Solver comparison for LCvx-based powered descend guidance problems

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

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

DOI: 10.13111/2066-8201.2025.17.3.1

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

Abstract: *This article aims to analyze the differences between 3 of the CVX toolbox canonical solvers: SDPT3, SeDuMi and ECOS; for problem formulations based on lossless convexification technique (LCvx). Without loss of generality, the simulations were conducted on the first stage of the launcher Falcon 9, while the goal of the optimization process is to minimize the fuel consumption through minimization of the total thrust under restrictions based on physical limitations of the vehicle.*

Key Words: *CVX, lossless convexification, optimal trajectory, PDG, glideslope angle, SOCP framework, optimization solvers, generated trajectory*

1. CONVEXIFICATION TECHNIQUES: THEORETICAL ASPECTS

The modeling technique considered in this work for addressing a nonconvex problem through transformation into a convex formulation is LCvx. The convexification is achieved through a relaxation or a revised formulation of the initial problem that guarantees a globally optimal solution for the initial problem via Portryagin's maximum principle [1]. The foundational idea was introduced in [2], which developed an optimal (minimal fuel) thrust program for the terminal phase of a lunar soft-landing starting from the equation of vertical motion of a vehicle (1-DoF). The trajectory results based on identifying an admissible $\boldsymbol{u}(t)$ that maximizes the Hamiltonian and an appropriate switching function, more unequivocally detailed as a relation $f(x_1,x_2)=0$ that guarantees soft landing of the vehicle if thrusters switch to full force upon validation of this equation.

Since the 21st century, the most prominent application of lossless techniques has been convexifying the thrust magnitude constraint. The nonconvexity arises from the lower bound for thrust and the solution is introduced in [3] by Açıkmeşe and Ploen:

$$\rho_1 \leq \|\boldsymbol{T_C}(t)\|_2 \leq \rho_2 \tag{1}$$

where ρ_1 and ρ_2 represent the minimum and maximum thrust the vehicle is capable of generating. The convexification procedure consist of a lift of the feasible input set, while rigorously proving that the optimal input from the lifted problem corresponds to a feasible input of the initial problem. From a geometric viewpoint, the constraint is expanded into an additional dimension corresponding to the slack variable:

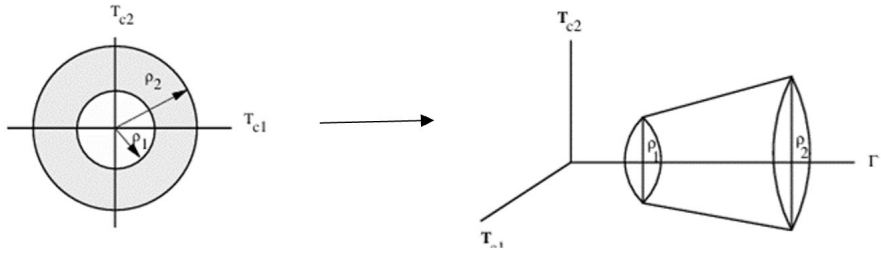

Figure 1. Lossless convexification of thrust magnitude constraints [3]

The lossless convexification procedure for a 3-DoF powered descend problem was developed in [4] and starts with the following non-convex mathematical framework:

Problem 1

$$\begin{gathered}\min_{\boldsymbol{T_c}(\cdot)} \int_0^{t_f} \|\boldsymbol{T_c}(t)\|_2 dt \\ \text{subject to } \ddot{\boldsymbol{r}}(t) = \boldsymbol{g} + \frac{\boldsymbol{T_C}(t)}{m(t)} \\ \dot{m}(t) = -\alpha\|\boldsymbol{T_C}(t)\|_2 \\ \rho_1 \le \|\boldsymbol{T_C}(t)\|_2 \le \rho_2 \\ \boldsymbol{r_1}(t) \ge 0, \boldsymbol{r}(0) = \boldsymbol{r}_0, \dot{\boldsymbol{r}}(0) = \dot{\boldsymbol{r}}_0, m(0) = m_{\text{wet}} \\ \boldsymbol{r}(t_f) = \dot{\boldsymbol{r}}(t_f) = 0,\end{gathered} \tag{2}$$

The initial convex problem is augmented with a slack variable Γ that incorporates the information for the thrust magnitude. To handle the nonconvexities posed by the dynamic equations, the following substitutions are performed:

$$\boldsymbol{u}(t) = \frac{\boldsymbol{T_C}(t)}{m(t)}, \sigma(t) = \frac{\Gamma(t)}{m(t)}, z(t) = ln\,(m(t)) \tag{3}$$

Based on those and the approximation for the new sigma bound constraint using the first 2 and 3 terms of Taylor's expansion to ensure the problem stays within the SOCP framework, the convex formulation based on LCvx technique is obtained in [4]:

Problem 2

$$\begin{gathered}\min_{\sigma(.)} \int_0^{t_f} \sigma(t) dt \\ \text{subject to } \ddot{\boldsymbol{r}}(t) = \boldsymbol{g} + \boldsymbol{u}(t) \\ \dot{z}(t) = -\alpha\sigma(t) \\ \|\boldsymbol{u}(t)\|_2 \le \sigma(t) \\ \rho_1 e^{-z_0(t)}\left[1 - (z(t) - z_0(t)) + \frac{(z(t) - z_0(t))^2}{2}\right] \le \sigma(t) \le \rho_2 e^{-z_0(t)}[1 - (z(t) - z_0(t))] \\ ln\,(m_{wet} - \alpha\rho_2 t) \le z(t) \le ln\,(m_{wet} - \alpha\rho_1 t) \\ z(0) = ln\,(m_{wet}), \boldsymbol{r}(0) = \boldsymbol{r_0}, \dot{\boldsymbol{r}}(0) = \dot{\boldsymbol{r}}_0, \boldsymbol{r}(t_f) = \dot{\boldsymbol{r}}(t_f) = 0\end{gathered} \tag{4}$$

2. MATHEMATICAL MODEL TO SOLVER FORM

This section develops a solver-compatible mathematical framework for a 3-DoF fixed-final time powered descent guidance problem subjected to the following constraints:

- Initial position, velocity and mass
- Final position and velocity
- Dynamics
- Minimum mass consumption
- Lower and upper bound for thrust magnitude
- Maximum velocity
- Vertical landing
- Maximum gimbal angle
- Maximum glideslope angle

Based on the procedure described in [4] and the above desired constraints, the following continuous framework was developed:

Problem 3

$$
\begin{gathered}
\min_{\sigma(.)} \int_0^{t_f} \sigma(t)dt \\
\text{subject to } \ddot{\boldsymbol{r}}(t) = \boldsymbol{g} + \boldsymbol{u}(t) \\
\dot{z}(t) = -\alpha\sigma(t) \\
\|\boldsymbol{u}(t)\|_2 \leq \sigma(t) \\
\rho_1 e^{-z_0(t)}\left[1 - (z(t) - z_0(t)) + \frac{(z(t) - z_0(t))^2}{2}\right] \leq \sigma(t) \leq \rho_2 e^{-z_0(t)}[1 - (z(t) - z_0(t))] \\
\left\|\boldsymbol{S}_j x(t) - \boldsymbol{v}_j\right\| + \boldsymbol{c}_j^T x(t) + a_j \leq 0 \\
z(0) = ln\,(m_{\text{total}}), r(0) = r_0, \dot{r}(0) = \dot{r}_0, r(t_f) = \dot{r}(t_f) = 0 \\
m(t_{\text{final}}) \geq m_{dry} \\
\boldsymbol{T}_x(t) \geq cos\,(\theta_{\max})\boldsymbol{T}(t) \\
\boldsymbol{T}_y(t_{\text{final}}) + \boldsymbol{T}_z(t_{\text{final}}) \to 0 \\
\|\boldsymbol{v}(t)\| \leq v_{max}
\end{gathered}
\tag{5}
$$

As is the case with most optimization solvers, the solvers in the CVX toolbox require a discrete formulation of the problem.

The discretization algorithm adopted from reference [4] is distinctive in its use of basic functions to decompose both the state and control variables, which are then used to define the convex optimization problem. The process begins by discretizing the continuous time domain into a set of time nodes.

$$t_k = k\Delta t, \qquad k = 0..N \tag{6}$$

The discretized states include positions and velocities, denoted by $\boldsymbol{x_k}$, and logarithm of mass, denoted by z_k.

Likewise, the control vector consists of the discretized new thrust $\boldsymbol{u_k}$ and the new slack variable σ_k. The discrete state-space system is then formulated as follows:

$$\begin{bmatrix}\boldsymbol{x}_k\\ z_k\end{bmatrix} = \xi_k + \psi_k\eta, k = 1 \ldots . N, \text{where } \xi_k = \Phi_k \boldsymbol{y}_0 + \Lambda_k \begin{bmatrix}\boldsymbol{g}\\ 0\end{bmatrix}$$
$$\begin{bmatrix}\boldsymbol{u}_k\\ \sigma_k\end{bmatrix} = \Upsilon_k\eta, k = 0 \ldots . N \tag{7}$$

where the matrices ψ_k, Φ_k and Λ_k are expanded matrices of the state transition matrix and control input matrix:

$$\Phi_k = \boldsymbol{A}^k$$
$$\Lambda_k = \boldsymbol{B} + \boldsymbol{AB} + \cdots + \boldsymbol{A}^{k-1}\boldsymbol{B}$$
$$\psi_k = \begin{bmatrix} 0 & 0 & 0 & 0 & 0\\ \boldsymbol{B} & 0 & 0 & 0 & 0\\ \boldsymbol{AB} & \boldsymbol{B} & 0 & 0 & 0\\ \ldots & \boldsymbol{AB} & \boldsymbol{B} & 0 & 0\\ \boldsymbol{A}^{k-1}\boldsymbol{B} & \ldots & \ldots & \ldots & 0\end{bmatrix} \tag{8}$$

Here, η represents the optimization variable that encapsulates the thrust and sigma values at each discretization point and Υ_k selects a specific timestep thrust component:

$$\eta = \begin{bmatrix}u_0\\ \sigma_0\\ \cdots\\ u_N\\ \sigma_N\end{bmatrix} \qquad \Upsilon_k = [0 \quad \ldots \quad \boldsymbol{I} \quad \ldots \quad 0] \tag{9}$$

To have a complete view of the dynamics, the expanded formulation is the following:

$$\begin{bmatrix}\boldsymbol{r}_k\\ \dot{\boldsymbol{r}}_k\\ \ln m_k\end{bmatrix} = \boldsymbol{A}^k \begin{bmatrix}\boldsymbol{r}_0\\ \dot{\boldsymbol{r}}_0\\ \ln m_0\end{bmatrix} + (\boldsymbol{B} + \boldsymbol{AB} + \cdots \boldsymbol{A}^{k-1}\boldsymbol{B}) \begin{bmatrix}g\\ 0\\ 0\end{bmatrix} +$$
$$\begin{bmatrix} 0 & 0 & 0 & 0 & \ldots & 0\\ \boldsymbol{B} & 0 & 0 & 0 & \ldots & 0\\ \boldsymbol{AB} & \boldsymbol{B} & 0 & 0 & \ldots & 0\\ \ldots & \ldots & \ldots & \ldots & \ldots & 0\\ \boldsymbol{A}^{k-1}\boldsymbol{B} & \ldots & \ldots & \ldots & \ldots & \ldots\\ 0 & 0 & 0 & 0 & \ldots & 0\end{bmatrix} \begin{bmatrix}u_0\\ \sigma_0\\ \ldots\\ u_{N-1}\\ \sigma_{N-1}\end{bmatrix} \tag{10}$$

where *A* and *B* are the discretized state transition matrix and control input matrix:

$$\boldsymbol{A} = e^{\boldsymbol{A_c}\Delta t}, \boldsymbol{A_c} = \begin{bmatrix}0 & \boldsymbol{I} & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix},$$
$$\boldsymbol{B} = \int_0^{\Delta t} e^{\boldsymbol{A_c}(\Delta t - s)}\boldsymbol{B_c}ds, \boldsymbol{B_c} = \begin{bmatrix}0 & 0\\ \boldsymbol{I} & 0\\ 0 & -\alpha\end{bmatrix} \tag{11}$$

Regarding the objective function, minimizing the thrust needed to sustain the trajectory is reformulated as the minimization of the sigma components of η, multiplied by scalar coefficients, which are defined by the chosen integration technique applied.

Using the information presented above, Problem 3 is transformed into the final discrete formulation suitable for CVX implementation:

Problem 4

$$\min_{\eta} \omega^T \eta$$

$$\text{subject to} \quad \boldsymbol{E}_r(\xi_1 + \psi_1\eta) = \mathbf{r}_0 \; \boldsymbol{E}_v(\xi_1 + \psi_1\eta) = \mathbf{v}_0 \; \boldsymbol{E}_m(\xi_1 + \psi_1\eta) = m_{total}$$

$$\boldsymbol{E}_r(\xi_N + \psi_N\eta) = \mathbf{r}_{final} \; \boldsymbol{E}_v(\xi_N + \psi_N\eta) = \mathbf{v}_{final} \; \boldsymbol{E}_m(\xi_N + \psi_N\eta) \geq m_{dry}$$

$$\|\boldsymbol{E}_u \Upsilon_k \eta\| \leq \boldsymbol{E}_\sigma \Upsilon_k \eta$$

$$\mu_1(t_k)\left[1 - \left(\boldsymbol{F}(\xi_k + \psi_k\eta) - z_0(t_k)\right) + \frac{\left(F(\xi_k + \psi_k\eta) - z_0(t_k)\right)^2}{2}\right] \leq \tag{12}$$

$$\boldsymbol{E}_\sigma \Upsilon_k \eta \leq \mu_2(t_k)\left[1 - \left(\boldsymbol{F}(\xi_k + \psi_k\eta) - z_0(t_k)\right)\right]$$

$$\left\|\mathbf{S}_j \boldsymbol{E}_{rv}(\xi_k + \psi_k\eta) - \mathbf{v}_j\right\| + \mathbf{c}_j^T(\xi_k + \psi_k\eta) + a_j \leq 0$$

$$\boldsymbol{E}_{u_x} \Upsilon_k \eta \geq \cos(\theta_{max})\|\boldsymbol{E}_u \Upsilon_k \eta\|$$

$$\boldsymbol{E}_{u_y} \Upsilon_N \eta + \boldsymbol{E}_{u_z} \Upsilon_N \eta \to 0$$

$$\|\boldsymbol{E}_v(\xi_k + \psi_k\eta)\| \leq v_{max}$$

3. PARTICULARITIES OF EACH SOLVER

From the CVX Toolbox, a comparative analysis is conducted between SDPT3 [5], SeDuMi [6] and ECOS [7]. Although ECOS is designed to be memory-efficient and to exploit problem sparsity (properties well-suited for real-time use), its applicability is limited to linear programming (LP) and second-order cone programming (SOCP) problems, while the other 2 can handle semidefinite constraints (SDP). The complexity of dealing with hard nonconvexity goes against ECOS's design philosophy, whose simplicity brings it closer to first-order methods.

Nonetheless, all three have at their core second-order primal-dual interior point methods, but SeDuMi has the particularity that the self-dual embedded approach transforms the primal-dual problem into a single feasibility problem. As for SDPT3, it is optimized by using Mehrotra's predictor-corrector to converge faster and solve complex problems quicker.

For the problem defined in this analysis, SeDuMi was not able to handle the glideslope constraint, therefore an approximation was considered.

The cone was approximated as a pyramid, with its base inscribed in the square as can be seen in the following representation:

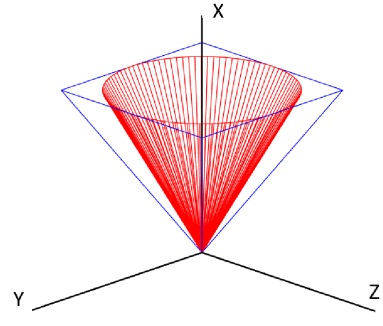

Figure 2. Cone to pyramid approximation for SeDuMi

4. SIMULATION RESULTS

The simulations were performed in the MATLAB 2023b environment. For the SDPT3 was used version 4.0, for SeDuMi version 1.3.4 and for ECOS version 2.0.7. The first stage of Falcon 9 was considered as the modelling vehicle and the following structural data served as inputs for the problem:

Stage 1. Falcon 9 (Block 5) [8]	
Dry mass, (kg)	25,600
Total mass, (kg)	421300
Specific impulse, (s)	~282
Maximum thrust, (MN)	~7.6
Minimum thrust, (MN)	~2.4

As for the other constraints, the mission is defined as starting from [1000, 300, -300] [m] with [-1 1 -1] [m/s] and ends when reaching position [0, 0, 0] [m] with [0, 0, 0] [m/s] velocity. Additionally, a maximum gimbal angle of 20 degrees, a maximum glideslope of 30 degrees and a maximum velocity of 50 m/s were considered based on physical requirements defined for a successful mission. For all 3 solvers, the optimal value (the objective function value) was almost identical – approximately 302.202 – highlighting the proximity of the generated trajectories and therefore of the solvers, regardless of the interior search methods employed.

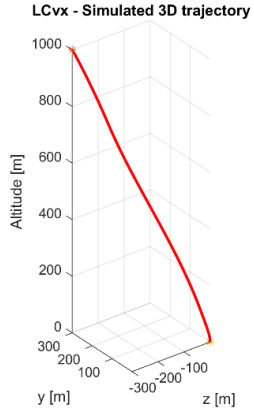

Figure 3. SDPT3 – 3D Trajectory

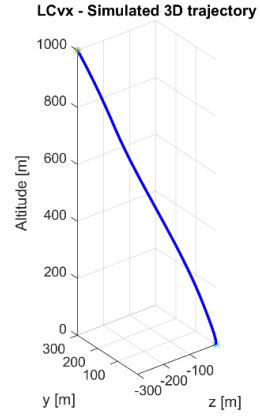

Figure 4. SeDuMi – 3D Trajectory

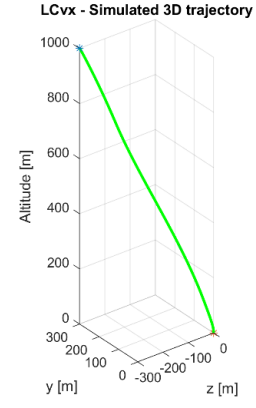

Figure 5. ECOS – 3D Trajectory

From the 3D viewpoint, the trajectories generated by all solvers are very similar, as no difference can be discerned between Figure 3, Figure 4 and Figure 5.

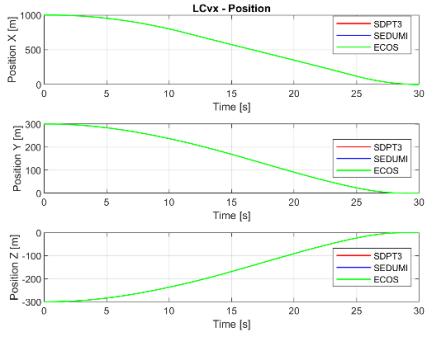

Figure 6. LCvx – Position

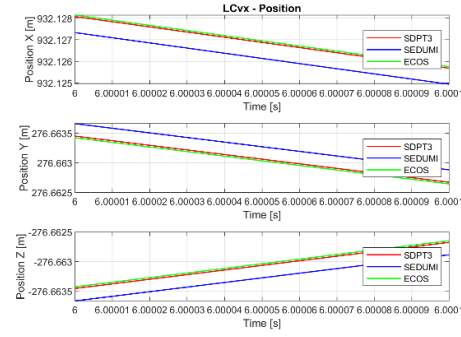

Figure 7. LCvx – Position – Zoom in

In Figure 6, positions of all three solver were plotted overlapping each other, and the solutions are so close only the last solver's solution can be seen in Figure 6. On the right side, a zoom-in view was provided for the interval [6, 6.0001] [s] to differentiate each solver solution. As it can be seen in Figure 7, SeDuMi deviates by 0.001 m from the others two.

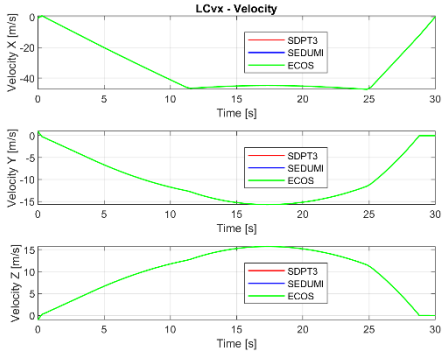

Figure 8. LCvx - Velocity

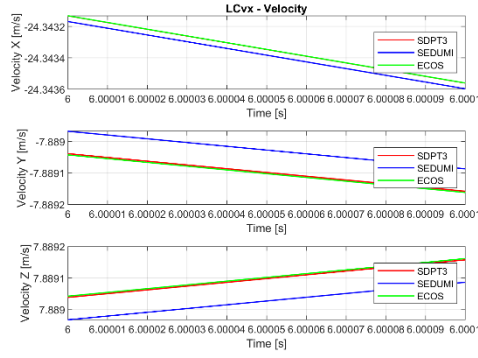

Figure 9. LCvx – Velocity – Zoom in

The same observation is valid for velocities throughout the mission (Figure 8), while differences of 10^{-4} are observed when zoomed into the same interval as for the positions (Figure 9). For velocities, it is notable how the vertical velocity stays below the 50 m/s mark for almost half of the mission, in order to respect the maximum velocity constraint.

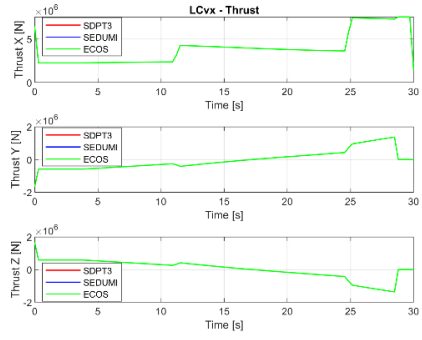

Figure 10. LCvx - Thrust

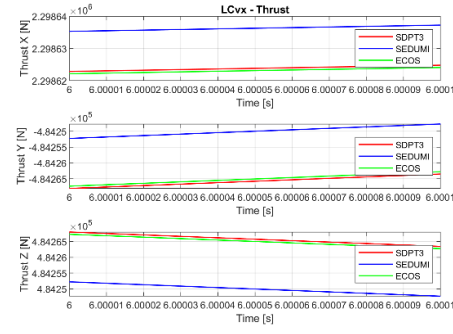

Figure 11. LCvx – Thrust – Zoom in

All thrusts generated stay withing the interval defined by the minimum and maximum physical limitations for the thrust (Figure 10). When zooming in, the SeDuMi deviates by several tens of Newtons from the other two solvers, which remain tightly clustered (Figure 11).

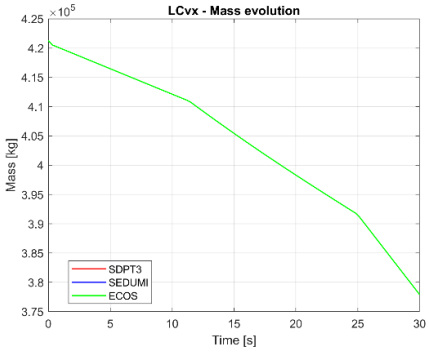

Figure 12. LCvx – Mass evolution

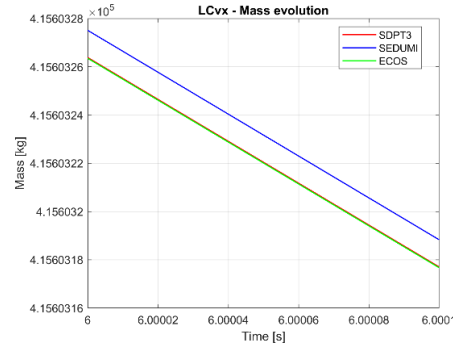

Figure 13. LCvx – Mass evolution – Zoom in

The mass evolution follows a smooth decreasing slope for each of the 3 solvers (Figure 12). Even when looking in depth (Figure 13), the difference between solvers for the interval analyzed is less than 1 kg, an insignificant quantity compared to the approximately 40000 kg required to perform the mission.

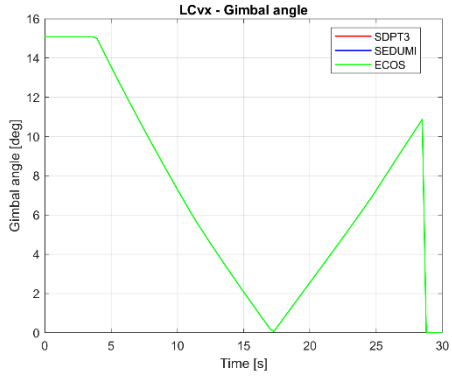

Figure 14. LCvx – Gimbal angle

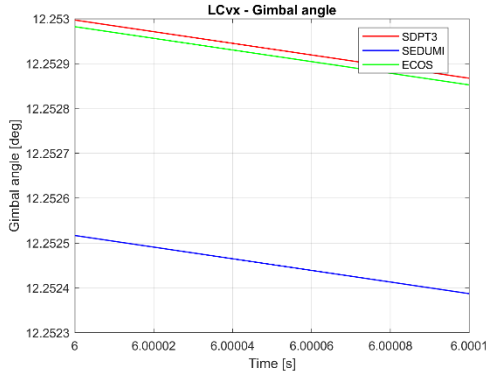

Figure 15. LCvx – Gimbal angle – Zoom in

The maximum gimbal angle of 20 degrees constraint is respected throughout the mission, as can be seen in Figure 14, while it tends to reach 0 in the final stages of the mission, corresponding to the vertical landing required. As for difference between solvers, SDPT3 and ECOS give almost identical results, while SeDuMi deviates by 0.001 degrees (Figure 15).

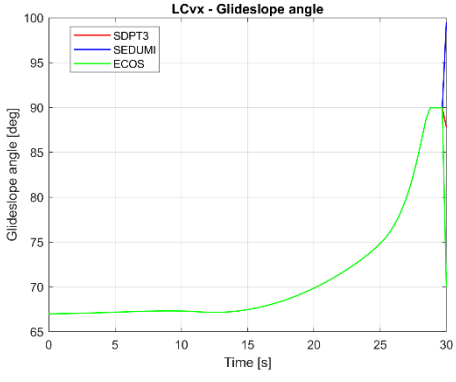

Figure 16. LCvx – Glideslope angle

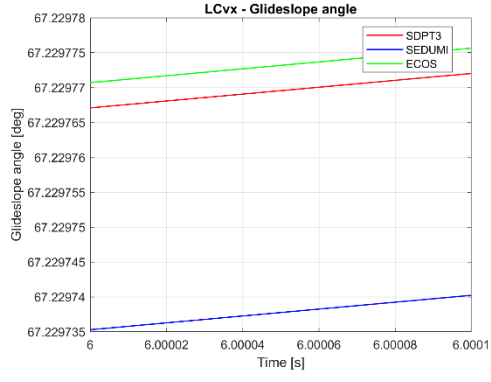

Figure 17. LCvx – Glideslope angle – Zoom in

The solely observable difference when representing the entire mission is found in the glideslope representation, due to the division of very small values for the final step which is further converted in degrees, accumulating more errors (Figure 16). Except for the final step, differences between solvers are of 0.0001 degrees (Figure 17).

5. CONCLUSIONS

Based on the analysis conducted, insignificant differences in values are observed when using one of the 3 canonical solvers of CVX: SDPT3, SeDuMi and ECOS for the powered descend guidance problem based on lossless convexification techniques defined in Chapter 2. The observation might not hold for successive convexification implementations, where the iterative process can enlarge the gap between solutions, so a separate analysis must be conducted to assess the effect of using a particular solver. Nonetheless, an informed decision

must be performed when choosing the solver. Some are more robust , while others display sensitiveness to problem scaling. Some are well-design for embedded optimization but more restrictive when formulating the problem, while others can solve complex problems but require a proportionally heavy computational workload. Therefore, the complexity of the problem and requirements in terms of robustness and computational efficiency must be carefully traded.

REFERENCES

[1] L. Pontryagin, *The Mathematical Theory of Optimal Processes*, 1986.

[2] J. Meditch, On the problem of optimal thrust programming for a lunar soft landing, in: *IEEE Transactions on Automatic Control*, Volume: **9**, Issue: 4, October 1964.

[3] S. Ploen and B. Açıkmeşe, A Powered Descent Guidance Algorithm for Mars Pinpoint Landing, DOI:10.2514/6.2005-6288, *Conference: AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005.

[4] S. Ploen and B. Açıkmeşe, Convex Programming Approach to Powered Descent Guidance for Mars Landing, *Journal of Guidance, Control, and Dynamics*, Vol. **30**, No. 5, September–October 2007.

[5] K. C. Toh, M. J. Todd and R. H. Tütüncü, SDPT3 - *a MATLAB software package for semidefinite programming*, 1999.

[6] J. F. Sturm, *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*, 1999.

[7] A. Domahidi, E. Chu and S. Boyd, *ECOS: An SOCP solver for embedded systems*, 2013 European Control Conference (ECC), July 17-19, 2013, Zürich, Switzerland.

[8] * * * https://www.spacex.com/vehicles/falcon-9/.