Spacecraft Landing Trajectory Optimization

May 29, 2019

Michael Spieler
Robotics and Autonomous Systems
michael.spieler@epfl.ch

Nicolas Gandar
Microengineering
nicolas.gandar@epfl.ch

Nicolas Lesimple
Computational Science & Engineering
nicolas.lesimple@epfl.ch

Abstract

In this paper we discuss trajectory optimization for propulsive spacecraft landing with minimal fuel consumption by formulating and solving a convex optimization problem using CVX and CVXPY software packages. In a second part the minimal time trajectory and effects on fuel consumption are explored.

1 Introduction

Convex optimization applied in control systems is an enabling technology for precision spacecraft landing. In 2012 the Mars Science Laboratory (MSL) achieved the most accurate Martian landing within an ellipse of 20 by 7 km using a powered descent stage [4]. This technology allows companies like SpaceX to successfully recover and reuse the first stage of their Falcon9 rocket by a controlled propulsive landing with high precision. In the following we discuss a minimal-fuel trajectory optimization problem for a simplified spacecraft model.

2 Optimal spacecraft landing problem

We consider exercise *13.2 Optimal spacecraft landing* from the *Additional Exercises for Convex Optimization* by Stephen Boyd and Lieven Vandenberghe [2] described in the following:

2.1 Problem Description

We consider the problem of optimizing the thrust profile for a spacecraft to carry out a landing at a target position. The spacecraft dynamics are

$$m\ddot{p} = f - mge_3,$$

where $m > 0$ is the spacecraft mass, $p(t) \in \mathbb{R}^3$ is the spacecraft position, with 0 the target landing position and $p_3(t)$ representing height, $f(t) \in \mathbb{R}^3$ is the thrust force, and $g > 0$ is the gravitational acceleration. For simplicity we assume that the spacecraft mass is constant. This is not always a good assumption, since the mass decreases with fuel use. We will also ignore any atmospheric friction. We must have

$p(T^{td}) = 0$ and $p'(T^{td}) = 0$, where T^{td} is the touchdown time. The spacecraft must remain in a region given by

$$p_3(t) \geq \alpha \left\| (p_1(t), p_2(t)) \right\|_2 ,$$

where $\alpha > 0$ is a given minimum glide slope. The initial position p(0) and velocity p'(0) are given.

The thrust force $f(t)$ is obtained from a single rocket engine on the spacecraft, with a given maximum thrust; an attitude control system rotates the spacecraft to achieve any desired direction of thrust. The thrust force is therefore characterized by the constraint $\left\| f(t) \right\|_2 \leq F^{max}$. The fuel use rate is proportional to the thrust force magnitude, so the total fuel use is

$$\int_0^{T^{td}} \gamma \left\| f(t) \right\|_2 dt,$$

where $\gamma > 0$ is the fuel consumption coefficient. The thrust force is discretized in time, i.e., it is constant over consecutive time periods of length $h > 0$, with $f(t) = f_k$ for $t \in [(k–1)h, kh)$, for k=1,...,K, where $T^{td} = Kh$. Therefore we have

$$v_{k+1} = v_k + (h/m)f_k – hge_3,$$

$$p_{k+1} = p_k + (h/2)(v_k + v_{k+1}),$$

where p_k denotes $p((k–1)h)$, and v_k denotes $p'((k–1)h)$. We will work with this discrete-time model. For simplicity, we will impose the glide slope constraint only at the times t = 0, h, 2h, . . . , Kh.

2.2 Problem formulation

The problem description above can be formulated mathematically in Problem 1. The decision variables are the position $\mathbf{p}$, velocity $\mathbf{v}$ and thrust $\mathbf{f}$ vectors at each of the K time steps (Notation: bold letters for vectors, subscript for time index). The function to minimize is the total fuel consumption as defined in section 2.1. The constraints are: *Max Thrust*, *Glide Cone*, initial state, final state and discretized dynamics (same order as in Problem 1). We observe that all equality constraints are linear, and the inequality constraints can be formulated as second-order cone constraints.

$$\begin{array}{lll}
\underset{\mathbf{p},\mathbf{v},\mathbf{f}}{\text{minimize}} & \sum_{i=1}^{K} h\gamma \left\| \mathbf{f}_i \right\|_2 & \\
\text{subject to} & \left\| \mathbf{f}_i \right\|_2 \leq F_{max} & i = 1, ..., K \\
& \mathbf{p}_i^z \geq \alpha \left\| \mathbf{p}_i^{x,y} \right\|_2 & i = 1, ..., K \\
& \mathbf{p}_1 = \mathbf{p}_{init} & \\
& \mathbf{v}_1 = \mathbf{v}_{init} & \\
& \mathbf{p}_K = 0 & \\
& \mathbf{v}_K = 0 & \\
& \mathbf{v}_{i+1} = \mathbf{v}_i + \frac{h}{m}\mathbf{f}_i - hg\mathbf{e_3} & i = 1, ..., K-1 \\
& \mathbf{p}_{i+1} = \mathbf{p}_i + \frac{h}{2}(\mathbf{v}_i + \mathbf{v}_{i+1}) & i = 1, ..., K-1
\end{array} \tag{1}$$

Table 1 lists the parameter values used for initial numerical evaluation.

Parameter	Value	Description
γ	1.0	Fuel consumption coefficient
α	0.5	Glide cone slope
h	1.0 s	Discretization time
K	35	Number of time steps
m	10 kg	Mass
$\mathbf{g}$	$0.1\,\mathrm{m/s^2}$	Gravity
F_{max}	10 N	Maximal thrust
$\mathbf{p}_{init}$	$\begin{bmatrix}50 & 50 & 100\end{bmatrix}$ m	Initial position
$\mathbf{v}_{init}$	$\begin{bmatrix}-10 & 0 & -10\end{bmatrix}$ m/s	Initial velocity

Table 1: Parameter values and description

2.2.1 Convexity and SOCP formulation

It is easy to show that problem 1 is convex. The cost function is a sum of norms, which is convex. The maximal thrust and glide cone constraints are quadratic and second-order cone constraints respectively, where both are convex. Finally, the equality constraints are linear and are therefore convex as well. [1]

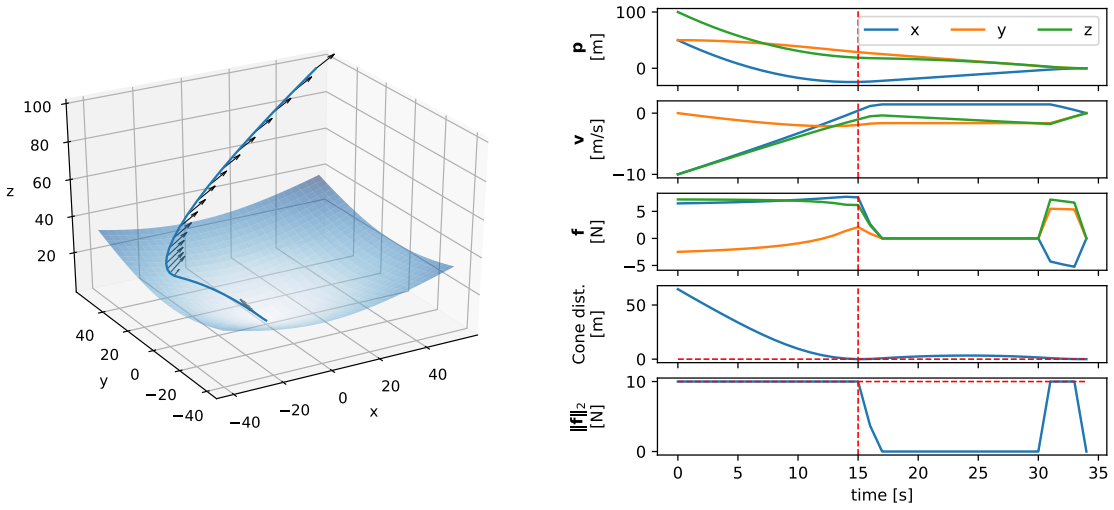
By introducing a variable $\mathbf{t}$ we can reformulate the problem as a Second-Order Cone Program (SOCP) shown in 2. There exist solvers that can solve SOCPs efficiently. Our CVXPY implementation is using ECOS, an interior-point solver for second-order cone programming [3]. Finally, we notice that without the glide cone constraint, the problem could be formulated as a QCQP.

$$
\begin{aligned}
\underset{\mathbf{p},\mathbf{v},\mathbf{f},\mathbf{t}}{\text{minimize}} \quad & \mathbf{1}^T\mathbf{t} \\
\text{subject to} \quad & \|\mathbf{f}_i\|_2 \leq \mathbf{t}_i && i = 1...K \\
& \|\mathbf{f}_i\|_2 \leq F_{max} && i = 1...K \\
& \|\mathbf{p}_i^{x,y}\|_2 \leq \tfrac{1}{\alpha}\mathbf{p}_i^z && i = 1...K \\
& \mathbf{p}_1 - \mathbf{p}_{init} = 0 \\
& \mathbf{v}_1 - \mathbf{v}_{init} = 0 \\
& \mathbf{p}_K = 0 \\
& \mathbf{v}_K = 0 \\
& \mathbf{v}_i - \mathbf{v}_{i+1} + \tfrac{h}{m}\mathbf{f}_i - hg\mathbf{e_3} = 0 && i = 1,...,K-1 \\
& \mathbf{p}_i - \mathbf{p}_{i+1} + \tfrac{h}{2}(\mathbf{v}_i + \mathbf{v}_{i+1}) = 0 && i = 1,...,K-1
\end{aligned}
\tag{2}
$$

3 Solution

Using CVX we solve Problem 2 using parameters from Table 1. The resulting trajectory and thrust profile are studied in the following. The Matlab code to solve the problem in CVX can be found in the Appendix A.

3.1 Trajectory and Constraint analysis



(a) Spacecraft trajectory and thrust vectors with the glide cone constraint.

(b) Evolution of position, velocity and thrust with constraint margins

Figure 1: Trajectory and variable evolution for $K = 35\,\mathrm{s}$.

In Figure 1a, we observe that the spacecraft starts by decelerating strongly to avoid leaving the allowed region. After almost touching the *Glide Cone* constraint border, it enters a parabolic trajectory towards the target where no thrust is applied until a final burn shortly before touchdown.

3.1.1 Constraint Activity

In Figure 1b we analyze the trajectory and thrust profile and the margins to the *Max Thrust* and *Glide Cone* constraints. One can visually verify that both constraints (in red) are always respected.

Figure 2 shows the dual variables of *Max Thrust* and *Glide Cone* constraints. The corresponding constraint is active where the duals are non-zero, which supports the findings from Figure 1a. The *Max Thrust* constraint is active during deceleration until the spacecraft reaches the cone border, where thrust is turned off. It is active again for final deceleration. The *Glide Cone* constraint is only active when approaching the cone at time $K = 15$ and upon landing.

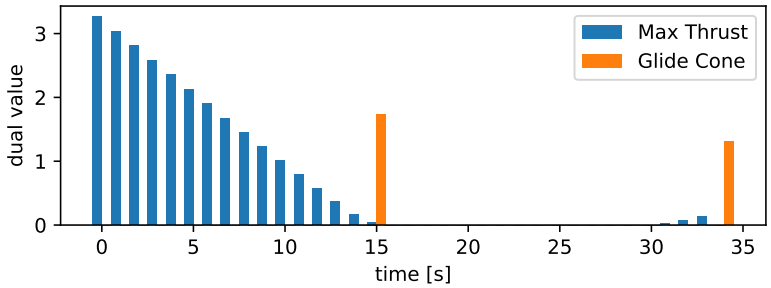
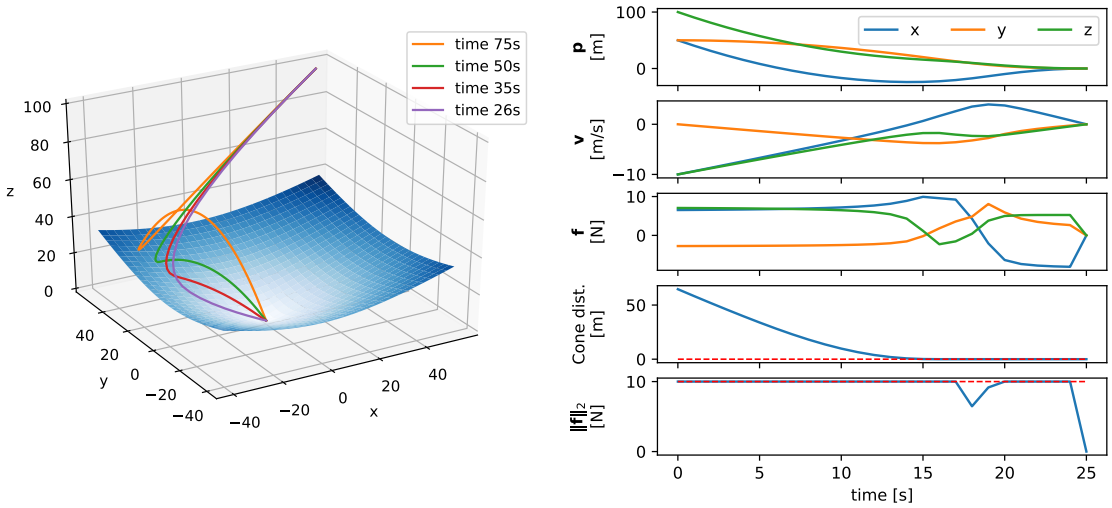


Figure 2: Dual variables of *Max Thrust* and *Glide Cone* constraints

3.2 Minimum Landing Time

To find the minimum landing time we search for a minimal K for which Problem 2 is still feasible. This is achieved by iteratively decreasing K and solving the problem until it becomes infeasible. Note: This can be done efficiently using binary search. The solution for $K_{min} = 26s$ is shown in Figure 3b.



(a) Trajectories for different landing times. The shortest feasible time $K_{min} = 26\,\mathrm{s}$

(b) Evolution of position, velocity, thrust with constraint margins. $K = 26\,\mathrm{s}$

Figure 3: Trajectories and variable evolution.

Comparing Figure 3b with Figure 1b we observe that the thrust is always active and most of the time saturated. It is likely due to the *Max Thrust* constraint that the problem becomes infeasible at $K = 25\,\mathrm{s}$ because the spacecraft can not accelerate and decelerate fast enough to reach the target in time.

In Figure 3a multiple trajectories with different feasible landing times are compared. We notice that the trajectory for small K lead more directly to the target, while for large K the coasting phase with zero thrust is longer. This hints at a weakness in the problem formulation: The actual landing is not modelled, where the spacecraft is supported by the ground. Therefore, it is more fuel-efficient to enter a slow parabolic trajectory instead of heading straight to the target and hovering there until timestep K.

3.3 Fuel Consumption

Finally, we look at the actual fuel consumption for different landing times which is shown in figure 4. We observe that for the minimum landing time the fuel consumption is maximal, which makes sense because hard acceleration/deceleration is costly. However, after a certain point fuel consumption increases again for increasing K because staying longer in the air requires more fuel as well. We identify minimal fuel consumption for landing time $K = 41\,\mathrm{s}$.

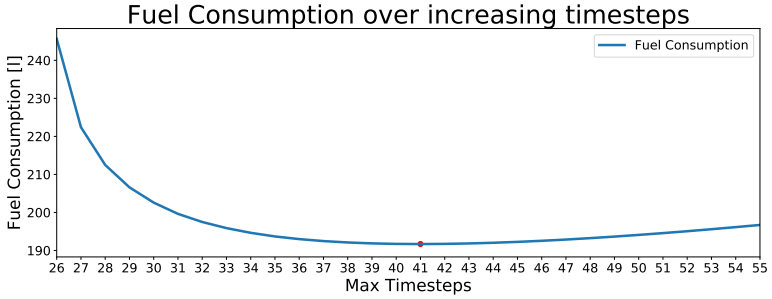


Figure 4: Fuel consumption for different landing times.

4 Conclusion

In this project we formulated and solved a minimum-fuel trajectory optimization problem for a simplified spacecraft model. We analyzed the evolution of fuel consumption for different landing times and highlighted that the fastest landing is not the most fuel efficient. Possible improvements could be to model the decreasing mass due fuel consumption or to model the ground supporting the spacecraft. However, those changes could lead to a non-convex problem which CVX can not solve.

References

[1] Stephen Boyd, Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, ISBN 0521833787, 2004.

[2] Stephen Boyd, Lieven Vandenberghe. *Additional Exercises for Convex Optimization*. `http://web.stanford.edu/~boyd/cvxbook/bv_cvxbook_extra_exercises.pdf`

[3] Alexander Domahidi, Eric Chu, Stephen Boyd. *ECOS: An SOCP solver for embedded systems*. European Control Conference (ECC), p3071-3076, 2013.

[4] NASA MSL *Entry, Descent, and Landing* Retrieved 2019 from `https://mars.nasa.gov/msl/mission/timeline/edl/`

A Matlab code

```
h = 1;
g = 0.1;
m = 10;
Fmax = 10;
p0 = [50,50,100]';
v0 = [-10,0,-10]';
alpha = 0.5;
gamma = 1;
K = 35;

cvx_begin
    % position, velocity, thrust vectors
    variables p(3,K) v(3,K) f(3,K);

    fuel = 0;
    for k = 1:K
        fuel = fuel + gamma * h * norm(f(:,k),2);
    end

    minimize ( fuel );
    subject to
        % Initial state
        p(:,1) == p0;
        v(:,1) == v0;
        % Target
        p(:,K) == zeros(3,1);
        v(:,K) == zeros(3,1);
        for k = 1:K
            % Maximal thrust
            norm(f(:,k), 2) <= Fmax;
            % Glide cone. The spacecraft must remain in this region
            p(3,k) >= alpha * norm(p(1:2,k), 2)
        end
        % Spacecraft dynamics constraints
        for k = 1:K-1
            v(:,k+1) == v(:,k) + h/m*f(:,k) - h*[0,0,g]';
            p(:,k+1) == p(:,k) + h/2*(v(:,k) + v(:,k+1));
        end
cvx_end

x = linspace(-40,55,30); y = linspace(0,55,30);
[X,Y] = meshgrid(x,y);
Z = alpha*sqrt(X.^2+Y.^2);
figure; colormap autumn; surf(X,Y,Z);
axis([-40,55,0,55,0,105]);
grid on; hold on;
plot3(p(1,:),p(2,:),p(3,:),'b','linewidth',1.5);
quiver3(p(1,1:K),p(2,1:K),p(3,1:K),...
        f(1,:),f(2,:),f(3,:),0.3,'k','linewidth',1.5);
```

B Python implementation

While initial tests were done in Matlab, we switched to Python with CVXPY for more flexibility. The project source code for generating the plots is published on GitHub: `https://github.com/nuft/cvx-optimal-spacecraft-landing`