AE4850 – Applied Dynamic Optimization

Final Class Project – Starship Landing Trajectory

Jeffrey A. Mays

2022-12-15

I. INTRODUCTION & PURPOSE

This effort is an attempt to both satisfy the requirements of AE4850's final class project, as well as have fun deriving a landing trajectory that is similar to SpaceX Starship 2nd stage's previous landing attempts. This project is **not meant** to accurately depict the landing guidance logic, but simply to execute an optimal control problem that relates to an interesting aerospace vehicle. The trajectory solution in this document is exceptionally trivial compared to the one likely used onboard Starship and should be viewed as a "fun toy problem" rather than an accurate representation of the real vehicle's landing guidance logic.

This project scopes the Starship's 2nd stage from its ignition of the Raptor engines and stowing of the aft flaps, causing a pitching torque on the vehicle. It will be up to this optimal control problem to derive the thrust and thrust vector controlled (TVC) gimbal angle of the engine to dampen the dynamics and land the vehicle safely at the desired location. The program used to solve the nonlinear convex optimal control program is DIDO, which will be explained in more detail later.

Figure 1: Starship 2nd stage launch and landing [1]

Given a low order Starship 2nd stage vehicle dynamical model operating in a 2-dimensional (2D) plane, the problem to solve will be to find the optimal landing trajectory from the time of the attitude reorient to the upright soft landing on the ground. An illustration for this problem is shown in Figure 2.

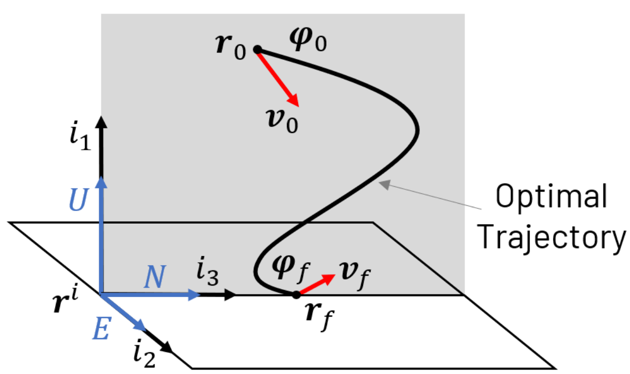

Figure 2: Trajectory to be solved in a quasi-inertial frame

II. RESOURCES

1. Starship Image
2. Starship Wiki
3. Raptor Wiki
4. SN10 Livestream Video
5. Drag Coefficients of a cylinder

III. ASSUMPTIONS & LIMITATIONS

The assumptions and limitations of this work is documented below:

- All Starship and engine properties were derived from publicly available sources [2][3], with no certainty in the accuracy of these parameters.
- The dynamics in this project are 2D in the Up-North plane
- The derived Starship dynamics model is low fidelity
  - Rigid body and propellant assumptions
  - A rudimentary aerodynamic mode was derived. It's intended purpose was not to be correct, but to present dynamic behavior that would likely plague the system during landing.
  - Aerodynamic effectors were assumed to be in their static position at the entrance of the landing trajectory, and would not change during the trajectory of interest, nor be part of the optimized trajectory's control inputs
  - Starship 2nd stage's three engines and TVC's were consolidated into a singular thrust vector in a 2D plane.
- Wind was not considered
- Atmospheric gradients were not considered due to the relatively short altitude change
- Propellant mass is not considered

IV. DERIVING STARSHIP LANDING DYNAMICS

Starship is currently still under development, and likely going through many design cycles of landing strategy for a myriad of planetary and payload configuration landings. Still, assumptions can be made, and a rudimentary model can be easily derived.

Landing Sequence of Events:

Before deriving dynamics, the landing maneuver should be sequenced. Based on YouTube videos [4], we can make a few noteworthy assumptions about the sequence of events for an atmospheric planetary landing:

1. Starship 2nd stage enters the atmosphere horizontally, using its forward and aft flaps to impose aerodynamic forces and torques on the vehicle, allowing it to control attitude and glide itself over a particular landing zone, while also limiting decent velocity.
2. During descent, propellant is bled for a determinant and low-mass landing
3. Three Raptor engine are ignited, allowing up to 1.5e6 lbf of sea level (SL) thrust to be produced.
4. The aft flaps slip out of the freestream, leaving the forward fins fully extended, imposing a positive aerodynamic pitching torque on the vehicle. At the same time, the Raptor engine's thrust vector control (TVC) mounts also impose pitching torques on the vehicle.
5. Starship 2nd stage moves from a horizontal to a vertical attitude in the local geographic frame, noted as the reorient maneuver

6. Starship 2nd stage nulls vehicle velocities and angular rates, and decelerates enough where aerodynamic forces are negligible
7. Starship 2nd stage cuts off one or two Raptor engines
8. Starship 2nd stage lands

Raptor Engines and TVC:

A singular Raptor engine's performance is described here [3]. From this, we can derive the total max thrust from all three raptor engines are 1.5e6 lbf, or $F_{max} = \sim 6.7e6\ N$. Each engine is able to throttle down to 40% of its maximum throttle, meaning the minimum thrust Starship 2nd stage can produce, should it cut off two of its three engines, is 2e5 lbf, or 13.3% of the total max thrust. For simplicity, we will characterize the engine performance for Starship 2nd stage to have the following:

$$T = F_{max} T_{thr}$$

$$0.15 \leq T_{thr} \leq 1.0$$

Furthermore, this engine model will only include a single thrust vector, rather than the three thrust vectored sea level Raptor engines; the thrust vector derived in this project would theoretically make up the combined thrust vector for a higher fidelity simulation. This thrust vector will have bounds of $\pm 12\ deg$, or

$$|\delta_e| \leq 0.2094\ rad$$

Physical Parameters:

Starship is physically 50 m long and has a diameter of 9 m [2]. Furthermore, based on the SN10 YouTube video [4], it appears that only one Raptor engine (sea level config) was burning at the time of "landing," indicating that SpaceX's Starship 2nd stage vehicle weighed less than the thrust of a singular raptor engine. Since the Raptor engine is noted as producing ~5e5 lbf [3], we can assume SpaceX thought a singular engine would have the thrust and throttling capacity to maintain a steady state descent. If we assume a thrust to weight ratio of 1.25, Starship's 2nd stage would weight approximately 4e5 lbf, or have an initial wet mass of ~181e3 kg.

Initial State:

Based on the SN10 10km high altitude flight test, the vehicle took approximately ~75 seconds to fall from 10km to 2km based on the announcer's comments [4]. (Very) roughly speaking, that is an average decent speed of ~107 m/s, not accounting for the acceleration to that speed at the start of the descent phase. Being that it takes about 10 seconds for gravity to move an object from 0 m/s to 100 m/s, we will assume the descent would have taken ~65 seconds should the vehicle had not started from rest. This conservatively gives a constant landing descent velocity of ~123 m/s. Furthermore, looking at Figure 1, it appears the reorient maneuver occurs at approximately ~550 m AGL.

Aerodynamics:

Aerodynamics are extremely difficult to quantify, even for the company developing and operating the vehicle. However, in order to solve this problem, aerodynamics can be included (although in practice they are usually ignored through a justification such as no considerable atmosphere or operation in a low enough alpha region to not be significant for guidance purposes). Under Earth's atmosphere, Starship 2nd stage's final landing trajectory is well bounded (approximately 0m/s to ~120 m/s). Furthermore, the landing maneuver spans a very short atmospheric slice (~500 m), allowing us to assume many atmospheric qualities as constants. This greatly simplifies the aerodynamic envelop needed to qualify in this project. Since we are operating in a 2D plane, only body X and Z aerodynamic force coefficients, and the Y moment coefficient, need to be approximated.

At the time of the reorient maneuver, the forward flaps are extended, and the aft flaps are retracted. This moves the center of pressure (CP) forward towards the forward flaps, creating a positive static margin further aft of the center of mass w.r.t the free stream air. The system becomes dynamically stable in an upright attitude during landing with the engine pointing downward toward the ground. This change in CP is what initially adds energy into the pitch rate state.

Based on [5], we can derive the following aerodynamic tables for starship. Some work was performed to ensure these values produced feasible aerodynamic forces and moments (but not necessarily accurate nor correct). A sinusoid was used to continuously fill gaps between various points of estimation.

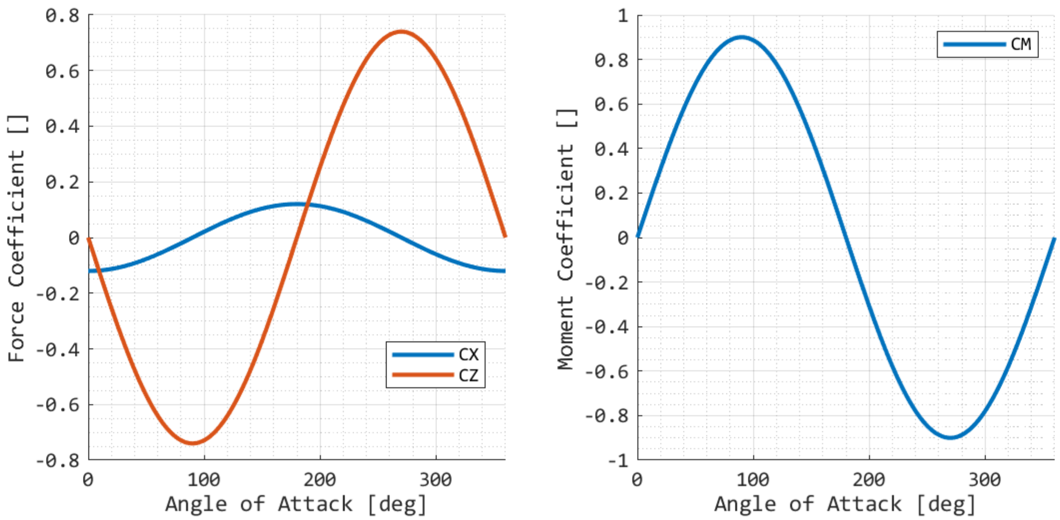

Figure 3: Rudimentary aerodynamic model

Our force and moment equations are then found through the following equations, where q denotes our dynamic pressure. Note that these coefficients were dimensionalized based on the reference in [5], and the parallel flow coefficient was refactored to be equivalent to the perpendicular flow coefficient.

$$\boldsymbol{F}_{aero}^{body} = qL_{ref}D_{ref}\begin{bmatrix} C_X(\alpha) \\ 0 \\ C_Z(\alpha) \end{bmatrix}$$

$$\boldsymbol{M}_{aero}^{body} = qL_{ref}\pi\left(\frac{D_{ref}}{2}\right)^2\begin{bmatrix} 0 \\ C_M(\alpha) \\ 0 \end{bmatrix}$$

L_{ref} and D_{ref} are the reference length and diameter, which are equivalent to the length and diameter of Starship 2nd stage previously noted. Angle of attack can be found as

$$\alpha = \begin{cases} atan2(v_z^b, v_x^b)\ if\ v_z^b > 0 \\ atan2(v_z^b, v_x^b) + 2\pi\ if\ v_z^b \leq 0 \end{cases}$$

Equations of Motion:

Given the following free-body-diagram (FBD) in Figure 4,

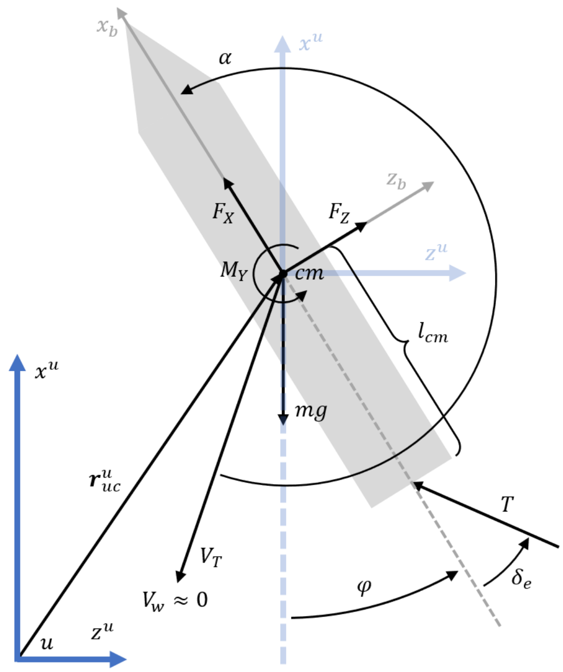

Figure 4: Starship 2nd stage free body diagram

we can derive the equations of motion for a vehicle in a 2D plane (or Up-North in the Up-East-Down (UEN) local geographic frame) as follows. Note the distinction between body, b, and UEN, u, expressed variables. These equations were written to easily transition them to the Hamiltonian-Adjoint-Minimize-Value-Evaluation-Transversality (HAMVET) method.

$$\dot{r}_x^u = v_x^u$$
$$\dot{r}_z^u = v_z^u$$
$$\dot{v}_x^u = -g + \frac{F_{tvc,x}^u + F_{aero,x}^u}{m}$$
$$\dot{v}_z^u = \frac{F_{tvc,z}^u + F_{aero,z}^u}{m}$$
$$\dot{\varphi}^b = \omega^b$$
$$\dot{\omega}^b = \frac{M_{tvc,y}^b + M_{aero,y}^b}{I}$$

The above equations are in their simplified form. Breaking out the components of the above equations yields the following UEN force components and body torque components.

$$F_{tvc,x}^u = F_{max} T_{thr} \cos(\varphi + \delta_e), \qquad F_{tvc,z}^u = F_{max} T_{thr} \sin(\varphi + \delta_e)$$

$$V_T = \sqrt{{v_x^u}^2 + {v_z^u}^2}, \qquad q = \frac{1}{2}\rho V_T^2, \qquad \alpha = \begin{cases} atan2(v_z^b, v_x^b)\ if\ v_z^b > 0 \\ atan2(v_z^b, v_x^b) + 2\pi\ if\ v_z^b \le 0 \end{cases}$$

$$F_{aero,x}^u = q L_{ref} D_{ref} (C_X(\alpha)\cos(\varphi) + C_Z(\alpha)\sin(\varphi)), \qquad F_{aero,z}^u = q L_{ref} D_{ref} (C_Z(\alpha)\cos(\varphi) - C_X(\alpha)\sin(\varphi))$$

$$M_{tvc,y}^b = -l_{cm} F_{max} T_{thr} \sin(\delta_e), \qquad M_{aero,y}^b = q L_{ref} \pi \left(\frac{D_{ref}}{2}\right)^2 C_M(\alpha)$$

V. OPTIMAL CONTROL PROBLEM FORMULATION

a. SCALING AND DERIVING THE OPTIMAL CONTROL PROBLEM

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 all similarly sized and not too egregious, which led to a solution that did not require scaling to be performed. However, the dynamics above were technically scaled from their canonical units to ambiguous scaled units as part of the assignment's requirements. More explicitly, the dynamics' units are as follows.

$$\tilde{r} = \frac{r}{\mathcal{L}}, \quad \tilde{t} = \frac{t}{\mathcal{T}}, \quad \tilde{m} = \frac{m}{\mathcal{M}}, \quad \tilde{v} = \frac{v\mathcal{T}}{\mathcal{L}}, \quad \tilde{\varphi} = \frac{\varphi}{\mathcal{R}}$$

$$\tilde{\omega} = \frac{\omega\mathcal{T}}{\mathcal{R}}, \quad \tilde{g} = \frac{g\mathcal{T}^2}{\mathcal{L}}, \quad \tilde{F} = \frac{F\mathcal{T}^2}{\mathcal{M}\mathcal{L}}, \quad \tilde{M} = \frac{M\mathcal{T}^2}{\mathcal{M}^2\mathcal{L}}, \quad \tilde{I} = \frac{I}{\mathcal{M}\mathcal{L}^2}$$

where $\tilde{r}$, $\tilde{t}$, $\tilde{m}$, $\tilde{v}$, $\tilde{\varphi}$, $\tilde{\omega}$, $\tilde{g}$, $\tilde{F}$, and $\tilde{M}$ are distance, time, mass, velocity, angle, angular rate, linear acceleration, force, moment and inertia "units," which are not described by any physical quantity. Rather, they are all described by their own unique units that can be transformed into the canonical forms from the equations above. More explicitly, scaling our dynamics creates

$$\begin{aligned}
\dot{\tilde{r}}_x^u &= \tilde{v}_x^u \\
\dot{\tilde{r}}_z^u &= \tilde{v}_z^u \\
\dot{\tilde{v}}_x^u &= -\tilde{g} + \frac{\tilde{F}_{tvc,x}^u + \tilde{F}_{aero,x}^u}{\tilde{m}} \\
\dot{\tilde{v}}_z^u &= \frac{\tilde{F}_{tvc,z}^u + \tilde{F}_{aero,z}^u}{\tilde{m}} \\
\dot{\tilde{\varphi}}^b &= \tilde{\omega}^b \\
\dot{\tilde{\omega}}^b &= \frac{\tilde{M}_{tvc,y}^b + \tilde{M}_{aero,y}^b}{\tilde{I}}
\end{aligned}$$

To greatly simplify the notation, **I chose to drop the ~ accent character from the above variables**, which can be seen as appropriate since no change to the equation logic was performed in the scaling of this assignment. In other words, all scaling parameters were unity.

Figure 5 provides a simple diagram of the initial and final states that are desired in this optimal control problem.

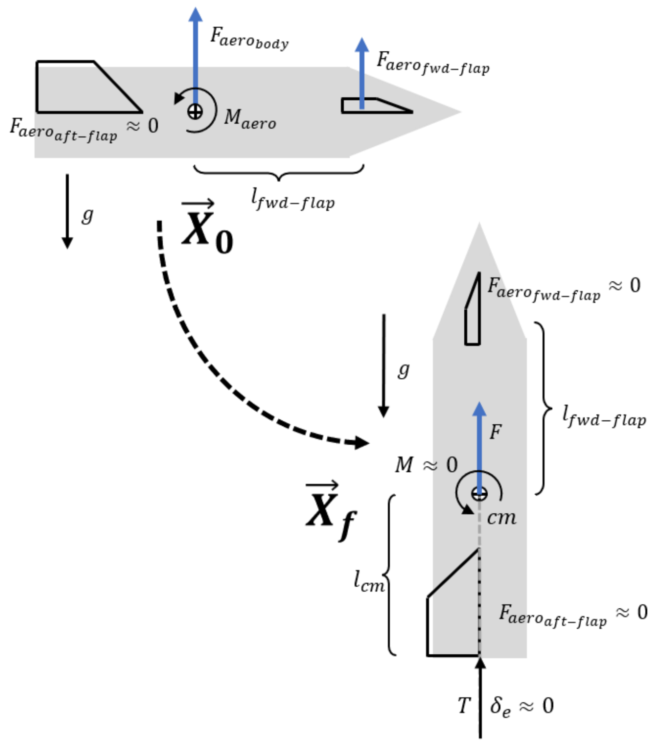

Figure 5: Initial and final free-body-diagrams

Based on previously noted items, as well as Figure 5, we can write the scaled optimal control problem for Starship 2[nd] stage landing concisely as

$$\boldsymbol{x}^T := [r_x \quad r_z \quad v_x \quad v_z \quad \varphi \quad \omega]^T \in \mathbb{R}^6$$
$$u^T := [T_{thr} \quad \delta_e]^T \in \mathbb{U}^2 := \{T_{thr} \in \mathbb{R} : 0.15 \leq T_{thr} \leq 1.0, \delta_e \in \mathbb{R} : |\delta_e| \leq 0.2094\}$$

Minimize $$J[\boldsymbol{x}(\cdot), \boldsymbol{u}(\cdot), t_f] = \frac{1}{2}\int_0^{t_f} \omega_{thr} T_{thr}^2 + \omega_{\delta_e}\delta_e^2 \quad dt$$

subject to

$$\dot{r}_x^u = v_x^u$$
$$\dot{r}_z^u = v_z^u$$
$$\dot{v}_x^u = -g + \frac{F_{tvc,x}^u + F_{aero,x}^u}{m}$$
$$\dot{v}_z^u = \frac{F_{tvc,z}^u + F_{aero,z}^u}{m}$$
$$\dot{\varphi} = \omega$$
$$\dot{\omega} = \frac{M_{tvc,y}^b + M_{aero,y}^b}{I}$$
$$t_0 = 0$$
$$\left(r_{x_0}, r_{z_0}, v_{x_0}, v_{z_0}, \varphi_0, \omega_0\right) = \left(550, 0, \ -123, 0, -\frac{\pi}{2}, 0\right)$$
$$\left(r_{x_f}, r_{z_f}, v_{x_f}, v_{z_f}, \varphi_f, \omega_f\right) = (l_{cm}, 0, 0, 0, 0, 0)$$

where the constant parameters are given as the following. These values should also be scaled prior to integrating into any optimal control problem solvers.

$$F_{max} = 6.7e6\ N, \qquad g = 9.81\frac{m}{s^2}, \qquad I = 3.7708e7\ kg - m^2$$

$$l_{cm} = 20\ m, \qquad \rho = 1.225\frac{kg}{m^3}, \qquad \omega_{thr},\ \omega_{\delta_e} = (1,1)$$

Note that the final desired r_{x_f} (corresponding to above ground level altitude) is l_{cm}. The equations of motion for this problem correspond to the center of mass (COM) of the vehicle in the UEN frame. The distance from the engine gimbal point to the COM is equivalent to the distance from the landing legs to the COM, meaning our optimal control problem produces a result that lands the vehicle correctly on its landing legs, as intended.

b. APPLYING PONTRYAGIN'S MINIMIZATION PRINCIPLE

We can solve this optimal control problem through the **HAMVET** method: **H**amiltonian, **A**djoint, **M**inimize, **V**alue, **E**valuation, and **T**ransversality.

1. Hamiltonian

Constructing the Lagrangian of the Hamiltonian for this optimization problem can be completed by manipulating the following equation

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

where $F(\boldsymbol{x},\boldsymbol{u})$ is the Langrange (running) cost element of the cost function, $f(\boldsymbol{x},\boldsymbol{u})$ are the system dynamics, $\boldsymbol{\lambda}^T$ are a vector of co-vectors, and $\boldsymbol{\mu}^T\boldsymbol{u}$ is included from the Karush-Kuhn-Tucker complimentary criterion due to the bounds on the control inputs. Constructing the Lagrangian of the Hamiltonian, we get the following. Note that we drop the cost weights since they are unity, and we also treat the "nonlinear" aerodynamic coefficient table as simply a function of angle of attack for simplicity.

$$\begin{aligned}\bar{H}(\boldsymbol{\mu},\boldsymbol{\lambda},\boldsymbol{x},\boldsymbol{u},t) = {} & \frac{1}{2}T_{thr}^2 + \frac{1}{2}\delta_e^2 + \lambda_{r_x^u}v_x^u + \lambda_{r_z^u}v_z^u \\ & + \lambda_{v_x}\left(-g + \frac{F_{max}T_{thr}\cos(\varphi+\delta_e) + \frac{1}{2}\rho\left(v_x^{u\,2}+v_z^{u\,2}\right)L_{ref}D_{ref}\left(C_X(\alpha)\cos(\varphi)+C_Z(\alpha)\sin(\varphi)\right)}{m}\right) \\ & + \lambda_{v_z}\left(\frac{F_{max}T_{thr}\sin(\varphi+\delta_e) + \frac{1}{2}\rho\left(v_x^{u\,2}+v_z^{u\,2}\right)L_{ref}D_{ref}\left(C_Z(\alpha)\cos(\varphi)-C_X(\alpha)\sin(\varphi)\right)}{m}\right) \\ & + \lambda_\varphi\omega + \lambda_\omega\left(\frac{-l_{cm}F_{max}T_{thr}\sin(\delta_e) + \frac{1}{2}\rho\left(v_x^{u\,2}+v_z^{u\,2}\right)L_{ref}\pi\left(\frac{D_{ref}}{2}\right)^2 C_M(\alpha)}{I}\right) + \mu_{T_{thr}}T_{thr} \\ & + \mu_{\delta_e}\delta_e\end{aligned}$$

2. Minimizing 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 gives us the following

$$\frac{\partial\bar{H}}{\partial T_{thr}} = 0 = T_{thr} + \lambda_{v_x}\left(\frac{F_{max}\cos(\varphi+\delta_e)}{m}\right) + \lambda_{v_z}\left(\frac{F_{max}\sin(\varphi+\delta_e)}{m}\right) + \lambda_\omega\left(\frac{-l_{cm}F_{max}\sin(\delta_e)}{I}\right) + \mu_{T_{thr}}$$

$$\begin{aligned}\frac{\partial\bar{H}}{\partial\delta_e} = 0 = \delta_e + \lambda_{v_x}\left(\frac{-F_{max}T_{thr}\sin(\varphi+\delta_e)}{m}\right) + \lambda_{v_z}\left(\frac{F_{max}\,T_{thr}\cos(\varphi+\delta_e)}{m}\right) + \lambda_\omega\left(\frac{-l_{cm}F_{max}T_{thr}T_{thr}\cos(\delta_e)}{I}\right) \\ + \mu_{\delta_e}\end{aligned}$$

which completes the stationary condition minimization. From the Karush-Kuhn-Tucker complimentary criterion, we can also say

$$\mu_{T_{thr}} \begin{cases} \leq 0 \; if \; T_{thr} = 0.15 \\ = 0 \; if \; 0.15 < T_{thr} < 1.0 \\ \geq 0 \; if \; T_{thr} = 1.0 \end{cases}$$

$$\mu_{\delta_e} \begin{cases} \leq 0 \; if \; \delta_e = -0.2094 \\ = 0 \; if \; |\delta_e| \leq 0.2094 \\ \geq 0 \; if \; \delta_e = 0.2094 \end{cases}$$

3. **Constructing the Adjoint Equations**

We know our Adjoint equation is our time rate of change of our co-vectors as defined by the negative time rate of change of our Hamiltonian, or

$$\frac{\partial \bar{H}}{\partial \boldsymbol{x}} = -\dot{\boldsymbol{\lambda}}$$

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

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

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

$$\frac{\partial \bar{H}}{\partial v_x^u} = -\dot{\lambda}_{v_x^u} = \lambda_{r_x^u} + \lambda_{v_x^u}\left(\frac{\rho v_x^u L_{ref} D_{ref}(C_X(\alpha)\cos(\varphi) + C_Z(\alpha)\sin(\varphi))}{m}\right) + \lambda_{v_z}\left(\frac{\rho v_x^u L_{ref} D_{ref}(C_Z(\alpha)\cos(\varphi) - C_X(\alpha)\sin(\varphi))}{m}\right) + \lambda_\omega\left(\frac{\rho v_x^u L_{ref}\pi\left(\frac{D_{ref}}{2}\right)^2 C_M(\alpha)}{I}\right)$$

$$\frac{\partial \bar{H}}{\partial v_z^u} = -\dot{\lambda}_{v_z^u} = \lambda_{r_z^u} + \lambda_{v_x^u}\left(\frac{\rho v_z^u L_{ref} D_{ref}(C_X(\alpha)\cos(\varphi) + C_Z(\alpha)\sin(\varphi))}{m}\right) + \lambda_{v_z}\left(\frac{\rho v_z^u L_{ref} D_{ref}(C_Z(\alpha)\cos(\varphi) - C_X(\alpha)\sin(\varphi))}{m}\right) + \lambda_\omega\left(\frac{\rho v_z^u L_{ref}\pi\left(\frac{D_{ref}}{2}\right)^2 C_M(\alpha)}{I}\right)$$

$$\frac{\partial \bar{H}}{\partial \varphi} = -\dot{\lambda}_\varphi = \lambda_{v_x}\left(\frac{-F_{max}T_{thr}\sin(\varphi + \delta_e) - \frac{1}{2}\rho\left({v_x^u}^2 + {v_z^u}^2\right)L_{ref}D_{ref}(C_X(\alpha)\sin(\varphi) + C_Z(\alpha)\cos(\varphi))}{m}\right) + \lambda_{v_z}\left(\frac{F_{max}T_{thr}\cos(\varphi + \delta_e) - \frac{1}{2}\rho\left({v_x^u}^2 + {v_z^u}^2\right)L_{ref}D_{ref}(C_Z(\alpha)\sin(\varphi) - C_X(\alpha)\cos(\varphi))}{m}\right) + \lambda_\omega\left(\frac{-l_{cm}F_{max}T_{thr}\cos(\delta_e)}{I}\right)$$

$$\frac{\partial \bar{H}}{\partial \omega} = -\dot{\lambda}_\omega = \lambda_\varphi$$

This completes the adjoint equations for the system.

4. **Transversality Conditions**

Our endpoint errors at the time t_f can be written as

$$\boldsymbol{e}(x_f) = \begin{bmatrix} r^u_{x_f} - r^{u\,f}_x \\ r^u_{z_f} - r^{u\,f}_z \\ v^u_{x_f} - v^{u\,f}_x \\ v^u_{z_f} - v^{u\,f}_z \\ \varphi_f - \varphi^f \\ \omega_f - \omega^f \end{bmatrix} \Longrightarrow \nu \in \mathbb{R}^6$$

The Endpoint Lagrangian is defined as

$$\bar{E}(\boldsymbol{x}_f, \boldsymbol{\nu}, t_f) = E(t_f) + \boldsymbol{\nu}^T \boldsymbol{e}(x_f)$$

where $E(t_f)$ is our endpoint cost, $\boldsymbol{\nu}$ are our endpoint convectors, and $\boldsymbol{e}(x_f)$ are our endpoint error equations. From the proposed optimal control problem, we know $E(t_f) = 0$, therefore

$$\bar{E}(\boldsymbol{x}_f, \boldsymbol{\nu}, t_f) = \nu_1 \left(r^u_{x_f} - r^{u\,f}_x\right) + \nu_2 \left(r^u_{z_f} - r^{u\,f}_z\right) + \nu_3 \left(v^u_{x_f} - v^{u\,f}_x\right) \\ + \nu_4 \left(v^u_{z_f} - v^{u\,f}_z\right) + \nu_5 (\varphi_f - \varphi^f) + \nu_6 (\omega_f - \omega^f)$$

where ν are denoted with the respective state equation number they are associated with. When simplified and values are added, we get

$$\bar{E}(\boldsymbol{x}_f, \boldsymbol{\nu}, t_f) = \nu_1 \left(r^u_{x_f} - l_{cm}\right) + \nu_2 r^u_{z_f} + \nu_3 v^u_{x_f} \\ + \nu_4 v^u_{z_f} + \nu_5 \varphi_f + \nu_6 \omega_f$$

The transversality condition notes that

$$\boldsymbol{\lambda}(t_f) = \frac{\partial \bar{E}}{\partial \boldsymbol{x}_f}$$

therefore, giving us the answers to the partial derivatives of the Endpoint Lagrangian with respect to the state vector at the final conditions

$$\frac{\partial \bar{E}}{\partial r^u_{x_f}} = \nu_1, \qquad \frac{\partial \bar{E}}{\partial r^u_{z_f}} = \nu_2, \qquad \frac{\partial \bar{E}}{\partial v^u_{x_f}} = \nu_3$$

$$\frac{\partial \bar{E}}{\partial v^u_{z_f}} = \nu_4, \qquad \frac{\partial \bar{E}}{\partial \varphi_f} = \nu_5, \qquad \frac{\partial \bar{E}}{\partial \omega_f} = \nu_6$$

The above equations do not yield any additional information about the boundary value to be solved, which is appropriate given that all the final states are already explicitly defined in the proposed optimal control problem (no ambiguity).

5. **Hamiltonian Evolution and Value Conditions**

The Hamiltonian Value condition states

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

The partial derivative of the Endpoint Lagrangian with respect to the final time is null since our optimal control problem is not a minimum time problem, therefore

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

When solving for optimality, we should expect to see the optimal solution to be fairly close to zero at the final time. The Hamiltonian Evolution is then

$$\frac{d\mathcal{H}}{dt} = \frac{\partial \bar{H}}{\partial t} = 0$$

since $\bar{H}$ is not a function of time.

VI. DIDO IMPLEMENTATION

In this assignment, we are using a program called "DIDO" to solve the optimal control problem formulated earlier. To solve this optimal control problem using DIDO, scripts and lookup tables were developed and are listed below. All code used to support DIDO and the rest of this assignment will be shown in this report's Appendix.

1. `StarshipProblem.m`
2. `StarshipPreamble.m`
3. `StarshipPath.m`
4. `StarshipEvents.m`
5. `StarshipDynamics.m`
6. `StarshipCost.m`
7. `StarshipConstants.m`
8. `StarshipDynamics_Wrapper.m`
9. `create_aero_tables.m`
10. `eulR_to_DCM.m`

Furthermore, an animation script was created to animate the trajectory. The animation video file will be submitted with this report.

1. `animate_2D_starship.m`

Once the DIDO infrastructure was setup, an internal DIDO tool was used to verify that the problem had been correctly formatted and was ready to attempt solving.

```
>> check(Starship);


***************
Congratulations! You MAY have successfully completed STEP # 1
in using DIDO.
Please move on to STEP # 2:
    Apply Pontryagin"s Principle to check your
    problem formulation.
    Use pencil and paper.
    Ignore this step and die!
    REMEMBER: You are trying to solve an optimal control problem.
    Aren"t you?
****************
>>
```

Figure 6: DIDO Check

The last remaining step before running DIDO was to determine how many nodes to place throughout the trajectory. 100 nodes were decided over an iterative method, mainly to provide enough fidelity/smoothness in the produced trajectories. Solver time was not an explicit requirement, although the author noted that going much beyond 100 nodes caused an exponential increase in the required solver time.

VII. RESULTS

Once the problem was formulated in DIDO, the Starship 2nd stage landing trajectory could be generated. The results of this optimal control problem will be presented in the following sections as topics of discussion over optimality. First, the Hamiltonian will be illustrated to show that it aligns with the Hamiltonian Value condition. Then the solver time will be investigated, followed by figures of the states and co-vectors of the system. Then, Verification and Validation (V&V) will be performed by integrating the dynamics in a separate solver, while using the optimally found control vectors.

Note that explicitly unscaled figures do not exist in this section. Since the scaling was unity, this addition would make the unscaled plots redundant. The canonical form of the states and control, as they were originally derived for this problem, are shown.

1. Hamiltonian

The Hamiltonian Value condition stated we should have a value near zero at the final time. According to Figure 7, this appears to be true, as the final value is approximately $-0.1E-3$.

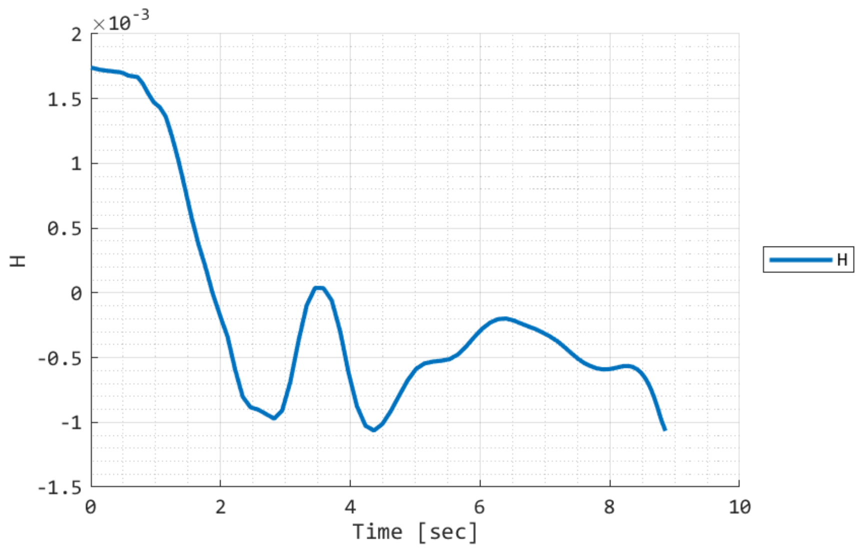

Figure 7: Hamiltonian during solved optimal trajectory

2. Solver Time

This optimal control problem had no set requirements for how long it could take before the solution converged. Therefore, no explicit care was given to shorten the run time. For this formulated problem, it took on average ~50.5 seconds to provide a solution, which is appropriate for this particular problem.

3. Solution states

Viewing the states, with focus on the initial and final states of the system, Figure 8 provides a clear illustration of the system achieving its desired goal of using its control effectors, along with the help from vertically stabilizing aerodynamics, to reorient the vehicle from the initial horizontal attitude ($\varphi = -90\ deg$) to the final vertical attitude ($\varphi = 0\ deg$). The vehicle also decelerated the system such that it finished the trajectory with the landing legs at $(0\ m, 0\ m)$ in the local geographic Up-North frame, along with zero vertical and horizontal velocities. The trajectory took approximately 8.85 seconds to perform.

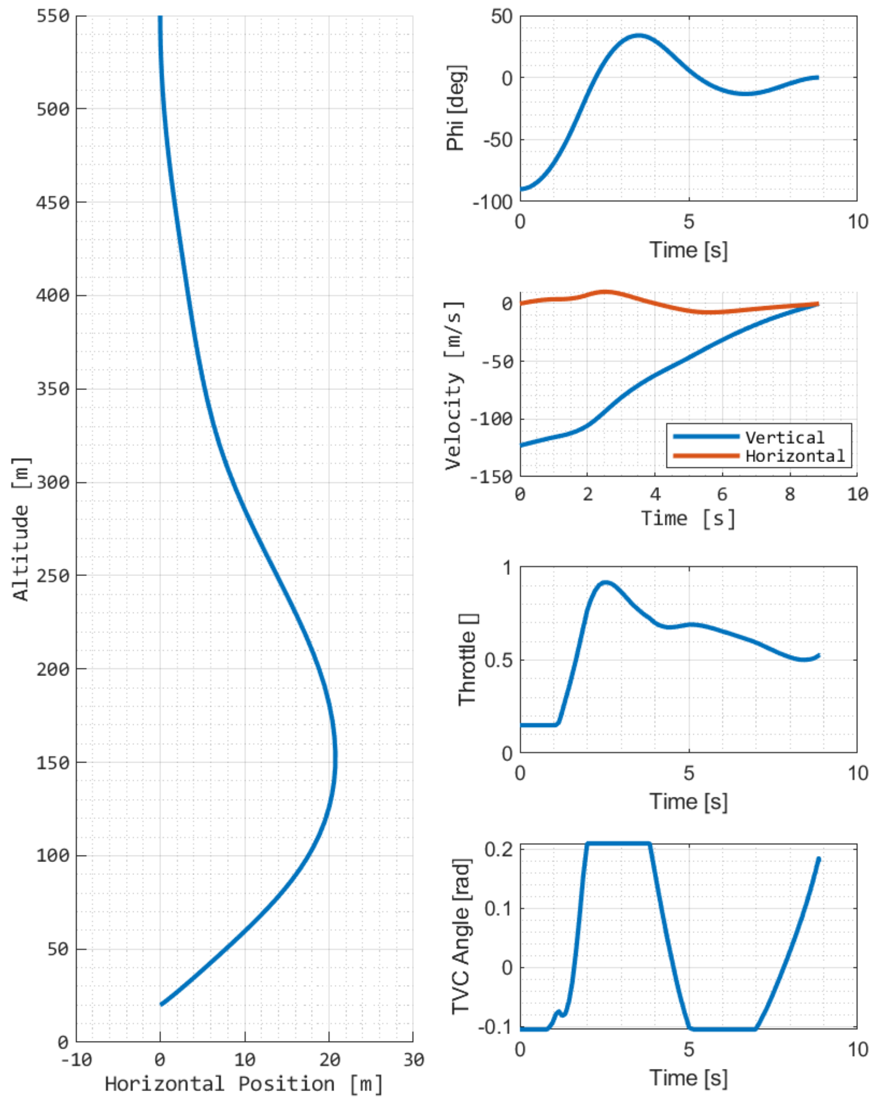

Figure 8: Dynamic states and control trajectories of the system

Furthermore, we can also plot the co-vector dynamics as well. Note that $\lambda_{r_x^u}$ and $\lambda_{r_v^u}$ are shown as constants in the following plot. Furthermore, λ_φ vaguely mimics λ_ω as expected from the derivation of the adjoint equations.

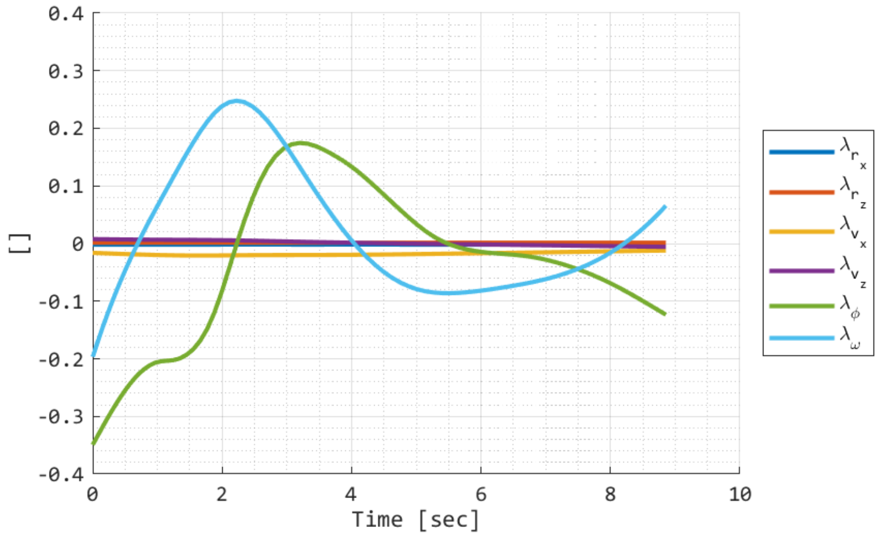

Figure 9: Co-vectors

4. V&V

For verification and validation, the control vector from the optimally solved optimal control problem was taken and injected into a separate tool that also contained the Starship 2nd stage dynamics. The optimal control trajectory was linearly interpolated to provide the system with a higher fidelity lookup. Using Maltab's ODE45 propagator, the system produced the following "open-loop" simulation.

Figure 10 illustrates the system as propagated by the optimal control solution. Note that while we do not see a perfectly mimicked solution, it is close enough to determine that our optimal control problem is solved, and that the solution produced is applicable to the system with the previously described dynamics. Changes to the problem formulation would lead to a less differential comparison (such as increasing the DIDO node count, better control trajectory lookup interpolation, integration bias from the propagator, etc.). This solution is also open loop, meaning it cannot correct for deviations once the dynamics start propagating. Should this problem be required to be used in practice, it is commonplace to either use the pre-solved trajectory as a table lookup, or create an optimal control problem that is linearized or simplified in some way where it can be iteratively solved at a set timestep, allowing it to be able to null out differences between the sensed system states and the optimal trajectory over time.

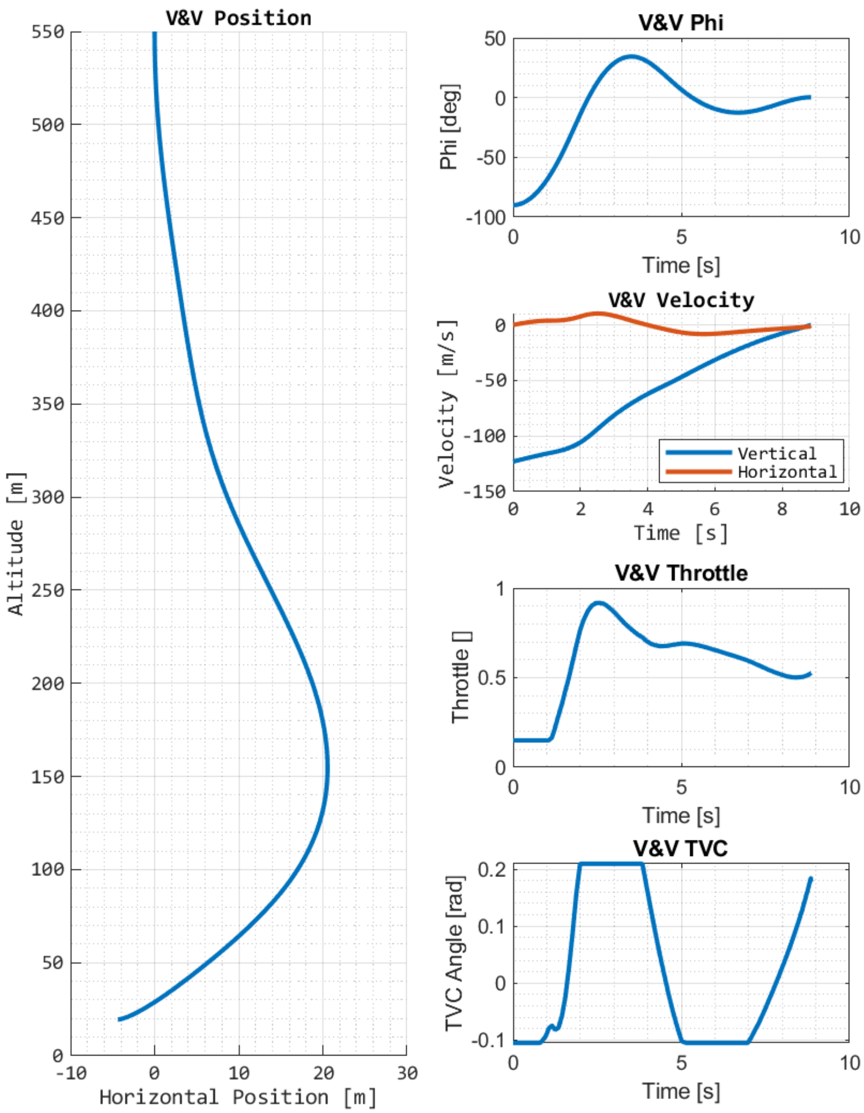

Figure 10: V&V States

VIII. CONCLUSION & FUTURE WORK

In this project, the 6 state equations of motion were derived for a Starship 2nd stage like vehicle that describe the landing segment of its real-life counterpart. Once we derived the equations of motion, an optimal control problem was formulated such that a trajectory would guide the vehicle from its horizontal attitude descent to a vertical attitude soft landing on the ground. We also specified that it needed to do this with as minimum thrust and minimum TVC rotation as possible (as to theoretically preserve fuel and dampen the TVC use and therefore minimize angular velocity). The optimal control problem was first scaled and then solved by the HAMVET process. Then, it was integrated into DIDO, where a solution was produced which was investigated for optimality and then V&V'd. The derived solution was found to be acceptable for the problem statement.

One large pitfall of this project is not having a specified glide slope requirement, a common addition in landing optimal control problems. This would have prevented many of the solutions from descending below the ground during the initial design phase of the problem statement. Furthermore, after some investigation and research, it appears other Entry-Decent-Landing (EDL) optimal control problems are posed in such a way where aerodynamics can be ignored, or at least vastly simplified. It is very plausible that similar justifications were made to the real Starship 2nd stage vehicle, especially if the solution is solved onboard.

This project should fulfill the requirements for AE4850's final project. The next course to be taken related to optimization is AE4881 – Aerospace Trajectory Planning and Guidance. This project could easily be expanded to include far more states, a higher fidelity plant, three dimensional dynamics, inner control loops, Raptor engine cutoff logic, upper atmospheric flap control guidance and control prior to the reorient maneuver, propellant consumption and consumption constraints, pre-landing propellant dumping logic, landing state requirements, glide slope constraints, and/or many more.

Furthermore, an animation of this solved control problem is included with the submission of this report. Please let the author know if the file of note is unable to be viewed. Figure 11 illustrates the animation.

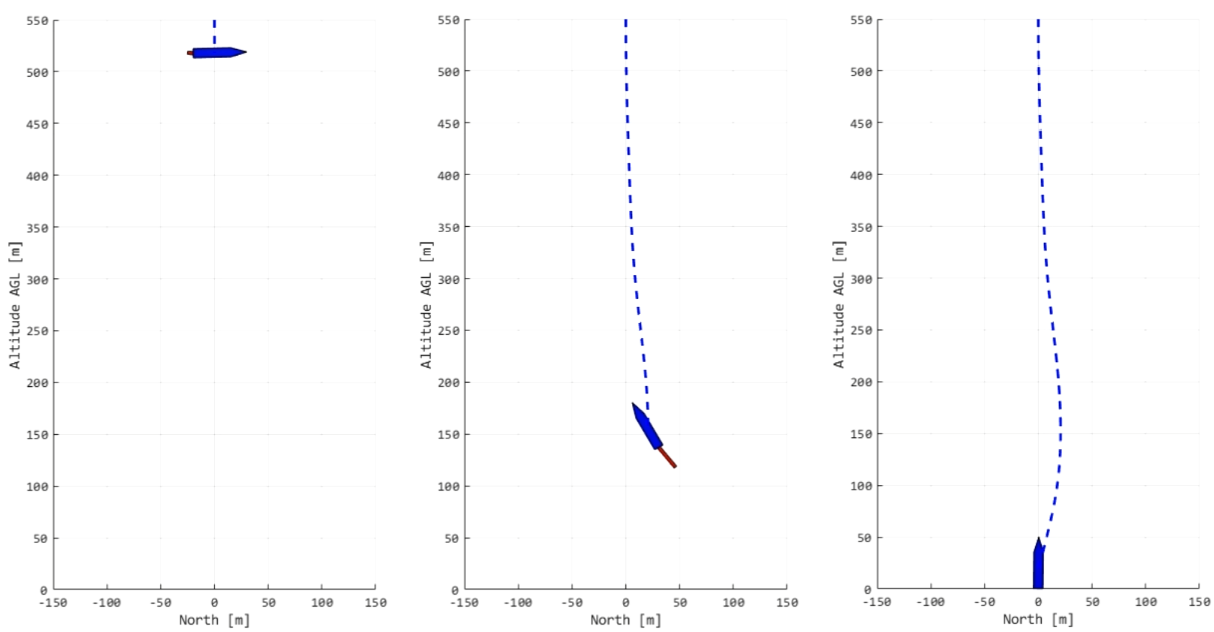

Figure 11: Starship 2nd stage landing animation screen capture

IX. APPENDIX

This version of the document does not contain the project source code over concerns with export control. Thank you for your understanding.