75th International Astronautical Congress (IAC), Milan, Italy, 14-18 October 2024.

IAC–24–C1.3.4

AUTONOMOUS GUIDANCE, NAVIGATION, AND CONTROL FOR CLOSE FORMATIONS THROUGH SEQUENTIAL CONVEX PROGRAMMING AND INTER-SATELLITE RANGING

Enrico Belloni [a*], **Francesco De Cecio** [a], **Michèle Lavagna** [a]

[a] *Department of Aerospace Science and Technology, Politecnico di Milano, Italy*

[*] Corresponding Author: enrico.belloni@polimi.it

A trending topic for future space missions is to distribute complex tasks among several agents, instead of relying on large monolithic spacecraft. In these scenarios, the acquisition and maintenance of the formation geometry becomes a critical aspect to be addressed that cannot be solved with traditional ground-based techniques. Hence, the development of novel and autonomous guidance, navigation, and control methods plays a driving role in enabling such mission concepts. In this paper, distributed algorithms are explored to accomplish both tasks in an organic and cooperative manner, trying to investigate all the key challenges of the problem. An innovative strategy for the relative state estimation and control of large spacecraft formations is proposed, based on a convex model predictive control strategy and radio-frequency navigation via inter-satellite link. More in detail, the guidance and control of the distributed system are addressed with sequential convex programming techniques that allow the computation of optimal control profiles for the formation considering goal-oriented objectives and safety-related constraints in a computationally efficient algorithmic routine. On the other hand, the selected navigation technique exploits a radio-frequency network architecture established between agents based on code division multiple access. Range measurements obtained through a GNSS-like signal transmitted by each agent are simulated. A real-time estimate of the formation state can thus be obtained with an increase in accuracy with respect to only using ECI-based measurements by GNSS sensors. The proposed distributed design optimizes the time and fuel spent by each agent for formation acquisition and keeping while retaining with higher fidelity the features of the problem like noisy, asynchronous measurement collection.

Keywords: formation flying, Sequential Convex Programming, Model Predictive Control, Inter-satellite ranging

Nomenclature

$\mathbf{A}$	: dynamics plant matrix
a	: semi-major axis
$\mathbf{B}$	: control matrix
B	: ballistic coefficient
e	: eccentricity
i	: inclination
M	: mean anomaly
n	: mean motion
R	: radius
T	: thrust
u	: argument of latitude
v	: velocity
θ	: true anomaly
ρ	: density
Ω	: right-ascension of the ascending node
ω	: argument of perigee

Abbreviations

CH	: control horizon
DEM	: digital elevation map
ECI	: earth-centered inertial
EKF	: extended kalman filter
GNC	: guidance, navigation and control
GNSS	: global navigation satellite system
ISL	: inter-satellite link
LTAN	: local time of the ascending node
LTV	: linear time variant
LVLH	: local-vertical local-horizontal
MPC	: model predictive control
OCP	: optimal control problem
OOE	: osculating orbital elements
PH	: prediction horizon
ROE	: relative orbital elements
SSO	: sun-synchronous orbit
UKF	: unscented Kalman filter

1. Introduction

The miniaturization trend of space systems is progressively encouraging a switch from big monolithic spacecraft towards distributed systems of smaller satellites acting and cooperating together to perform complex tasks, such interferometry, in-orbit assembly, target inspection, and so on. One of the most complex and important objectives to achieve is therefore to design safe and robust guidance, navigation, and control algorithms for these formations to guarantee the desired level of accuracy, safety, and autonomy level. Traditional ground-based techniques for spacecraft operations are insufficient to manage the continuous and autonomous control of formation flying, making the development of novel, onboard algorithms essential for future mission concepts. In distributed spacecraft systems, maintaining the desired formation geometry becomes a critical requirement for mission success. The relative positioning and control of each agent must be managed autonomously, in real-time, and under the strict computational and resource constraints of space processors. These systems also need to be robust to uncertainties, such as asynchronous data collection and noisy measurements.

This paper proposes an innovative approach to formation control and navigation, leveraging convex model predictive control (MPC) and radio-frequency (RF) navigation based on inter-satellite links (ISL). The goal is to develop a distributed, autonomous framework that is suitable for small satellite formations that optimizes both the computational burden and the performance of the formation-keeping process. Sequential convex programming (SCP) techniques are used to compute optimal control profiles, while a code division multiple access (CDMA) RF network facilitates relative state estimation. This strategy aims to provide a robust and efficient solution for maintaining large spacecraft formations in challenging space environments. Through several simulations, this paper evaluates the performance and robustness of the proposed system by testing the proposed algorithmic framework in challenging mission scenarios with that require precise formation control.

2. Satellite dynamics

The relative motion between each satellite of the formation is defined with respect to a common virtual reference placed in the centroid, and it is modeled by exploiting a quasi-nonlinear Relative Orbital Elements (ROE) state representation [1]. ROE are nonlinear combinations of mean orbital elements (MOE) that can be linearly propagated by proper plant matrices to include the effects of keplerian motion and most relevant perturbations [2] [3]. A ROE representation is chosen due to its more direct connection to the physics of the problem and relative motion geometry and for their smaller linearization errors in non-circular orbits and for large spacecraft separations, showing several advantages with respect to the traditionally used Hills-Clohessy-Wiltshire equations [4], which imply a cartesian representation in the referenced-centered LVLH reference frame. Moreover, their slowly-varying nature is beneficial to computational efficiency. Furthermore, in order to include the effects of differential drag in the relative orbital motion, the state is augmented with a differential ballistic term ΔB. The augmented ROE state is then computed as:

$$\underline{\mathbf{x}} = \begin{pmatrix} \delta a \\ \delta\lambda \\ \delta e_x \\ \delta e_y \\ \delta i_x \\ \delta i_y \\ \Delta B \end{pmatrix} = \begin{pmatrix} (a-a_c)/a_c \\ u-u_c+(\Omega-\Omega_c)\cdot\cos(i_c) \\ e\cdot\cos(\omega)-e_c\cdot\cos(\omega_c) \\ e\cdot\sin(\omega)-e_c\cdot\sin(\omega_c) \\ i-i_c \\ (\Omega-\Omega_c)\cdot\sin(i_c) \\ (B-B_c)/B \end{pmatrix} \tag{1}$$

Using this state vector, the natural relative dynamics with respect to the reference trajectory can then be linearly propagated in state-space form as:

$$\dot{\underline{\mathbf{x}}}(t) = \mathbf{A}(t)\underline{\mathbf{x}}(t) + \mathbf{B}(t)\underline{\mathbf{u}}(t) \tag{2}$$

where the control vector $\underline{\mathbf{u}}$ is expressed in *xyz* coordinates in the chief-centered LVLH frame and the plant matrix for the natural dynamics is given by the sum of keplerian motion plus the most relevant disturbances in LEO, namely, drag and J2 effects:

$$\mathbf{A}(t) = \mathbf{A_{kep}} + \mathbf{A_{J2}}(t) + \mathbf{A_{drag}}(t) \tag{3}$$

$$\mathbf{A_{kep}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{3}{2}n_c & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

$$\mathbf{A_{J2}} = \kappa_{J2} \cdot \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{7}{2}EP & 0 & e_xGFP & e_yGFP & -FS & 0 & 0 \\ \frac{7}{2}e_yQ & 0 & -4e_xe_yGQ & -(1+4Ge_y^2)Q & 5e_yS & 0 & 0 \\ -\frac{7}{2}e_yQ & 0 & (1+4Ge_x^2)Q & 4e_xe_yGQ & -5e_xS & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{7}{2}S & 0 & -4e_xGS & -4e_yGS & 2T & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5}$$

$$\mathbf{A_{drag}} = B\rho v^2 \cdot \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \frac{a^2 v}{\mu} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{(e+\cos(\theta))\cdot\cos(\omega)-\sin(\theta)\sin(\omega)}{v} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{(e+\cos(\theta))\cdot\sin(\omega)+\sin(\theta)\cos(\omega)}{v} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (6)$$

where the terms in the $\mathbf{A_{J2}}$ matrix are defined as in the following expressions:

$$\eta = \sqrt{1-e_c^2}, \quad \kappa_{J2} = \frac{3}{4}\frac{J_2 R_E^2 \sqrt{\mu}}{a^{3.5}\eta^4}, \quad E = 1+\eta,$$
$$F = 4+3\eta, \quad G = \frac{1}{\eta^2}, \quad P = 3\cos^2(i_c)-1,$$
$$Q = 5\cos^2(i_c)-1, \quad S = \sin(2i_c), \quad T = \sin^2(i_c)$$

The formulations for the three different plant matrices are taken respectively from [5], [2], and [3]. The density term ρ in the drag plant matrix can be computed onboard with any atmospheric model of choice, naturally, a better model improves the accuracy and optimality of the algorithm. The differential ballistic coefficient in the state vector is assumed to be equal to 1 when control is performed to a virtual nondecaying reference, whereas it can be computed starting from physical characteristics for any couple of satellites . For what concerns the control matrix **B** that maps the control actions in the LVLH frame into the correspondent variation of ROE, the used formulation is taken from [6] and is reported in the following:

$$\mathbf{B} = \frac{1}{a_c n_c} \cdot \begin{bmatrix} \frac{2}{\eta}e_c\sin(\theta_c) & \frac{2}{\eta}(1+e_c\cos(\theta_c)) & 0 \\ -\frac{2\eta^2}{1+e_c\cos(\theta_c)} & 0 & 0 \\ \eta\cdot\sin(u_c) & \eta\frac{(2+e_c\cos(\theta_c))\cos(u_c)+e_x}{1+e_c\cos(\theta_c)} & \frac{\eta e_y}{\tan(i_c)}\frac{\sin(u_c)}{1+e_c\cos(\theta_c)} \\ -\eta\cdot\cos(u_c) & \eta\frac{(2+e_c\cos(\theta_c))\sin(u_c)+e_y}{1+e_c\cos(\theta_c)} & -\frac{\eta e_x}{\tan(i_c)}\frac{\sin(u_c)}{1+e_c\cos(\theta_c)} \\ 0 & 0 & \eta\frac{\cos(u_c)}{1+e_c\cos(\theta_c)} \\ 0 & 0 & \eta\frac{\sin(u_c)}{1+e_c\cos(\theta_c)} \\ 0 & 0 & 0 \end{bmatrix} \quad (7)$$

3. GNC formulation

3.1 GNSS and ranging-based navigation

3.1.1 Measurement types

The navigation filter is designed to process two typologies of measurements, the improvement that each of these can give to navigation accuracy is evaluated in Section 4:

1. *Indirect GNSS ranging:* the relative state is computed by taking the difference between two time-tagged GNSS solutions (i.e. absolute position and velocity in the Earth-Centered Inertial reference frame) from different agents, assumed that each of them is sharing its own measurements among the formation via the ISL.

2. *One-way direct ranging:* a direct, high-precision, measurement of the distance between each two agents is provided through a one-way ranging scheme. From a technological standpoint, this measure can be obtained with an ISL signal designed to carry both a pseudo-random noise (PRN) code and a navigation message [7].

To be able to use these measurements in the ROE-based dynamics model these measurements, which find their natural representation in the ECI and LVLH reference frames respectively, some information post-processing is needed. In particular, a filtering technique is employed to recover the relative state of interest from noisy data coming to GNSS sensors. First, the output of the GNSS receiver is filtered with an Unscented Kalman Filter (UKF) exploiting the knowledge of the full nonlinear perturbed model, then it is converted first to Osculating Orbital Elements (OOE), and next to Mean Orbital Elements (MOE) via Brouwer transformation [8]. The retrieved MOE state is then used for the computation of the relative state with respect to the reference. It should be pointed out that Brouwer transformation does not provide the most accurate solution in terms of MOE, semi-major axis in particular, and that more precise methods have been proposed in recent literature [9], [10]. At this point, the MOE sets of the agents for which a ROE set is sought, are synchronized to the present epoch by propagating them. Propagation time is computed thanks to the timestamp associated with each GNSS measurement. The inaccuracies due to Brouwer transformation mostly cancel out in the ROE evaluation due to the very small distances of the formation agents with respect to the chief [11]. A representation of the transformation process is represented in Figure 1. The ranging can instead be included in the navigation filter using the nonlinear measurement model presented in the following section.

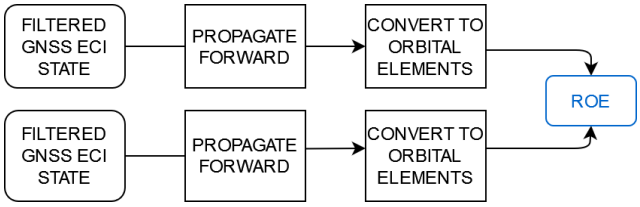

Fig. 1: Simplified pipeline of ROE computation from GNSS state estimates.

3.1.2 Measurement models

The obtained GNSS-based ROE measurements are then fed to the main EKF for relative state estimation. The innovation that is computed from such measure can be computed as follows:

$$\begin{aligned} \underline{\mathbf{y}}_k &= \underline{\mathbf{ROE}}_k - \mathbf{H}_k \underline{\hat{\mathbf{x}}}_{k|k-1}(z) \\ \mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \\ \mathbf{H}_k &= \mathbf{I}_{3x3} \end{aligned} \tag{8}$$

It should be remarked that, in this case, the filter architecture corresponds to a standard Kalman filter. As anticipated, To use the range measure in the filter, a nonlinear measurement model, mapping the state expressed in the reference-centered LVLH frame to the square of the relative position vector, has been derived. The transformation of the state from ROE to LVLH representation is computed through a linear function. The innovation in this case can be expressed as:

$$\begin{aligned} \underline{\mathbf{y}}_k &= R_k^2 - h(\mathbf{T}_k \underline{\hat{\mathbf{x}}}_{k|k-1}(z)) \\ \mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \\ \mathbf{H}_k &= 2 \cdot [1,1,1]^T \mathbf{T}_k \end{aligned} \tag{9}$$

where the ROE-to-LVLH transformation matrix $\mathbf{T}$ is derived introducing an intermediate change of coordinates exploiting the classical orbital elements differences vector $\Delta OE = [\Delta a, \Delta M, \Delta\omega, \Delta e, \Delta i, \Delta\Omega]$ as follows:

$$\mathbf{T} = \frac{\partial \underline{\mathbf{x}}_{\mathbf{LVLH}}}{\partial \Delta OE} \cdot \frac{\partial \Delta OE}{\partial \delta \underline{\alpha}} \tag{10}$$

The first-order approximation of the mapping between LVLH state and classical osculating orbital elements difference is extrapolated from the equations described in [1]:

$$\begin{cases} x = \frac{r}{a}\Delta a - a \cdot \cos(\theta)\Delta e + \frac{a \cdot e \cdot \sin(\theta)}{\sqrt{1-e^2}}\Delta M \\ y = \left(a + \frac{r}{1-e^2}\right)\sin(\theta)\Delta e + \frac{a^2}{r}\eta\Delta M + r\Delta\omega \; + \; r \cdot \cos(i)\Delta\Omega \\ z = r \cdot \sin(u)\Delta i - r \cdot \sin(i)\cos(u)\Delta\Omega \end{cases} \tag{11}$$

From the previous equations, the first transformation matrix can be retrieved. Since only the relative positions are of interest, the last four rows can be set equal to zero to avoid useless computations. The same yields for the last column, as no dependence on the differential ballistic coefficient is present.

$$\frac{\partial \underline{\mathbf{x}}_{\mathbf{LVLH}}}{\partial \Delta OE} = r \cdot \begin{bmatrix} 1/a & \frac{a \cdot e \cdot \sin(\theta)}{r\sqrt{1-e^2}} & 0 & -\frac{a}{r} \cdot \cos(\theta) & 0 & 0 & 0 \\ 0 & \frac{a^2}{r^2}\eta & 1 & \left(\frac{a}{r} + \frac{1}{1-e^2}\right)\sin(\theta) & 0 & \cos(i) & 0 \\ 0 & 0 & 0 & 0 & \sin(u) & -\sin(i)\cos(u) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{12}$$

In order to obtain the full mapping, the Jacobian of the transformation from classical orbital elements to quasi-nonsingular ROE is necessary. This is obtained from the definition of $\delta\underline{\alpha}$ for $\delta OE \to 0$ as:

$$\frac{\partial \Delta OE}{\partial \delta \underline{\alpha}} = \begin{bmatrix} a & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \frac{\sin(\omega)}{e} & -\frac{\cos(\omega)}{e} & 0 & \frac{\cos(i)}{\sin(i)} & 0 \\ 0 & 0 & -\frac{\sin(\omega)}{e} & \frac{\cos(\omega)}{e} & 0 & 0 & 0 \\ 0 & 0 & \cos(\omega) & \sin(\omega) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sin(i) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{13}$$

3.2 Guidance and control

The guidance and control strategy for the formation relies on Model Predictive Control (MPC), to process the relative state output of the navigation chain and exploit the knowledge of the dynamics to solve an Optimal Control Problem (OCP) in real-time, taking into account all the imposed constraints. The control profile is computed over a so-called Prediction Horizon (PH), discretized into K time steps defined by the sample time Δt and applied over a shorter Control Horizon (CH), after which a new optimization starts and

an updated control action is computed. The dynamics of the problem can be easily expressed in convex form, taking advantage of the previously introduced linear dynamics formulation. On the other hand, collision avoidance brings a constraint that is inherently non-convex. Many strategies have been proposed to convexify this constraint using affine hyperplane representations, however, in this paper Sequential Convex Programming (SCP) is preferred in order not to overconstrain the agents, especially in presence of large formations. The specific SCP routine that is selected is Guaranteed Sequential Trajectory Optimization (GuSTO) [12], a trust region based algorithm that finds a convergence proof in the Pontryagin principle. The peculiar choice of this strategy is to include trust region and path constraint violation terms in the cost function, as will be better clarified in the following section.

3.2.1 Convexified optimal control subproblem

The objective of the MPC is to find the trajectory, and correspondent control profile, that minimizes a weighted function of tracking error and fuel cost, along with the other terms that must be added to ensure SCP convergence.

Dynamics: The dynamics of the system can be directly inserted in the cost function since, with the LTV formulation previously introduced, the state evolution of the system can be written in the form:

$$\underline{\mathbf{x}}_{k+1} = \Phi_k \underline{\mathbf{x}}_k + \Psi_k \underline{\mathbf{u}}_k \tag{14}$$

where:

$$\begin{aligned} \Phi_k &= \mathbf{A}_k \Delta t + \mathbf{I} \quad &\in \mathbb{R}^{7\times 7} \\ \Psi_k &= \mathbf{B}_k \Delta t \quad &\in \mathbb{R}^{7\times 3} \end{aligned} \tag{15}$$

Thus, the entire trajectory evolution of the system over the propagated horizon can be simply obtained as:

$$\underline{\mathbf{X}} = \mathbf{M}\underline{\mathbf{U}} + \mathbf{N}\underline{\mathbf{x}}_0 \tag{16}$$

in which $\underline{\mathbf{U}}$ is a column vector of dimension $N = 3(K-1)$ containing the prescribed control action at each time step over the prediction horizon, and the matrices $\mathbf{M}$ and $\mathbf{N}$ are defined as:

$$\mathbf{M} = \begin{bmatrix} \Psi_1 & & & & \\ \Phi_2\Psi_1 & \Psi_2 & & & \\ \Phi_3\Phi_2\Psi_1 & \Phi_2\Psi_2 & \Psi_3 & & \\ \vdots & & & \ddots & \\ (\prod_{k=2}^{K-1}\Phi_k)\Psi_1 & \cdots\cdots\cdots\cdots & & & \Psi_{K-1} \end{bmatrix} \in \mathbb{R}^{M\times N} \tag{17}$$

$$\mathbf{N} = \begin{bmatrix} \Phi_1 \\ \vdots \\ \\ \prod_{k=1}^{K-1}\Phi_k \end{bmatrix} \in \mathbb{R}^{M\times 7} \tag{18}$$

where $M = 7(K-1)$.

Dynamics: The original cost functional of the problem to be minimized is a weighted function of the ΔV and tracking error of the entire trajectory with respect to the desired final ROE state:

$$\Gamma(\underline{\mathbf{X}}, \underline{\mathbf{U}}) = ||\underline{\mathbf{U}}||_1 + ||\mathbf{P}\cdot(\underline{\mathbf{X}} - \underline{\tilde{\mathbf{X}}})||_1 \tag{19}$$

in which $\mathbf{P}$ is a diagonal weighting matrix on the tracking error term of the cost function that allows to prioritize some ROE with respect to others. In the minimization algorithm, $\underline{\mathbf{U}}$ is the decisional vector to be found. In fact, the state can be expressed as a linear function of the control profile and initial conditions as specified in eq. (16). In any case, the dependence on both parameters is kept in the notation for clarity.

As previously mentioned, GuSTO also requires the inclusion of soft penalties on both trust region violation, in this case, collision avoidance nonconvex constraint violation. These are included via a convex and non-decreasing penalty function $h_\lambda : \mathbb{R} \to \mathbb{R}_+$ that depends on a scalar weight λ. The goal of is to penalize any positive value and ignore non-positive values. The chosen solution is to simply take the positive part of the argument:

$$h_\lambda(\mathbf{z}) = \lambda([\mathbf{z}]^+) \tag{20}$$

Hence, the soft penalty on the non-convex path constraint violation can be expressed as:

$$g_{pc}(\underline{\mathbf{X}}) \triangleq h_\lambda(s(\underline{\mathbf{X}})) \tag{21}$$

Where $s(\mathbf{x}, p)$ is the function that defines the nonconvex path contraint:

$$s(\underline{\mathbf{X}}) \triangleq R_{\text{koz}}^2 - A_{\text{ss}}(\mathbf{A}_{\text{ca}}\mathbf{J}\underline{\mathbf{X}})^{\mathrm{T}}(\mathbf{A}_{\text{ca}}\mathbf{J}\underline{\mathbf{X}}) \leq 0 \tag{22}$$

where $\mathbf{J}$ is the transformation matrix to retrieve the positional components in the LVLH frame starting from ROE, according to the formulation in [3], $\mathbf{A}_{\mathrm{CA}}$ is a matrix that takes the diference between every spacecraft couple in the formation, and $\mathbf{A}_{\mathrm{ss}}$ is a matrix that computes the sum of squares of the positional components, three by three. In order to convexify the nonconvex expression, the following Jacobian is defined:

$$\hat{C} = \nabla_x s(\bar{\underline{\mathbf{X}}}) \tag{23a}$$

So that the convex approximation of $g_{pc}(\mathbf{x}, p)$ can be expressed as:

$$\breve{g}_{pc}(\underline{\mathbf{X}}) = h_\lambda\left(s(\bar{\underline{\mathbf{X}}}) + \hat{C}\delta\underline{\mathbf{X}}\right) \tag{24}$$

On the other hand, the term penalizing trust region violation to limit artificial unboundedness is added as:

$$g_{tr}(\mathbf{U}) \triangleq h_\lambda\left(\|\delta\underline{\mathbf{U}}\|_\infty - \eta\right) \tag{25}$$

where $\delta\underline{\mathbf{U}}$ defines the variation of the control profile between two consecutive iterations of the SCP algorithm. Now the full convex expression of the OCP cost function can be written by integrating in the $[0,1]$ time interval as:

$$L_\lambda(\underline{\mathbf{X}}, \underline{\mathbf{U}}) = \Gamma(\underline{\mathbf{U}}) + \breve{g}_{pc}(\underline{\mathbf{X}}) + g_{tr}(\underline{\mathbf{U}}) \tag{26}$$

3.2.2 Iterative update strategy

Being a Sequential Convex Programming routine, GuSTO starts from an arbitrary initial guess and needs an iterative update procedure to converge to the final optimal solution. This procedure is thoroughly described in [13], whereas here only a basic description is provided for conciseness. Whenever an iteration takes place, the algorithm checks if the trust region is violated and, if this is the case, it proceeds by discarding the result and increasing the penalty violation coefficient λ. On the other hand, if the solution lies in the trust region, a term $\rho \in (0,1)$ is computed to measure the discrepancy between the nonlinear original problem formulation and the convexified one with respect to the current iteration solution $(\underline{\mathbf{X}}^*, \underline{\mathbf{U}}^*)$:

$$\rho \triangleq \frac{|\mathscr{J}(\underline{\mathbf{X}}^*, \underline{\mathbf{U}}^*) - L_\lambda(\underline{\mathbf{X}}^*, \underline{\mathbf{U}}^*)|}{|L_\lambda(\underline{\mathbf{X}}^*, \underline{\mathbf{U}}^*)|} \tag{27}$$

where the nonlinear augmented cost $\mathscr{J}(\underline{\mathbf{X}}^*, \underline{\mathbf{U}}^*)$ in the numerator is a temporally discretized version of the nonlinear expression of L_λ. The algorithm then proceeds comparing ρ to two user-defined constants $\rho_0, \rho_1 \in (0,1)$: if $\rho \leq \rho_0$ the convexification accuracy is deemed good, the trajectory is kept and used as the next initial guess, and the trust region can be expanded by a factor β_{gr}; if $\rho_0 < \rho \leq \rho_1$, the convex approximation is still considered satisfactory and the trajectory is stored, but the trust region is kept unchanged; finally, if $\rho > \rho_1$, the convexification not considered reliable, the trajectory is discarded, and the trust region is reduced by a factor β_{sh}. After this first update, a secondary step is needed, since GuSTO convergence relies on the fact that the trust region eventually shrinks to zero [12]. However, when the algorithm starts converging to closer and closer trajectories, the convexification accuracy increases and the trust region instead keeps becoming larger. Therefore, an exponential update is included at the end to ensure that the trust region eventually starts shrinking:

$$\eta \leftarrow \mu^{[1+k-k_*]^+}\eta \tag{28}$$

where $\mu \in (0,1)$ is the exponential shrink coefficient and k_* is a parameter which allows to decide when to stop the exploration phase and start shrinking the trust region.

3.2.3 Parameter selection

A key guideline in MPC design is to make the prediction horizon (PH) as large as possible, until further increases yield no significant performance improvements. Conversely, the sampling time and control horizon (CH) should be kept small to enhance the controller's accuracy and convergence. Additionally, minimizing computational time is crucial for onboard implementation, which requires using a smaller PH, larger CH, and longer sampling time, resulting in smaller optimization problems and less frequent recalculations. All these factors, along with simulation results, were considered during the selection of the MPC parameters for this application. The selected values are reported in Table 1 together with the parameters used to tune GuSTO SCP routine.

Table 1: Selected parameters for MPC design.

Parameter	**Value**
Sample time, Δt	100 s
Prediction Horizon, PH	2900 s
Control Horizon, CH	200 s
ρ_0	0.1
η_M	1000
β_{sh}	1.2
β_{gr}	1.2
λ_0	1
γ_{fail}	5
μ	0.85
k_*	3
$\lambda_{\max}$	10^{12}

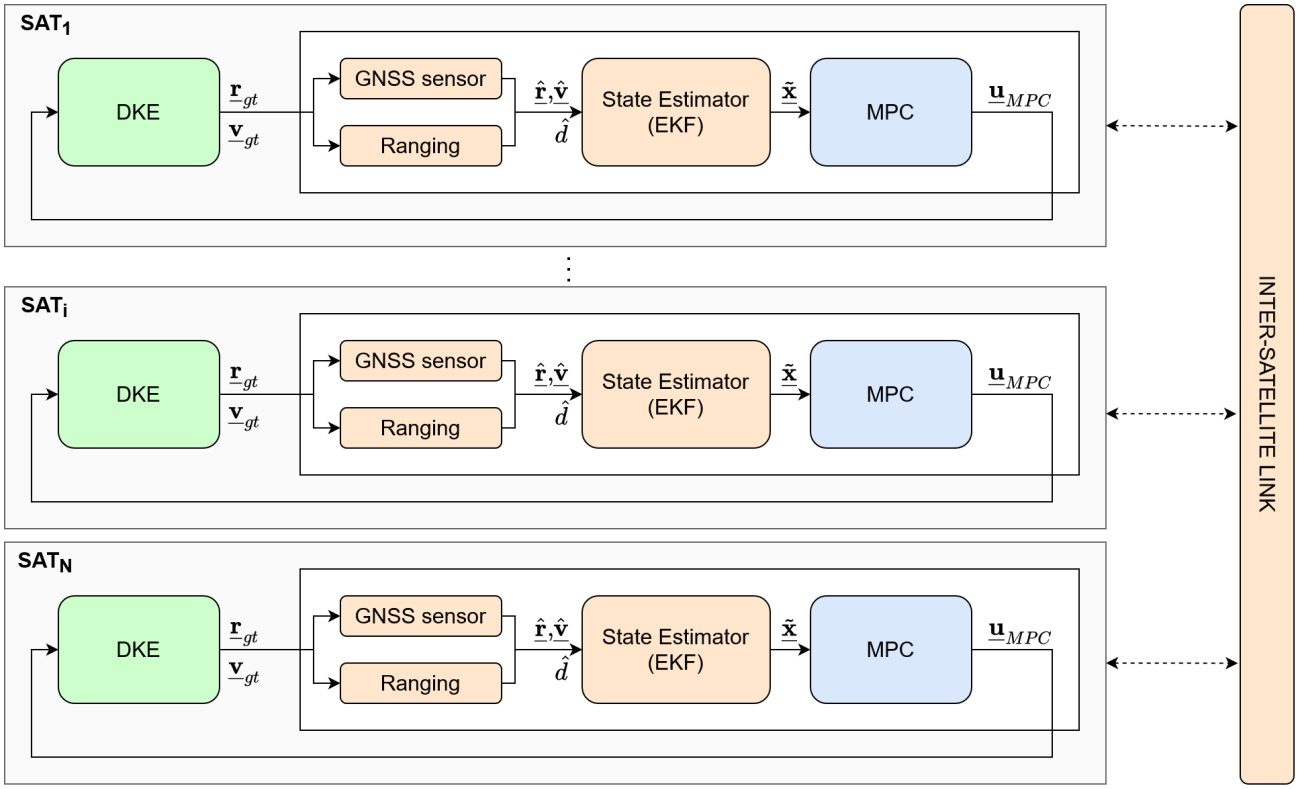

Fig. 2: Simplified pipeline of the simulation environment.

4. Simulations and results

The performance of the algorithm is tested numerically in Simulink in loop with a high-fidelity orbital propagator, considered as ground-truth for the evolution of the dynamics. The measurements are then computed from the ground truth adding a noise profile according to the sensor. Then, the measurements enter the previously described Extended Kalman Filter that computed the state estimation input to the guidance generation routine. A flowchart of the used pipeline is represented graphically in Figure 2. The mean keplerian orbital elements of the reference orbit of the formation are reported in Table 2. The reference trajectory is a 500 km frozen Sun-synchronous orbit.

Table 2: Reference mean keplerian parameters of the simulated formation.

h	e	i	ω
500 km	0.001	97.05°	90°

The physical properties of the spacecraft that are used in all simulations resemble the ones of modern CubeSats, to put an emphasis on possible application of these algorithms on nanosatellites. These are reported in Table 3.

Table 3: Assumed spacecraft and engine properties.

Parameter	**Value**
Mass	20 kg
Drag area	0.1 m^2
SRP area	0.1 m^2
CD	2.1
Max thrust	1 mN
Specific impulse	550 s

Simulations are performed for both formation maintenance and acquisition scenarios, in order to compare the results and assess differences in achievable performance. In both cases three navigation scenarios are analyzed, according to what measurements, modelled as previously anticipated, are included into the observer. In the first case, only exchanged GNSS information is exploited in the filter, differentiating the absolute measurements in the ECI frame in order to recover the relative orbital elements. In the sec-

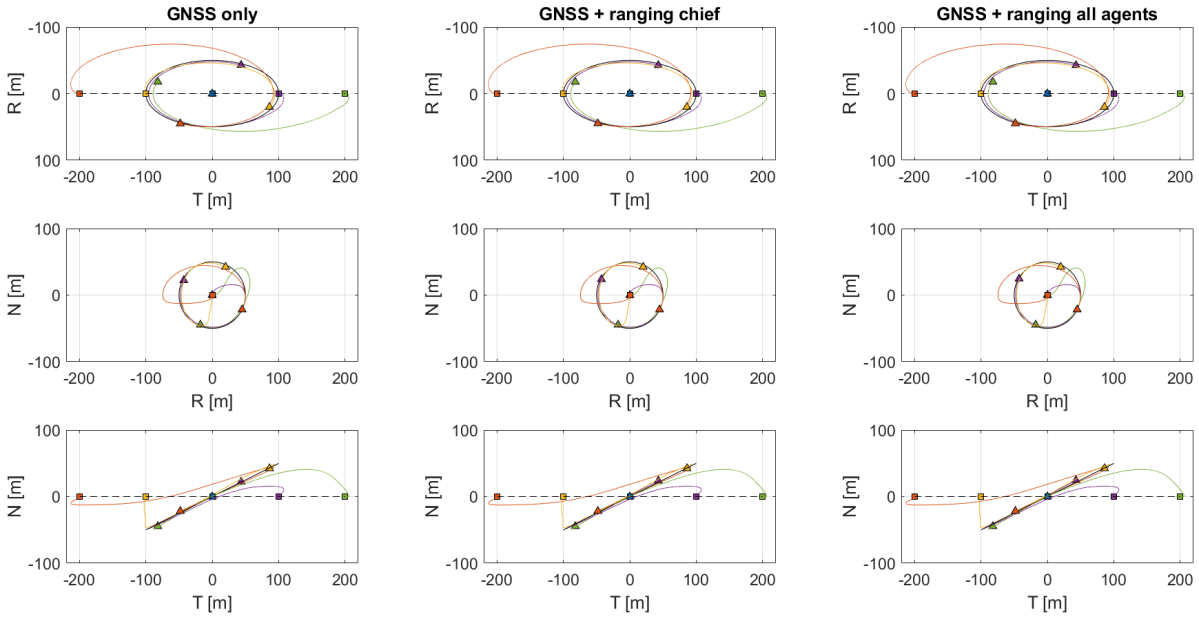

Fig. 3: Trajectory evolution during formation acquisition.

ond case, on the other hand, the range measure with respect to the reference is also introduced in the filter. Finally, in the third navigation scenario, the ranges with respect to all other formation agents are used to estimate the position of each spacecraft in the chief-centered reference frame.

4.1 Formation acquisition

The formation acquisition scenario is designed to simulate a configuration and geometry shift from an in-line train formation to a disposition in which the four satellites move to an out-of-plane periodic holding orbit around the reference. The starting and target relative orbital elements for the simulation are reported in Table 4. To ensure collision avoidance, during the transfer, the spacecraft are not allowed to enter a spherical zone of 30 meters around each other satellite.

Table 4: Starting and target relative states for tetrahedron formation acquisition and maintenance.

	Sat A	**Sat B**	**Sat C**	**Sat D**
$a\delta a$ [m]	$0 \to 0$	$0 \to 0$	$0 \to 0$	$0 \to 0$
$a\delta\lambda$ [m]	$-200 \to 0$	$-100 \to 0$	$100 \to 0$	$-200 \to 0$
$a\delta e_x$ [m]	$0 \to 50$	$0 \to 0$	$0 \to -50$	$0 \to 0$
$a\delta e_y$ [m]	$0 \to 0$	$0 \to 50$	$0 \to 0$	$0 \to -50$
$a\delta i_x$ [m]	$0 \to 50$	$0 \to 0$	$0 \to -50$	$0 \to 0$
$a\delta i_y$ [m]	$0 \to 0$	$0 \to 50$	$0 \to 0$	$0 \to -50$

The results of the formation acquisition scenario show a fast converge rate of the Sequential Convex Programming routine, capable of bringing the satellites towards the nominal configuration in about one orbit, with a low and almost equally distributed values of fuel consumption. Naturally, the satellites placed initially at the extreme points of the formation need a slightly larger ΔV to reach the target. The values of ΔV correspondent to each agent are reported in Table 5.

Table 5: Required ΔV for the four satellites to acquire the desired formation geometry.

	ΔV case 1	ΔV case 2	ΔV case 3
Sat A	0.1369 m/s	0.1372 m/s	0.1372 m/s
Sat B	0.1188 m/s	0.1189 m/s	0.1189 m/s
Sat C	0.1107 m/s	0.1152 m/s	0.1137 m/s
Sat D	0.1214 m/s	0.1212 m/s	0.1213 m/s

It can be seen from the previous image and tables, that in all navigation scenarios, the desired formation can be acquired with sufficient accuracy while satisfying the included collision avoidance constraint. What is of interest however is the different levels of navigation accuracy that are reached during the evolution of the formation geometry. This is represented for each case in Figure 4. First of all, it can be noticed how the inclusion of the range in the observer strongly

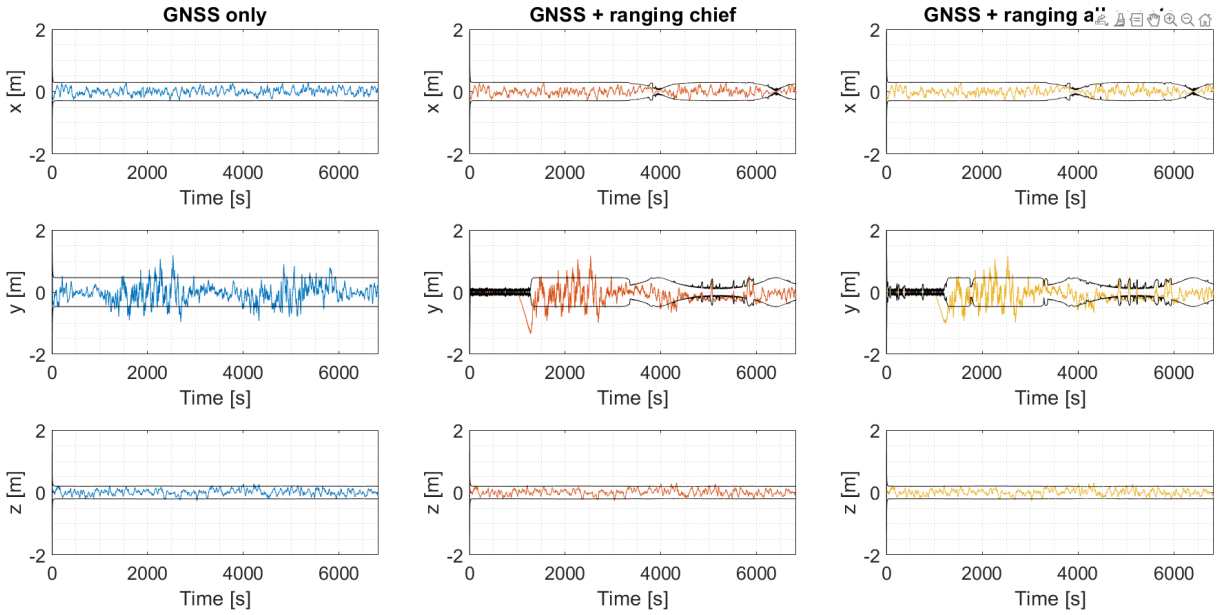

Fig. 4: Navigation accuracy during formation acquisition.

increases the accuracy in the range direction. This is mostly evident at the beginning of the transfer, when the satellites are still placed in a line formation, with a noticeable error shrink in the tangential y direction of the LVLH frame. The accuracy increase is however lost during long firing windows, in particular in correspondence of normal thrusting accelerations. The accuracy is then recovered at the end of the transfer, when control actions are small and more sparse. In this case, however, the increase in accuracy is periodically distributed in radial and tangential direction, according to the geometry of the target holding orbit. Nevertheless, the navigation scenario exploiting the range measurements with respect to all agents does not show any substantial improvement with respect to the case in which only the range with respect to the reference is exploited. This is due to the filter architecture described previously. More in detail, the range with the reference can be exploited with far more certainty since the chief is placed exactly in the origin of the reference frame by definition, i.e., its position is known with certainty. On the other hand, to exploit the range information with respect to all other agents, their position information is fundamental to correctly locate the direction of the range in three-dimensional space. Thus, the accuracy that the measurement is able to provide does not correspond to the range measure accuracy, but to the much more inaccurate position determination accuracy of the agents. A tentative solution was to include an adaptive covariance scheme described in the previous sections, however, this countermeasure proved to be insufficient in improving the navigation solution in the multi-agent case.

4.2 Formation maintenance

Using the same algorithmic framework, also formation maintenance for long periods of operations can be automatically achieved without further tuning nor implementation of additional measures. The trajectory evolution in the reference-centered LVLH frame during the two orbits following formation acquisition is displayed in Figure 5. In the graphs, it can be noticed how the spacecraft are able to track the target holding orbit precisely over the simulated time span. As for the previous example, it however more interesting to have a look at the achievable position estimation accuracy during the maneuvers, which is shown instead in Figure 6. Also in this case, the same behaviour that was noticed for the last fraction of the formation acquisition simulation can be observed. Indeed, an accuracy increase due to the range inclusion in the filter is again visible and distributes periodically in radial or tangential direction according to the evolution of the formation geometry. Even in this simulation case, however, the navigation solution cannot be improved by including multi-agent range measurements with respect to including only the range with respect to the central reference, due to the reasons reported in the previous section.

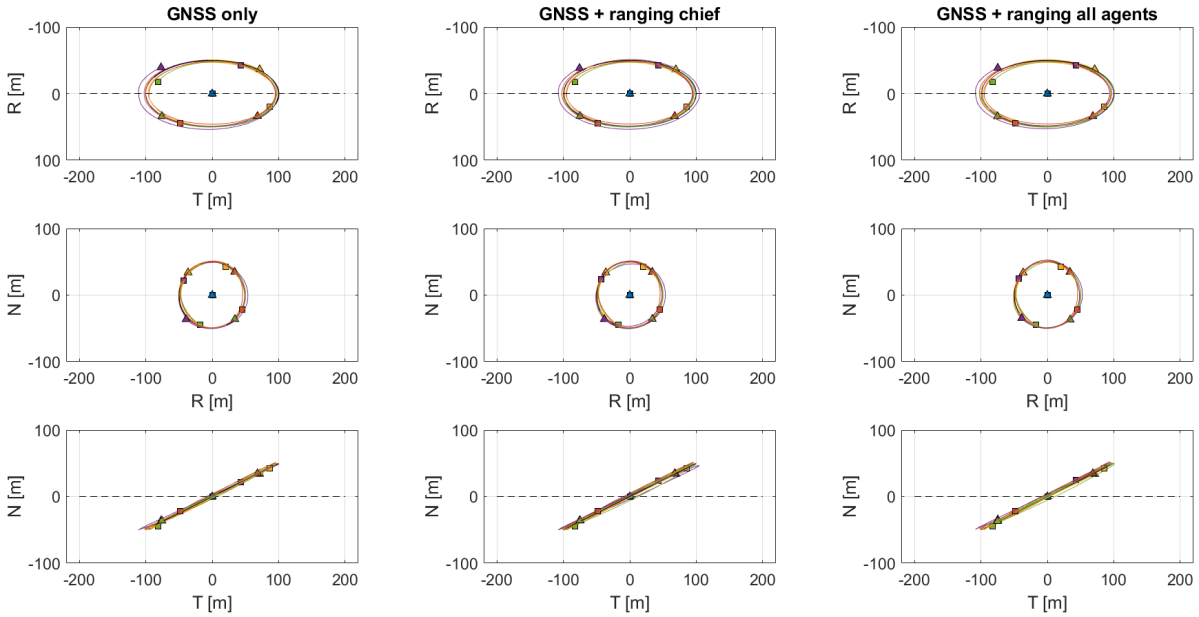

Fig. 5: Trajectory evolution during formation maintenance.

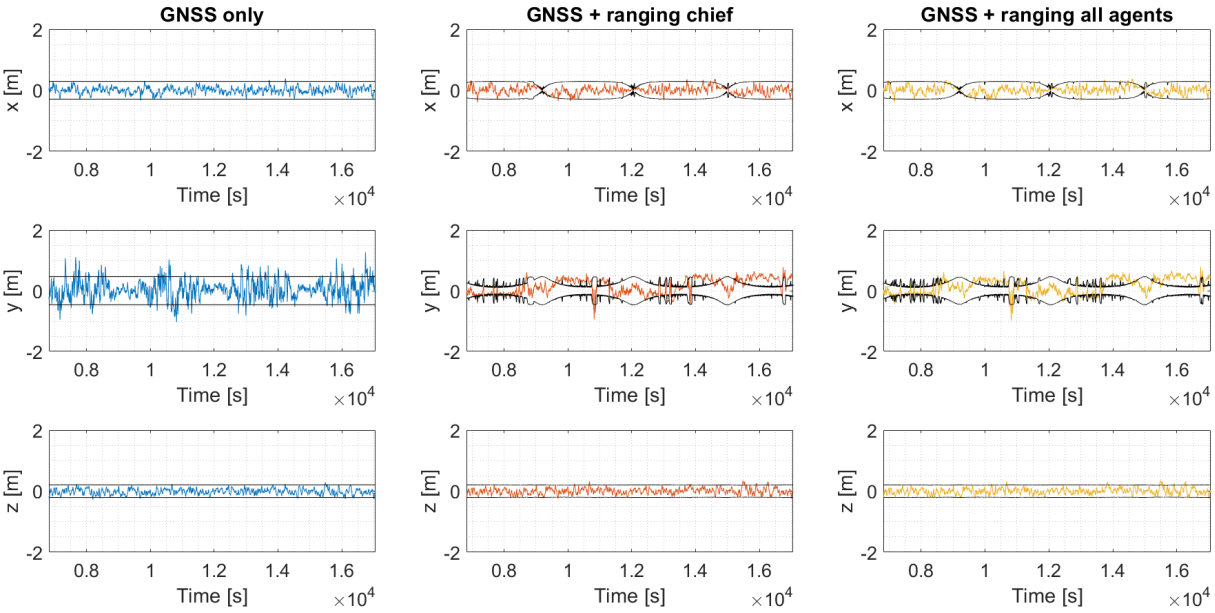

Fig. 6: Navigation accuracy during formation maintenance.

5. Conclusions

This paper presents an innovative approach to the guidance, navigation, and control of spacecraft formations, focusing on distributed algorithms to address the increasing complexity of future multi-agent missions. By employing model predictive control with Sequential Convex Programming (SCP) and radio-frequency navigation through inter-satellite links, the proposed strategy effectively handles formation acquisition maintenance in a cooperative

and autonomous manner. The guidance and control methods developed with SCP generate optimal control solutions that consider both mission objectives and safety constraints while being computationally efficient. Furthermore, the navigation system based on a radio-frequency network and code division multiple access architecture is capable of providing reliable relative state estimation increasing the performance of small GNSS sensors for precise positioning tasks. The proposed design optimizes computational effort while maintaining high performance and addressing computational burden on flight hardware. Robustness assessments via high-fidelity simulations further validated the reliability of the proposed methods and strategy. In conclusion, this work highlights the potential of distributed, autonomous methods for future space missions, contributing to the development of scalable and efficient guidance, navigation, and control systems for large spacecraft formations. Future work could focus on further optimizing the scalability of the algorithm by spreading the computational effort on multiple agents and increasing the robustness in more challenging space environments.

References

[1] S. D'Amico, "Autonomous formation flying in low earth orbit," Ph.D. dissertation, TU Delft, 2010.

[2] A. W. Koenig, T. Guffanti, and S. D'Amico, "New state transition matrices for spacecraft relative motion in perturbed orbits," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 7, pp. 1749–1768, 2017. DOI: 10.2514/1.G002409. [Online]. Available: https://doi.org/10.2514/1.G002409.

[3] E. Belloni, S. Silvestrini, J. Prinetto, and M. Lavagna, "Relative and absolute on-board optimal formation acquisition and keeping for scientific activities in high-drag low-orbit environment," *Advances in Space Research*, 2023, ISSN: 0273-1177. DOI: https://doi.org/10.1016/j.asr.2023.07.051. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0273117723006002.

[4] W. H. Clohessy and R. S. Wiltshire, "Terminal guidance system for satellite rendezvous," *Journal of the Aerospace Sciences*, vol. 27, no. 9, pp. 653–658, 1960. DOI: 10.2514/8.8704. [Online]. Available: https://doi.org/10.2514/8.8704.

[5] T. Guffanti, S. D'Amico, and M. Lavagna, "Long-term Analytical Propagation of Satellite Relative Motion in Perturbed Orbits," *27th AAS/AIAA Space Flight Mechanics Meeting*, Feb. 2017.

[6] L. Steindorf, S. D'Amico, J. Scharnagl, F. Kempf, and K. Schilling, "Constrained Low-Thrust Satellite Formation-Flying Using Relative Orbit Elements," *27th AAS/AIAA Space Flight Mechanics Meeting*, Jan. 2017.

[7] F. De Cecio, E. Belloni, G. Zanotti, M. Lavagna, C. Pirat, and G. Leccese, "Enabling on-board relative ranging with commercial off-the-shelf software-defined radios: The vulcain mission inter-satellite iod," in *Proceedings of the International Astronautical Congress, IAC24*, vol. 2024-October, 2024.

[8] D. Brouwer, "Solution of the problem of artificial satellite theory without drag," *The Astronomical Journal*, vol. 64, p. 378, 1959. [Online]. Available: https://api.semanticscholar.org/CorpusID:121292445.

[9] W. Zhong and P. Gurfil, "Mean orbital elements estimation for autonomous satellite guidance and orbit control," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 6, pp. 1624–1641, 2013. DOI: 10.2514/1.60701. [Online]. Available: https://doi.org/10.2514/1.60701.

[10] G. Gaias, C. Colombo, and M. Lara, "Accurate osculating/mean orbital elements conversions for spaceborne formation flying," in *27th International Symposium on Space Flight Dynamics (ISSFD)*, Feb. 2019.

[11] E. Belloni and M. Lavagna, "Formation keeping in very low-earth orbit: The vulcain mission case study," in *Proceedings of the International Astronautical Congress, IAC23*, Cited by: 0, vol. 2023-October, 2023. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85188004284&partnerID=40&md5=98f2c8d12a5cb8270de508019015e590.

[12] R. Bonalli, A. Cauligi, A. Bylard, and M. Pavone, "Gusto: Guaranteed sequential trajectory optimization via sequential convex programming," May 2019, pp. 6741–6747. DOI: 10.1109/ICRA.2019.8794205.

[13] D. Malyuta, T. P. Reynolds, M. Szmuk, *et al.*, "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. DOI: 10.1109/MCS.2022.3187542.