Model Predictive Control for the Non-Convex Soft Landing Problem

Fausto Vega
Robotics Institute
Carnegie Mellon University
Pittsburgh, USA
fvega@andrew.cmu.edu

***Abstract*—The soft landing of a spacecraft on a planetary surface is an important phase in Entry, Descent, and Landing (EDL) missions. This allows for the safe landing of scientific equipment on planets for exploration purposes. The problem is posed as an optimization problem with state and control constraints. The main challenges in this problem are the non convex control constraints due to the lower bound on the thruster, and the non linear rocket dynamics due to time varying mass. However, the convexification of the non convex constraint transform the problem into a convex optimization problem, and a change of variables convert the dynamics to linear form which allows the problem to be solved in real time. An optimal trajectory and set of control inputs that minimize landing error and fuel consumption are generated from a model predictive controller that reasons about the constraints and external disturbances caused by winds or fuel sloshing.**

***Index Terms*—soft landing, rocket dynamics, optimal control, trajectory optimization.**

I. Introduction

The soft landing problem refers to the guidance of a spacecraft to a target location with an accuracy of less than hundreds of meters and a terminal velocity of zero. The spacecraft has control authority by using thrusters, and the main goal is precise landing along with optimizing fuel. This problem occurs during the powered descent phase of and Entry, Descent, Landing mission as the spacecraft maneuvers to the target location safely. This problem can be formulated as an optimization problem that optimizes landing error and fuel consumption subject to a set of state and control constraints. The landing phase of the mission is essential for deploying scientific tools such as rovers that are calibrated on Earth and deployed on other planets for exploration purposes. Therefore, the soft landing problem can be considered as one of the most important optimal control problems to revolutionize planetary exploration.

There have been multiple examples of soft landing in the past decade. The National Aeronautical Space Agency (NASA) has developed a sky crane that maneuvers to a safe distance from the ground and releases the rover off a tether. The principles of formulating the soft landing problem as a trajectory optimization problem has landed Curiosity (2012) and Perseverance (2021) on Mars. SpaceX has also used this result to design a spaceflight business model that allows the re-usability of rockets and boosters which help minimize the cost of space travel. Within the last 10 years, SpaceX has had success on landing rockets as the Falcon 9 rocket has 111 total landing to this day.

The soft landing of rocket is a complicated problem and has been studied in detail by [1], [2]. The main challenges are the following:

- The control constraints are non-convex.
- The dynamics of the rocket are non-linear.

The control constraints are non convex because the thrusters cannot be throttled off after ignition, and the dynamics are non linear since the mass is time varying. In [1], [2] the authors propose a solution to the soft landing problem by formulating a convex relaxation of the original problem and showing that the solution to the relax problem is the solution to the original problem. A change of variables and the addition of a slack variable transform the problem to a convex finite horizon optimal control problem. The authors propose to minimize a landing error to encourage the model to land at the target location while taking into account the mass reduction dynamics and state/control constraints.

In Sec. II, the problem formulation are described in detail. Although, previous works [1], [2] are able to address the above challenges, the goal of these works are to generate an optimal trajectory that is able to obey the constraints. However, this report includes additional disturbances resulting from wind and fuel sloshing. A model predictive controller is implemented to generate the optimal trajectory subject to external disturbances.

The report is organized as follows. In Section II, the dynamics and problem formulation of the soft landing optimization problem is described. Section III shows the algorithm and plots of landing simulations. Finally, a discussion and analysis on the obtained results is included in Section IV.

II. Model

In this work, the soft landing problem is posed as a convex optimization problem which computes the optimal trajectory and controls to reach the target state. This section will describe the problem formulation in detail. The final solution consists of solving two convex optimization problems, which minimize the landing error and the fuel consumption of the

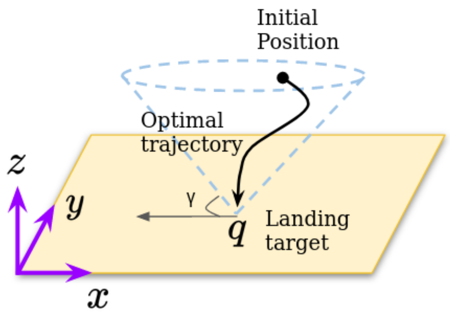

Fig. 1. Constrained soft landing

spacecraft.

The dynamics of the rocket are modeled using Eq. 1. Here, $\mathbf{x} \in \mathbb{R}^6$ is the state vector (position, velocity), $T \in \mathbb{R}^3$ is the thrust vector, m is the mass, $g \in \mathbb{R}^3$ is the constant planet gravity vector, and $\omega \in \mathbb{R}^3$ is the constant angular velocity vector of the planet. Gravity is dependent on the position of the spacecraft; however, since the powered descent phase of the spacecraft starts at a low altitude, a uniform gravity assumption is valid. The aerodynamic forces during the power descent phase are much lower than the forces experienced during the atmospheric entry phase, therefore the forces are treated as disturbances w. The state space matrices A and B are derived in the Appendix [5]. The rocket's mass also depletes over time and it is modeled in Eq. 2. In Eq. 2, $\alpha > 0$ and this constant describes the fuel consumption rate.

$$\dot{\mathbf{x}} = A\mathbf{x} + B\left(\mathbf{g} + \frac{T(t)}{m(t)}\right) + \mathbf{w} \tag{1}$$

$$\dot{m}(t) = -\alpha||T(t)|| \tag{2}$$

The cost function shown in Eq. 3 minimizes the norm of the planar landing error subject to a set of constraints.

$$\mathbf{min}_T||r^{(x,y)}(t_f) - q^{(x,y)}|| \tag{3}$$

where, $r = (x, y, z)$ is the current position of the rocket and $q = (x, y, 0)$ is the target location of the rocket on the planetary surface (both in Earth Centered Earth Fixed coordinates).

The rocket is subject to an upper and lower thrust limit (Eq. 4a) as well as a thrust pointing constraint which limits the thrust to a cone with defined by angle θ and a unit direction vector n which points to the positive z axis (Eq. 4b). The attitude dynamics are not included in this formulation; however, by restricting the directions the thrust points, the attitude can be constrained. Finally, Eq. 5 and 6 describe the initial (k_r) and final state constraints.

$$0 \leq T_{min} \leq ||T|| \leq T_{max} \qquad \hat{n}^T T(t) \geq ||T||\mathbf{cos}(\theta) \tag{4}$$

$$m(0) = k_m \qquad r(0) = k_r \qquad \dot{r}(0) = k_{\dot{r}} \tag{5}$$

$$r^{(z)}(t_f) = 0 \qquad \dot{r}(t_f) = 0 \tag{6}$$

The constraint described in Eq. 4a is *non-convex* due to the lower bound on the norm of the thrust and the dynamics described in Eq. 1 are *non-linear* because of the thrust divided by the time varying mass. Therefore, these challenges are addressed by using lossless convexification [1], [2] and a change of variables described in Sec. II-A. Unlike [1], [2], process noise $\mathbf{w}$ is included to account for disturbances caused by fuel-sloshing, aerodynamic forces, and wind disturbances that affect the velocity of the spacecraft. A model predictive controller is used to plan the entire trajectory and execute the first control action, and resolving the problem with the new state as the initial state constraint. The re planning of the trajectory helps reduce the affect of errors in the model and disturbances that accumulate over time.

Fig. 1 depicts the state constraints of the spacecraft. This constraint requires the spacecraft to remain inside a cone that is defined by a minimum slope angle, and this ensures it stays at a safe distance from the ground until it reaches the target. The velocities of the spacecraft are also constrained to be less than a maximum threshold to avoid super sonic velocities. The state constraints in equation form are shown in Eq. 7-9 where e_3 is the unit vector pointing in the positive z direction and γ is the glideslope angle that describes the cone in Figure 1.

$$||\dot{r}|| \leq V_{max} \tag{7}$$

$$||r^{(x,y)} - r^{x,y}(tf)|| - c^T(r^{(x,y)} - r^{x,y}(tf)) \leq 0 \tag{8}$$

$$c = \frac{e_3}{tan(\gamma_{gs})} \tag{9}$$

The solution to the landing error minimization is then used as a constraint for the fuel optimization problem. The cost function for the fuel optimization problem is shown in Equation 10.

$$\mathbf{min}_T \int_0^{t_f} \alpha||T||dt \tag{10}$$

The fuel optimization problem is subject to the same dynamics and controls constraints described for the minimum landing error problem (Eq. 1,2,4,5,6). However an additional constraint (Eq. 11) is added which limits the spacecraft final position based on the final position of the minimum landing error solution (d^*).

$$||r^{x,y}(t_f) - q|| \leq ||d^* - q|| \tag{11}$$

A. Lossless Convexification and Change of Variables

The cost functions and dynamics describes thus far are non convex and nonlinear which does not guarantee a global optimum, and take more time to compute a solution. The key result in [1] is the relaxation of the non convex constraints to make them convex and a change of variables to obtain linear dynamics. A slack variable Γ replaces $||T||$ and introduces a new constraint $||T|| < \Gamma$. In [1], it is proven that the optimal solution to the relaxed problem is also the optimal solution for the *non-convex* problem. The following change of variables in Eq. 12 was also performed to convert to linear dynamics.

$$\sigma = \frac{\Gamma}{m} \quad u = \frac{T}{m} \quad z = \mathbf{ln}(m) \tag{12}$$

The change of variables makes the problems linear convex optimization problems. The fuel optimization cost function becomes Eq. 13.

$$\mathbf{min}_T \int_0^{t_f} \sigma dt \tag{13}$$

Additional details on the substitution of these variables and approximations are included in References [1] and [2]. The mass dynamics after these transformations are shown in Eq. 14 and were included into the A matrix to incorporate mass into the state.

$$\dot{z} = -\alpha\sigma \tag{14}$$

B. Discretization of Problem and Solution Algorithm

The continuous dynamics derived in the Appendix are discretized via the exponential map [6], and the derivation is shown in Eq. 15-19. The dynamics are in the form of an affine ODE (15) , and can be written in matrix form (16)

$$\dot{x} = Ax + Bu + d \tag{15}$$

$$\begin{bmatrix} \dot{x} \\ \dot{u} \\ \dot{d} \end{bmatrix} = \begin{bmatrix} A & B & I \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ u \\ d \end{bmatrix} \tag{16}$$

The system is then discretized with a time step of dt in the same way as a homogeneous linear ODE.

$$\begin{bmatrix} x_{k+1} \\ u_{k+1} \\ d_{k+1} \end{bmatrix} = exp(\begin{bmatrix} A & B & I \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} dt) \begin{bmatrix} x_k \\ u_k \\ d_k \end{bmatrix} \tag{17}$$

The state transition matrices in Eq. 19, A_d, B_d, and D_d come from the matrix exponential and they describe the discrete dynamics of the system (Eq. 19).

$$\begin{bmatrix} A_d & B_d & D_d \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} = \mathbf{exp} \left(\begin{bmatrix} A & B & I \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} dt \right) \tag{18}$$

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ u_{k+1} \\ d_{k+1} \end{bmatrix} = \begin{bmatrix} A_d & B_d & D_d \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ u_k \\ d_k \end{bmatrix} \tag{19}$$

A zero-order hold is applied to the controls and the variable d is a constant. Once the dynamics are discretized, they are used as a constraint for the convex optimization problem to develop a dynamically feasible trajectory. The state $\mathbf{x} \in \mathbb{R}^7$ consists of the position, velocity, and a variable z (representing the mass), and the control $u \in \mathbb{R}^4$ consists of the thrust in 3 directions and the term σ which arises in the mass dynamics (Eq. 14).

III. Results

These problems were then implemented in the Julia programming language [3], using the Convex.jl optimization library [7] along with the MOSEK solver [4]. A Model Predictive Controller (MPC) was applied to the problem to obtain the optimal trajectory at every time step subject to noise $\mathbf{w}$ in the dynamics. The noise was from a random Gaussian distribution with a mean of 0.1 m/s, and applied to the velocities to model disturbances resulting from winds and fluid sloshing. In all the experiments, the following values were used for model parameters: $dt = 0.5$ s, $m_0 = 2000$ kg, $m_f = 300$ kg, $T_{min} = 3600$ N, $T_{max} = 19200$ N, and $\alpha = 5{\times}10^{-4}$ s/m. Martian gravity $g = (0, 0, -3.71)\ m/s^2$ and the Mars rotation vector $\omega = (0, 6.62 \times 10^{-5}, 2.53 \times 10^{-5})$ rad/s were also used. The initial state of the spacecraft for the simulation are the following: $r_0 = (450, -330, 2400)$ m and $\dot{r}_0 = (-40, 10, -10)$ m/s. For the state constraints, a maximum velocity of 90 m/s was used as well as a glide slope angle of 30 degrees. For the thrust pointing constraint, an angle of 70 degrees from the normal direction was implemented. The results of the landing error optimization are shown in Fig. 2-5. Fig. 2 shows the optimal trajectory with the start position being the orange marker while Fig. 3 depicts the spacecraft states over time. Fig. 4 shows the mass depletion of the spacecraft, and Fig. 5 shows $||T||$ over time as well as the upper and lower thrust limits as horizontal lines. The system converged within the set window 48 seconds and used 234 kg of mass. However, the lower bound thrust constraint was violated at some time-steps and this effect can be mitigated by adding the additional cost function that minimizes fuel. The results of the fuel optimization problem are shown in Fig. 6-9. The same parameters from the minimum landing error were used except for the time step and number of knot points. A time step (dt) of 1 s was chosen because position controller on rockets runs at only several Hz, and the number of knot points was changed to 48. The fuel optimization trajectory utilized 195 kg of fuel which is approximately saves 40 kg of fuel compared to the landing error trajectory. The optimal control for the fuel optimization problem also follows the upper and lower constraints tighter than the previous solution even though the upper and lower limits are approximated with second order cones (more details on these approximations are available in [1]).

IV. Discussion

The results in this work confirm the results of [2], as the Fig. 5 and 9 show that a bang bang solution is the optimal solution for this problem. However, the time step and amount of knot points are arbitrarily chosen in this work, and the next step is to include the time step into the optimization problem and formulate a minimum time problem. Overall the plots match the physical intuition as the mass depleted at a faster rate once the thrusters were fired aggressively as the rate is proportional to the norm of the thrust.

Additional research directions include exploring different dynamic models and cost functions to analyze the effect they

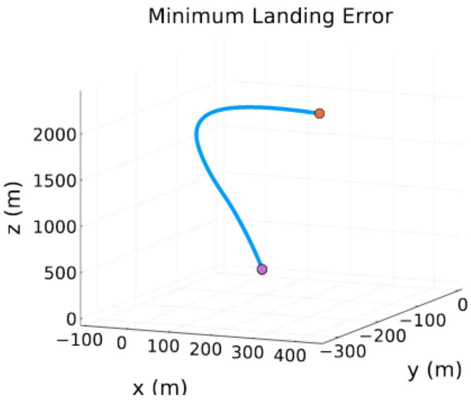

Fig. 2. Minimum Landing Error Trajectory

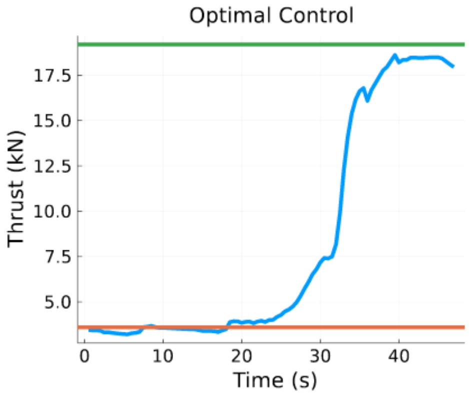

Fig. 5. Spacecraft Thrust for Landing Error Problem

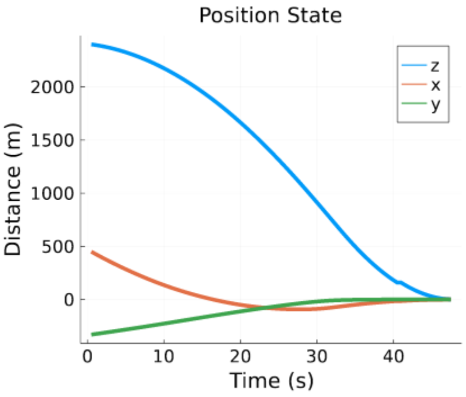

Fig. 3. Spacecraft Position State for Landing Error Problem

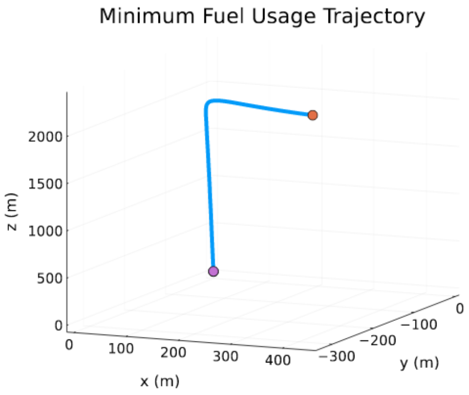

Fig. 6. Fuel Optimization Trajectory

have on the solution. By making the soft landing problem convex, this algorithm can be ran on-board as current technology is able to solve a convex optimization problem in real-time. Implementing the non linear problem and comparing the solution of the relaxed convex problem is also another research direction that can potentially be explored. This is because the relaxed convex problem does not reason about attitude dynamics and uses a constraint to restrict the attitude; however, in the nonlinear dynamics problem, both the position and attitude dynamics would be a part of the problem. Solvers like ALTRO [8] can be used to formulate this problem and will be explored in future work.

Conclusion

In this work, a method for soft landing of single-thruster spacecraft on the surface of planets such as Mars was explored using a point mass dynamics model and constrained optimization of a predefined cost function. A method of loseless

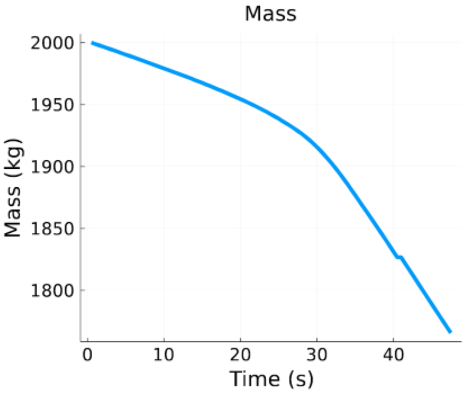

Fig. 4. Spacecraft Mass Depletion for Landing Error Problem

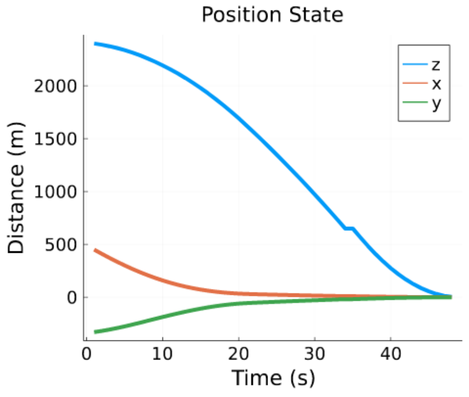

Fig. 7. Spacecraft Position State for Fuel Optimization Problem

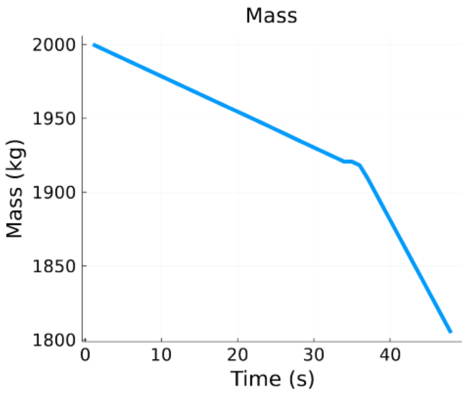

Fig. 8. Spacecraft Mass Depletion for Fuel Optimization Problem

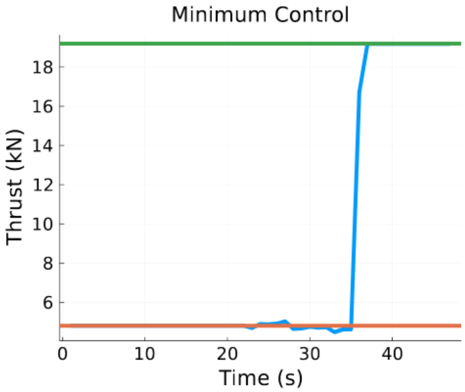

Fig. 9. Spacecraft Thrust for Fuel Optimization Problem

convexification and change of variables proposed by [1] [2] was implemented and the results show that the spacecraft is able to land at the target location softly and satisfy constraints regardless of noise in the dynamics. This work enables precise pin point landing on scientifically interesting locations for planetary exploration.

Appendix

A. Cartesian Dynamics Derivation

In this work, the dynamics were represented in the Cartesian space. The planet body frame is denoted as as B and an inertial frame as N, and both frames are aligned in the z axes. The planet angular velocity is assumed to be constant about the positive z axis which results in the following angular velocity vector,

$$^N\omega^B = \begin{bmatrix}0 & 0 & \omega_z\end{bmatrix} \tag{20}$$

The notation $^N\omega^B$ denotes the angular velocity of the body B in the inertial frame N, and it will be denoted as ω for simplicity. A body frame Q is placed on the spacecraft and a position vector in the planet body frame from the center of the planet to the spacecraft is defined as $^Br^Q$. The velocity of the spacecraft in the inertial frame is derived by differentiating the position vector using the kinematic transport theorem.

$$^Nv^Q = \frac{^Nd(^Br^Q)}{dt} \tag{21}$$

$$^Nv^Q = \frac{^Bd(^Br^Q)}{dt} + \omega \times {}^Br^Q \tag{22}$$

$$^Nv^Q = {}^Bv^Q + \omega \times {}^Br^Q \tag{23}$$

The acceleration of the spacecraft with respect to the inertial frame N is found by differentiating the velocity expression in Eq. 23. The kinematic transport theorem was also used when calculating the derivative.

$$^Na^Q = \frac{^Nd(^Nv^Q)}{dt} \tag{24}$$

$$^Na^Q = \frac{^Bd(^Nv^Q)}{dt} + \omega \times {}^Nv^Q \tag{25}$$

The velocity expression in Eq. 23 is substituted in Eq. 25 to yield Eq. 26.

$$^Na^Q = \frac{^Bd(^Bv^Q + \omega \times {}^Br^Q)}{dt} + \omega \times {}^Nv^Q \tag{26}$$

The time derivatives are then separated and computed independently. The cross product term derivative is calculated using the product rule.

$$^Na^Q = \frac{^Bd(^Bv^Q)}{dt} + \frac{^Bd(\omega \times {}^Br^Q)}{dt} + \omega \times {}^Nv^Q \tag{27}$$

$$^Na^Q = {}^Ba^Q + \omega \times {}^Bv^Q + \omega \times {}^Nv^Q \tag{28}$$

The velocity expression is once again substituted into Eq. 28 to yield the equations representing the acceleration of the spacecraft in the inertial frame.

$$^Na^Q = {}^Ba^Q + \omega \times {}^Bv^Q + \omega \times ({}^Bv^Q + \omega \times {}^Br^Q) \tag{29}$$

$$^Na^Q = {}^Ba^Q + 2(\omega \times {}^Bv^Q) + \omega \times (\omega \times {}^Br^Q) \tag{30}$$

Eq. 30 is rearranged to solve for the acceleration of the spacecraft in the planet body frame.

$$^Ba^Q = {}^Na^Q - 2(\omega \times {}^Bv^Q) - \omega \times (\omega \times {}^Br^Q) \tag{31}$$

The state of the spacecraft is now parameterized with the position vector $(^Br^Q)$ and velocity vector $(^Bv^Q)$ expressed in the planet body frame. The cross product matrix is used to describe the dynamics in state space form ($\dot{x} = Ax$). The spacecraft is assumed to have a single thruster (T) which only affects the accelerations. The full dynamics model is shown below.

$$\begin{bmatrix}v\\a\end{bmatrix} = \begin{bmatrix}0 & I\\ -S(\omega)^2 & -2S(\omega)\end{bmatrix}\begin{bmatrix}r\\v\end{bmatrix} + \begin{bmatrix}0\\I\end{bmatrix}(g + \frac{T(t)}{m(t)}) \tag{32}$$

$$S = \begin{bmatrix}0 & -\omega_3 & \omega_2\\ \omega_3 & 0 & -\omega_1\\ -\omega_2 & \omega_1 & 0\end{bmatrix} \tag{33}$$

This dynamics model does not include aerodynamics forces, however a constraint is set on the velocities to reduce the high effect of aerodynamic forces.

References

[1] B. Açıkmeşe, J. M. Carson and L. Blackmore, “Lossless Convexification of Nonconvex Control Bound and Pointing Constraints of the Soft Landing Optimal Control Problem,” in IEEE Transactions on Control Systems Technology, vol. 21, no. 6, pp. 2104-2113, Nov. 2013.

[2] B. Açıkmeşe. and Ploen, S.R., 2007. “Convex programming approach to powered descent guidance for mars landing”. Journal of Guidance, Control, and Dynamics, 30(5), pp.1353-1366.

[3] Jeff Bezanson, Stefan Karpinski, Viral B. Shah and Alan Edelman, “Julia: A Fast Dynamic Language for Technical Computing”, arXiv 1209.5145, 2012

[4] A. Mosek. “The MOSEK optimization software at https://www.mosek.com”. 2010.

[5] Tracy, K. and Manchester, Z., CPEG: A Convex Predictor-corrector Entry Guidance Algorithm.

[6] Tracy, K. and Manchester, Z., 2020, August. Model-predictive attitude control for flexible spacecraft during thruster firings. In AAS/AIAA Amer. Inst. Aeronaut., Astrodynamics Specialist Conf..

[7] Udell, M., Mohan, K., Zeng, D., Hong, J., Diamond, S. and Boyd, S., 2014, November. Convex optimization in Julia. In 2014 First Workshop for High Performance Technical Computing in Dynamic Languages (pp. 18-28). IEEE.

[8] Howell, T.A., Jackson, B.E. and Manchester, Z., 2019, November. Altro: A fast solver for constrained trajectory optimization. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 7674-7679). IEEE.