraint, the position co-vector states change as a function of time as the vehicle positionally translates, but only if  $\mu_{\gamma_{gs}}$  is non-zero. From the KKT condition, we also can infer the following in Eqn. 49.

$$\mu_{\gamma_{gs}} \begin{cases} \leq 0 & \text{if } \gamma_{gs}(\mathbf{x}, \mathbf{p}) = 85 \text{ deg} \\ = 0 & \text{if } 85 \text{ deg} < \gamma_{gs}(\mathbf{x}, \mathbf{p}) < 90 \text{ deg} \\ \geq 0 & \text{if } \gamma_{gs}(\mathbf{x}, \mathbf{p}) = 90 \text{ deg} \end{cases} \quad (49)$$

Since the glide slope cannot mathematically be greater than  $90 \text{ deg}$ , then we should expect to see  $\mu_{\gamma_{gs}} \leq 0$ .

### Landing Transversality Conditions

The Endpoint Lagrangian is for the Landing trajectory is defined differently than the previous trajectories due to the removal of the time minimization and the fully defined final position and velocity states.

$$e(\mathbf{x}_0, \mathbf{x}_f) = \begin{bmatrix} \mathbf{v}_0 - \mathbf{v}^0 \\ \mathbf{r}_0 - \mathbf{r}^0 \\ \mathbf{v}_f - \mathbf{v}^f \\ \mathbf{r}_f - \mathbf{r}^f \end{bmatrix} \Rightarrow \mathbf{v} \in \mathbb{R}^{12}$$

From the proposed optimal control problem, we know  $E(t_f) = 0$ , therefore our Endpoint Lagrangian becomes

$$\bar{E}(\mathbf{x}_0, \mathbf{x}_f, \mathbf{v}, t_f) = [v_1 \ v_2 \ v_3] \mathbf{v}_0 + [v_4 \ v_5 \ v_6] \mathbf{r}_0 + [v_7 \ v_8 \ v_9] \mathbf{v}_f + [v_{10} \ v_{11} \ v_{12}] \mathbf{r}_f$$

The above Endpoint Lagrangian can be used to find the terminal transversality conditions by using Eqn. 36.

$$\frac{\partial \bar{E}}{\partial v_{f,x}} = \lambda_{v_{f,x}}(t_f) = v_7$$

$$\frac{\partial \bar{E}}{\partial v_{f,y}} = \lambda_{v_{f,y}}(t_f) = v_8$$

$$\frac{\partial \bar{E}}{\partial v_{f,z}} = \lambda_{v_{f,z}}(t_f) = v_9$$

$$\frac{\partial \bar{E}}{\partial r_{f,x}} = \lambda_{r_{f,x}}(t_f) = v_{10}$$

$$\frac{\partial \bar{E}}{\partial r_{f,y}} = \lambda_{r_{f,y}}(t_f) = v_{11}$$

$$\frac{\partial \bar{E}}{\partial r_{f,z}} = \lambda_{r_{f,z}}(t_f) = v_{12}$$

With the complementarity condition, we know that all  $\mathbf{v}$  above are unrestricted in value, and therefore the transversality conditions provide no additional mathematical information.

### Landing Hamiltonian Value Conditions

The HVC for the Landing trajectory is found by performing Eqn. 38 on the Landing Endpoint Lagrangian. However, due to it not being a function of the final time, the expected final Hamiltonian value is

$$\mathcal{H}[\@t_f] = -\frac{\partial \bar{E}}{\partial t_f} = 0$$

### Landing Hamiltonian Evolution Equation

Like the Ascent, Corridor, and Reentry trajectories, due to the Hamiltonian not being an explicit function of time, the HEE is equivalent to Eqn. 40.

### Landing Trajectory Results

The Landing trajectory from the Optimal Problem 4: Landing trajectory problem formulation was implemented in DIDO and solved for under the previously described initial, endpoint, and path constraints, leading to Fig. 26, which contains the solved position and velocity states of the vehicle. Note that this figure also contains a post-simulation fidelity check as previously described in the Ascent trajectory's results section. A plot of the glide slope can also be shown, illustrated in Fig. 27, to prove the path constraint more easily was met.

Furthermore, Fig. 28 of the control vector can be made as well, which shows the engine control effector state as non-zero, and the grid fin control effector states as zero, per the path constraints.

There are a few elements of note about the above solution that prove the solution matches what was desired by the original formulation:

- The initial states are as defined in the optimal control problem, which are equivalently the final states of the Reentry trajectory. The vehicle starts at an initial velocity of  $\mathbf{v}_{uc}^u = [-31.56 \quad -3.99 \quad 0.17]^T m/s$  in the UEN frame. The initial position in the UEN frame is  $\mathbf{r}_{uc}^u = [100.0 \quad 2.44 \quad -0.22]^T m$ .

- As the vehicle descends, it implements some force from its engine to keep the vehicle within the defined cone, such that it does not exceed its glide slope path constraint. The final glide slope angle going to zero in Fig. 27 is due to the vehicle being at the final desired position, therefore making the angle hard to define. Otherwise, the glide slope path constraint was met.
- The vehicle descends  $\sim 90 m$  and reaches the desired endpoint conditions in 4.30 s. The final position is  $\mathbf{r}_{uc}^u = [10.19 \quad 0.0 \quad 0.0]^T m$  and the final velocity is  $\mathbf{v}_{uc}^u = [-0.95 \quad 0.37 \quad -0.21]^T m$ , which is close to what was desired for the start of the T=W trajectory.
- To minimize thrust usage, the vehicle waits until  $\sim 2$  seconds have passed in the Landing trajectory before setting a full throttle condition, sometimes known in industry as a “decel” burn.
- The control vector elements never exceed their control vector path constraints.
- The feasibility simulation somewhat closely mimics the provided DIDO solution, indicating the solution is applicable to the system of interest. Deviations between the two would reduce by increasing the node count of the DIDO tool.

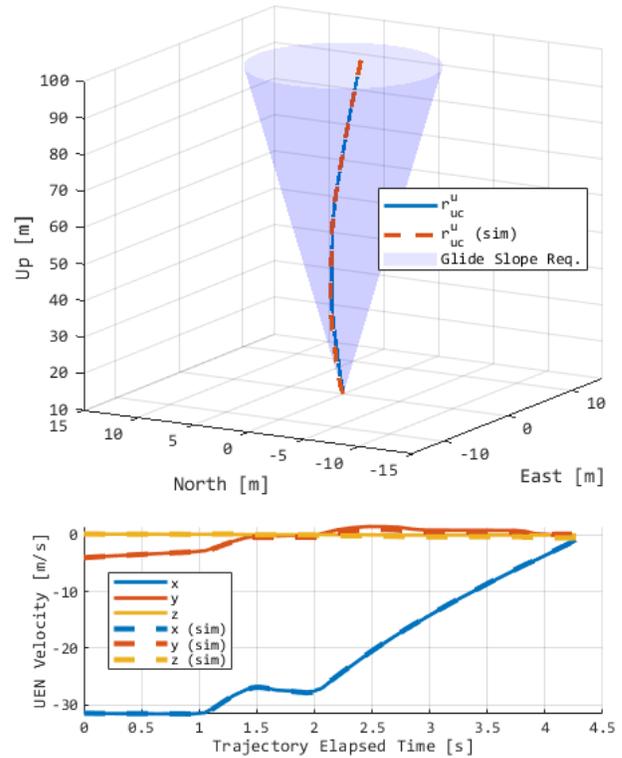


Fig. 26 Landing state and feasibility simulation states

Analysis will now turn to V&V of the optimality. One point of V&V is an investigation of the Hamiltonian evolution from DIDO. The DIDO Hamiltonian is shown below in Fig. 29.

The Hamiltonian supports a few conclusions from the previous section. Since the optimal control problem is no longer

minimum time, the above supports the theory by being close to a final value of 0. From the HEE, it can also be seen that the Hamiltonian does not evolve through time (relatively). This solution supports the theory. The above was solved with 70 nodes within DIDO in order to provide a closer solution to  $H[@t_f] = 0$ , whereas many of the other trajectories were solved with 50 nodes. The feasibility solution, as it applies to the theory, would theoretically be improved if this node count increased even more, but the above solution is acceptable for this application.

Investigation can also be made into the HMC, where the stationary and complimentary conditions are added into Hamiltonian to minimize path constraints on controls. Fig. 30 illustrates the KKT co-vectors for the path constraints.

The complementarity conditions during the optimal control trajectory support the path constraints as defined in the conditionals previously illustrated while minimizing the Hamiltonian. The thrust is shown to be non-zero most of the time, and this is due to the “bang-bang” control logic implemented from the cost functional. Furthermore, looking at the  $\varphi$  and  $\beta$  controls during the Landing trajectory, it clearly shows saturation of the control limits at specific points line up with the KKT co-vector state deviations from 0. Since the Landing trajectory completely constrains the grid fin control effectors, it makes sense that their control co-vector is latched at zero for all time. Lastly, the  $\mu_{\gamma_{gs}}$  path constraint co-vector is only non-zero when the glide slope touches the 85 deg limit as seen from Fig. 27, which matches our expected conditional statement in Eqn. 49. Furthermore, since the glide slope cannot mathematically be greater than 90 deg, then based on the KKT in Eqn. 49, we see supporting evidence that  $\mu_{\gamma_{gs}}$  is only ever  $\leq 0$  during the trajectory.

The co-states can also be investigated to help V&V the solution in comparison with the theory. The co-states are shown in Fig. 31. The co-states in this problem are governed by the derived adjoint equations. The vertical and east velocity co-states are non-zero, which is expected given the bulk of the trajectory is resolved to the Up-East plane. The position co-states are seemingly linear and can be supported as so given the  $\mu_{\gamma_{gs}}$  value is zero for most of the trajectory. Looking back at the adjoint equations for Landing velocity, a zero value for  $\mu_{\gamma_{gs}}$  would produce a constant position costate. Once the path constraint is limited and  $\mu_{\gamma_{gs}}$  becomes non-zero, shown in Fig. 30, then the velocity co-states are no longer constant and are allowed to change over time.

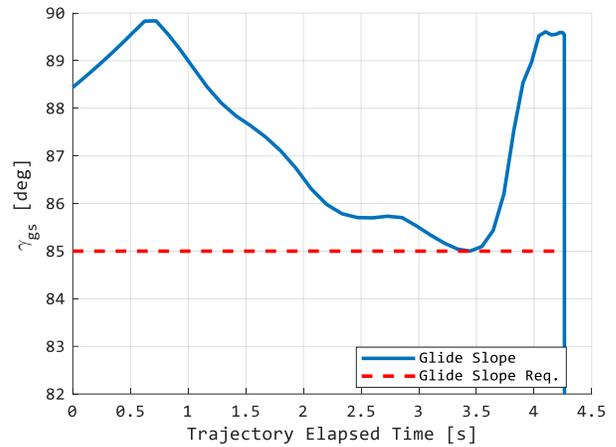


Fig. 27 Landing trajectory glide slope

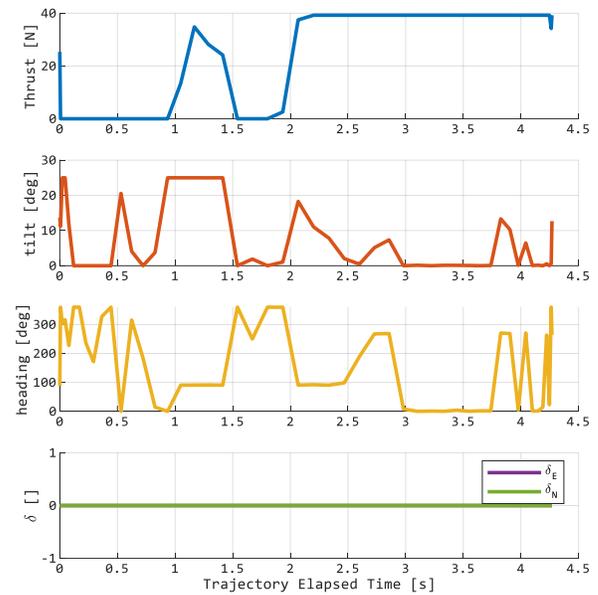


Fig. 28 Landing trajectory control vectors

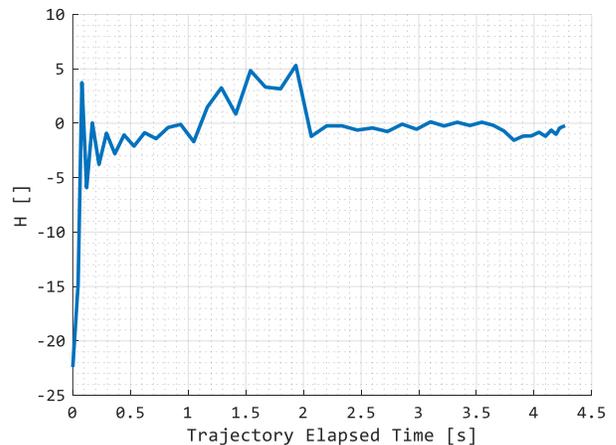


Fig. 29 Landing Hamiltonian

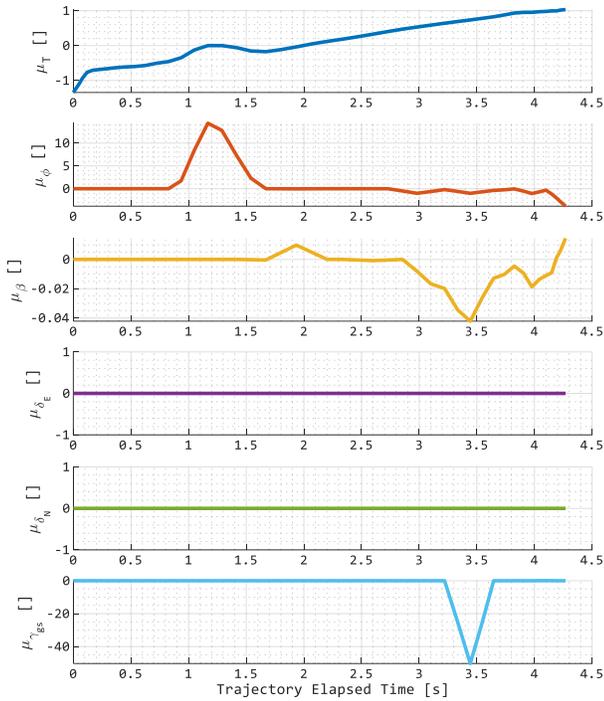


Fig. 30 Landing control co-vectors

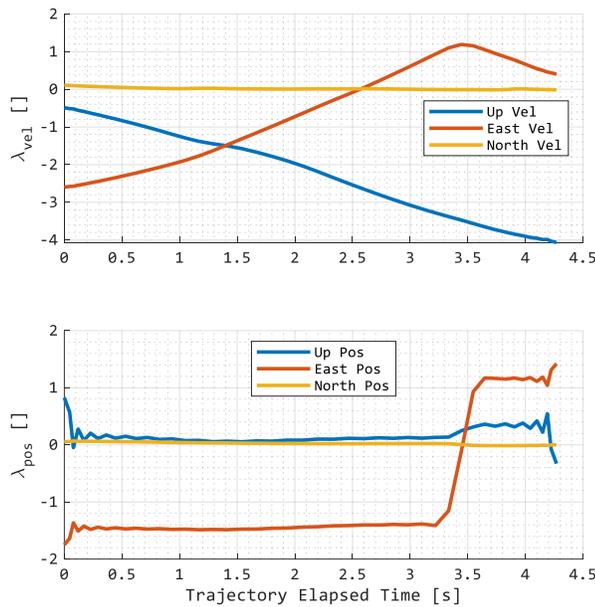


Fig. 31 Landing co-states

### G. $T=W$ Trajectory

The  $T=W$  trajectory is simply a flight phase that continues its dynamics from the Landing trajectory handoff to touchdown. The  $T=W$  begins at a descent rate of  $-1\text{ m/s}$  and at  $10\text{ m}$  directly above the ground as previously described in the desired final state of the Landing trajectory problem statement. In the true system, there would be a closed loop engine controller meant to retain the “slow” vertical descent velocity of the vehicle such that the touchdown event does not exceed any

landing gear structural requirements. There would also be attitude and lateral position control to help constrain the system given unforeseen lateral velocities and/or wind near the pad. Therefore, since this trajectory phase does not require any optimization, it is simply stated here for the future inner-loop control designer’s knowledge. This trajectory then completes the complete mission statement for this vehicle.

## V. CONCLUSION

This section concludes the entirety of the above set of optimal trajectories. It is broken up into three parts: An overview of the entire solution, the conclusion of the project, and areas and improvements for next steps.

### A. Compiled Solution

The previous sections contained the problem formulations and solutions for the various non-optimized/optimized trajectories as they apply to the theoretical vehicle of note in this assignment. Fig. 32 illustrates all trajectories combined into the vehicle’s full mission flight profile. This plot represents an end-to-end liftoff to landing trajectory that the true system could be tasked to follow. First, the vehicle ascends from the pad using its engine. The grid fins are stowed and disabled in this flight region. Then, after one second passes, the vehicle vectors eastward to the Corridor trajectory entrance circular waypoint as fast as the dynamics will allow. Then, the vehicle flies through the cylindrical corridor path as defined in the Corridor trajectory under minimum time and at full throttle. At the end of the Corridor trajectory, the engine is shut off, and the vehicle is allowed to ballistically reach apogee, denoted as the Coast trajectory. Once the vehicle hits apogee, the Reentry trajectory starts, signaling the grid fins to deploy and control the descent path back towards the pad. During the Reentry trajectory, the engine is disabled. At  $100\text{ m}$  altitude, the Landing trajectory starts, and the engine is reenabled and the grid fin control is disabled. The engine then tries to both guide the vehicle to its desired final location above the pad, but also maintain a prespecified glide slope. The set of trajectories then end at the entrance to the  $T=W$  trajectory, which would softly return the vehicle back onto the pad from which it started. Assuming the  $T=W$  time takes  $\sim 10$  seconds, the total mission takes only  $\sim 35$  seconds to complete, with the corridor being successfully flown through at just before 9 seconds total mission elapsed time. This completes the mission profile of this vehicle, assuming the user wanted to measure the atmospheric corridor specified (shown in green in Fig. 32). This trajectory could be easily expanded to fly a myriad of other configurations, spanning different altitudes and waypoints.

### B. Conclusion

This project’s task was to develop an optimal trajectory for a theoretical drone that would allow it to fly through an atmospheric region and then land itself back on the pad from which it came. A complete set of trajectories were successfully developed to do so, utilizing the dynamics and control effectors of the proposed system. These trajectories were all formulated and solved using Pontryagin’s minimization principle to align the theory and provide a feasibility/V&V check for the solution

provided from DIDO. All trajectories developed aligned with the theory, and therefore, the solution in this paper can be deemed optimal given the proposed constraints.

In relation to radiosondes, this drone flew through this atmospheric corridor perfectly, and in as minimum time as possible give the cost functionals and vehicle dynamics as previously described. Assuming an average radiosonde flight ascent velocity, it would have taken an entire minute for a radiosonde to have reached an altitude of 250 m, whereas this drone could do it in just 9 seconds. This is a sixth of the time, and an exact measure of the desired atmospheric path, and large improvement from the radiosonde system.

Furthermore, as an aside, the tooling was setup in such a way that the trajectories could be easily reconfigured. Should the trajectory designer desire a different trajectory than what was developed in this paper, the simple change of a few constraints could allow the trajectory designer to fly the vehicle through different atmospheric corridors and at different altitudes. Furthermore, all trajectories in software were written to depend on the prior trajectory’s final state, meaning that each trajectory will always initialize from where the vehicle was prior.

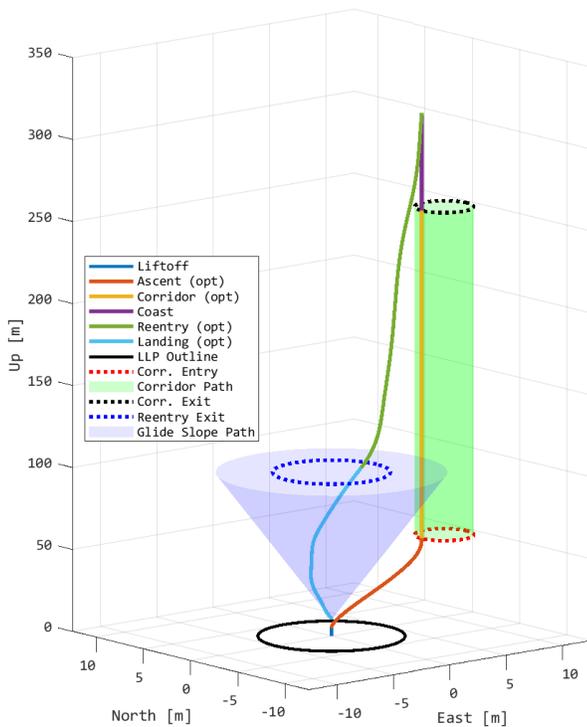


Fig. 32 Full mission

C. Areas of Improvement

Lastly, there are a few areas of improvement. These improvement areas could be seen as future development tasks, or understandable model improvements. Either way, they need to be discussed prior to the completion of this paper.

- Coriolis and centripetal acceleration were not included in the governing dynamics. While a justifiable simplification for the trajectory designed in this paper,

the effects from these sources would be non-negligible if higher altitudes/velocities were desired.

- The grid fins are current disabled in the Landing trajectory. However, most of that trajectory’s dynamic pressure is still strong enough to drive a large amount of effectiveness out of the grid fins. Therefore, moving forward, it would make sense to include the grid fin control effectors along with the engine in the landing algorithm.
- The dynamic pressure in the optimal control solver causes some odd transient behavior in the reentry trajectory. Likely, at this point in the flight, the engineers would be more worried about getting the vehicle back over the pad rather than if the trajectory was completely optimal (feasible) in completing Reentry in minimum time. Therefore, it’s possible the optimal dynamic model for the grid fins, or the design of the Reentry trajectory, may need some additional work.
- The Landing trajectory should contain only a single transient for the engine from zero to full throttle. Some updates to the desired endpoint bounds and control effectors could allow this to happen (such as keeping grid fin control enabled in Landing). Too many on/off thrust events of the engine during Landing could cause undesired effects in the real system.

More improvement ideas may come to mind later, but the above should be sufficient for completion of this project.

REFERENCES

- [1] S. J. Fletcher, “Observations,” in *Data Assimilation for the Geosciences*. Fort Collins, CO, USA: Elsevier Inc. 2017. Ch14, pp 599-626. Accessed: Jan 10, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780128044445000143>
- [2] Mobile Weather Drones, n.d. <https://www.meteomatics.com/en/meteodrones-weather-drones/>
- [3] NOAA deploys new Altius drone into the eye of Hurricane Ian, n.d. <https://www.aoml.noaa.gov/altius-drone-flies-hurricane-ian/>
- [4] B. L. Stevens, F. L. Lewis, R. N. Johnson, *Aircraft Control and Simulation*. Hoboken, New Jersey: John Wiley & Sons, Inc., 2016
- [5] “Use ExtMass and ExtMassCylinder Features with Cables.” [documentation.dsaocean.com/tutorials/Tutorials/PDS-ACP.html](https://documentation.dsaocean.com/tutorials/Tutorials/PDS-ACP.html)
- [6] I. M. Ross, *A Primer on Pontryagin’s Principle in Optimal Control, Second Edition*. Collegiate Publishers, San Francisco, CA, 2015