SUCCESSIVE CONVEXIFICATION FOR ON-BOARD SCHEDULING OF SPACECRAFT RENDEZVOUS MISSIONS

Thomas Claudet[1, 2], **Jérémie Labroquère**[3], **Damiana Losa**[2, 4], **Francesco Sanfedino**[1], **Daniel Alazard**[1]

[1] *ISAE-SUPAERO, Toulouse, 31400, France, {thomas.claudet, francesco.sanfedino, daniel.alazard}@isae-supaero.fr*
[2] *IRT Saint-Exupéry, 3 Rue Tarfaya, 31400, Toulouse, France, {thomas.claudet, damiana.losa}@irt-saintexupery.com*
[3] *Thales Alenia Space, 26 Avenue Jean François Champollion, 31100, Toulouse, France, jeremie.labroquere@thalesaleniaspace.com*
[4] *Thales Alenia Space, 5 All. des Gabians, 06150 Cannes, France, damiana.losa@thalesaleniaspace.com*

ABSTRACT

Generating fast and optimal trajectories for rendezvous missions is a challenging problem with a wide range of applications from commercial or scientific on-orbit servicing and manufacturing, to agile military operations. In particular, in the case of interplanetary missions or for non-cooperative or aggressive targets, human-in-the-loop operations seem extremely limited, while a greater deal of autonomy will become increasingly mandatory. In this paper, the on-board optimization of rendezvous trajectories has been studied, focusing on state-triggered constraints implemented within a successive convexification scheme. Convergence properties are discussed while numerical validations demonstrate the performance of the proposed approach.

1 INTRODUCTION

Rendezvous (RDV) missions have been a topic of interest in the field of aerospace engineering for decades, with applications ranging from docking and servicing of spacecraft to asteroid deflection [1]. In recent years, there has been an increasing interest in developing efficient and reliable methods for solving the optimal control problems associated with such missions.

Solving non-convex optimization problems can be quite difficult in their general form. This is particularly true for guidance design in hyper-constrained proximity operations, which involve complex coupled dynamics. Finding a propellant or time-optimal solution for such scenarios can require significant computational effort.

Successive Convexification (SCvx) [2], a type of Sequential Convex Programming algorithm, proposes an alternative approach to solving non-linear and non-convex optimization problems compared to traditional optimization strategies. The aim of these algorithms is to iteratively solve a sequence of convex optimization sub-problems that are part of the original non-convex optimization problem. This is done in order to find an approximate near-optimal solution to the original problem[3, 4].

Convex optimization problems are guaranteed to converge towards a unique global solution, and there are polynomial complexity solvers available to efficiently solve them [5]. This means that, provided that the transcription of the original non-convex problem into the convex sub-problems is not too computationally expensive, it is possible to develop very efficient algorithms to approximate solutions of non-convex problems.

However, it is important to note that the fundamental limitations of these methods are similar to those of traditional direct methods [3]. While SCvx methods will rapidly converge to locally optimal solutions, they may struggle to find global solutions of the original non-convex problem [3]. As a result, SCvx should be primarily used when near-optimal solutions are considered an acceptable trade-off or in hyper-constrained scenarios where finding a near-optimal solution that satisfies all the constraints with low computational effort is crucial. For example, SCvx techniques can be particularly useful in autonomous RDV systems and proximity operations to design a high-frequency refresh rate guidance law. Additionally, these techniques can also be applied to more traditional trajectory design to quickly access feasible near-optimal trajectories in non-linear dynamics.

Successive convexification has been applied successfully to a range of control problems, including rocket trajectory optimization [6] and formation flying. It has been found to be particularly effective when dealing with complex non-linear systems, where traditional optimization methods may struggle.

This paper focuses on the application of successive convexification to the problem of rendezvous missions, with the aim of developing a robust and efficient solution approach. Specifically, the problem of bringing two spacecraft into close proximity while minimizing fuel consumption as well as ensuring constraints on trajectory and control action is considered.

Mainly inherited from [4], the contributions of this work are twofold. First, relaxed version of the quaternion end-point constraint was provided for increased feasibility. Then, a linear relaxation of the keep-out zone to enter from the top of the approach cone was provided.

The structure of this paper is as follows: Section 2 formulates the problem and Section 3 focuses on the implemented Successive Convexification algorithm. Section 4 presents the achieved results and Section 5 finally draws conclusions on this work.

2 PROBLEM FORMULATION

2.1 Reference Frame Definition

The reference frames used in this work are defined with the following notations:

- $\mathcal{R}_I$, the ECI (Earth-Centered Inertial) frame.
- $\mathcal{R}_{LVLH}$, the Local-Vertical Local-Horizontal frame (LVLH), centered at the target's position and defined as in [1].
  $\overrightarrow{X}$, also called V-bar, with the associated vector in the direction of the orbital velocity, but not necessarily aligned with it. It is computed as orthogonal to the other two.
  $\overrightarrow{Y}$, also called H-bar, with the associated vector in the opposite direction of the angular momentum vector of the orbit.
  $\overrightarrow{Z}$, also called R-bar, with the associated vector in the radial direction from the spacecraft center of mass to the center of the Earth.

- $\mathcal{R}_C$, the body reference frame of the chaser.
- $\mathcal{R}_T$, the body reference frame of the target.

2.2 Optimization Variables

The number of simulation intervals is denoted by N and the reader is invited to refer to Fig. 1 for visual definitions.

2.2.1 Physical Variables

- $\left[\overrightarrow{O_T O_C}\right]^i_{\mathcal{R}_{LVLH}} \triangleq \boldsymbol{r_{ct}^{(i)}} \quad \forall i \in [0, \dots, N]$ the position vector of the chaser in the LVLH frame at time step i.
- $\left[\frac{d}{dt}\overrightarrow{O_T O_C}\Big|_{\mathcal{R}_I}\right]^i_{\mathcal{R}_{LVLH}} \triangleq \boldsymbol{\dot{r}_{ct}^{(i)}} \quad \forall i \in [0, \dots, N]$ the velocity of the chaser with respect to the target, computed in the inertial frame, and expressed in the LVLH frame at time step i.
- $[\boldsymbol{F_C}]^i_{\mathcal{R}_{LVLH}} \triangleq \boldsymbol{F_c^{(i)}} \quad \forall i \in [0, \dots, N-1]$ the control force acting on the chaser at O_C and expressed in the LVLH frame at time step i.
- $\left[\boldsymbol{q}_{\mathcal{R}_C/\mathcal{R}_I}\right]^i_{\mathcal{R}_I} \triangleq \boldsymbol{q_c^{(i)}} \quad \forall i \in [0, \dots, N]$ the quaternion representing the orientation of the chaser with respect to the inertial frame.
- $\left[\boldsymbol{\omega}_{\mathcal{R}_C/\mathcal{R}_I}\right]^i_{\mathcal{R}_I} \triangleq \boldsymbol{\omega_c^{(i)}} \quad \forall i \in [0, \dots, N]$ the angular velocity of the chaser with respect to the inertial frame, expressed in the inertial frame.
- $[\boldsymbol{T_C}]^i_{\mathcal{R}_I} \triangleq \boldsymbol{T_c^{(i)}} \quad \forall i \in [0, \dots, N-1]$ the control torque acting on the chaser at point O_C and expressed in the inertial frame at time step i.

2.2.2 Slack Variables

Slack variables are often used in optimization. Here, they serve as substitutes for the actuation and state norms to fit the linear cost function. Indeed, when minimizing the energy, which is the 2-norm of the actuation vector, the square root of the squared components will appear, which is by essence not linear with respect to the components. On the other hand, adding slack variables as new optimization parameters constrained to represent norms and summing them over time in the cost function will preserve linearity.

- $\eta_F^{(i)} \quad \forall i \in [0, \dots, N-1]$ the slack variable at time step i associated to the force at time step i.
- $\eta_T^{(i)} \quad \forall i \in [0, \dots, N-1]$ the slack variable at time step i associated to the torque at time step i.
- $\eta_\omega^{(i)} \quad \forall i \in [0, \dots, N-1]$ the slack variable at time step i associated to the angular velocity at time step i.
- $\epsilon_q^{(N)}$ the slack variable associated to the final quaternion soft constraint.

The optimization vector $\boldsymbol{x}$ is then defined as:

$$\begin{aligned}\boldsymbol{x} = \Big[& \boldsymbol{r_{ct}^{(0)}}^T \quad \boldsymbol{\dot{r}_{ct}^{(0)}}^T \quad \dots \quad \boldsymbol{r_{ct}^{(N)}}^T \quad \boldsymbol{\dot{r}_{ct}^{(N)}}^T \quad \boldsymbol{F_c^{(0)}}^T \quad \dots \quad \boldsymbol{F_c^{(N-1)}}^T \\ & \boldsymbol{q_c^{(0)}}^T \quad \boldsymbol{\omega_c^{(0)}}^T \quad \dots \quad \boldsymbol{q_c^{(N)}}^T \quad \boldsymbol{\omega_c^{(N)}}^T \quad \boldsymbol{T_c^{(0)}}^T \quad \dots \quad \boldsymbol{T_c^{(N-1)}}^T \\ & \eta_F^{(0)} \quad \dots \quad \eta_F^{(N-1)} \quad \eta_T^{(0)} \quad \dots \quad \eta_T^{(N-1)} \quad \eta_\omega^{(0)} \quad \dots \quad \eta_\omega^{(N)} \quad \epsilon_q^{(N)} \Big]^T .\end{aligned} \tag{1}$$

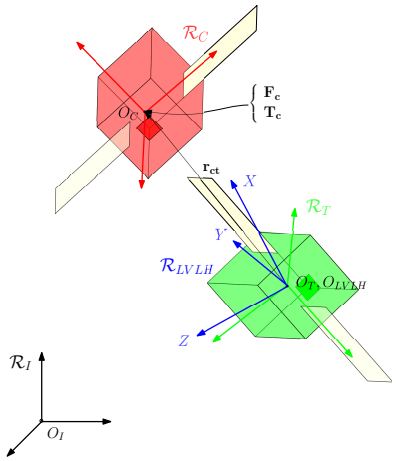


Figure 1: Reference frame definition

2.3 Dynamical and Boundary Constraints

In this subsection, the linear constraints which are part of the problem formulation are defined.

2.3.1 Translation

- The dynamical equations for the translation are those of Hill [7] projected in the LVLH frame as defined by [1]:

$$\begin{bmatrix} \dot{\boldsymbol{r}}_{\boldsymbol{ct}} \\ \ddot{\boldsymbol{r}}_{\boldsymbol{ct}} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{0_{3\times3}} & \boldsymbol{I_{3\times3}} \\ 0 & 0 & 0 & 0 & 0 & 2\omega_t \\ 0 & -\omega_t^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3\omega_t^2 & -2\omega_t & 0 & 0 \end{bmatrix}}_{{}^t\boldsymbol{A_c}} \begin{bmatrix} \boldsymbol{r}_{\boldsymbol{ct}} \\ \dot{\boldsymbol{r}}_{\boldsymbol{ct}} \end{bmatrix} + \underbrace{\begin{bmatrix} \boldsymbol{0_{3\times3}} \\ \frac{1}{m_C}\boldsymbol{I_{3\times3}} \end{bmatrix}}_{{}^t\boldsymbol{B_c}} \boldsymbol{F_c}, \tag{2}$$

with ${}^t\boldsymbol{A_c}$ the continuous state matrix for the translation, ${}^t\boldsymbol{B_c}$ the associated input matrix, ω_t the orbit pulsation of the target, m_c the mass of the chaser.

To fit the numerical convex framework, one must discretize them. To do so, the Taylor series development of the exponential at the first order was used as in [8], denoting ΔT as the constant discretization time. Increasing precision can be achieved by further pushing the development order, but at the cost of a longer computation time. For ΔT close to 0, it is obtained:

$$ {}^t\boldsymbol{A_d} = e^{{}^t\boldsymbol{A_c}\Delta T} = \boldsymbol{I} + {}^t\boldsymbol{A_c}\Delta T, \tag{3} $$

$$ {}^t\boldsymbol{B_d} = {}^t\boldsymbol{A_c}^{-1}({}^t\boldsymbol{A_d} - \boldsymbol{I}){}^t\boldsymbol{B_c} = {}^t\boldsymbol{B_c}\Delta T. \tag{4} $$

The discrete dynamics are then:

$$ \forall k \in [0, \dots, N-1],\ \begin{bmatrix} \boldsymbol{r_{ct}^{(k+1)}} \\ \boldsymbol{\dot{r}_{ct}^{(k+1)}} \end{bmatrix} = {}^t\boldsymbol{A_d} \begin{bmatrix} \boldsymbol{r_{ct}^{(k)}} \\ \boldsymbol{\dot{r}_{ct}^{(k)}} \end{bmatrix} + {}^t\boldsymbol{B_d}\boldsymbol{F_c^{(k)}}, \tag{5} $$

or in matrix form:

$$ \begin{bmatrix} {}^t\boldsymbol{A_d} & -\boldsymbol{I_{6\times6}} & \boldsymbol{0_{6\times6}} & \dots & \boldsymbol{0_{6\times6}} \\ \boldsymbol{0_{6\times6}} & {}^t\boldsymbol{A_d} & -\boldsymbol{I_6} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \boldsymbol{0_{6\times6}} \\ \boldsymbol{0_{6\times6}} & \dots & \boldsymbol{0_{6\times6}} & {}^t\boldsymbol{A_d} & -\boldsymbol{I_{6\times6}} \end{bmatrix} \begin{bmatrix} {}^t\boldsymbol{B_d} & \boldsymbol{0_{6\times6}} & \dots & \boldsymbol{0_{6\times6}} & \boldsymbol{0_{6\times6}} & \dots \\ \boldsymbol{0_{6\times6}} & {}^t\boldsymbol{B_d} & \ddots & \vdots & \boldsymbol{0_{6\times6}} & \dots \\ \vdots & \ddots & \ddots & \boldsymbol{0_{6\times6}} & \vdots & \dots \\ \boldsymbol{0_{6\times6}} & \dots & \boldsymbol{0_{6\times6}} & {}^t\boldsymbol{B_d} & \boldsymbol{0_{6\times6}} & \dots \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{6} $$

- Initial relative positions and velocities are inputs of the problem. They are identified with a hat in the following.

$$ \boldsymbol{r_{ct}^{(0)}} = \boldsymbol{\hat{r}_{ct}^{(0)}}, \tag{7} $$

$$ \boldsymbol{\dot{r}_{ct}^{(0)}} = \boldsymbol{\hat{\dot{r}}_{ct}^{(0)}}. \tag{8} $$

- The final position in the relative frame is the origin of the LVLH frame, thus of zero-coordinates. The final relative velocity is zero as well.

$$ \boldsymbol{r_{ct}^{(N)}} = \boldsymbol{\hat{r}_{ct}^{(N)}} = \boldsymbol{0_{3\times1}}, \tag{9} $$

$$ \boldsymbol{\dot{r}_{ct}^{(N)}} = \boldsymbol{\hat{\dot{r}}_{ct}^{(N)}} = \boldsymbol{0_{3\times1}}. \tag{10} $$

2.3.2 Rotation

- As for translation, the attitude dynamics are those of Newton-Euler, in their quaternionic version:

$$ \dot{\boldsymbol{q}}_{\boldsymbol{c}} = \frac{1}{2}\boldsymbol{\Omega_c}\boldsymbol{q_c}, \tag{11} $$

$$ \boldsymbol{I_c}\,\dot{\boldsymbol{\omega}}_{\boldsymbol{c}} + \boldsymbol{\omega_c} \times \boldsymbol{I_c}\,\boldsymbol{\omega_c} = \boldsymbol{T_c}, \tag{12} $$

with

$$ \boldsymbol{\Omega_c} = \begin{bmatrix} 0 & -\omega_c^z & \omega_c^y \\ \omega_c^z & 0 & -\omega_c^x \\ -\omega_c^y & \omega_c^x & 0 \end{bmatrix}. \tag{13} $$

These equations are time-varying and nonlinear. Indeed $\boldsymbol{\omega_c}$ is a function of time and the product of it by itself or $\boldsymbol{q_c}$ creates the nonlineary in the optimization variables. The equations were then

linearized at every point along the trajectory and as many state and command matrices were obtained. The linearization about a point $\bar{\boldsymbol{x}}$ reads:

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}(\bar{\boldsymbol{x}}) + \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\bar{\boldsymbol{x}}} (\boldsymbol{x} - \bar{\boldsymbol{x}}) + o\left((\boldsymbol{x} - \bar{\boldsymbol{x}})^2\right), \tag{14}$$

which, in the multidimensional multivariate case, still at order 1, and in the neighborhood of the reference, gives:

$$\begin{bmatrix} \dot{\boldsymbol{q}}_{\boldsymbol{c}} \\ \dot{\boldsymbol{\omega}}_{\boldsymbol{c}} \end{bmatrix} = \begin{bmatrix} f_{q_c}(\boldsymbol{q}_{\boldsymbol{c}}, \boldsymbol{\omega}_{\boldsymbol{c}}, \boldsymbol{T}_{\boldsymbol{c}}) \\ f_{\omega_c}(\boldsymbol{q}_{\boldsymbol{c}}, \boldsymbol{\omega}_{\boldsymbol{c}}, \boldsymbol{T}_{\boldsymbol{c}}) \end{bmatrix} = \begin{bmatrix} f_{q_c}(\bar{\boldsymbol{q}}_{\boldsymbol{c}}, \bar{\boldsymbol{\omega}}_{\boldsymbol{c}}, \bar{\boldsymbol{T}}_{\boldsymbol{c}}) \\ f_{\omega_c}(\bar{\boldsymbol{q}}_{\boldsymbol{c}}, \bar{\boldsymbol{\omega}}_{\boldsymbol{c}}, \bar{\boldsymbol{T}}_{\boldsymbol{c}}) \end{bmatrix} + \left.\begin{bmatrix} \frac{\partial f_{q_c}}{\partial \boldsymbol{q}_{\boldsymbol{c}}} & \frac{\partial f_{q_c}}{\partial \boldsymbol{\omega}_{\boldsymbol{c}}} & \frac{\partial f_{q_c}}{\partial \boldsymbol{T}_{\boldsymbol{c}}} \\ \frac{\partial f_{\omega_c}}{\partial \boldsymbol{q}_{\boldsymbol{c}}} & \frac{\partial f_{\omega_c}}{\partial \boldsymbol{\omega}_{\boldsymbol{c}}} & \frac{\partial f_{\omega_c}}{\partial \boldsymbol{T}_{\boldsymbol{c}}} \end{bmatrix}\right|_{\bar{\boldsymbol{q}}_{\boldsymbol{c}}, \bar{\boldsymbol{\omega}}_{\boldsymbol{c}}, \bar{\boldsymbol{T}}_{\boldsymbol{c}}} \begin{bmatrix} \boldsymbol{q}_{\boldsymbol{c}} - \bar{\boldsymbol{q}}_{\boldsymbol{c}} \\ \boldsymbol{\omega}_{\boldsymbol{c}} - \bar{\boldsymbol{\omega}}_{\boldsymbol{c}} \\ \boldsymbol{T}_{\boldsymbol{c}} - \bar{\boldsymbol{T}}_{\boldsymbol{c}} \end{bmatrix}, \tag{15}$$

which can also be written as:

$$\begin{aligned} \begin{bmatrix} \dot{\boldsymbol{q}}_{\boldsymbol{c}} \\ \dot{\boldsymbol{\omega}}_{\boldsymbol{c}} \end{bmatrix} = \begin{bmatrix} f_{q_c}(\boldsymbol{q}_{\boldsymbol{c}}, \boldsymbol{\omega}_{\boldsymbol{c}}, \boldsymbol{T}_{\boldsymbol{c}}) \\ f_{\omega_c}(\boldsymbol{q}_{\boldsymbol{c}}, \boldsymbol{\omega}_{\boldsymbol{c}}, \boldsymbol{T}_{\boldsymbol{c}}) \end{bmatrix} &= \left.\begin{bmatrix} \frac{\partial f_{q_c}}{\partial \boldsymbol{q}_{\boldsymbol{c}}} & \frac{\partial f_{q_c}}{\partial \boldsymbol{\omega}_{\boldsymbol{c}}} \\ \frac{\partial f_{\omega_c}}{\partial \boldsymbol{q}_{\boldsymbol{c}}} & \frac{\partial f_{\omega_c}}{\partial \boldsymbol{\omega}_{\boldsymbol{c}}} \end{bmatrix}\right|_{\bar{\boldsymbol{q}}_{\boldsymbol{c}}, \bar{\boldsymbol{\omega}}_{\boldsymbol{c}}, \bar{\boldsymbol{T}}_{\boldsymbol{c}}} \begin{bmatrix} \boldsymbol{q}_{\boldsymbol{c}} \\ \boldsymbol{\omega}_{\boldsymbol{c}} \end{bmatrix} + \left.\begin{bmatrix} \frac{\partial f_{q_c}}{\partial \boldsymbol{T}_{\boldsymbol{c}}} \\ \frac{\partial f_{\omega_c}}{\partial \boldsymbol{T}_{\boldsymbol{c}}} \end{bmatrix}\right|_{\bar{\boldsymbol{q}}_{\boldsymbol{c}}, \bar{\boldsymbol{\omega}}_{\boldsymbol{c}}, \bar{\boldsymbol{T}}_{\boldsymbol{c}}} \boldsymbol{T}_{\boldsymbol{c}} \\ &+ \begin{bmatrix} f_{q_c}(\bar{\boldsymbol{q}}_{\boldsymbol{c}}, \bar{\boldsymbol{\omega}}_{\boldsymbol{c}}, \bar{\boldsymbol{T}}_{\boldsymbol{c}}) \\ f_{\omega_c}(\bar{\boldsymbol{q}}_{\boldsymbol{c}}, \bar{\boldsymbol{\omega}}_{\boldsymbol{c}}, \bar{\boldsymbol{T}}_{\boldsymbol{c}}) \end{bmatrix} - \left.\begin{bmatrix} \frac{\partial f_{q_c}}{\partial \boldsymbol{q}_{\boldsymbol{c}}} & \frac{\partial f_{q_c}}{\partial \boldsymbol{\omega}_{\boldsymbol{c}}} & \frac{\partial f_{q_c}}{\partial \boldsymbol{T}_{\boldsymbol{c}}} \\ \frac{\partial f_{\omega_c}}{\partial \boldsymbol{q}_{\boldsymbol{c}}} & \frac{\partial f_{\omega_c}}{\partial \boldsymbol{\omega}_{\boldsymbol{c}}} & \frac{\partial f_{\omega_c}}{\partial \boldsymbol{T}_{\boldsymbol{c}}} \end{bmatrix}\right|_{\bar{\boldsymbol{q}}_{\boldsymbol{c}}, \bar{\boldsymbol{\omega}}_{\boldsymbol{c}}, \bar{\boldsymbol{T}}_{\boldsymbol{c}}} \begin{bmatrix} \bar{\boldsymbol{q}}_{\boldsymbol{c}} \\ \bar{\boldsymbol{\omega}}_{\boldsymbol{c}} \\ \bar{\boldsymbol{T}}_{\boldsymbol{c}} \end{bmatrix}. \end{aligned} \tag{16}$$

By regrouping the terms and denoting attitude as ${}^{a}(.)$, one finally obtain:

$$\begin{bmatrix} \dot{\boldsymbol{q}}_{\boldsymbol{c}} \\ \dot{\boldsymbol{\omega}}_{\boldsymbol{c}} \end{bmatrix} = {}^{a}\boldsymbol{A}_{\boldsymbol{c}} \begin{bmatrix} \boldsymbol{q}_{\boldsymbol{c}} \\ \boldsymbol{\omega}_{\boldsymbol{c}} \end{bmatrix} + {}^{a}\boldsymbol{B}_{\boldsymbol{c}}\boldsymbol{T}_{\boldsymbol{c}} + {}^{a}\boldsymbol{R}_{\boldsymbol{c}}, \tag{17}$$

with

$${}^{a}\boldsymbol{A}_{\boldsymbol{c}} = \left.\begin{bmatrix} \frac{\partial \dot{\boldsymbol{q}}_{\boldsymbol{c}}}{\partial \boldsymbol{q}_{\boldsymbol{c}}} & \frac{\partial \dot{\boldsymbol{q}}_{\boldsymbol{c}}}{\partial \boldsymbol{\omega}_{\boldsymbol{c}}} \\ \frac{\partial \dot{\boldsymbol{\omega}}_{\boldsymbol{c}}}{\partial \boldsymbol{q}_{\boldsymbol{c}}} & \frac{\partial \dot{\boldsymbol{\omega}}_{\boldsymbol{c}}}{\partial \boldsymbol{\omega}_{\boldsymbol{c}}} \end{bmatrix}\right|_{\bar{\boldsymbol{q}}_{\boldsymbol{c}}, \bar{\boldsymbol{\omega}}_{\boldsymbol{c}}, \bar{\boldsymbol{T}}_{\boldsymbol{c}}}, \tag{18}$$

$${}^{a}\boldsymbol{B}_{\boldsymbol{c}} = \left.\begin{bmatrix} \frac{\partial \dot{\boldsymbol{q}}_{\boldsymbol{c}}}{\partial \boldsymbol{T}_{\boldsymbol{c}}} \\ \frac{\partial \dot{\boldsymbol{\omega}}_{\boldsymbol{c}}}{\partial \boldsymbol{T}_{\boldsymbol{c}}} \end{bmatrix}\right|_{\bar{\boldsymbol{q}}_{\boldsymbol{c}}, \bar{\boldsymbol{\omega}}_{\boldsymbol{c}}, \bar{\boldsymbol{T}}_{\boldsymbol{c}}}, \tag{19}$$

and

$${}^{a}\boldsymbol{R}_{\boldsymbol{c}} = \begin{bmatrix} f_{q_c}(\bar{\boldsymbol{q}}_{\boldsymbol{c}}, \bar{\boldsymbol{\omega}}_{\boldsymbol{c}}, \bar{\boldsymbol{T}}_{\boldsymbol{c}}) \\ f_{\omega_c}(\bar{\boldsymbol{q}}_{\boldsymbol{c}}, \bar{\boldsymbol{\omega}}_{\boldsymbol{c}}, \bar{\boldsymbol{T}}_{\boldsymbol{c}}) \end{bmatrix} - \left.\begin{bmatrix} \frac{\partial f_{q_c}}{\partial \boldsymbol{q}_{\boldsymbol{c}}} & \frac{\partial f_{q_c}}{\partial \boldsymbol{\omega}_{\boldsymbol{c}}} & \frac{\partial f_{q_c}}{\partial \boldsymbol{T}_{\boldsymbol{c}}} \\ \frac{\partial f_{\omega_c}}{\partial \boldsymbol{q}_{\boldsymbol{c}}} & \frac{\partial f_{\omega_c}}{\partial \boldsymbol{\omega}_{\boldsymbol{c}}} & \frac{\partial f_{\omega_c}}{\partial \boldsymbol{T}_{\boldsymbol{c}}} \end{bmatrix}\right|_{\bar{\boldsymbol{q}}_{\boldsymbol{c}}, \bar{\boldsymbol{\omega}}_{\boldsymbol{c}}, \bar{\boldsymbol{T}}_{\boldsymbol{c}}} \begin{bmatrix} \bar{\boldsymbol{q}}_{\boldsymbol{c}} \\ \bar{\boldsymbol{\omega}}_{\boldsymbol{c}} \\ \bar{\boldsymbol{T}}_{\boldsymbol{c}} \end{bmatrix}. \tag{20}$$

The term ${}^{a}\boldsymbol{R}_{\boldsymbol{c}}$ represents the zeroth-order of the linearization along a trajectory.

The discretization of linear time-varying systems makes use of the Peano-Baker series in the general case. However, in the case studied here, the matrices are piecewise constant which drastically simplifies computation. By discretizing in the same way as in Eq. (3) and (4) to fit the numerical framework, for ΔT sufficiently close to 0, one obtains:

$${}^{a}\boldsymbol{A}_{\boldsymbol{d}}^{(\boldsymbol{k})} = e^{{}^{a}\boldsymbol{A}_{\boldsymbol{c}}^{(\boldsymbol{k})}\Delta T} = \boldsymbol{I} + {}^{a}\boldsymbol{A}_{\boldsymbol{c}}^{(\boldsymbol{k})}\Delta T, \tag{21}$$

$${}^{a}\boldsymbol{B}_{\boldsymbol{d}}^{(\boldsymbol{k})} = {}^{a}\boldsymbol{A}_{\boldsymbol{c}}^{(\boldsymbol{k})^{-1}}({}^{a}\boldsymbol{A}_{\boldsymbol{d}}^{(\boldsymbol{k})} - \boldsymbol{I}){}^{a}\boldsymbol{B}_{\boldsymbol{c}}^{(\boldsymbol{k})} = {}^{a}\boldsymbol{B}_{\boldsymbol{c}}^{(\boldsymbol{k})}\Delta T, \tag{22}$$

$$ {}^{a}\boldsymbol{R}_{\boldsymbol{d}}^{(\boldsymbol{k})} = {}^{a}\boldsymbol{A}_{\boldsymbol{c}}^{(\boldsymbol{k})^{-1}}({}^{a}\boldsymbol{A}_{\boldsymbol{d}}{}^{(k)} - \boldsymbol{I}){}^{a}\boldsymbol{R}_{\boldsymbol{c}}^{(\boldsymbol{k})} = {}^{a}\boldsymbol{R}_{\boldsymbol{c}}^{(\boldsymbol{k})}\Delta T, \tag{23} $$

with

$$ {}^{a}\boldsymbol{A}_{\boldsymbol{c}}^{(\boldsymbol{k})} = \left.\begin{bmatrix} \frac{\partial \dot{\boldsymbol{q}}_{\boldsymbol{c}}}{\partial \boldsymbol{q}_{\boldsymbol{c}}} & \frac{\partial \dot{\boldsymbol{q}}_{\boldsymbol{c}}}{\partial \boldsymbol{\omega}_{\boldsymbol{c}}} \\ \frac{\partial \dot{\boldsymbol{\omega}}_{\boldsymbol{c}}}{\partial \boldsymbol{q}_{\boldsymbol{c}}} & \frac{\partial \dot{\boldsymbol{\omega}}_{\boldsymbol{c}}}{\partial \boldsymbol{\omega}_{\boldsymbol{c}}} \end{bmatrix}\right|_{\bar{\boldsymbol{q}}_{\boldsymbol{c}}^{(\boldsymbol{k})}, \bar{\boldsymbol{\omega}}_{\boldsymbol{c}}^{(\boldsymbol{k})}, \bar{\boldsymbol{T}}_{\boldsymbol{c}}^{(\boldsymbol{k})}}, \tag{24} $$

$$ {}^{a}\boldsymbol{B}_{\boldsymbol{c}}^{(\boldsymbol{k})} = \left.\begin{bmatrix} \frac{\partial \dot{\boldsymbol{q}}_{\boldsymbol{c}}}{\partial \boldsymbol{T}_{\boldsymbol{c}}} \\ \frac{\partial \dot{\boldsymbol{\omega}}_{\boldsymbol{c}}}{\partial \boldsymbol{T}_{\boldsymbol{c}}} \end{bmatrix}\right|_{\bar{\boldsymbol{q}}_{\boldsymbol{c}}^{(\boldsymbol{k})}, \bar{\boldsymbol{\omega}}_{\boldsymbol{c}}^{(\boldsymbol{k})}, \bar{\boldsymbol{T}}_{\boldsymbol{c}}^{(\boldsymbol{k})}}, \tag{25} $$

and

$$ {}^{a}\boldsymbol{R}_{\boldsymbol{c}}^{(\boldsymbol{k})} = \begin{bmatrix} f_{q_c}(\bar{\boldsymbol{q}}_{\boldsymbol{c}}^{(\boldsymbol{k})}, \bar{\boldsymbol{\omega}}_{\boldsymbol{c}}^{(\boldsymbol{k})}, \bar{\boldsymbol{T}}_{\boldsymbol{c}}^{(\boldsymbol{k})}) \\ f_{\omega_c}(\bar{\boldsymbol{q}}_{\boldsymbol{c}}^{(\boldsymbol{k})}, \bar{\boldsymbol{\omega}}_{\boldsymbol{c}}^{(\boldsymbol{k})}, \bar{\boldsymbol{T}}_{\boldsymbol{c}}^{(\boldsymbol{k})}) \end{bmatrix} - \left.\begin{bmatrix} \frac{\partial f_{q_c}}{\partial \boldsymbol{q}_{\boldsymbol{c}}} & \frac{\partial f_{q_c}}{\partial \boldsymbol{\omega}_{\boldsymbol{c}}} & \frac{\partial f_{q_c}}{\partial \boldsymbol{T}_{\boldsymbol{c}}} \\ \frac{\partial f_{\omega_c}}{\partial \boldsymbol{q}_{\boldsymbol{c}}} & \frac{\partial f_{\omega_c}}{\partial \boldsymbol{\omega}_{\boldsymbol{c}}} & \frac{\partial f_{\omega_c}}{\partial \boldsymbol{T}_{\boldsymbol{c}}} \end{bmatrix}\right|_{\bar{\boldsymbol{q}}_{\boldsymbol{c}}^{(\boldsymbol{k})}, \bar{\boldsymbol{\omega}}_{\boldsymbol{c}}^{(\boldsymbol{k})}, \bar{\boldsymbol{T}}_{\boldsymbol{c}}^{(\boldsymbol{k})}} \begin{bmatrix} \bar{\boldsymbol{q}}_{\boldsymbol{c}}^{(\boldsymbol{k})} \\ \bar{\boldsymbol{\omega}}_{\boldsymbol{c}}^{(\boldsymbol{k})} \\ \bar{\boldsymbol{T}}_{\boldsymbol{c}}^{(\boldsymbol{k})} \end{bmatrix}. \tag{26} $$

The dynamical equations are then:

$$ \forall k \in [0, \dots, N-1],\ \begin{bmatrix} \boldsymbol{q}_{\boldsymbol{c}}^{(\boldsymbol{k+1})} \\ \boldsymbol{\omega}_{\boldsymbol{c}}^{(\boldsymbol{k+1})} \end{bmatrix} = {}^{a}\boldsymbol{A}_{\boldsymbol{d}}^{(\boldsymbol{k})} \begin{bmatrix} \boldsymbol{q}_{\boldsymbol{c}}^{(\boldsymbol{k})} \\ \boldsymbol{\omega}_{\boldsymbol{c}}^{(\boldsymbol{k})} \end{bmatrix} + {}^{a}\boldsymbol{B}_{\boldsymbol{d}}^{(\boldsymbol{k})}\boldsymbol{T}_{\boldsymbol{c}}^{(\boldsymbol{k})} + {}^{a}\boldsymbol{R}_{\boldsymbol{d}}^{(\boldsymbol{k})}, \tag{27} $$

$$ \begin{bmatrix} \dots & \boldsymbol{0_{7\times7}} & {}^{a}\boldsymbol{A}_{\boldsymbol{d}}^{(\boldsymbol{0})} & -\boldsymbol{I_{7\times7}} & \boldsymbol{0_{7\times7}} & \dots & \boldsymbol{0_{7\times7}} \\ \dots & \boldsymbol{0_{7\times7}} & \boldsymbol{0_{7\times7}} & {}^{a}\boldsymbol{A}_{\boldsymbol{d}}^{(\boldsymbol{1})} & -\boldsymbol{I_{7\times7}} & \ddots & \vdots \\ \dots & \vdots & \vdots & \ddots & \ddots & \ddots & \boldsymbol{0_{7\times7}} \\ \dots & \boldsymbol{0_{7\times7}} & \boldsymbol{0_{7\times7}} & \dots & \boldsymbol{0_{7\times7}} & {}^{a}\boldsymbol{A}_{\boldsymbol{d}}^{(\boldsymbol{N-1})} & -\boldsymbol{I_{7\times7}} \end{bmatrix} $$
$$ \begin{bmatrix} {}^{a}\boldsymbol{B}_{\boldsymbol{d}}^{(\boldsymbol{0})} & \boldsymbol{0_{7\times7}} & \dots & \boldsymbol{0_{7\times7}} & \boldsymbol{0_{7\times7}} & \dots \\ \boldsymbol{0_{7\times7}} & {}^{a}\boldsymbol{B}_{\boldsymbol{d}}^{(\boldsymbol{1})} & \ddots & \vdots & \boldsymbol{0_{7\times7}} & \dots \\ \vdots & \ddots & \ddots & \boldsymbol{0_{7\times7}} & \vdots & \dots \\ \boldsymbol{0_{7\times7}} & \dots & \boldsymbol{0_{7\times7}} & {}^{a}\boldsymbol{B}_{\boldsymbol{d}}^{(\boldsymbol{N-1})} & \boldsymbol{0_{7\times7}} & \dots \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} -{}^{a}\boldsymbol{R}_{\boldsymbol{d}}^{(\boldsymbol{0})} \\ -{}^{a}\boldsymbol{R}_{\boldsymbol{d}}^{(\boldsymbol{1})} \\ \vdots \\ -{}^{a}\boldsymbol{R}_{\boldsymbol{d}}^{(\boldsymbol{N-1})} \end{bmatrix}. \tag{28} $$

- The initial quaternions and velocities are also set beforehand. It is added a constraint to have them satisfied.

$$ \boldsymbol{q}_{\boldsymbol{c}}^{(\boldsymbol{0})} = \hat{\boldsymbol{q}}_{\boldsymbol{c}}^{(\boldsymbol{0})}, \tag{29} $$

$$ \boldsymbol{\omega}_{\boldsymbol{c}}^{(\boldsymbol{0})} = \hat{\boldsymbol{\omega}}_{\boldsymbol{c}}^{(\boldsymbol{0})}. \tag{30} $$

- The final quaternion constraint is chosen to be conic as its linear equality counterpart was leading to infeasibilities.

$$ \|\boldsymbol{q}_{\boldsymbol{c}}^{(\boldsymbol{N})} - \hat{\boldsymbol{q}}_{\boldsymbol{c}}^{(\boldsymbol{N})}\| \leq \epsilon_{q}^{(N)}. \tag{31} $$

Indeed, imposing a unitary-norm quaternion as the initial and final states is not compatible with norm drift usually encountered during propagation. This drift can, for example, be caused by the fact that a tangent plane (i.e. by use of the Jacobian) on the quaternion unit-norm sphere makes the linearization leave the sphere, thus making the quaternion norm slowly drifting away from being unitary. Methods to solve this could be the use of another attitude representation like Rodrigues parameters as discussed in [9].

2.4 Maximum Control and State Constraints

Upper-bounds for the control and/or state constraints have the general form:

$$\|*\| \leq *_{max}, \tag{32}$$

with $*$ a general quantity.

As explained in §2.2.2, it is introduced slack variables to linearize the formulation and fit the linear cost function. They are constrained in the following way:

$$\|*\| \leq \eta_* \leq *_{max}. \tag{33}$$

The norm is still less than the maximum value, and minimizing the slack variables by adding them in the linear cost function will therefore minimize the norm.

- The constraints to force the slack variables to be less than the maximum allowed thrust and torque are the following:

$$\forall i \in [0, \dots, N-1], \|\boldsymbol{F_c^{(i)}}\| \leq \eta_F^{(i)}, \tag{34}$$

$$\forall i \in [0, \dots, N-1], \eta_F^{(i)} \leq F_{max}. \tag{35}$$

- The torque follows the equivalent formulation:

$$\forall i \in [0, \dots, N-1], \|\boldsymbol{T_c^{(i)}}\| \leq \eta_T^{(i)}, \tag{36}$$

$$\forall i \in [0, \dots, N-1], \eta_T^{(i)} \leq T_{max}. \tag{37}$$

- Likewise, the angular velocity also has to be less than a maximum preset value.

$$\forall i \in [0, \dots, N], \|\boldsymbol{\omega_c^{(i)}}\| \leq \eta_\omega^{(i)}, \tag{38}$$

$$\forall i \in [0, \dots, N], \eta_\omega^{(i)} \leq \omega_{max}. \tag{39}$$

2.5 State-Triggered Constraints

State-triggered constraints are if-else constraints. They act as an approximating to heavy Mixed-Integer Linear Programming (MILP) formations [10] and are compatible with the convex framework. They allow constraints to only activate in certain configurations of the states. For instance, if the chaser in sufficiently close to the target, then it must always see it from its camera. In order to know whether or not the state configuration is achieved, the idea is to look at the current reference trajectory and enforce or relax the constraints for the current iteration.

2.5.1 Station Keeping in Hold Point

Suppose it is asked the chaser to hold (i.e. to have null relative velocity) during ΔT_{hold} after entering a sphere of radius d_{hold} around the target.

First, by inspecting the reference trajectory (previous optimal trajectory), it is detected the index i_{hold} at which the relative distance goes below the distance threshold. Then, it is added the following constraints:

$$\forall i \in [i_{hold}, i_{hold} + \Delta i_{hold}], \quad \dot{\boldsymbol{r}}_{\boldsymbol{ct}}^{(\boldsymbol{i})} = \mathbf{0}_{\mathbf{3\times 1}}. \tag{40}$$

with Δi_{hold} the number of indices representing the amount of time ΔT_{hold}.

2.5.2 Line-of-Sight (LOS) from Chaser to Target

This constraint ensures that the chaser sees the target from its camera by a predefined angle α_c and from a certain relative distance d_{LOS}. The is particularly useful to prevent collisions and estimate the correct attitude from navigation algorithms for the docking to happen properly.

Again, first it is found the index at which the reference trajectory gets into the range of the camera, denoted as i_{LOS}. Then, with respect to Fig. 1, it is added the constraints:

$$\forall i \in [i_{LOS}, \ldots, N], \ \alpha_{d_C T} \leq \alpha_C, \tag{41}$$

with $\alpha_{d_C T}$ the angle between the chaser's docking port and the target, and α_C the angle of the cone for the chaser's camera. It follows that:

$$\cos(\alpha_{d_C T}^{(i)}) \geq \cos(\alpha_C). \tag{42}$$

To make the optimization variables appear, it is used the definition of the scalar product of the two vectors that define $\alpha_{d_C T}^{(i)}$:

$$\frac{({}^{\boldsymbol{I}}\boldsymbol{r}_{\boldsymbol{t}}^{(\boldsymbol{i})} - {}^{\boldsymbol{C}}\boldsymbol{r}_{\boldsymbol{d_c}}^{(\boldsymbol{i})} - {}^{\boldsymbol{I}}\boldsymbol{r}_{\boldsymbol{c}}^{(\boldsymbol{i})}).{}^{\boldsymbol{C}}\boldsymbol{r}_{\boldsymbol{d_c}}^{(\boldsymbol{i})}}{\|{}^{\boldsymbol{I}}\boldsymbol{r}_{\boldsymbol{t}}^{(\boldsymbol{i})} - {}^{\boldsymbol{C}}\boldsymbol{r}_{\boldsymbol{d_c}}^{(\boldsymbol{i})} - {}^{\boldsymbol{I}}\boldsymbol{r}_{\boldsymbol{c}}^{(\boldsymbol{i})}\|\|{}^{\boldsymbol{C}}\boldsymbol{r}_{\boldsymbol{d_c}}^{(\boldsymbol{i})}\|} \geq \cos(\alpha_C). \tag{43}$$

Reorganizing the terms, one obtains:

$$\|-{}^{\boldsymbol{I}}\boldsymbol{r}_{\boldsymbol{ct}}^{(\boldsymbol{i})} - {}^{\boldsymbol{C}}\boldsymbol{r}_{\boldsymbol{d_c}}^{(\boldsymbol{i})}\| \cos(\alpha_c) \leq (-{}^{\boldsymbol{I}}\boldsymbol{r}_{\boldsymbol{ct}}^{(\boldsymbol{i})} - {}^{\boldsymbol{C}}\boldsymbol{r}_{\boldsymbol{d_c}}^{(\boldsymbol{i})}).\frac{{}^{\boldsymbol{C}}\boldsymbol{r}_{\boldsymbol{d_c}}^{(\boldsymbol{i})}}{\|{}^{\boldsymbol{C}}\boldsymbol{r}_{\boldsymbol{d_c}}^{(\boldsymbol{i})}\|}, \tag{44}$$

with

$${}^{\boldsymbol{I}}\boldsymbol{r}_{\boldsymbol{ct}}^{(\boldsymbol{i})} = {}^{\boldsymbol{I}}\boldsymbol{r}_{\boldsymbol{c}}^{(\boldsymbol{i})} - {}^{\boldsymbol{I}}\boldsymbol{r}_{\boldsymbol{t}}^{(\boldsymbol{i})}. \tag{45}$$

Since the geometry is the same in any frame, the constraint can be rephrased, expressing it in the LVLH frame:

$$\|-\boldsymbol{r}_{\boldsymbol{ct}}^{(\boldsymbol{i})} - {}^{\boldsymbol{LVLH}}\boldsymbol{r}_{\boldsymbol{d_c}}^{(\boldsymbol{i})}\| \cos(\alpha_c) \leq (-\boldsymbol{r}_{\boldsymbol{ct}}^{(\boldsymbol{i})} - {}^{\boldsymbol{LVLH}}\boldsymbol{r}_{\boldsymbol{d_c}}^{(\boldsymbol{i})}).\frac{{}^{\boldsymbol{LVLH}}\boldsymbol{r}_{\boldsymbol{d_c}}^{(\boldsymbol{i})}}{\|{}^{\boldsymbol{LVLH}}\boldsymbol{r}_{\boldsymbol{d_c}}^{(\boldsymbol{i})}\|}. \tag{46}$$

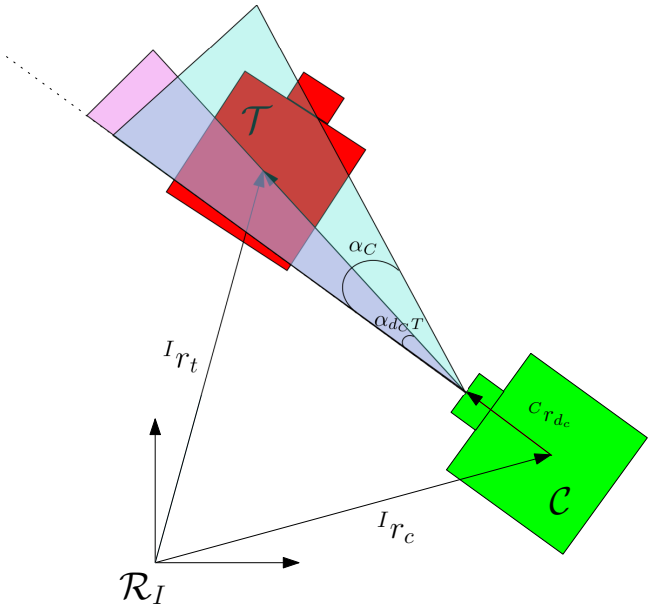


Figure 2: Geometric representation of the line-of-sight constraint from the chaser to the target, satisfied in this case.

2.5.3 Line-of-Sight from Target to Chaser

This version is the mirror of the previous one, namely asking for the target to see the chaser by an angle from a certain distance on.

$$\forall i \in [i_{LOS}, \dots, N],\ \|\boldsymbol{r}_{\boldsymbol{ct}}^{(i)} - {}^{\boldsymbol{LVLH}}\boldsymbol{r}_{\boldsymbol{d_t}}^{(i)}\| \cos(\alpha_T) \leq (\boldsymbol{r}_{\boldsymbol{ct}}^{(i)} - {}^{\boldsymbol{LVLH}}\boldsymbol{r}_{\boldsymbol{d_t}}^{(i)}).\frac{{}^{\boldsymbol{LVLH}}\boldsymbol{r}_{\boldsymbol{d_t}}^{(i)}}{\|{}^{\boldsymbol{LVLH}}\boldsymbol{r}_{\boldsymbol{d_t}}^{(i)}\|}. \tag{47}$$

2.5.4 Keep-Out Zone

To force the chaser to enter the cone from the top and not from the sides, a keep-out zone constraint is added to the optimization problem. It has the following form:

$$\forall i \in [0, \dots, i_{LOS}],\ \|\boldsymbol{r}_{\boldsymbol{ct}}^{(i)}\| \geq d_{LOS}. \tag{48}$$

On the one hand, at i_{LOS}, the chaser must be at the border or outside of the circle of radius d_{LOS}. On the other hand, it must be in the cone. Therefore, this ensures to enter from the top of the cone.

This constraint is then convexified into a linear one by squaring the previous equation and replacing the norm squared by the dot product of the vector by itself. Then, by using the reference in order to make it linear, one gets:

$$-\bar{\boldsymbol{r}}_{\boldsymbol{ct}}^{(i)}\boldsymbol{r}_{\boldsymbol{ct}}^{(i)} \leq -d_{LOS}^2. \tag{49}$$

2.6 Trust Region Constraints

The trust region constraint is one of the specifics of the successive convexification optimization that will be better explained in Sec. 3. It says that the new optimal trajectory should not be too far from

the reference one, as linearization was computed along it. One have:

$$\forall i \in [0, \dots, N], \quad \left\| \begin{matrix} \frac{1}{\tilde{L}} \begin{pmatrix} \boldsymbol{r}_{\boldsymbol{ct}}^{(\boldsymbol{i})} - \hat{\boldsymbol{r}}_{\boldsymbol{ct}}^{(\boldsymbol{i})} \end{pmatrix} \\ \frac{\tilde{T}}{\tilde{L}} \begin{pmatrix} \dot{\boldsymbol{r}}_{\boldsymbol{ct}}^{(\boldsymbol{i})} - \hat{\dot{\boldsymbol{r}}}_{\boldsymbol{ct}}^{(\boldsymbol{i})} \end{pmatrix} \end{matrix} \right\| \leq {}^{t}\rho^{(i)}, \tag{50}$$

$$\forall i \in [0, \dots, N], \quad \left\| \begin{matrix} \boldsymbol{q}_{\boldsymbol{c}}^{(\boldsymbol{i})} - \hat{\boldsymbol{q}}_{\boldsymbol{c}}^{(\boldsymbol{i})} \\ \tilde{T} \begin{pmatrix} \dot{\boldsymbol{\omega}}_{\boldsymbol{c}}^{(\boldsymbol{i})} - \hat{\dot{\boldsymbol{\omega}}}_{\boldsymbol{c}}^{(\boldsymbol{i})} \end{pmatrix} \end{matrix} \right\| \leq {}^{a}\rho^{(i)}, \tag{51}$$

with ${}^{t}\rho^{(i)}$ and ${}^{a}\rho^{(i)}$ the trust regions radius for the translation and attitude states respectively. $\tilde{T}$ and $\tilde{L}$ are the scaling factors, equal respectively to the orbital period of the chaser and its semi-major axis.

2.7 Objective Function

It is desired to minimize the energy over a fixed time horizon, i.e., the transfer time. The cost function will then be the sum of all the thrust and torque norms at every step. However, since the function has to be linear, one cannot directly use them. That is why one instead sums their associated slack variables. Moreover, to ensure an upper angular velocity bound at each time, the related slack variables are also added. Finally, as one wants to minimize the error between the optimal final quaternion and the required final quaternion, the slack variable for the specific constraint are added. Note that inputing the hard constraint instead of its relaxed counterpart may lead to infeasibilities with non-unitary quaternion norms. Normalization factors are added in the criterion.

The objective function J to be minimized is therefore defined as:

$$J = \frac{1}{F_{max}} \sum_{i=0}^{N-1} \eta_F^{(i)} + \frac{1}{T_{max}} \sum_{i=0}^{N-1} \eta_T^{(i)} + \frac{1}{\Omega_{max}} \sum_{i=0}^{N} \eta_\omega^{(i)} + \epsilon_q^{(N)}. \tag{52}$$

3 THE SUCCESSIVE CONVEXIFICATION ALGORITHM

3.1 Pseudo-Code

The algorithm used in this work can be summed up by Alg. 1.
In the following, it will be explained each phase of the algorithm, applied in the specific case of relative position and attitude control between chaser and target satellites.

Initialization The initialization consists in providing a translational and a rotational trajectory for the target as well as an initial guess for the chaser.

Scaling To improve the efficiency of the solver, a pre-scaling step is necessary. It consists in adimensionalizing all units in the physical constants. The scaling distance unit is given by the semi-major axis of the target a_T in circular orbit, while the time is its orbital period T_T.

$$\tilde{L} = a_T, \tag{53}$$

$$\tilde{T} = T_T. \tag{54}$$

Algorithm 1 Successive Convexification (SCvx)

```
Initialization
Scaling (Eq. 53, 54)
while Not Accurate or Not Converged do
    Linearization (Eq. 14)
    Discretization (Eq. 21, 22, 23)
    Constraints update
    Solving of the convexified sub-problem
    Non-linear repropagation (Eq. 55)
    if Not Feasible then
        Dilate trust region (Eq. 58)
    else
        if Not Accurate then
            Update trust region (Eq. 58, 59)
        else
            if Not Converged then
                Update trust region (Eq. 58, 59)
                Update reference (Eq. 60)
            end if
        end if
    end if
end while
```

Nonlinear Repropagation After the solving step has occured, in case of feasibility and optimal solution found, a nonlinear repropagation of the solution is performed. The forces and torques appearing in the equations are now the optimal ones from the solver. This allows to compare the precision between the convexified problem and the non-convex original problem.

$$\begin{cases} \ddot{\boldsymbol{r}}_{\boldsymbol{c}} = -\frac{\mu}{r_c^2}\frac{\boldsymbol{r_c}}{\|\boldsymbol{r_c}\|} + \frac{1}{m_c}\boldsymbol{R}_{\boldsymbol{\mathcal{I}}\leftarrow\boldsymbol{LVLH}}(t)\boldsymbol{F}_{\boldsymbol{c}}^{*} \\ \dot{\boldsymbol{q}}_{\boldsymbol{c}} = \frac{1}{2}\boldsymbol{\Omega}_{\boldsymbol{c}}\boldsymbol{q}_{\boldsymbol{c}} \\ \boldsymbol{I}_{\boldsymbol{c}}\dot{\boldsymbol{\omega}}_{\boldsymbol{c}} + \boldsymbol{\omega}_{\boldsymbol{c}} \times \boldsymbol{I}_{\boldsymbol{c}}\boldsymbol{\omega}_{\boldsymbol{c}} = \boldsymbol{T}_{\boldsymbol{c}}^{*} \end{cases} \tag{55}$$

with $\boldsymbol{R}_{\boldsymbol{\mathcal{I}}\leftarrow\boldsymbol{LVLH}}(t)$ the frame transformation from the LVLH to the inertial, and μ the gravitational parameter of the Earth.

Accuracy and Convergence Criteria In order to make sure the subproblem begin solved stays sufficently close to the nonconvex one, the SCvx algorithm employs a notion of accuray. It is defined as:

$$^{t}\epsilon_a = \max_{i=0\ldots N} \left\| \begin{matrix} \frac{1}{\tilde{L}}\left(\boldsymbol{r}_{\boldsymbol{ct}}^{*(i)} - \boldsymbol{r}_{\boldsymbol{ct}}^{\boldsymbol{NL}(i)}\right) \\ \frac{\tilde{T}}{\tilde{L}}\left(\dot{\boldsymbol{r}}_{\boldsymbol{ct}}^{*(i)} - \dot{\boldsymbol{r}}_{\boldsymbol{ct}}^{\boldsymbol{NL}(i)}\right) \end{matrix} \right\|, \quad ^{a}\epsilon_a = \max_{i=0\ldots N} \left\| \begin{matrix} \boldsymbol{q}_{\boldsymbol{c}}^{*(i)} - \boldsymbol{q}_{\boldsymbol{c}}^{\boldsymbol{NL}(i)} \\ \tilde{T}\left(\dot{\boldsymbol{\omega}}_{\boldsymbol{c}}^{*(i)} - \dot{\boldsymbol{\omega}}_{\boldsymbol{c}}^{\boldsymbol{NL}(i)}\right) \end{matrix} \right\|, \tag{56}$$

with $^{t}\epsilon_a$, and $^{a}\epsilon_a$ the accuracy criteria for the translation and attitude respectively.

Furthermore, the solution is only taken out if it has converged. The criteria related to it are the following:

$$^{t}\epsilon_c = \max_{i=0\ldots N} \left\| \begin{matrix} \frac{1}{\tilde{L}}\left(\boldsymbol{r}_{\boldsymbol{ct}}^{*(i)} - \hat{\boldsymbol{r}}_{\boldsymbol{ct}}^{(i)}\right) \\ \frac{\tilde{T}}{\tilde{L}}\left(\dot{\boldsymbol{r}}_{\boldsymbol{ct}}^{*(i)} - \hat{\dot{\boldsymbol{r}}}_{\boldsymbol{ct}}^{(i)}\right) \end{matrix} \right\|, \quad ^{a}\epsilon_c = \max_{i=0\ldots N} \left\| \begin{matrix} \boldsymbol{q}_{\boldsymbol{c}}^{*(i)} - \hat{\boldsymbol{q}}_{\boldsymbol{c}}^{(i)} \\ \tilde{T}\left(\dot{\boldsymbol{\omega}}_{\boldsymbol{c}}^{*(i)} - \hat{\dot{\boldsymbol{\omega}}}_{\boldsymbol{c}}^{(i)}\right) \end{matrix} \right\|, \tag{57}$$

with ${}^t\epsilon_c$, and ${}^a\epsilon_c$ the convergence criteria for the translation and attitude respectively.

Trust Region Update A trust region is defined as a search region for the next iteration of the optimal solution. To make sure that the convexified sub-problem is close to the original one, the optimizer is constrained to find solutions which are not too far away from the reference trajectory. Indeed, the linearization step is valid around a neighborhood of the reference.

When the problem is infeasible, it could be because the trust region radius is too small. In that case, it is dilated until the problem becomes feasible again and a new trajectory can be obtained.

$$\rho^{(i+1)} = K^+ \rho^{(i)}, K^+ \geq 1, \tag{58}$$

with K the dilation radius gain.

On the other hand, when the solution is feasible, one wants to reduce the trust radius. The update is performed as a function of the optimal and the nonlinear cost functions. The closer they are, the smaller the new trust radius will be:

$$\rho^{(i+1)} = K^- |f^{(i)*} - f^{(i)NL}| \, \rho^{(i)}, \tag{59}$$

with K^- the contraction radius gain.

Reference Update As long as the convergence between 2 iterations of the optimal solution does not fall under a certain threshold, the optimal solution becomes the new reference to linearize along:

$$\bar{\boldsymbol{x}}^{(i+1)} = \boldsymbol{x}^{(i)*}. \tag{60}$$

4 NUMERICAL RESULTS

The values used in this simulation are given by Table. 1. The reference optimization vector was initiated with zero thrust and torque, following the initial values provided. Matlab R2021a was employed with Mosek [5] as the optimization solver. No perturbations were added in this setup.

Table 1: Simulation Values

Mission	
m_c [kg]	1000
I_c, I_t [kg.m^2]	$1000.I_3$
a_t [km]	42165
F_{max} [N]	1
T_{max} [N.m]	1
ω_{max} [rad/s]	0.1
α_C [°]	30
α_T [°]	30
d_{hold} [m]	1000
ΔT_{hold} [s]	100
d_{LOS} [m]	300

Simulation	
T_{sim} [s]	5000
N	100
ϵ_a	1
ϵ_c	10^{-7}
K^-	0.1
K^+	10

Initial states	
$\hat{r}_{ct}^{(0)}$ [m]	[-2000, 500, 200]^T
$\hat{\dot{r}}_{ct}^{(0)}$ [m]	[0, 0, 0]^T
$\hat{r}_{ct}^{(N)}$ [m]	[0, 0, 0]^T
$\hat{\dot{r}}_{ct}^{(N)}$ [m]	[0, 0, 0]^T
$\hat{q}_c^{(0)}$	$[\sqrt{1-0.3^2}, 0, -0.3, 0]^T$
$\hat{\omega}_c^{(0)}$ [rad/s]	[0, 0, 0.0001]^T
$\hat{q}_t^{(0)}$	$[\sqrt{1-0.3^2}, 0, 0.3, 0]^T$
$\hat{\omega}_t^{(0)}$ [rad/s]	[0, 0, 0.01]^T

In Fig. 3, the optimal thrust norm is shown. The observe the satisfaction of the maximum norm constraint of 1N. The thrust is either full or null. We depict the initial acceleration, then the deceleration

to hold at null relative velocity, then a new acceleration to stop holding, and the final deceleration to arrive at null velocity at the target.

Then, in Fig. 4a, the distance of the chaser to the target over time is displayed. The hold point of 100s at 1000m is observed. Then, in Fig. 4b, the satisfaction of the approach cone angle constraint is shown. The chaser slides along the surface of the cone to minimize its energy consumption.

Moreover, in Fig. 5, the translation convergence of the algorithm over time is plotted. The algorithm stops as soon as this metric reaches below the given threshold of 10^{-7}.

Finally, Fig. 6 shows two different views of another example representing the chaser rotating around the target to enter its approach cone with a hold point. The family of curves displayed here is the iterations of SCvx, ending with the red bold one.

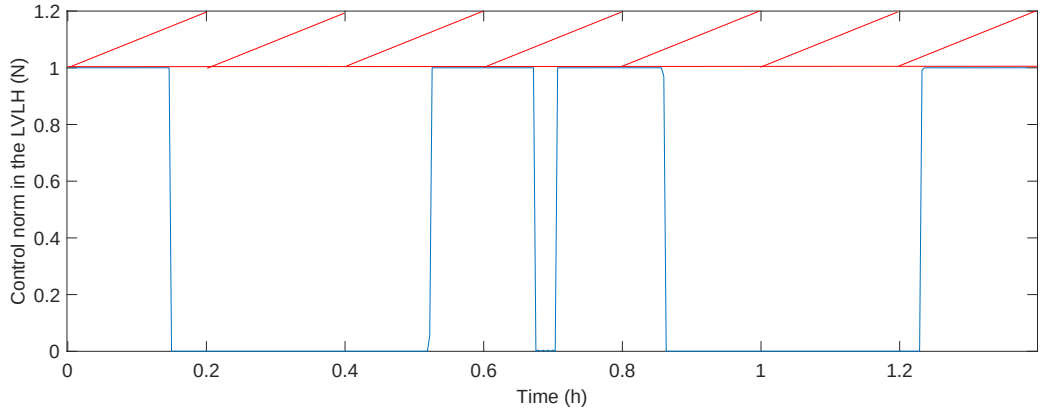
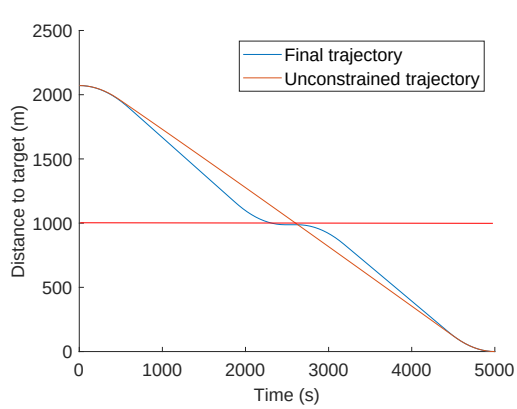
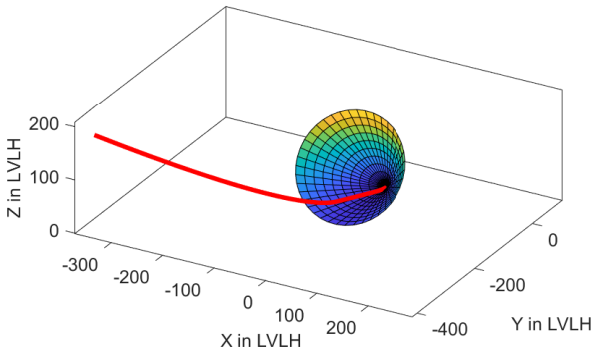


Figure 3: Optimal thrust of the chaser. It satisfies the maximum norm constraint of 1N.



(a) Distance of the chaser to the target over time. The hold point of 100s at 1000m is observed.



(b) 3D approach of the chaser in the LVLH frame. The approach cone angle constraint is shown.

Figure 4: Hold point and approach cone angle constraint.

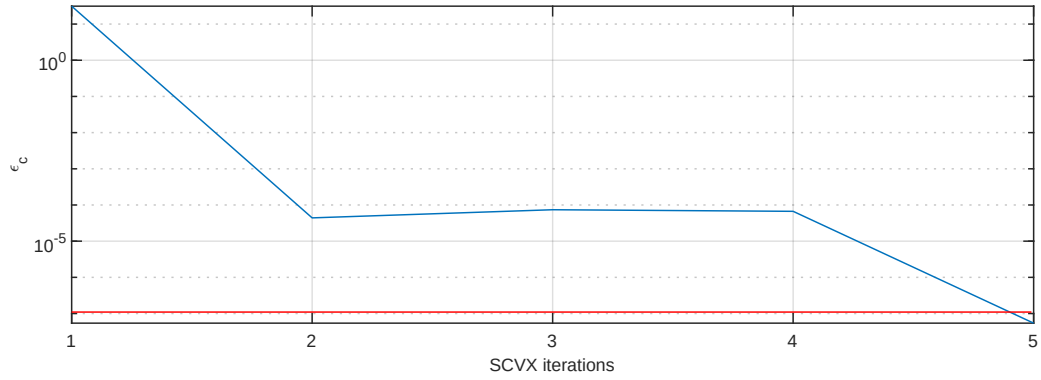


Figure 5: Convergence of the algorithm over time. The algorithm stops as soon as this metric reaches below the given threshold of 10^{-7}.

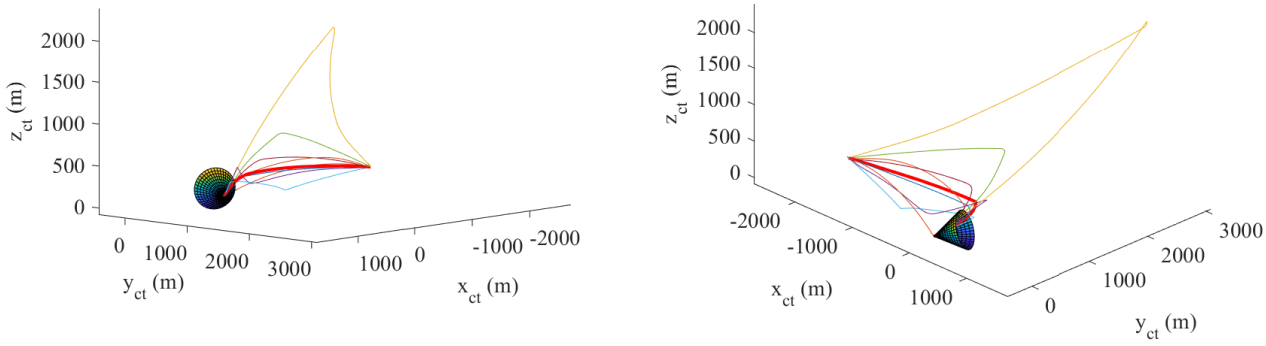


Figure 6: Two different views of another example representing the chaser rotating around the target to enter its approach cone with a hold point. The family of curves shows the iterations of SCvx, ending with the red bold one.

5 CONCLUSION

In this paper, a relaxed version of the quaternion end-point constraint was implemented to improve feasibility. Also, a linear relaxation of the keep-out zone constraint to enter from the top of the approach cone was implemeted. The effectiveness of this approach was demonstrated through numerical simulations.

REFERENCES

[1] W. Fehse, *Automated Rendezvous and Docking of Spacecraft,* Cambridge, UK: Cambridge University Press, 2003, vol. 16.

[2] H. Jiang, M. S. Drew, and Z.-N. Li, “Matching by linear programming and successive convex-

ification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 959–975, 2007.

[3] D. Malyuta, T. P. Reynolds, M. Szmuk, T. Lew, R. Bonalli, M. Pavone, and B. Açıkmeşe, "Convex optimization for trajectory generation: A tutorial on generating dynamically feasible trajectories reliably and efficiently," *IEEE Control Systems Magazine*, vol. 42, no. 5, pp. 40–113, 2022.

[4] E. Blazquez, "Rendezvous optimization and gnc design for proximity operations on cis-lunar near rectilinear halo orbits," Ph.D. dissertation, Institut Supérieur de l'Aéronautique et de l'Espace (ISAE), Toulouse, France, 2021.

[5] M. ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 10.0.*, 2023. Available: http://docs.mosek.com/9.0/toolbox/index.html

[6] D. Malyuta, T. Reynolds, M. Szmuk, B. Acikmese, and M. Mesbahi, "Fast trajectory optimization via successive convexification for spacecraft rendezvous with integer constraints," in *AIAA Scitech 2020 Forum*, 2020, p. 0616.

[7] W. Clohessy and R. Wiltshire, "Terminal guidance system for satellite rendezvous," *Journal of the Aerospace Sciences*, vol. 27, no. 9, pp. 653–658, 1960.

[8] D. Malyuta, T. Reynolds, M. Szmuk, M. Mesbahi, B. Acikmese, and J. M. Carson, "Discretization performance and accuracy analysis for the rocket powered descent guidance problem," in *AIAA Scitech 2019 Forum*, 2019, p. 0925.

[9] B. E. Jackson, K. Tracy, and Z. Manchester, "Planning with attitude," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5658–5664, 2021.

[10] T. Claudet, R. Alimo, E. Goh, M. D. Johnston, R. Madani, and B. Wilson, "Δ-milp: Deep space network scheduling via mixed-integer linear programming," *IEEE Access*, vol. 10, pp. 41 330–41 340, 2022.