AIAA SciTech Forum
4-8 January 2016, San Diego, California, USA
AIAA Guidance, Navigation, and Control Conference

10.2514/6.2016-0378

Successive Convexification for Fuel-Optimal Powered Landing with Aerodynamic Drag and Non-Convex Constraints

Michael Szmuk and Behçet Açıkmeşe

Dept. of Aeronautics & Astronautics, University of Washington, Seattle, WA 98195, USA

Andrew W. Berning Jr.

Dept. of Aerospace Engr. & Engr. Mechanics, University of Texas at Austin, Austin, TX 78712, USA

In this paper we present a convex programming approach for a fuel-optimal powered landing problem in the presence of aerodynamic drag forces and new types of non-convex control constraints. This problem presents a challenge for real-time guidance applications due to its non-convex control constraints, and to the nonlinearities introduced by aerodynamic drag, mass-depletion dynamics, and free final time. To make the problem tractable, we employ lossless convexification to address the minimum-thrust constraint, and successive convexification to eliminate the remaining non-convexities. The latter constitutes the main contribution of this paper, and relies on the use of linearization, trust regions, and relaxations. Through successive convexification, we formulate the problem as a sequence of iteratively solved second-order cone programming (SOCP) problems that is initialized using only a coarse guess of the time of flight. Through numerical simulations, we show that the proposed algorithm converges reliably after only a small number of successive convexification iterations, and is robust to a wide range of time of flight inputs. Furthermore, our algorithm can be implemented using powerful Interior Point Method (IPM) solvers, thus making it suitable for autonomous real-time applications.

Nomenclature

DOF	Degrees-of-Freedom	$\hat{\mathbf{e}}_u$	Up-pointing unit vector
IPM	Interior Point Method	m_{dry}	Vehicle dry mass
SC	Successive Convexification	m	Vehicle mass
SOCP	Second-Order Cone Programming	$\mathbf{r}$	Vehicle position vector
VTVL	Vertical Takeoff, Vertical Landing	$\mathbf{v}$	Vehicle velocity vector
i	Index of SC iteration	$\mathbf{a}$	Vehicle acceleration vector
t	Time since beginning of landing trajectory	$\mathbf{T}$	Vehicle thrust vector
Δt	Discretization sampling time	$\mathbf{D}$	Vehicle drag vector
t_k	Time at $t = k\Delta t$	$\mathbf{g}$	Local gravity vector
$\dot{x}$	Time rate of change of quantity x	g_0	Standard gravity
x_0	Quantity x at initial time	ρ	Local air density
x_f	Quantity x at final time	P_{amb}	Local air pressure
$x(t)$	Value of quantity x at time t	A_{nozzle}	Nozzle exit area of vehicle rocket motor
$x[k]$	Value of quantity x at time $t = t_k$	I_{sp}	Specific impulse of vehicle rocket motor
$x_i[k]$	Value of quantity $x[k]$ on ith SC iteration	C_D	Vehicle coefficient of drag
$\delta x_i[k]$	Used to denote $x_i[k] - x_{i-1}[k]$	S_D	Vehicle drag reference area
$\mathbf{\Psi}_{\mathbf{x}}[k]$	Used to denote $\left[\mathbf{x}^T[k] \;\; \mathbf{x}^T[k+1]\right]^T$	θ	Vehicle tilt angle off vertical
$\lvert\lvert\mathbf{x}\rvert\rvert$	2-norm of vector quantity $\mathbf{x}$	N	Number of time discretization points
$\hat{\mathbf{x}}$	Unit vector parallel to vector quantity $\mathbf{x}$	n_{SC}	Number of SC iterations

I. Introduction

In this paper we propose a successive convexification (SC) algorithm suitable for real-time guidance applications of the fuel-optimal powered landing problem with aerodynamic drag and non-convex constraints. The problem consists of a rapidly descending vehicle that uses a throttleable rocket engine to decelerate and maneuver the vehicle to a specified landing position and velocity. During its descent, the vehicle is subject to nonlinear aerodynamic drag forces, has finite propellant, and has an unspecified time of flight. Additionally, throughout the trajectory, the rocket motor must maintain a non-zero thrust inside of a specified thrust interval, while maintaining the vehicle's tilt angle below a prescribed maximum value.

In recent years, vertical takeoff, vertical landing (VTVL) rocket capabilities have come to the forefront of the aerospace industry.[1–3] While a number of reusable rocket concepts have been proposed, the VTVL concept owes its prominence to its potential scalability, as evidenced by its concurrent development on multiple suborbital and orbital launch vehicles. Two key challenges posed by the powered landing problem are the uncertainty in initial conditions prior to landing phase commencement, and the sensitivity of mission availability to landing divert capabilities. Our proposed guidance algorithm addresses both of these challenges. First, since the algorithm is suitable for real-time applications, it can be executed in close temporal proximity to the beginning of the powered landing phase. Thus, the initial conditions used in computing the guidance trajectory will have higher fidelity to the true state of the vehicle. Second, since our algorithm solves an optimal control problem, it is able to generate better performing *feasible* trajectories when compared to alternative heuristic guidance strategies. Thus, the vehicle can increase its landing divert performance, which in turn can alleviate operating envelope restrictions.

Our algorithm solves the aforementioned landing problem by employing lossless[4–11] and successive convexification.[12, 13] We use the latter convexification technique to generate a sequence of second-order cone programming (SOCP) problems,[14, 15] for which we can use powerful Interior Point Method (IPM) solvers.[16–18] This successive convexification technique consists of repeatedly linearizing the problem about a previously obtained trajectory, and iterating until a user-defined convergence criteria is met. To aid convergence, we introduce quadratic trust regions to keep the solution bounded, and add a relaxation to the dynamics to ensure feasibility throughout the convergence process. Consequently, our algorithm can accommodate non-convexities such as nonlinear aerodynamic drag, mass-depletion dynamics, and free final time, and can do so at a speed suitable for real-time applications.[19, 20]

The paper is organized as follows. In Section II, we formally state the continuous-time non-convex problem. In Section III, we discretize the problem, apply lossless convexification, and describe the successive convexification algorithm. Lastly, in Section IV, we present simulation results for an example powered landing scenario.

II. Problem Description

The objective of the method described in this paper is to obtain a trajectory that satisfies the specified boundary conditions, does not violate the specified state and control constraints, and minimizes the propellant used. Unless otherwise specified, we will use the word *trajectory* to refer to a time history of mass, position, velocity, and thrust.

In this section, we describe the continuous-time non-convex fuel-optimal powered landing problem. We begin by describing the vehicle model, after which we pose the continuous-time problem formulation consisting of the objective function, boundary conditions, and state and control constraints.

II.A. Vehicle Model

The vehicle is modeled as a point mass subject to 3-DOF translational motion. At any point in time, we denote the vehicle's mass, position, and velocity as $m(t)$, $\mathbf{r}(t)$, and $\mathbf{v}(t)$, respectively.

We assume that gravity, $\mathbf{g}$, air pressure, P_{amb}, and air density, ρ, are constant over the course of the trajectory. This is a reasonable assumption since the powered landing phase subsumes only a short portion of the entry, descent, and landing phase of the mission.

The vehicle's rocket motor produces a thrust vector $\mathbf{T}(t)$ that is assumed to be aligned with the vehicle's longitudinal axis. The direction of the thrust vector is used as a surrogate for the vehicle's attitude, thus maintaining the translational nature of the problem.

We model the vehicle as a sphere with a constant coefficient of drag, C_D, and drag reference area, S_D. Consequently, we assume lift is negligible, and that the magnitude of the drag force is agnostic to the direction of the relative wind. Thus, we express the aerodynamic drag force as

$$\mathbf{D}(t) = -\frac{1}{2}\rho S_D C_D ||\mathbf{v}(t)||\mathbf{v}(t) \tag{1}$$

In a vacuum, the mass-depletion dynamics can be assumed to be linearly proportional to the commanded thrust magnitude.[5–7] However, for applications where the ambient pressure and density are not negligible, we must modify the model to account for back-pressure losses. Denoting the specific impulse of the rocket motor as I_{sp}, standard gravity as g_0, and the exit area of the rocket nozzle as A_{nozzle}, the back-pressure compensated mass-depletion dynamics are given by

$$\alpha \triangleq \frac{1}{I_{sp}g_0} \tag{2}$$

$$\dot{m}_{bp} \triangleq \frac{P_{amb}A_{nozzle}}{I_{sp}g_0} \tag{3}$$

$$\dot{m}(t) = -\alpha||\mathbf{T}(t)|| - \dot{m}_{bp} \tag{4}$$

Lastly, the 3-DOF translational motion of the vehicle is modeled as a double integrator driven by an acceleration term, $\mathbf{a}(t)$, as shown below.

$$\dot{\mathbf{r}}(t) = \mathbf{v}(t) \tag{5}$$

$$\dot{\mathbf{v}}(t) = \mathbf{a}(t) \tag{6}$$

$$\mathbf{a}(t) = \frac{1}{m(t)}\left[\mathbf{T}\left(t\right) + \mathbf{D}\left(t\right)\right] + \mathbf{g} \tag{7}$$

II.B. Continuous-Time Formulation

The continuous-time problem formulation is summarized in Problem 1. The objective function maximizes the final mass of the vehicle, thus minimizing the propellant used during the powered landing.

The vehicle's initial mass, position vector, velocity vector, and thrust vector are specified as fixed parameters. Likewise, the vehicle's final position and velocity vectors are fixed. $\hat{\mathbf{n}}_0$ and $\hat{\mathbf{n}}_f$ are used to specify the direction of the initial and final thrust vectors, respectively. Note that the vehicle's final mass and thrust *magnitude* are left unconstrained.

The dynamics equations capture the mass-depletion dynamics of Eq. (4), and the 3-DOF translational dynamics from Eqs. (5-7). State constraints ensure that at every point along the trajectory the vehicle mass remains at or above the vehicle's dry mass, and that the vehicle remains above a prescribed glide-slope cone with a half-cone angle of γ_{gs}.[5,6] Lastly, the control constraints are used to ensure that: (a) the commanded thrust does not violate the vehicle's minimum and maximum thrust limits, (b) the thrust magnitude rate does not exceed the maximum throttling rate capabilities of the rocket motor, and (c) the vehicle tilt does not exceed a maximum tilt angle of θ_{max}.

III. Convex Formulation

In this section we will introduce three modifications to Problem 1. First, we will formulate the problem in discrete-time. Second, we will apply lossless convexification to handle the minimum-thrust control constraint.[5–7] Third, we will linearize the remaining non-convex elements of the problem to arrive at an SOCP formulation that can be solved repeatedly to achieve convergence.

The third step, which we refer to as *successive convexification*, involves the introduction of trust regions, the addition of relaxation terms, and the augmentation of the objective function. The primary contribution of this paper is the reliable implementation of the successive convexification process for the powered landing problem with nonlinear aerodynamic drag, mass-depletion dynamics, and free final time.

Problem 1.

Objective Function:

$$\max_{t_f, \mathbf{T}} \quad m(t_f) \quad \text{subject to:} \tag{8}$$

Boundary Conditions:

$$m(0) = m_0, \mathbf{r}(0) = \mathbf{r}_0, \mathbf{v}(0) = \mathbf{v}_0, \mathbf{T}(0) = T_0\hat{\mathbf{n}}_0 \tag{9}$$

$$\mathbf{r}(t_f) = \mathbf{0}, \mathbf{v}(t_f) = \mathbf{0}, \mathbf{T}(t_f) = ||\mathbf{T}(t_f)||\hat{\mathbf{n}}_f \tag{10}$$

Dynamics:

$$\dot{m}(t) = -\alpha||\mathbf{T}(t)|| - \dot{m}_{bp} \tag{11}$$

$$\dot{\mathbf{r}}(t) = \mathbf{v}(t) \tag{12}$$

$$\dot{\mathbf{v}}(t) = \mathbf{a}(t) \tag{13}$$

$$\mathbf{a}(t) = \frac{1}{m(t)}\left[\mathbf{T}(t) + \mathbf{D}(t)\right] + \mathbf{g} \tag{14}$$

State Constraints:

$$m_{dry} \leq m(t) \tag{15}$$

$$||\mathbf{r}(t)||\cos(\gamma_{gs}) \leq \hat{\mathbf{e}}_u^T\mathbf{r}(t) \tag{16}$$

Control Constraints:

$$0 \leq T_{min} \leq ||\mathbf{T}(t)|| \leq T_{max} \tag{17}$$

$$||\mathbf{T}(t)||\cos(\theta_{max}) \leq \hat{\mathbf{e}}_u^T\mathbf{T}(t) \tag{18}$$

$$\dot{T}_{min} \leq \frac{d}{dt}||\mathbf{T}(t)|| \leq \dot{T}_{max} \tag{19}$$

III.A. Discretization

To discretize Problem 1, we divide the trajectory into $N-1>0$ time segments of (equal) length, Δt. Thus, we are left with $N>1$ points at which the trajectory must be computed. Since N is specified, we can relate Δt to t_f via the following relation.

$$k_f \triangleq N - 1 \tag{20}$$

$$t_f = k_f\Delta t \tag{21}$$

For the remainder of the problem formulations, we will solve for Δt in lieu of t_f. We will use the index k to refer to time $t = k\Delta t$. Assume that $k \in [0, k_f]$ unless otherwise specified.

Discretization will most notably effect the dynamics equations, Eqs. (11-14), and the control constraint equations containing time derivatives, namely Eq. (19). Eqs. (11-14) are discretized under the assumption that thrust and acceleration are linearly interpolated between t_k and t_{k+1}. We perform linear interpolation to improve the accuracy of the numerical integration (compared to a zero-order-hold) while still obtaining affine relations suitable for SOCP formulations. We discretize Eq. (19) via forward differencing. The discretized versions of Eqs. (11-14, 19) are reflected in Problem 2 below.

Problem 2.

Objective Function:

$$\max_{\Delta t, \mathbf{T}} \quad m[k_f] \quad \text{subject to:} \tag{22}$$

Boundary Conditions:

$$m[0] = m_0, \mathbf{r}[0] = \mathbf{r}_0, \mathbf{v}[0] = \mathbf{v}_0, \mathbf{T}[0] = T_0\hat{\mathbf{n}}_0 \tag{23}$$

$$\mathbf{r}[k_f] = \mathbf{0}, \mathbf{v}[k_f] = \mathbf{0}, \mathbf{T}[k_f] = ||\mathbf{T}[k_f]||\hat{\mathbf{n}}_f \tag{24}$$

Dynamics:

$$m[k+1] = m[k] - \left[\frac{\alpha}{2}\left(||\mathbf{T}[k]|| + ||\mathbf{T}[k+1]||\right) + \dot{m}_{bp}\right]\Delta t \qquad k \in [0, k_f) \tag{25}$$

$$\mathbf{r}[k+1] = \mathbf{r}[k] + \mathbf{v}[k]\Delta t + \frac{1}{3}\left(\mathbf{a}[k] + \frac{1}{2}\mathbf{a}[k+1]\right)\Delta t^2 \qquad k \in [0, k_f) \tag{26}$$

$$\mathbf{v}[k+1] = \mathbf{v}[k] + \frac{1}{2}\left(\mathbf{a}[k] + \mathbf{a}[k+1]\right)\Delta t \qquad k \in [0, k_f) \tag{27}$$

$$\mathbf{a}[k] = \frac{1}{m[k]}\left(\mathbf{T}[k] + \mathbf{D}[k]\right) + \mathbf{g} \tag{28}$$

State Constraints:

$$m_{dry} \leq m[k] \tag{29}$$

$$||\mathbf{r}[k]||\cos(\gamma_{gs}) \leq \hat{\mathbf{e}}_u^T\mathbf{r}[k] \tag{30}$$

Control Constraints:

$$0 \leq T_{min} \leq ||\mathbf{T}[k]|| \leq T_{max} \tag{31}$$

$$||\mathbf{T}[k]||\cos(\theta_{max}) \leq \hat{\mathbf{e}}_u^T\mathbf{T}[k] \tag{32}$$

$$\dot{T}_{min} \leq \frac{||\mathbf{T}[k+1]|| - ||\mathbf{T}[k]||}{\Delta t} \leq \dot{T}_{max} \qquad k \in [0, k_f) \tag{33}$$

III.B. Lossless Convexification

Eqs. (31-32) constitute a set of non-convex constraints, where the latter is convex for $\theta_{max} \leq 90°$, and non-convex otherwise. To circumvent this obstacle, we replace Eqs. (31-32) with the following lossless relaxation.[5–7]

$$||\mathbf{T}[k]|| \leq \Gamma[k] \tag{34}$$

$$0 \leq T_{min} \leq \Gamma[k] \leq T_{max} \tag{35}$$

$$\Gamma[k]\cos(\theta_{max}) \leq \hat{\mathbf{e}}_u^T\mathbf{T}[k] \tag{36}$$

Performing lossless convexification ensures that the optimal solution for the relaxed problem recovers the optimal solution of the original non-convex problem. In recognition of the fact that we are extending the application beyond the scope of the original proof, we ensure that our numerical simulations verify that this relaxation indeed remains lossless (see Section IV.A).

We further modify this problem formulation by including Γ as a free variable in the objective function maximization, adding an initial condition for Γ to Eq. (23), and replacing all instances of $||\mathbf{T}[k]||$ in Eqs. (24-25, 33) with $\Gamma[k]$. Thus, we arrive at the problem formulation summarized in Problem 3.

Problem 3.

Objective Function:

$$\max_{\Delta t, \mathbf{T}, \Gamma} \quad m[k_f] \quad \text{subject to:} \tag{37}$$

Boundary Conditions:

$$m[0] = m_0, \mathbf{r}[0] = \mathbf{r}_0, \mathbf{v}[0] = \mathbf{v}_0, \mathbf{T}[0] = \Gamma[0]\hat{\mathbf{n}}_0, \Gamma[0] = \Gamma_0 \tag{38}$$

$$\mathbf{r}[k_f] = \mathbf{0}, \mathbf{v}[k_f] = \mathbf{0}, \mathbf{T}[k_f] = \Gamma[k_f]\hat{\mathbf{n}}_f \tag{39}$$

Dynamics:

$$m[k+1] = m[k] - \left[\frac{\alpha}{2}\left(\Gamma[k] + \Gamma[k+1]\right) + \dot{m}_{bp}\right]\Delta t \qquad k \in [0, k_f) \tag{40}$$

$$\mathbf{r}[k+1] = \mathbf{r}[k] + \mathbf{v}[k]\Delta t + \frac{1}{3}\left(\mathbf{a}[k] + \frac{1}{2}\mathbf{a}[k+1]\right)\Delta t^2 \qquad k \in [0, k_f) \tag{41}$$

$$\mathbf{v}[k+1] = \mathbf{v}[k] + \frac{1}{2}\left(\mathbf{a}[k] + \mathbf{a}[k+1]\right)\Delta t \qquad k \in [0, k_f) \tag{42}$$

$$\mathbf{a}[k] = \frac{1}{m[k]}\left(\mathbf{T}[k] + \mathbf{D}[k]\right) + \mathbf{g} \tag{43}$$

State Constraints:

$$m_{dry} \leq m[k] \tag{44}$$

$$||\mathbf{r}[k]|| \cos(\gamma_{gs}) \leq \hat{\mathbf{e}}_u^T \mathbf{r}[k] \tag{45}$$

Control Constraints:

$$||\mathbf{T}[k]|| \leq \Gamma[k] \tag{46}$$

$$0 \leq T_{min} \leq \Gamma[k] \leq T_{max} \tag{47}$$

$$\Gamma[k] \cos(\theta_{max}) \leq \hat{\mathbf{e}}_u^T \mathbf{T}[k] \tag{48}$$

$$\dot{T}_{min} \leq \frac{\Gamma[k+1] - \Gamma[k]}{\Delta t} \leq \dot{T}_{max} \qquad k \in [0, k_f) \tag{49}$$

III.C. Successive Convexification

In this section we present the main contribution of this paper. This method is in the same spirit of methods presented in [12, 13, 21]. However, our method differs in the way we perform time of flight search, in our implementation of trust regions for non-convex control constraints and nonlinear dynamics, and in our addition of a relaxation to the dynamics to avoid artificial infeasibility. Our successive convexification procedure requires us to solve a sequence of n_{SC} SOCP problems, where $n_{SC} > 1$. During the first iteration ($i = 0$), we solve a simplified problem which imposes very mild assumptions on the nature of the trajectory. For the remaining ($n_{SC} - 1$) iterations, we linearize the i^{th} iteration about the trajectory computed in the $(i-1)^{th}$ iteration, and solve the full (linearized) problem.

The remainder of this section is outlined as follows. In Sections III.C.1-III.C.3 we describe the three modifications required to cast Problem 3 into SOCP form: (a) linearization, which applies only to iterations where $i > 0$, (b) the introduction of quadratic trust region constraints that ensure the problem remains bounded during every iteration, and (c) the addition of a relaxation that prevents artificial infeasibility potentially induced by linearization. In Section III.C.4 we discuss the assumptions and simplifications used in the first iteration to initialize the algorithm in a robust manner. Lastly, in Section III.C.5 we summarize the final problem formulation and provide an outline of our successive convexification algorithm.

III.C.1. Linearization

The linearization process we discuss here pertains to iterations where $i > 0$. The formulation presented in Problem 3 has three sources of nonlinearity: (a) the Δt terms present in Eqs. (40-42), (b) the drag term defined in Eq. (1) and present in Eq. (43), and (c) the mass term present in the denominator of Eq. (43). The first item is a consequence of including the final time as a solution variable, while the last item is due to the positive $\dot{m}_{bp}$ term induced in Eq. (40) by non-negligible back-pressure losses. Note that we cannot circumvent the mass nonlinearity in Eq. (43) using the change of variables proposed in [5–7] due to the vehicle's changing ballistic coefficient. The linearization process gives rise to two issues, both of which are addressed in Sections III.C.2 and III.C.3.

III.C.2. Trust Region Constraints

The first concern we introduce when linearizing is rendering the problem unbounded. To mitigate this risk, we want to ensure that variables involved in the linearization do not deviate significantly from their values obtained in the previous iteration. In doing so, we get the added benefit of keeping the solution closer to the valid region of the linearization.

For notational convenience, we will denote the difference between a quantity in the i^{th} iteration and its counterpart in the $(i-1)^{th}$ iteration as

$$\delta x_i[k] \triangleq x_i[k] - x_{i-1}[k] \qquad i > 0 \tag{50}$$

Our approach is to apply the trust region constraints on $\delta\Delta t_i$ and $\delta\mathbf{T}_i$. The former is constrained to ensure that the final time does not undergo excessively large changes across iterations. The latter is constrained in recognition of the fact that keeping $||\delta\mathbf{T}_i||$ bounded guarantees that $|\delta\Gamma_i|$ and $||\delta\mathbf{v}_i||$ follow suite. This observation follows from the guarantees of lossless convexification, and from the problem's controllability property.

To impose the quadratic trust region constraints, we introduce a new set of η-variables to bound the δ-quantities from above pointwise in time.

$$\delta\Delta t_i^2 \leq \eta_{\Delta t} \qquad i > 0 \tag{51}$$

$$\delta\mathbf{T}_i^T[k]\delta\mathbf{T}_i[k] \leq \eta_{\mathbf{T}}[k] \qquad i > 0 \tag{52}$$

Later, in Section III.C.5, we will augment the η-variables to the objective function, and add Eqs. (51-52) to the new problem formulation. As indicated in Eqs. (51-52), the trust regions will only be applied for iterations where $i > 0$.

III.C.3. Relaxation

The second concern we introduce when linearizing is artificial infeasibility. At various points in the solution space, the linearized formulation can generate an infeasible problem, even if the underlying nonlinear problem is feasible. In such situations, the infeasibility obstructs the iteration process and prevents convergence. The most evident example of such an infeasible condition arises when the problem is linearized about a time of flight that is unrealistically short (i.e. Δt is very small). In such a case, we intuitively understand that there is no feasible control sequence that can satisfy the prescribed boundary conditions.

To prevent artificial infeasibility, we introduce an additive relaxation term, $\mathbf{a}_R$, to Eq. (43). The relaxation can be understood as a synthetic acceleration that acts on the vehicle when the set of feasible $\mathbf{T}$'s is not sufficient. This modification is applied to all iterations, and partakes in the linearization process when $i > 0$. We bound the magnitudes of the $\mathbf{a}_R$ terms pointwise in time using a set of κ-variables, as shown below.

$$||\mathbf{a}_R[k]|| \leq \kappa_{\mathbf{a},R}[k] \tag{53}$$

As with the η-variables in Section III.C.2, we will augment the κ-variables to the objective function, and add Eq. (53) to the new problem formulation. This will be done in Section III.C.5.

Lastly, we express the relaxation in this fashion in order to leverage the controllability of the problem, hence obtaining a good trade-off between robustness and problem size. However, while this relaxation guards against a vast set of artificial infeasibility conditions, it does not make the problem foolproof. Additional relaxations can be added to make the formulation more robust, depending on the requirements of the application.

III.C.4. Initialization

Before we present the finalized problem formulation, we address the simplifications and assumptions applied to the first successive convexification iteration. Generally, prior to solving the guidance problem, we do not have enough a-priori knowledge to generate a good estimate of the full trajectory. In the absence of a trajectory estimate, we must employ a different method in the first iteration than we do in the succeeding ones. Since we are interested in making our algorithm's initialization as robust as possible, we elect to rid the first iteration of explicit linearization, and to make some mild assumptions about key scalar quantities in the problem. Specifically, the first iteration requires an estimate of (a) the time of flight (or equivalently Δt), (b) the mass profile, and (c) the speed profile.

During the first iteration, we will use $\Delta\tau$ to denote a constant version of Δt. We can compute $\Delta\tau$ from a user-specified guess of the time of flight, $t_{f,s}$, as shown in Eq. (54).

$$\Delta\tau = t_{f,s}/k_f \tag{54}$$

Denoting the reference mass and speed as μ and s, respectively, we forgo linearization and simplify Eqs. (1, 43) as shown below.

$$\mathbf{D}[k] = -\frac{1}{2}\rho S_D C_D s[k]\mathbf{v}[k] \tag{55}$$

$$\mathbf{a}[k] = \frac{1}{\mu[k]}\left(\mathbf{T}[k] + \mathbf{D}[k]\right) + \mathbf{a}_R[k] + \mathbf{g} \tag{56}$$

Note that Eq. (56) includes the relaxation term introduced in Section III.C.3. As a consequence of our simplifications, Eqs. (40-42, 55-56) can now be formulated in the SOCP framework.

For simplicity and generality, we assume that μ and s vary linearly from m_0 to m_{dry}, and from $||\mathbf{v}_0||$ to $||\mathbf{v}_f||$, respectively. Thus, with only a coarse user-specified guess of the time of flight, this initialization process allows us to generate an estimate of the full trajectory with which to initiate the remaining sequence of $(n_{SC} - 1)$ iterations.

III.C.5. Successive Convexification Algorithm

In this section we will summarize the finalized problem formulation and the successive convexification algorithm. The formulation used in the first iteration is shown in Problem 4, and the formulation used in the remaining $(n_{SC} - 1)$ iterations is provided in Problem 5. The algorithm is outlined in Algorithm 1. Before proceeding, note the following:

a). As before, assume that $k \in [0, k_f]$ unless otherwise specified.

b). Define $||\eta_{\mathbf{T}}||$ and $||\kappa_{\mathbf{a},R}||$ as shown in Eqs. (57-58).

$$||\eta_{\mathbf{T}}|| \triangleq \left[\sum_{k=0}^{k_f}\Big(\eta_{\mathbf{T}}[k]\Big)^2\right]^{1/2} \tag{57}$$

$$||\kappa_{\mathbf{a},R}|| \triangleq \left[\sum_{k=0}^{k_f}\Big(\kappa_{\mathbf{a},R}[k]\Big)^2\right]^{1/2} \tag{58}$$

c). We will use a new set of w-variables to denote weighting parameters in the objective function. These variables are assumed to be strictly positive.

d). Given a generic variable $\mathbf{x}$, we will define the following notation for use in Problem 5.

$$\mathbf{\Psi}_{\mathbf{x}}[k] \triangleq \Big[\mathbf{x}^T[k] \quad \mathbf{x}^T[k+1]\Big]^T \qquad k \in [0, k_f) \tag{59}$$

Problem 4.

Objective Function:

$$\min_{\mathbf{T},\Gamma} \quad -w_{m,f}m[k_f] + w_{\kappa,\mathbf{a},R}||\kappa_{\mathbf{a},R}|| \quad \text{subject to:} \tag{60}$$

Boundary Conditions:

$$m[0] = m_0, \mathbf{r}[0] = \mathbf{r}_0, \mathbf{v}[0] = \mathbf{v}_0, \mathbf{T}[0] = \Gamma[0]\hat{\mathbf{n}}_0, \Gamma[0] = \Gamma_0 \tag{61}$$

$$\mathbf{r}[k_f] = \mathbf{0}, \mathbf{v}[k_f] = \mathbf{0}, \mathbf{T}[k_f] = \Gamma[k_f]\hat{\mathbf{n}}_f \tag{62}$$

Dynamics:

$$m[k+1] = m[k] - \left[\frac{\alpha}{2}\left(\Gamma[k] + \Gamma[k+1]\right) + \dot{m}_{bp}\right]\Delta\tau \qquad k \in [0, k_f) \tag{63}$$

$$\mathbf{r}[k+1] = \mathbf{r}[k] + \mathbf{v}[k]\Delta\tau + \frac{1}{3}\left(\mathbf{a}[k] + \frac{1}{2}\mathbf{a}[k+1]\right)\Delta\tau^2 \qquad k \in [0, k_f) \tag{64}$$

$$\mathbf{v}[k+1] = \mathbf{v}[k] + \frac{1}{2}\left(\mathbf{a}[k] + \mathbf{a}[k+1]\right)\Delta\tau \qquad k \in [0, k_f) \tag{65}$$

$$\mathbf{a}[k] = \frac{1}{\mu[k]}\left(\mathbf{T}[k] - \frac{1}{2}\rho S_D C_D s[k]\mathbf{v}[k]\right) + \mathbf{a}_R[k] + \mathbf{g} \tag{66}$$

State Constraints:

$$m_{dry} \leq m[k] \tag{67}$$

$$||\mathbf{r}[k]||\cos(\gamma_{gs}) \leq \hat{\mathbf{e}}_u^T\mathbf{r}[k] \tag{68}$$

Control Constraints:

$$||\mathbf{T}[k]|| \leq \Gamma[k] \tag{69}$$

$$0 \leq T_{min} \leq \Gamma[k] \leq T_{max} \tag{70}$$

$$\Gamma[k]\cos(\theta_{max}) \leq \hat{\mathbf{e}}_u^T\mathbf{T}[k] \tag{71}$$

$$\dot{T}_{min}\Delta\tau \leq \Gamma[k+1] - \Gamma[k] \leq \dot{T}_{max}\Delta\tau \qquad k \in [0, k_f) \tag{72}$$

SC Modifications:

$$||\mathbf{a}_R[k]|| \leq \kappa_{\mathbf{a},R}[k] \tag{73}$$

IV. Simulation Results

In this section we present an example scenario of a powered rocket landing, subject to the constraints discussed in this paper. Our algorithm was implemented in MATLAB using CVX,[17] SDPT3,[15] and the parameters listed in Tables 1 and 2. Note that the thrust parameters $T_{min,vac}$, $T_{max,vac}$, and $\Gamma_{0,vac}$ are expressed as vacuum thrusts, and that vector quantities are expressed in an Up-East-North reference frame.

We present the results in two parts. In Section IV.A we will look at the case where $t_{f,s} = 15$ [s], and will focus on the trajectory generated in the final successive convexification iteration. In particular, we will discuss the nature of the trajectory, and the constraints activated during the flight. In Section IV.B we will investigate the convergence behavior of the successive convexification process by looking at (a) position, velocity, and thrust to assess the degree of convergence, (b) the behavior of the relaxation term throughout the convergence process, and (c) the robustness of the process to different time of flight initializations.

Problem 5.

Objective Function:

$$\min_{\Delta t,\mathbf{T},\Gamma} \quad -w_{m,f}m[k_f] + w_{\eta,\Delta t}\eta_{\Delta t} + w_{\eta,\mathbf{T}}||\eta_{\mathbf{T}}|| + w_{\kappa,\mathbf{a},R}||\kappa_{\mathbf{a},R}|| \tag{74}$$

subject to:

Boundary Conditions:

$$m[0] = m_0, \mathbf{r}[0] = \mathbf{r}_0, \mathbf{v}[0] = \mathbf{v}_0, \mathbf{T}[0] = \Gamma[0]\hat{\mathbf{n}}_0, \Gamma[0] = \Gamma_0 \tag{75}$$

$$\mathbf{r}[k_f] = \mathbf{0}, \mathbf{v}[k_f] = \mathbf{0}, \mathbf{T}[k_f] = \Gamma[k_f]\hat{\mathbf{n}}_f \tag{76}$$

Dynamics:

$$\boldsymbol{\Psi}[k] \triangleq \begin{bmatrix}\Delta t & \boldsymbol{\Psi}_m^T[k] & \boldsymbol{\Psi}_\Gamma^T[k] & \boldsymbol{\Psi}_\mathbf{v}^T[k] & \boldsymbol{\Psi}_\mathbf{T}^T[k] & \boldsymbol{\Psi}_{\mathbf{a},R}^T[k]\end{bmatrix}^T \qquad k \in [0, k_f) \tag{77}$$

$$f_m(\boldsymbol{\Psi}[k]) \triangleq -\left[\frac{\alpha}{2}\left(\Gamma[k] + \Gamma[k+1]\right) + \dot{m}_{bp}\right]\Delta t \qquad k \in [0, k_f) \tag{78}$$

$$\mathbf{f}_\mathbf{r}(\boldsymbol{\Psi}[k]) \triangleq \mathbf{v}[k]\Delta t + \frac{1}{3}\left(\mathbf{a}[k] + \frac{1}{2}\mathbf{a}[k+1]\right)\Delta t^2 \qquad k \in [0, k_f) \tag{79}$$

$$\mathbf{f}_\mathbf{v}(\boldsymbol{\Psi}[k]) \triangleq \frac{1}{2}\left(\mathbf{a}[k] + \mathbf{a}[k+1]\right)\Delta t \qquad k \in [0, k_f) \tag{80}$$

$$m[k+1] = m[k] + f_m(\boldsymbol{\Psi}_{i-1}[k]) + \left.\frac{\partial f_m}{\partial \boldsymbol{\Psi}}\right|_{\boldsymbol{\Psi}_{i-1}[k]} \delta\boldsymbol{\Psi}_i[k] \qquad k \in [0, k_f) \tag{81}$$

$$\mathbf{r}[k+1] = \mathbf{r}[k] + \mathbf{f}_\mathbf{r}(\boldsymbol{\Psi}_{i-1}[k]) + \left.\frac{\partial \mathbf{f}_\mathbf{r}}{\partial \boldsymbol{\Psi}}\right|_{\boldsymbol{\Psi}_{i-1}[k]} \delta\boldsymbol{\Psi}_i[k] \qquad k \in [0, k_f) \tag{82}$$

$$\mathbf{v}[k+1] = \mathbf{v}[k] + \mathbf{f}_\mathbf{v}(\boldsymbol{\Psi}_{i-1}[k]) + \left.\frac{\partial \mathbf{f}_\mathbf{v}}{\partial \boldsymbol{\Psi}}\right|_{\boldsymbol{\Psi}_{i-1}[k]} \delta\boldsymbol{\Psi}_i[k] \qquad k \in [0, k_f) \tag{83}$$

$$\mathbf{a}[k] = \frac{1}{m[k]}\left(\mathbf{T}[k] + \mathbf{D}[k]\right) + \mathbf{a}_R[k] + \mathbf{g} \tag{84}$$

State Constraints:

$$m_{dry} \leq m[k] \tag{85}$$

$$||\mathbf{r}[k]||\cos(\gamma_{gs}) \leq \hat{\mathbf{e}}_u^T\mathbf{r}[k] \tag{86}$$

Control Constraints:

$$||\mathbf{T}[k]|| \leq \Gamma[k] \tag{87}$$

$$0 \leq T_{min} \leq \Gamma[k] \leq T_{max} \tag{88}$$

$$\Gamma[k]\cos(\theta_{max}) \leq \hat{\mathbf{e}}_u^T\mathbf{T}[k] \tag{89}$$

$$\dot{T}_{min}\Delta t \leq \Gamma[k+1] - \Gamma[k] \leq \dot{T}_{max}\Delta t \qquad k \in [0, k_f) \tag{90}$$

SC Modifications

$$\delta\Delta t_i^2 \leq \eta_{\Delta t} \tag{91}$$

$$\delta\mathbf{T}_i^T[k]\delta\mathbf{T}_i[k] \leq \eta_\mathbf{T}[k] \tag{92}$$

$$||\mathbf{a}_R[k]|| \leq \kappa_{\mathbf{a},R}[k] \tag{93}$$

Algorithm 1.

1. Specify vehicle and environmental parameters (e.g. m_{dry}, θ_{max}, P_{amb}, etc.), boundary conditions (e.g. $\mathbf{r}_0$, Γ_0, m_0, etc.), and algorithm parameters (e.g. N, n_{SC}, $w_{m,f}$, etc.).
2. Specify a time of flight guess, $t_{f,s}$, and compute $\Delta\tau$ using Eq. 54.
3. Compute mass and speed profiles for first iteration using Eqs. (94-95), for integers $k \in [0, k_n]$.

$$\mu[k] = \left(\frac{k_n - k}{k_n}\right) m_0 + \left(\frac{k}{k_n}\right) m_{dry} \tag{94}$$

$$s[k] = \left(\frac{k_n - k}{k_n}\right) ||\mathbf{v}_0|| + \left(\frac{k}{k_n}\right) ||\mathbf{v}_f|| \tag{95}$$

4. Solve Problem 4 using $\Delta\tau$, μ, and s to obtain trajectory $\mathcal{T}_0$.
5. For $i = 1, 2, ..., n_{SC} - 1$
   (a) Solve Problem 5, linearizing about trajectory $\mathcal{T}_{i-1}$, to obtain trajectory $\mathcal{T}_i$.
   (b) Exit if $\mathcal{T}_i$ is within some acceptable tolerance of $\mathcal{T}_{i-1}$, or if $i = n_{SC} - 1$.

IV.A. Converged Solution

The parameters listed in Tables 1 and 2 represent a scenario where a vehicle is 500 [m] above the ground, and 500 [m] east of the landing pad. Initially the vehicle is loaded with 5000 [kg] of propellant, and is moving downwards and tangentially (relative to the pad) at a speed of 50 [m/s] in each direction. The vehicle is subjected to environmental conditions akin to those at sea-level on Earth, and to the following path and control constraints: (a) a minimum and maximum thrust constraint, (b) a minimum and maximum thrust magnitude rate constraint, (c) a thrust pointing limit of 15° off vertical, and (d) a glide-slope cone constraint with a half-angle of 80° (limiting the glide slope to 10° at sufficiently low altitudes).

Table 1. Simulation Parameters

Parameter	Value	Units
ρ	1.0	kg/m^3
P_{amb}	100	kPa
g_0	9.807	m/s^2
$\mathbf{g}$	$[-g_0 \quad 0 \quad 0]^T$	m/s^2
m_{dry}	10,000	kg
A_{nozzle}	0.5	m^2
I_{sp}	300	s
$T_{min,vac}$	100	kN
$T_{max,vac}$	250	kN
$\dot{T}_{min}$	-100	kN/s
$\dot{T}_{max}$	100	kN/s
θ_{max}	15	°
γ_{gs}	80	°
S_D	10	m^2
C_D	1.0	-

Table 2. B.C.'s and Algorithm Parameters

Parameter	Value	Units
$t_{f,s}$	15	s
m_0	15,000	kg
$\mathbf{r}_0$	$[500 \quad 500 \quad 0]^T$	m
$\mathbf{v}_0$	$[-50 \quad 0 \quad 50]^T$	m/s
$\Gamma_{0,vac}$	175	kN
$\hat{\mathbf{n}}_0$	$[1 \quad 0 \quad 0]^T$	-
$\hat{\mathbf{n}}_f$	$[1 \quad 0 \quad 0]^T$	-
N	30	-
n_{SC}	10	-
$w_{m,f}$	1.0	kg^{-1}
$w_{\eta,\Delta t}$	0.1	s^{-1}
$w_{\eta,\mathbf{T}}$	1×10^{-2}	kN^{-1}
$w_{\kappa,\mathbf{a},R}$	5×10^5	s^2/m

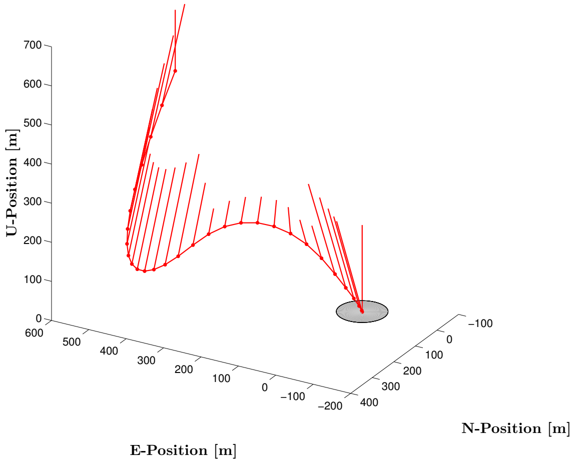

Figure 1. Three-dimensional trajectory produced by the last successive convexification iteration. *The dots along the trajectory indicate discretization points, and the lines intersecting the trajectory at the discretization points represent scaled commanded thrust vectors.*

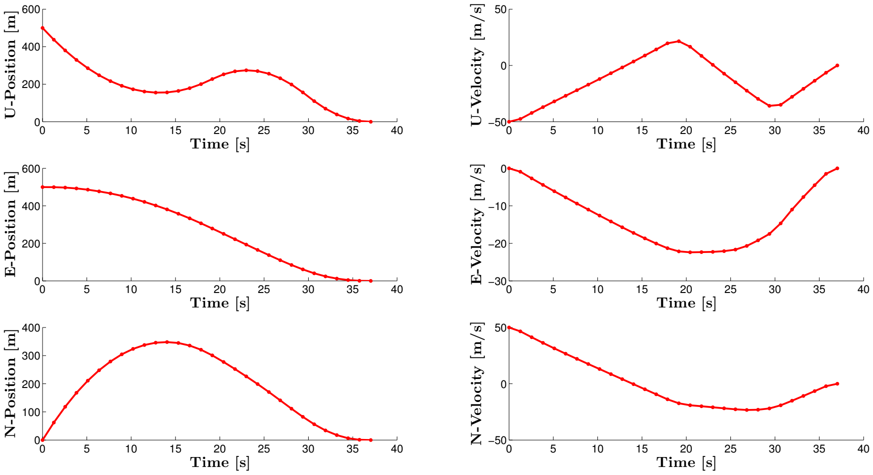

Figure 2. Up, east, and north components of the positions and velocities. *This figure shows a componentwise representation of the positions and velocities of the trajectory shown in Figure 1. The hop maneuver is evident in the top two plots.*

The thrust magnitude was initiated at half throttle. We arbitrarily elected to use 10 successive convexification iterations and 30 discretization points. The objective function weights were selected to disincentivize the use of the relaxation term $\mathbf{a}_R$, and to limit adverse effect of the trust regions on convergence speed.

The four figures presented in this section show the mass, position, velocity, and thrust profiles produced by the last successive convexification iteration for a run initialized with $t_{f,s} = 15$ [s]. Figure 1 shows the three-dimensional profile of the trajectory. The points along the trajectory represent the discretization points, and the lines that intersect the trajectory at the discretization points represent the scaled commanded thrust vectors. The thrust vectors are observed to be vertical at the beginning and end of the trajectory, in accordance with the specified pointing unit vectors $\hat{\mathbf{n}}_0$ and $\hat{\mathbf{n}}_f$. The vehicle's initial north-down velocity is evident from the commanded thrust input to the south-west, acting to slow the vehicle's tangential and vertical velocity, while simultaneously accelerating it radially towards the pad. In the vertical direction, after arresting its initial motion, the vehicle is observed to ascend again. This maneuver is due to the limited tilt authority afforded to the vehicle. As evidenced by the solution, under this circumstance it is more efficient to perform a "bang-coast-bang" maneuver, than it is to perform a constant-altitude translation to the pad. Figure 2 shows the time histories of the position and velocity components of the trajectory presented in Figure 1. The hop maneuver is evident in the up-position and -velocity plots.

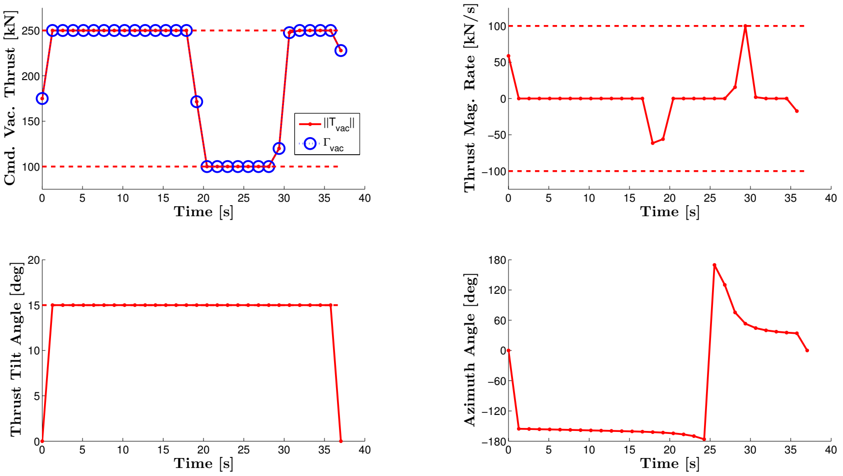

Figure 3. Thrust profile of the converged trajectory. *In the top left plot, the vehicle's vacuum thrust profile is shown, along with the variable* Γ*, and the minimum and maximum thrust constraints. In the top right plot, the thrust magnitude rate is shown along with its minimum and maximum bounds. The bottom left plot shows the tilt angle of the commanded thrust vector, as well as the 15° maximum tilt limit. Lastly, the bottom right plot shows the azimuth of the thrust vector.*

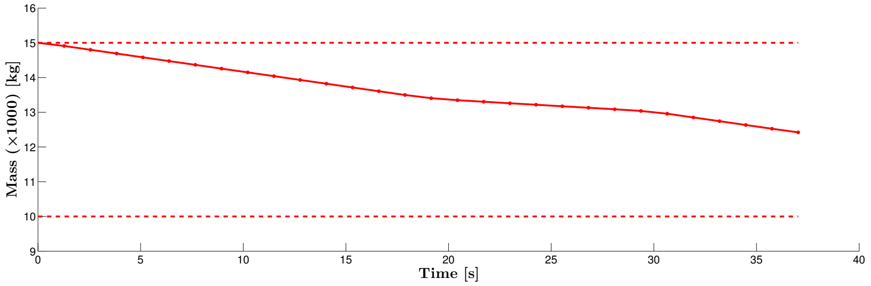

Figure 4. Vehicle mass as a function of time. *The dashed lines at the top and bottom of the figures represent the initial mass and the dry mass of the vehicle, respectively.*

Figure 3 shows the (vacuum) thrust profile commanded by our algorithm. The "bang-coast-bang" nature of the trajectory is readily evident in the throttle profile shown in the top left plot. The dashed lines at the top and bottom of the plot represent the maximum and minimum thrust constraints, respectively. Additionally, as mentioned in Section III.B, $||\mathbf{T}||$ and Γ are observed to retain the lossless nature of the convexification introduced in Eqs. (34-36).

The top right plot in Figure 3 shows the thrust magnitude rate, and the corresponding bounds enforced in the algorithm. Note that the thrust magnitude rate constraint is activated at $t \approx 30$ [s].

The bottom two plots show the thrust tilt and azimuth as functions of time. It is evident that the tilt activates the 15° constraint immediately, and keeps it activated until the second to last discretization point. At the same time, the azimuth plot shows us that the vehicle holds a nearly constant south-by-southwest azimuth during the first "bang" phase, and halfway through the "coast" phase. The remainder of the trajectory exhibits a rapid swing of the thrust azimuth through the south, where the vehicle proceeds past the east direction and reduces its tilt just prior to touchdown.

Lastly, Figure 4 shows the mass profile of the vehicle. The initial and dry masses of the vehicle are indicated by the dashed lines at the top and bottom, respectively. The mass profile is seen to satisfy its upper and lower bounds, and is observed to have a steeper slope during the high-thrust segments of the landing.

IV.B. Convergence Behavior

In this section we present three important observations. First, in Figure 5 we see that the trajectory discussed in the previous section indeed converged. Here, the quantity $\log \max_k \delta\mathbf{x}_i[k]$ (recall the definition in Eq. (50))

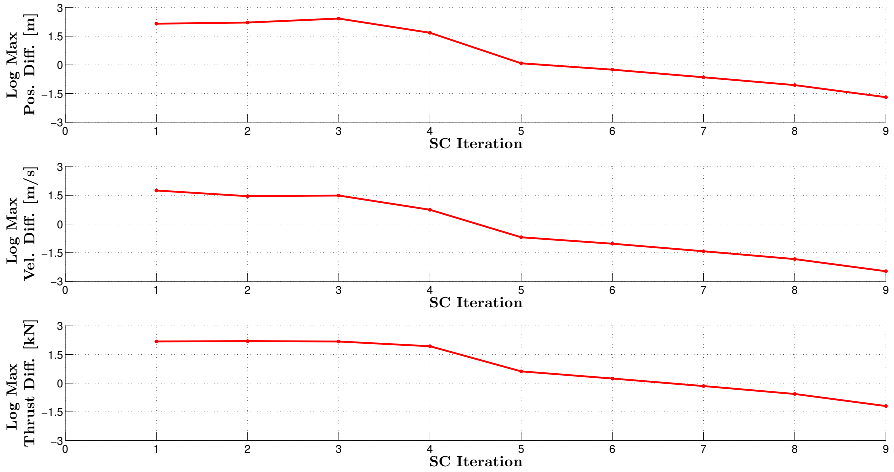

Figure 5. Iteration history of position, velocity, and thrust. *Each plot shows the quantity* $\log \max_k \delta\mathbf{x}_i[k]$ *at each SC iteration for which* $i > 0$*. From top to bottom, the plots show the maximum position, velocity, and thrust differences, respectively, over all* $k \in [0, k_f]$ *for each SC iteration.*

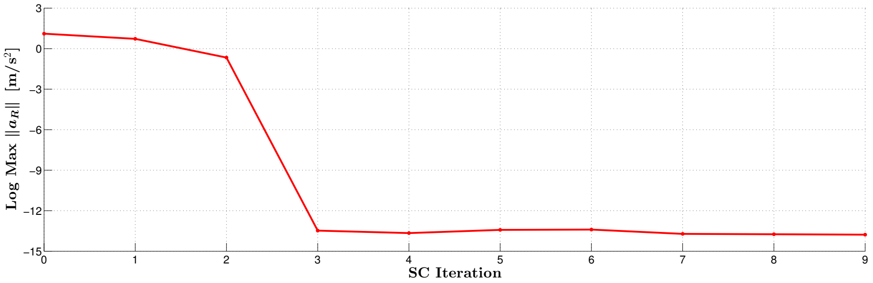

Figure 6. Iteration history of the relaxation term. *The figure shows the maximum value of* $||\mathbf{a}_R||$ *over all* $k \in [0, k_f]$ *for each SC iteration.*

is plotted versus SC iteration number for position, velocity, and thrust. The trends seen in the plots suggest that the trust region components of the cost, namely $\eta_{\Delta t}$ and $||\eta_{\mathbf{T}}||$, become negligible after about the sixth or seventh SC iteration.

Second, in Figure 6 we see a history of the maximum magnitude of $\mathbf{a}_R$ for each SC iteration. We observe that during the first three iterations, the relaxation term is heavily utilized. During these iterations, the time of flight is still relatively dynamic and small compared to its converged value. Thus, the problem is rendered infeasible until the linearization is performed in a more favorable part of the solution space. As soon as the artificial infeasibility subsides, the $\mathbf{a}_R$ term is reduced to a negligible magnitude, just as one would expect given the large value value of $w_{\kappa,\mathbf{a},R}$.

Lastly, to test the robustness of the algorithm's initialization, we initialized the scenario discussed in Section IV.A using four different times of flight: 7.5, 15, 30, and 60 seconds. Each run executed 50 successive convexification iterations. Three of the values specified for $t_{f,s}$ were less than the converged time of flight observed in the previous section, and one was significantly greater. The top plot of Figure 7 shows the log magnitudes of the difference between the time of flight of each iteration and the converged time of flight for the case where $t_{f,s} = 15$ [s]. The bottom plot shows the same information for the propellant mass consumed during each SC iteration. All four cases converged in about six or seven iterations, consistent with our observations from Figure 5. The times of flight of all solutions converged to within a hundredth of a second, while the consumed propellant mass converged to within a hundredth of a kilogram. The fact that all four cases generated virtually identical, physically meaningful, feasible trajectories is evidence that the algorithm is robust to a large set of time of flight initializations.

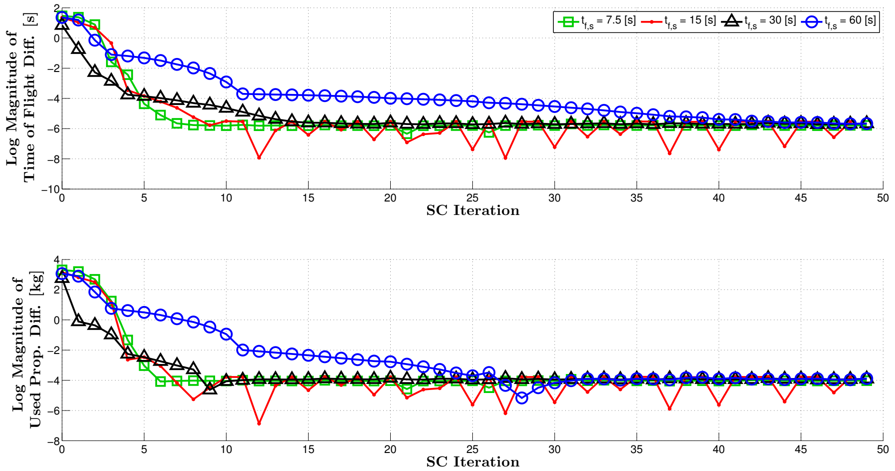

Figure 7. Iteration history of time of flight and propellant mass consumed. *This figure shows the convergence history of four runs, initialized with times of flight of 7.5, 15, 30, 60 [s]. Each run executed 50 SC iterations. The plot at the top compares the convergence of the time of flight, whereas the plot at the bottom compares the convergence of the propellant mass consumed during the landing. All differences are computed relative to the terminal values of the trajectory obtained in the last SC iteration of the* $t_{f,s} = 15$ *[s] case.*

V. Conclusion

In this paper, we presented a successive convexification algorithm designed for the fuel-optimal powered landing problem. The algorithm accounts for aerodynamic drag and various non-convex constraints, including free final time. The algorithm relies on lossless convexification to address the minimum thrust constraint, and on successive convexification to resolve the remaining non-convexities. The latter is comprised of a simplified initialization iteration, followed by a more complex (linearized) SOCP formulation that is solved repeatedly. In this fashion, our algorithm can solve a non-convex optimal control problem, while making very mild a-priori assumptions on the nature of the trajectory. The result is a guidance algorithm that is robust to initial conditions, and that can be solved quickly using state-of-the-art IPM solvers ideal for real-time applications.

Acknowledgments

Support for developing the theoretical basis of the successive convexification method was provided by the National Science Foundation grant CNS-1446520 and the Office of Naval Research grant N00014-14-1-0314.

References

[1] Ploen, S., Açıkmeşe, B., and Wolf, A., "A Comparison of Powered Descent Guidance Laws for Mars Pinpoint Landing," *AIAA GNC Conference and Exhibit, Keystone, CO*, 2006.

[2] Wolf, A. A., Tooley, J., Ploen, S., Ivanov, M., Acikmese, B., and Gromov, K., "Performance trades for Mars pinpoint landing," *Aerospace Conference, 2006 IEEE*, IEEE, 2006, pp. 16–pp.

[3] Carson, J. M., Açıkmeşe, B., Blackmore, L., and Wolf, A. A., "Capabilities of Onboard, Convex Powered-Descent Guidance Algorithms for Pinpoint and Precision Landing," *Aerospace Conference, 2011 IEEE*, IEEE, Big Sky, MT, 03/2011 2011.

[4] Açıkmeşe, B. and Ploen, S. R., "A powered descent guidance algorithm for Mars pinpoint landing," *AIAA GNC Conference and Exhibit, San Francisco*, 2005.

[5] Açıkmeşe, B. and Ploen, S. R., "Convex Programming Approach to Powered Descent Guidance for Mars Landing," *AIAA Journal of Guidance, Control and Dynamics*, Vol. 30, No. 5, 2007, pp. 1353–1366.

[6] Blackmore, L., Açıkmeşe, B., and Scharf, D. P., "Minimum Landing Error Powered Descent Guidance for Mars Landing using Convex Optimization," *AIAA Journal of Guidance, Control and Dynamics*, Vol. 33, No. 4, 2010.

[7] Açıkmeşe, B., Carson, J., and Blackmore, L., "Lossless Convexification of Non-convex Control Bound and Pointing Constraints of the Soft Landing Optimal Control Problem," *IEEE Transactions on Control Systems Technology*, Vol. 21, No. 6, 2013, pp. 2104–2113.

[8] Açıkmeşe, B. and Blackmore, L., "Lossless Convexification of a Class of Optimal Control Problems with Non-Convex Control Constraints," *Automatica*, Vol. 47, No. 2, 2011, pp. 341–347.

[9] Harris, M. and Açıkmeşe, B., "Lossless Convexification of Non-convex Optimal Control Problems for State Constrained Linear Systems," *Automatica*, Vol. 50, No. 9, 2014, pp. 2304–2311.

[10] Harris, M. W. and Açıkmeşe, B., "Maximum Divert for Planetary Landing Using Convex Optimization," *Journal of Optimization Theory and Applications*, 2013, pp. 1–21.

[11] Blackmore, L., Açıkmeşe, B., and Carson, J. M., "Lossless convexfication of control constraints for a class of nonlinear optimal control problems," *System and Control Letters*, Vol. 61, No. 4, 2012, pp. 863–871.

[12] Casoliva, J., *Spacecraft Trajectory Generation by Successive Approximation for Powered Descent and Cyclers*, Ph.D. thesis, University of California, Irvine, 2013.

[13] Liu, X. and Lu, P., "Solving Nonconvex Optimal Control Problems by Convex Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 750–765.

[14] Boyd, S. and Vandenberghe, L., *Convex Optimization*, Cambridge University Press, 2004.

[15] Toh, K., Todd, M., and Tutuncu, R., "SDPT3 — a Matlab Software Package for Semidefinite Programming," *Optimization Methods and Software*, Vol. 11, No. 1, 1999, pp. 545–581.

[16] Dueri, D., Zhang, J., and Açıkmeşe, B., "Automated Custom Code Generation for Embedded, Real-Time Second Order Cone Programming," *19th IFAC World Congress*, 2014, pp. 1605–1612.

[17] Grant, M. and Boyd, S., "CVX: Matlab Software for Disciplined Convex Programming, version 2.1," *http://cvxr.com/cvx*, March 2014.

[18] Domahidi, A., Zgraggen, A. U., Zeilinger, M. N., Morari, M., and Jones, C. N., "Efficient interior point methods for multistage problems arising in receding horizon control," *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, IEEE, 2012, pp. 668–674.

[19] Açıkmeşe, B., Aung, M., Casoliva, J., Mohan, S., Johnson, A., Scharf, D., Masten, D., Scotkin, J., Wolf, A., and Regehr, M. W., "Flight Testing of Trajectories Computed by G-FOLD: Fuel Optimal Large Divert Guidance Algorithm for Planetary Landing," *AAS/AIAA Spaceflight Mechanics Meeting*, 2013.

[20] Scharf, D. P., M. W. Regehr, D. D., Açıkmeşe, B., Vaughan, G. M., Benito, J., Ansari, H., M. Aung, A. J., Masten, D., Nietfeld, S., Casoliva, J., and Mohan, S., "ADAPT Demonstrations of Onboard Large-Divert Guidance with a Reusable Launch Vehicle," *IEEE Aerospace Conference*, 2014.

[21] Carson, J. and Açıkmeşe, B., "A model predictive control technique with guaranteed resolvability and required thruster silent times for small-body proximity operations," *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006.