Successive Convexification for Powered Descent Guidance: A Comparative Study of Fuel-Optimal Cost Functions

Stefan Dragos DRAGAN*,[1], Alexandra BOTEZ[1]

*Corresponding author
[1]Agile Systems Engineering,
10 Ing. Zablovschi, Bucharest, Romania,
stefan.dragan@agile-systems.ro, alexandra.botez@agile-systems.ro

DOI: 10.13111/2066-8201.2025.17.3.2

Received: 06 August 2025/ Accepted: 21 August 2025/ Published: September 2025

Abstract: *This paper presents a comparative study of two Successive Convexification (SCvx) formulations with free final time for the powered descent guidance (PDG) of Falcon 9's first stage. SCvx solves non-convex trajectory problems by linearizing dynamics and constraints, then solving convex subproblems via CVX in MATLAB. Two cost functions are investigated—maximizing final mass and minimizing total thrust both aiming to reduce fuel use. The 3 degrees of freedom (3-DoF) model includes realistic constraints such as thrust bounds, gimbal limits, glide slope, vertical landing and a novel implementation of vertical thrust rate constraint into a variable time problem. Simulations reveal how cost function choice affects fuel efficiency and trajectory design.*

Key Words: *Successive Convexification, Powered Descent Guidance, Trajectory Optimization, Falcon 9, First Stage Landing, Convex Programming, MATLAB, CVX, Vertical Landing, Rocket Reusability, Thrust Vectoring, Trust Region, Fuel Optimization*

1. INTRODUCTION

The recovery and reuse of launch vehicle stages has become a central objective in modern spaceflight, significantly reducing the cost of access to space. SpaceX's Falcon 9 first-stage booster exemplifies this capability through a complex autonomous landing procedure that demands precise real-time trajectory optimization under tight physical and operational constraints. To enable such guidance strategies, efficient and reliable algorithms capable of solving non-convex optimal control problems are essential.

This paper investigates two SCvx formulations for solving a 3-DoF powered descent guidance problem representative of Falcon 9 first-stage landings. SCvx addresses non-convex trajectory optimization by solving a sequence of convex subproblems, enabling potential real-time implementations under strict physical constraints.

The two formulations differ in their cost functions: one maximizes final mass, while the other minimizes total thrust - both aiming to reduce fuel consumption. Realistic constraints such as thrust and gimbal limits, glide slope, and vertical landing are enforced. Implemented in MATLAB using CVX, the approaches are compared via simulation to assess their impact on fuel efficiency, trajectory shape, and thrust behavior, offering practical insights for reusable rocket guidance design.

2. SUCCESSIVE CONVEXIFICATION ALGORITHMS ARCHITECTURE

The SCvx algorithm is a specific instance of the broader Sequential Convex Programming (SCP) framework, which addresses non-convex problems through iterative convex approximations. Another SCP-based method is Guaranteed Sequential Trajectory Optimization (GuSTO), which builds on similar principles [1]. All SCP algorithms share a core structure: at each iteration, a convex subproblem is solved and updated based on the current solution. As shown in Figure 1, SCP can be viewed as a predictor-corrector process - evaluating the current trajectory in the forward phase, and, if necessary, refining the approximation in the corrector phase to improve solution accuracy [1].

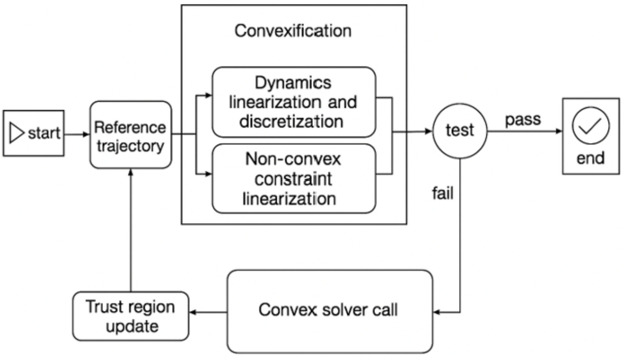

Figure 1. Block diagram of a typical SCP algorithm

SCP methods solve non-convex problems by iteratively refining convex approximations, or subproblems, based on a user-defined reference trajectory. This initial guess, often infeasible, guides local linearization of non-convex elements, while convex parts remain unchanged. Well-designed SCP algorithms ensure feasibility post-linearization and discretize the continuous-time subproblem for numerical solution. A convex solver computes the optimal solution, which is then checked against stopping criteria. If not met, the solution updates the trust region and serves as the next reference. Key trade-offs include how convex approximations are formed, solutions updated, and convergence assessed, all affecting performance and theoretical guarantee [2] s.

The most general form of the problems solved by SCvx is:

$$
\begin{gathered}
\min_{u,p} J(\boldsymbol{x},\boldsymbol{u},\boldsymbol{p}) \\
\text{s.t. } \dot{\boldsymbol{x}} = f(t,\mathbf{x}(t),\mathbf{u}(t),\mathbf{p}) \\
(x(t),p) \in X(t), \\
(u(t),p) \in U(t), \\
s(t,x(t),u(t),p) \leq 0, \\
g_i(x(0),p) = 0, \\
g_f(x(1),p) = 0.
\end{gathered} \tag{1}
$$

where $\mathbf{x} \in \mathbb{R}^n$ represents the state trajectory vector, $\mathbf{u} \in \mathbb{R}^m$ is the control vector and $\mathbf{p} \in \mathbb{R}^d$is a vector of parameters. The function $f: \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^d \to \mathbb{R}^n$ represents the dynamics, assumed to be continuously and differentiable.

Additionally, $\mathbf{x}$ and $\mathbf{u}$ are subject to constraints $\mathbf{x} \in \boldsymbol{X}$ and $\mathbf{u} \in \boldsymbol{U}$ where sets $\boldsymbol{X} \in \mathbb{R}^n$ and $\boldsymbol{U} \in \mathbb{R}^m$ are assumed to be convex. The nonconvex constraints are represented by the continuously differentiable function $s: \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^d \rightarrow \mathbb{R}^n$. The initial and final boundary constraints are introduced in the problem by functions $g_i: \mathbb{R}^n \times \mathbb{R}^d \rightarrow \mathbb{R}^{n_i}$ respectively $g_f: \mathbb{R}^n \times \mathbb{R}^d \rightarrow \mathbb{R}^{n_f}$. For the free final time applications, the problem formed by equations from (2.1) is defined over a normalized period $[0, 1]$ and with the help of a time dilation coefficient, the user can transform this normalized interval into the real time interval $\left[\mathrm{t}_0, \mathrm{t}_f\right]$.

All SCP methodologies work by addressing a series of local convex approximations to the optimization challenge defined by (1), which are referred to as subproblems. As illustrated in Figure 1, the algorithm needs a user-supplied initial trajectory guess. It will be denoted as $\{\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{p}\}$. A very common method for obtaining an initial trajectory is the straight-line interpolation. The first aspect needed to be addressed when performing an interpolation, is the selection of the initial and final states $\mathbf{x}_\mathrm{i}$ and $\mathbf{x}_\mathrm{f}$. In this scenario, those points represent the initial, respectively final boundary conditions of the problem. The state trajectory is defined as a linear interpolation between those two conditions [3]:

$$\boldsymbol{x}(t) = (1 - t)\mathbf{x}_i + t\mathbf{x}_\mathrm{f} \text{ for } t \in [0, 1] \tag{2}$$

But to have a complete set of data for the initial trajectory, the initial control vector needs to be computed. Following the same strategy as above, the user can select the initial and final controls based on the insight given by the physics of the problem.

For example, if the problem concerns the landing of a rocket, one of the states from $\boldsymbol{x}(t)$ will be the mass of the system, so each element of the initial control vector can be computed as $\boldsymbol{u}(t) = m(t)\mathbf{g}$. To complete the image of the trajectory initialization, an input value for $\boldsymbol{p}$ is needed as well. If the problem to be solved is described by a free final time, then $\boldsymbol{p}$ represents a guessed final time of the problem.

Following the structure of the block diagram in Figure 1, after the initial trajectory generation, the next step is convexifying the problem defined by (1). From the perspective of computational efficiency, the most effective approach is to approximate each nonconvex function in the problem using a first-order Taylor expansion [3]. This approximation yields a line tangent to the function at a specific point - hence the term “linearization”. To apply this process, it is necessary to select the points around which the linearization will be performed. This highlights the importance of having an initial reference trajectory. By replacing each nonconvex function with its first-order approximation around the corresponding reference trajectory points, the resulting subproblems become convex.

Before starting to explore the model any further, let us consider the formula of any first order Taylor approximation of a function $g(x)$, $x \in \mathbb{R}$ around a given point $a \in \mathbb{R}$:

$$g(x) \cong g(a) + g^{'}(x - a) \tag{3}$$

In the next formulas, the notation $\mathbf{x}^0, \mathbf{u}^0$ will denote the states and controls from the initial guessed trajectory and $\mathbf{x}, \mathbf{u}$ will now represent vectors. The initial guess will play the role of a from (3) and $f(\mathbf{x}, \mathbf{u})$ will represent the dynamics of the system. By using the first order Taylor Series approximation, the dynamics is linearized as (3):

$$f(\boldsymbol{x}, \boldsymbol{u}) \simeq f(\boldsymbol{x}^0, \boldsymbol{u}^0) + f'(\boldsymbol{x} - \boldsymbol{x}^0, \boldsymbol{u} - \boldsymbol{u}^0) \tag{4}$$

Where the term $g(a)$ from (3) is $f(\boldsymbol{x}, \boldsymbol{u})|_{x^0, y^0}$. Next, the term $g'(x - a)$ from (3) must be constructed, so the first order derivative of f with respect to the states $\boldsymbol{x}$ and the first order derivative of f with respect to the control $\boldsymbol{u}$ are needed:

$$\begin{aligned} \boldsymbol{A} &= \frac{\partial}{\partial x}(f(\boldsymbol{x},\boldsymbol{u}))\Big|_{\mathbf{x^0,u^0}} \\ \boldsymbol{B} &= \frac{\partial}{\partial u}(f(\boldsymbol{x},\boldsymbol{u}))\Big|_{\mathbf{x^0,u^0}} \end{aligned} \tag{5}$$

$\boldsymbol{A}$ represents the state matrix and $\boldsymbol{B}$ the control matrix of the system. They are numerically computed with values from the initial guessed trajectory. With those Jacobians computed, $f'(\boldsymbol{x}-\boldsymbol{x}^0,\boldsymbol{u}-\boldsymbol{u}^0)$ from (4) can be formulated as:

$$\begin{aligned} f'(\boldsymbol{x}-\boldsymbol{x}^0,\boldsymbol{u}-\boldsymbol{u}^0) = \underbrace{\frac{\partial}{\partial \boldsymbol{x}}(f(\boldsymbol{x},\boldsymbol{u}))\Big|_{\boldsymbol{x}^0,\boldsymbol{u}^0}}_{\boldsymbol{A}}\boldsymbol{x} - \underbrace{\frac{\partial}{\partial \boldsymbol{x}}(f(\boldsymbol{x},\boldsymbol{u}))\Big|_{\boldsymbol{x}^0,\boldsymbol{u}^0}}_{\boldsymbol{A}}\boldsymbol{x}^0 + \\ + \underbrace{\frac{\partial}{\partial \boldsymbol{u}}(f(\boldsymbol{x},\boldsymbol{u}))\Big|_{\boldsymbol{x}^0,\boldsymbol{u}^0}}_{\boldsymbol{B}}\boldsymbol{u} - \underbrace{\frac{\partial}{\partial \boldsymbol{u}}(f(\boldsymbol{x},\boldsymbol{u}))\Big|_{\boldsymbol{x}^0,\boldsymbol{u}^0}}_{\boldsymbol{B}}\boldsymbol{u}^0 \\ f'(\boldsymbol{x}-\boldsymbol{x}^0,\boldsymbol{u}-\boldsymbol{u}^0) = \boldsymbol{A}(\boldsymbol{x}-\boldsymbol{x}^0) + \boldsymbol{B}(\boldsymbol{u}-\boldsymbol{u}^0) \end{aligned} \tag{6}$$

At this time, the dynamic is completely linearized around the initial guessed trajectory and can be written as:

$$\begin{aligned} f(\boldsymbol{x},\boldsymbol{u}) &\simeq f(\boldsymbol{x}^0,\boldsymbol{u}^0) + f'(\boldsymbol{x}-\boldsymbol{x}^0,\boldsymbol{u}-\boldsymbol{u}^0) \\ f(\boldsymbol{x},\boldsymbol{u}) &\simeq f(\boldsymbol{x}^0,\boldsymbol{u}^0) + \frac{\partial}{\partial \boldsymbol{x}}(f(\boldsymbol{x},\boldsymbol{u}))\Big|_{\boldsymbol{x}^0,\boldsymbol{u}^0}\boldsymbol{x} + \frac{\partial}{\partial \boldsymbol{u}}(f(\boldsymbol{x},\boldsymbol{u}))\Big|_{\boldsymbol{x}^0,\boldsymbol{u}^0}\boldsymbol{u} - \\ &- \frac{\partial}{\partial \boldsymbol{x}}(f(\boldsymbol{x},\boldsymbol{u}))\Big|_{\boldsymbol{x}^0,\boldsymbol{u}^0}\boldsymbol{x}^0 - \frac{\partial}{\partial \boldsymbol{u}}(f(\boldsymbol{x},\boldsymbol{u}))\Big|_{\boldsymbol{x}^0,\boldsymbol{u}^0}\boldsymbol{u}^0 \end{aligned} \tag{7}$$

For simplicity, the following notations can be used:

$$\begin{aligned} f(\boldsymbol{x}^0,\boldsymbol{u}^0) &= \boldsymbol{\Sigma} \\ -\frac{\partial}{\partial \boldsymbol{x}}(f(\boldsymbol{x},\boldsymbol{u}))\Big|_{\boldsymbol{x}^0,\boldsymbol{u}^0}\boldsymbol{x}^0 - \frac{\partial}{\partial \boldsymbol{u}}(f(\boldsymbol{x},\boldsymbol{u}))\Big|_{\boldsymbol{x}^0,\boldsymbol{u}^0}\boldsymbol{u}^0 &= \boldsymbol{z} \end{aligned} \tag{8}$$

In the end, substituting the notations from (8) into (7), the linearized dynamics can be formulated as [3]:

$$f(\boldsymbol{x},\boldsymbol{u}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{\Sigma} + \boldsymbol{z} \tag{9}$$

Although linearization is an effective and computationally efficient method for approximating nonconvex functions, it comes with certain limitations such as artificial infeasibility and artificial unboundedness. To deal with those problems, the SCvx algorithm can be configured with virtual control terms and trust regions [1].

At the first iteration of the algorithm, the initial guessed trajectory represents the reference trajectory which is an approximation of a solution to problem (1). With this input, the convex solver finds a better solution to the problem (1). When a new, better solution is found, the solver stops and then the algorithm enters into the second part of the loop: with the new solution found by the solver, it computes the new dynamics representative for the new trajectory and then discretizes it again. The new set of information about the new trajectory and the new dynamics are the new inputs representing the new reference trajectory for a new iteration of the convex solver. This process in repeated all over again until the convergence test is passed. The convergence test verifies several aspects:

1. if the differences between the new found trajectory and the old trajectory are smaller than a threshold;
2. if the virtual control of the new found trajectory is smaller than a threshold;
3. if the difference between the new found final time and the old final time is smaller than a tolerance set by the user.

The conditions must be checked for the convergence to be achieved. A small enough difference between the states of the trajectory means that the solver is beginning to find closer and closer solutions which is a sign of convergence and a very small virtual control means that the new trajectory respects the dynamic and there is no danger of artificial infeasibility.

All thresholds are set by the user and if the test is passed, then the algorithm founds the optimal trajectory and it stops.

3. MATHEMATICAL FORMULATION OF THE PROBLEM

We will analyze the differences produced by having two differently ways of mathematically formulate the cost functions for a minimum fuel usage in a descending scenario: maximize final mass of the vehicle and minimize total thrust used by the engine. For both scenarios we will consider the time of flight as an optimization parameter which will be minimize by the algorithm. Before introducing the two equivalent cost function formulations, we first describe the 3-DoF dynamics and the associated constraints of the problem.

The vehicle operates in a constant gravitational field, with negligible aerodynamic forces. Thrust is provided by a single gimbaled engine, capable of varying between predefined minimum and maximum limits. The engine's direction can be adjusted via a controllable gimbal angle, which is also bounded. The states vector will be represented by mass, three components of position (one vertical coordinate and two lateral coordinates) and three components of velocity vector (one vertical component of the velocity vector and two lateral components of the velocity vector). The control vector will be composed by the three components of the thrust vector (one vertical component and two lateral components).

The vehicle consumes the fuel mass directly proportional with the thrust magnitude. This proportionality constant being [3]:

$$\alpha = \frac{1}{I_{\mathrm{sp}} \cdot 9.81} \tag{10}$$

where I_{sp} represent the specific impulse. Having the proportionality constant defined, the dynamics describing the mass depletion of the vehicle can be formulated as [3]:

$$\dot{m} = -\alpha \|\boldsymbol{T}(t)\|_2 \tag{11}$$

The translational dynamics can be formulated as:

$$\frac{d}{dt}\bar{\mathbf{r}}(t) = \bar{\mathbf{v}}(t) \tag{12}$$

$$\begin{gathered} \boldsymbol{T} + \mathbf{G} = m\bar{\mathbf{a}} \Rightarrow \bar{\mathbf{a}} = \frac{(\mathbf{T} + \mathbf{G})}{m} \Rightarrow \mathbf{a} = \frac{\mathbf{T}}{m} + \frac{m\mathbf{g}}{m} \Rightarrow \\ \Rightarrow \frac{d}{dt}\mathbf{v} = \frac{\mathbf{T}}{m} + \mathbf{g} \end{gathered} \tag{13}$$

Having the mathematical model of the vehicle dynamics, the desired constraints on the state and control are imposed.

First restriction is preventing the total final mass of the vehicle to be less than the mass of the vehicle structure:

$$m_{\text{dry}} \leq m(t) \tag{14}$$

The second constraint is the glide slope of the trajectory. In other words, the vehicle must exist at all time inside an upward facing cone that makes an angle $\gamma \in [0, \gamma_{\max})$ with respect to the horizontal line.

The convex constrain that is imposed for the vehicle to respect this condition has the following convex form:

$$\text{tg}(\gamma_{\max}) \leq \frac{\boldsymbol{r}_{\mathbf{x}}(t)}{\sqrt{\boldsymbol{r}_{\boldsymbol{y}}(t)^2 + \mathbf{r}_{\mathbf{z}}(t)^2}} \tag{15}$$

The next constraint is the maximum velocity constraint, ensuring that the vehicle lands smoothly, along the computed trajectory:

$$v_{\max} \geq ||\bar{\mathbf{v}}|| \tag{16}$$

The magnitude constraint of the thrust vector is addressed by:

$$\begin{gathered} T_{\max} \geq ||\mathbf{T}|| \\ T_{\min} \leq ||\text{T}|| \end{gathered} \tag{17}$$

Besides the magnitude range constraint of the thrust vector, the values of the gimbal angle must be bounded too [3]:

$$\cos(\delta_{\max})||\boldsymbol{T}||_2 \leq \mathbf{e}_1\mathbf{T} \tag{18}$$

where $\mathbf{e}_1$ represents the unit vector on the x axis (vertical axis), so $\mathbf{e}_1\mathbf{T}$ represents the vertical component of the thrust vector.

Supplementary, a vertical landing constraint will be imposed. The thrust vector must have very low values on the lateral component on the last ten nodes of the trajectory:

$$\begin{gathered} e_2\text{T}_{\text{N}-10,\text{N}-9,\dots,\text{N}} \leq 1 \\ e_3\text{T}_{\text{N}-10,\text{N}-9,\dots,\text{N}} \leq 1 \end{gathered} \tag{19}$$

A novel constraint implementation into a SCvx free final time algorithm is the vertical thrust rate constraint:

$$\dot{T}_x^{\max}\tau\sigma_{\text{old}}^{i-1} \leq |\mathbf{e}_1\mathbf{T}_{k+1} - \mathbf{e}_1\mathbf{T}_k| \tag{20}$$

where $\dot{T}_x^{max}$ represents the maximum vertical thrust rate accepted, τ is the time step, σ_{old}^{i-1} is the final time found by the previous SCvx iteration $(i-1)$ and k represents the time node from trajectory.

Before presenting the complete convex formulation suitable for implementation within an SCvx algorithm, several key aspects must be addressed.

First, the system dynamics and thrust magnitude constraints introduce non-convexities into the problem.

Second, once the problem has been convexified through linearization, the continuous-time dynamics must be discretized to enable numerical solution by the algorithm.

To handle these steps, first-order Taylor expansions are applied to linearize the non-convex constraints from dynamics equations and a Lossless Convexification technique is used for convexifying the thrust bounds constraint, while a first-order hold method is used for time discretization.

These procedures follow the approach outlined in [3].

Now, the whole problem is convex, discretized and ready to completely be formulated with the two variants of cost functions:

a) maximize final mass:

$$\text{minimize}\left(w_m\left(m_f - m_0\right) + w_\mu \|\mu\|_1 + w_{\text{traj}} \left\|\Delta_{\text{traj}}\right\|_2 + w_{\text{time}}\, \Delta t_f\right) \tag{21}$$

where w_m is a weight on the difference between final and initial mass, $w_\mu \|\mu\|_1$ is the virtual control term meant to handle artificial infeasibility [1] [3], $w_{\text{traj}} \left\|\Delta_{\text{traj}}\right\|_2$is a term guarding against artificial unboundedness playing the role of a trust region [1] [3] and $w_{\text{time}}\, \Delta t_f$ is the term which includes the final time of the trajectory as an optimization variable along with its numerical weight [3].

b) minimize total thrust:

$$\text{minimize}\left(w_T \boldsymbol{T} + w_\mu \|\mu\|_1 + w_{\text{traj}} \left\|\Delta_{\text{traj}}\right\|_2 + w_{\text{time}} \cdot \Delta t_f\right) \tag{22}$$

where w_T represents a numerical weight on the whole thrust used across trajectory.
The importance of minimizing a variable from the cost function is direct proportional with the value of its weight.

Subject to:

Bounded time condition:

$$0 \le t_f \tag{23}$$

Initial conditions:

$$\begin{gathered} m(0) = m_{\text{total}} \\ \boldsymbol{r}(0) = \boldsymbol{r}_i \\ \boldsymbol{v}(0) = \boldsymbol{v}_i \end{gathered} \tag{24}$$

Final conditions:

$$\begin{gathered} \boldsymbol{r}\left(t_f\right) = \boldsymbol{r}_f \\ \boldsymbol{v}\left(t_f\right) = \boldsymbol{v}_f \end{gathered} \tag{25}$$

Discrete dynamics [3]:

$$\boldsymbol{x}_{k+1} = \bar{\boldsymbol{A}}_k \boldsymbol{x}_k + \bar{\boldsymbol{B}}_k \boldsymbol{u}_k + \bar{\boldsymbol{C}}_{k+1} \boldsymbol{u}_{k+1} + \bar{\boldsymbol{\Sigma}}_k \sigma + \bar{\boldsymbol{z}}_k + \mu, k \in [1, K-1] \tag{26}$$

State Constraints:

$$\begin{gathered} m_{dry} \le m(t) \\ \operatorname{tg}(\gamma_{\max}) \le \frac{\boldsymbol{r}_x(t)}{\sqrt{\boldsymbol{r}_y(t)^2 + \boldsymbol{r}_z(t)^2}} \\ \mathrm{v}_{max} \ge \|\boldsymbol{v}\| \end{gathered} \tag{27}$$

Control constraints:

$$\begin{gathered}
\mathrm{T}_{max} \geq \|\boldsymbol{T}\| \\
\mathrm{T}_{min} \leq \frac{u^{0^T}}{\|u^0\|_2}\boldsymbol{u} \\
\cos\left(\delta_{\max}\right)\|\boldsymbol{T}\|_2 \leq \bar{\boldsymbol{e}}_1\boldsymbol{T} \\
\boldsymbol{e}_2\boldsymbol{T}_{N-10,N-9,\dots,N} \leq 1 \\
\boldsymbol{e}_3\boldsymbol{T}_{N-10,N-9,\dots,N} \leq 1 \\
\dot{T}_x^{max}\tau\sigma_{old}^{i-1} \leq \left|\boldsymbol{e}_1\boldsymbol{T}^{k+1} - \boldsymbol{e}_1\boldsymbol{T}^k\right|
\end{gathered} \tag{28}$$

4. RESULTS FROM SIMULATIONS

The results presented are the results of running the problem above using the specifications of the SpaceX Falcon 9 first step [4]. Specifically, it has a total mass of 396400 [kg], a maximum thrust of 7600 [kN] and a minimum thrust of 2400 [kN]. The gimbal angle can deflect the thrust vector away by the vertical direction with a maximum of 20 [deg] and the maximum thrust rate allowed for this example is set to 400 [kN/s]. The initial position of the mission is at 1000 [m] on vertical, respectively 200 [m] and 100 [m] on laterals. The initial velocity has the components -10 [m/s] on vertical, respectively -1 [m/s] and 1[m/s] on laterals. The target position is [0; 0; 0] [m] and the final velocity is set to be [0; 0; 0]. The maximum velocity that the vehicle is allowed to have along trajectory is 50 [m/s] and the vehicle is constrained to remain in a glideslope cone making 60 [deg] with respect to the horizontal.

For these simulations, a number of 100 discretization points was used and the algorithm was run in the Matlab R2021a environment using the CVX SDPT 3 solver for the convex optimization problem.

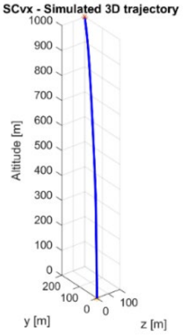

Figure 2. 3D trajectory - maximize final mass

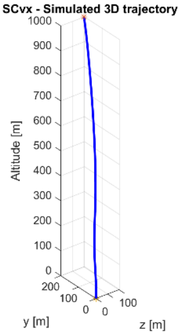

Figure 3. 3D trajectory - minimize total thrust

Figure 2 and Figure 3 reveal that both SCvx formulations—maximizing final mass and minimizing total thrust – yield smooth, feasible trajectories that meet the problem's initial and final conditions. Visually, the two trajectories appear very similar, indicating that both objective functions are effective in guiding the vehicle to its target while respecting dynamic and physical constraints. This suggests that, under the given conditions, the choice of cost function has minimal impact on the overall trajectory shape. However, as explored further in the simulation results, subtle differences may still arise in thrust usage, fuel efficiency, or control behavior, which are not immediately apparent from the trajectory plots alone.

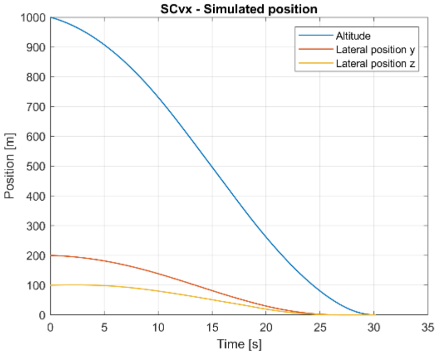

Figure 4. Positions - maximize final mass

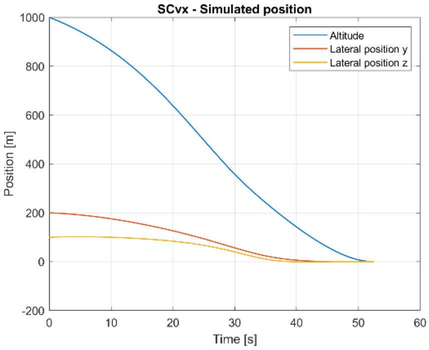

Figure 5. Positions - minimize total thrust

Maximizing final mass yields a faster, more fuel-efficient descent with sharper position changes (Figure 4). Minimizing total thrust produces a slower, more controlled trajectory that stretches the descent time to reduce instantaneous thrust demand (Figure 5). Both methods result in accurate landing at the origin.

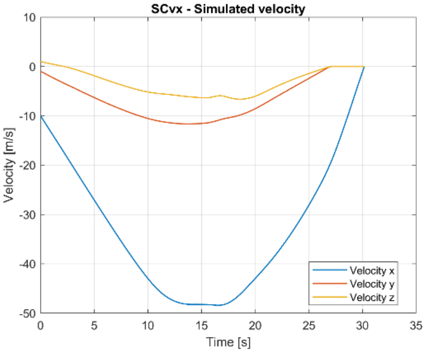

Figure 6. Velocity - maximize final mass

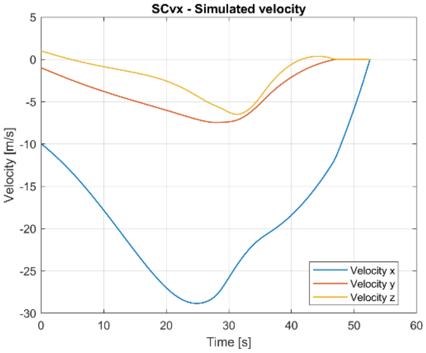

Figure 7. Velocity - minimize total thrust

As it can be observed in Figure 6 and Figure 7, maximizing final mass leads to a faster, more fuel-efficient descent, while minimizing total thrust results in a slower, more conservative trajectory with gentler velocity changes. Another observation is represented by the lateral velocities becoming closer and closer to zero several seconds before touchdown. This fact enforces the vertical landing constraint formulated in the problem, meaning that close to the landing site, the vehicle has only vertical motion.

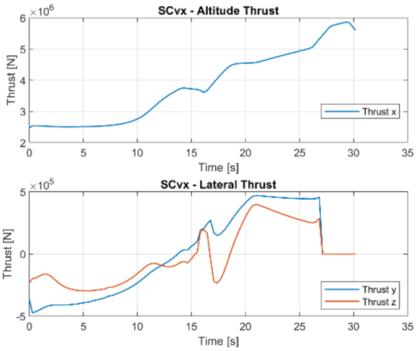

Figure 8. Thrust profile - maximize final mass

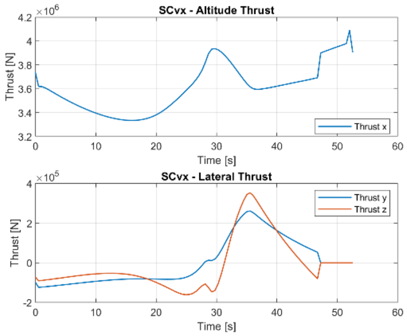

Figure 9. Thrust profile - minimize total thrust

The maximize final mass formulation applies more aggressive and time-varying thrust, particularly near landing, to minimize fuel use by shortening the burn duration (Figure 8). In contrast, the minimize total thrust formulation results in a smoother thrust profile with fewer sharp variations, favoring gradual control inputs and extended descent time to reduce overall thrust expenditure (Figure 9).

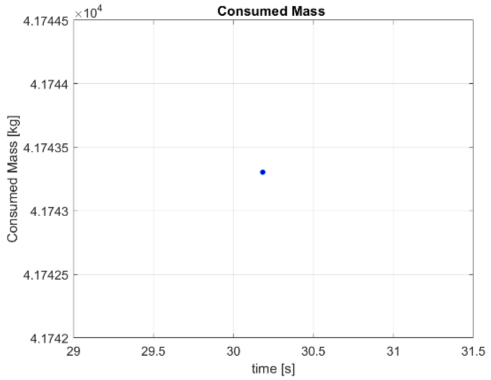

Figure 10. Consumed mass and final time - maximize final mass

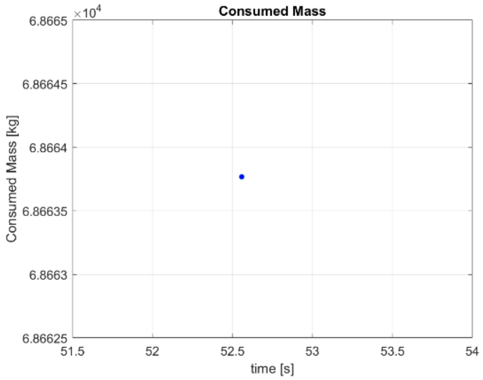

Figure 11. Conusmed mass and final time - minimize total thrust

Figure 10 and Figure 11 prove that maximizing final mass formulation achieves better fuel efficiency and faster landing, while minimizing total thrust reduces instantaneous control effort at the cost of higher total fuel use and longer flight time.

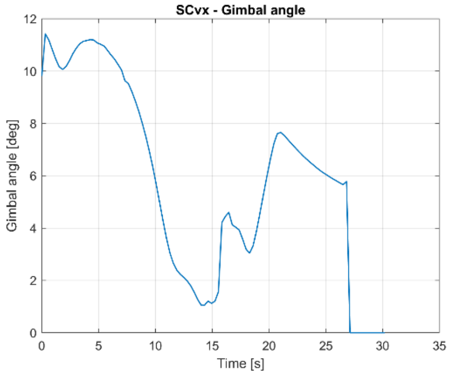

Figure 12. Gimbal angle evolution - maximize final mass

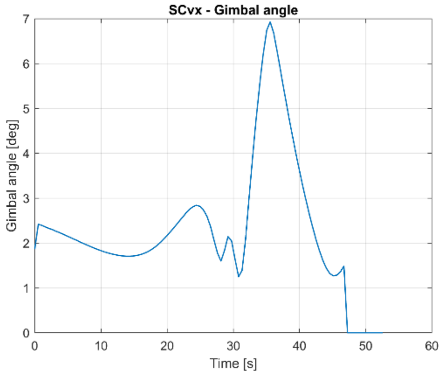

Figure 13. Gimbal angle evolution - minimize total thrust

Maximizing final mass results in larger and more dynamic gimbal deflections, enabling tighter maneuvering for rapid descent (Figure 12). In contrast, minimizing total thrust produces gentler, more stable gimbal behavior, prioritizing smoother control at the cost of longer flight time (Figure 13).

Both of them respect the gimbal constraint limit and in the final part they are both equal to zero.

This is another mathematical way of expressing that the vertical landing constraint is respected because the vehicle does not need any lateral forces to reach the final position in the last seconds of flight.

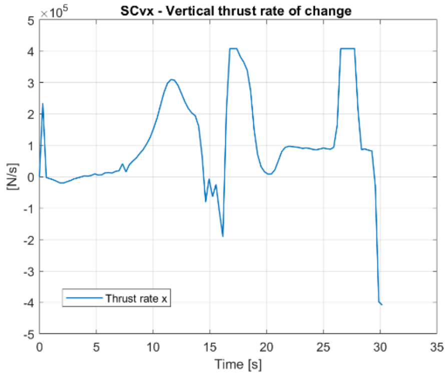

Figure 14. Vertical thrust rate - maximize final mass

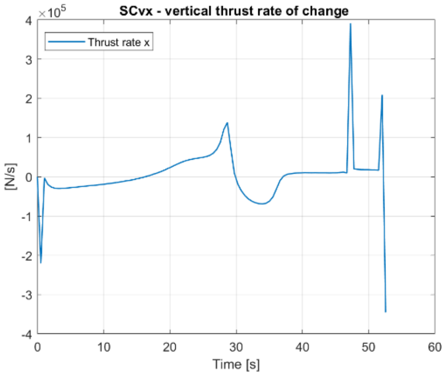

Figure 15. Vertical thrust rate - minimize total thrust

The maximize final mass case involves high-frequency, high-amplitude thrust changes, enabling quick trajectory corrections to save fuel (Figure 14). In contrast, the minimize total thrust strategy maintains a more stable thrust rate, reducing control effort and promoting a smoother descent (Figure 15). Also, it can be observed that the vertical thrust rate constraint is being respected.

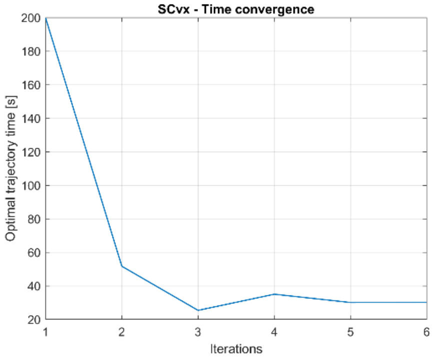

Figure 16. Time convergence - maximize final mass

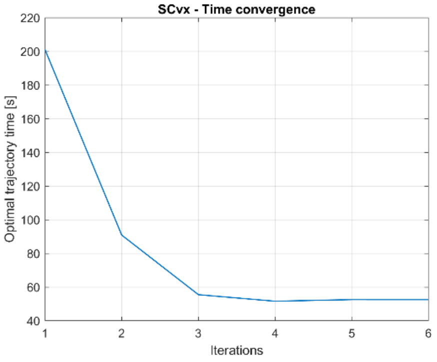

Figure 17. Time convergence - minimize total thrust

Figure 16 and Figure 17 show that both SCvx formulations demonstrate rapid and stable convergence in under six iterations. The method effectively refines the trajectory time from an initial rough guess to an optimized solution, confirming its efficiency and robustness for both fuel-saving and thrust-minimization objectives.

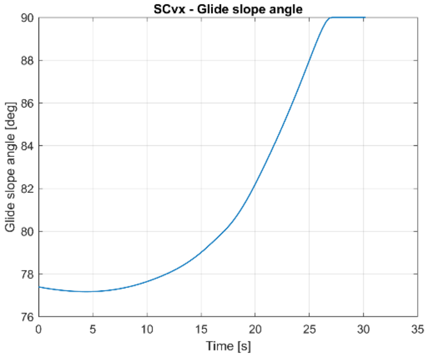

Figure 18. Glide slope evolution - maximize final mass

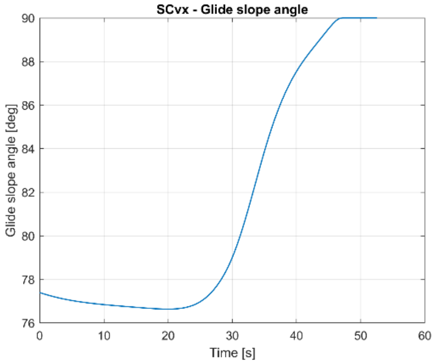

Figure 19. Glide slope evolution - minimize total thrust

Figure 18 and Figure 19 present the evolution of the glide slope angle along the generated trajectory. In this case, the glide slope angle is defined as the angle between the position vector of the vehicle and the horizontal plane. Because this angle can be calculated using only the coordinates of the vehicle on each axis, it can be verified that the evolution of this parameter is correct by looking at the initial starting point. The initial position is (1000, 200, 100) [m] which correlates with a glide slope angle of approximately 78 [deg] which is what the plot is showing. This angle converging to 90 [deg] as the vehicle is approaching the landing site is another mathematical formulation to describe a vertical landing on the last time period of the trajectory. The glide slope constraint enforced here was that the vehicle to be always inside of a cone whose generator makes a 60 [deg] angle with the horizontal plane. It can be cleary seen the constraint is respected.

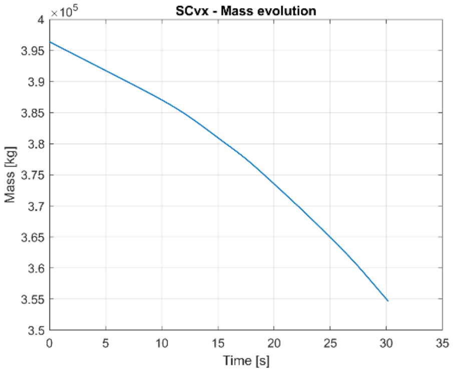

Figure 20. Mass evolution - maximize final mass

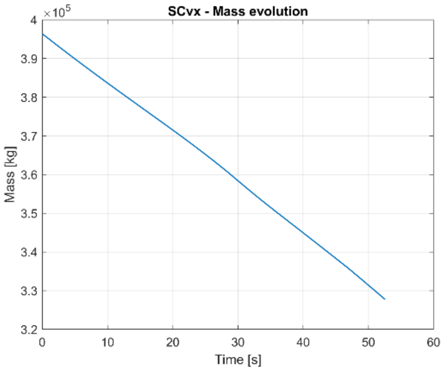

Figure 21. Mass evolution - minimize total thrust

Maximizing final mass leads to faster fuel-efficient burns with less total mass loss (Figure 20), while minimizing total thrust spreads out consumption, resulting in greater fuel usage over time despite smoother control actions (Figure 21). As observed, both trajectories exhibit an almost linear behavior in the initial phase. However, around the midpoint, the maximize final mass strategy must decelerate the vehicle from a higher velocity, requiring greater thrust—but only for a shorter duration.

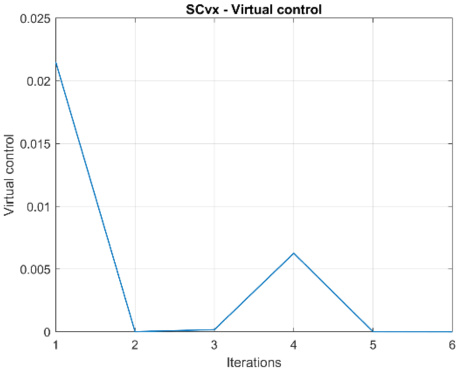

Figure 22. Virtual control evolution - maximize final mass

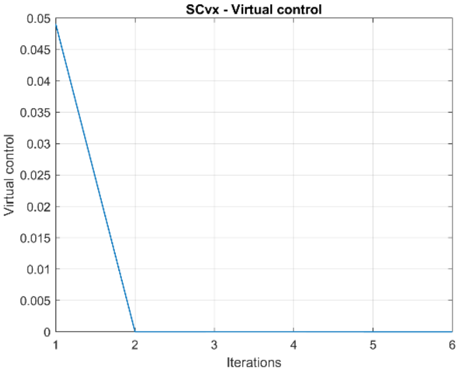

Figure 23. Virtual control evolution - minimize total thrust

The feasibility of the trajectory with respect to the dynamics and constraints is best described by the virtual control element displayed in Figure *22* and Figure *23*. Virtual control is a fail-safe against the artificial infeasibily. A feasible trajectory solution is described by a very small virtual control on the whole trajectory wich is the sum of all virtual controls on each point of the trajectory. Both formulations converge quickly, but the minimize total thrust approach shows faster feasibility recovery. The presence of small virtual control in the maximize final mass case suggests a slightly more challenging initial setup, though both converge to physically valid solutions within a few iterations.

5. CONCLUSIONS

This study presented a comparative analysis of two Successive Convexification (SCvx) formulations applied to the powered descent guidance (PDG) problem for a reusable launch vehicle, using the Falcon 9 first stage as a representative case. Both formulations - maximizing final mass and minimizing total thrust - produced feasible and dynamically consistent trajectories that satisfied all mission constraints.

Simulation results revealed that the maximize final mass formulation achieved faster descent and greater fuel efficiency by applying more aggressive thrust and control maneuvers over a shorter time horizon. In contrast, the minimize total thrust formulation favored smoother, more gradual control inputs, resulting in a longer flight duration but increased total fuel consumption.

The position, velocity, and thrust profiles showed that both strategies effectively guided the vehicle to the target while staying within glide slope, gimbal, and thrust constraints. However, the final mass maximization exhibited sharper changes in thrust rate and gimbal angle, which may be more demanding on actuators. On the other hand, thrust minimization offered a more stable control profile with lower instantaneous demands.

Virtual control and time convergence plots confirmed the robustness and efficiency of the SCvx algorithm in both cases. Virtual control dropped to near-zero within a few iterations, indicating high-quality convergence, especially for the thrust minimization case.

In summary, both SCvx formulations are valid and effective. The choice between them should be guided by mission priorities:

- Maximize final mass is ideal for fuel-critical missions requiring efficient fuel use.
- Minimize total thrust is better suited for missions prioritizing low control effort and smoother descent dynamics.

These findings offer valuable insights for designing real-time guidance algorithms for reusable rocket landings, balancing fuel use, control effort, and trajectory smoothness depending on operational goals.

REFERENCES

[1] D. Malyuta, Y. Yue, E. Purnanand and B. Ackmese, *Advances in Trajectory Optimization for Space Vehicle Control*, 2021.

[2] D. Malyuta, T. Reynolds, M. Szmuk, T. Lew, R. Bonalli, M. Pavone and B. Açıkmeşe, *Convex Optimization for Trajectory Generation: A Tutorial on Generating Dynamically Feasible Trajectories Reliably and Efficiently*, 2021.

[3] M. Szmuk and B. Acikmese, *Successive Convexification for 6-DoF Mars Rocket Powered Landing with Free-Final-Time, University of Washington*, Seattle, WA 98195-2400.

[4] * * * SpaceX, "www.spacex.com," SpaceX. [Online].