Distributed Convex Guidance Generation for Multi-Agent Mapping around Small Bodies

Enrico Belloni[1], Michèle Lavagna[2]

[1]*Politecnico di Milano*
Milan, Italy
Email: enrico.belloni@polimi.it

[2]*Politecnico di Milano*
Milan, Italy
Email: michelle.lavagna@polimi.it

Abstract **– This paper proposes a novel distributed Model Predictive Control approach for optimal mapping trajectory computation of multi-agent spacecraft formations around small bodies. Due to the nonlinear nature of imaging objectives, dynamics constraints, and collision avoidance, the proposed framework employs Sequential Convex Programming (SCP) using a trust-region solver with convergence guarantees. Each spacecraft solves a local SCP subproblem based on its own dynamics while coordinating shared trajectory guesses through parallel (Jacobi) or sequential (Gauss-Seidel) information exchange topologies. This goal-coupled formulation enables cooperation between the agents to achieve efficient and safe collaborative mapping without the computational bottleneck of a centralized solver. The improvement in performance is benchmarked against a Sun-Stabilized Terminator Orbit (SSTO), the typical approach to this type of mission. Monte Carlo results show consistent improvements across all runs, randomizing a set of points to be observed on the surface.**

I. INTRODUCTION

The exploration of small celestial bodies, such as near-Earth asteroids and comets, presents a unique set of astrodynamics and control challenges. Missions like OSIRIS-Rex [1] and Hayabusa2 [2] have demonstrated the high scientific value of close-proximity operations, but they historically relied on extensive ground-in-the-loop navigation and highly conservative, pre-planned flight paths. As the aerospace sector moves toward deploying swarms of resource-constrained SmallSats for increased return-to-cost ratio, relying on centralized ground control or single-node computation becomes a critical bottleneck. Multi-agent spacecraft formations offer distinct advantages for the mapping and characterization of highly irregular bodies. A distributed swarm can simultaneously acquire multi-angle observations, drastically reducing the time required to build high-fidelity shape and terrain reconstruction, while avoiding the self-occlusions inherent to complex asteroid geometries. However, real-time optimal guidance in this environment requires taking into account highly perturbed gravitational fields, irregular rotating obstacle geometries, and strict inter-agent collision avoidance (CA). To enable deterministic and sub-optimal trajectory generation onboard computationally limited flight units, a Model Predictive Control (MPC) approach based on Sequential Convex Programming (SCP) is selected. In contrast to reinforcement learning algorithms [3], which fail to explicitly exploit system dynamics models, or global and sampling-based optimization techniques [4], which are computationally prohibitive for embedded hardware, SCP provides a viable real-time solution. Standard centralized MPC frameworks would however also fail to scale to multi-agent swarms. Solving the dense Karush-Kuhn-Tucker (KKT) conditions for an N-dimensional problem may result in a worst-case computational complexity of $O(N^3)$ using Interior Point Methods [5], strictly precluding real-time onboard execution. This paper addresses this gap by introducing a distributed SCP guidance architecture. By distributing the computational burden, each agent solves a localized optimal control problem onboard based on the currently available information from the network. Through the implementation of Jacobi and Gauss-Seidel network topologies combined with SCP, the swarm achieves globally cooperative mapping behaviours while guaranteeing inter-agent collision avoidance.

II. SYSTEM MODELLING AND CONSTRAINTS

A. Orbital Dynamics and Perturbations

The motion of the spacecraft swarm is modelled in the body-fixed frame of the target asteroid to directly evaluate the geometry of surface features. For small bodies like Eros and Bennu, the gravitational field is often highly non-spherical. In this work, the acceleration a_{grav} is computed using the polyhedron gravity model [6], evaluating the exact potential field generated by the asteroid's triangular facet shape model. Furthermore, Solar Radiation Pressure (SRP) is included, as this represents a dominant perturbation in the micro-gravity environment of asteroids [7]. The continuous nonlinear dynamics of the i-th agent are defined as:

$$\dot{x}_i = f(x_i, u_i) = \begin{bmatrix} v_i \\ a_{grav}(r_i) - 2\omega \times v_i - \omega \times (\omega \times r_i) + a_{srp} + u_i \end{bmatrix} \quad (1)$$

Where $x_i = [r_i, v_i]$ is the state vector, ω is the asteroid's spin vector, and u_i is the control thrust subject to the actuator saturation bound $\|u_i\| \leq U_{max}$.

B. Mapping Objective Derivation

The primary mission objective is the rapid, high-quality observation of a set of P geological landmarks distributed across the surface of the target. For an observation to be considered valid for optical mapping, strict geometric constraints must be satisfied simultaneously:

1. *Line of Sight (LoS)*: The landmark must not be occluded by the asteroid's local topography.

2. *Illumination*: The Sun phase angle and local incidence angle must fall within specific bounds to guarantee sufficient surface contrast and prevent camera blinding.

3. *Range*: The distance to the target must satisfy the camera's resolution and focus criteria, $R_{min} \leq \|r_i - r_p\| \leq R_{max}$.

These conditions are aggregated into a highly non-convex, scalar mapping cost field $J_{map}(r_i)$. The objective of the guidance algorithm is to maximize exposure to regions where J_{map} is minimized. Following the work in [8], a continuously differentiable cost function is derived, to be able to use Sequential Convex Programming as a tool for fast and reliable guidance computation on spaceflight hardware. The objective is based on the Squareplus function, a quadratic and C^{∞} version of the Rectified Linear Unit (ReLU) and its derivative:

$$\phi(x) = \frac{x + \sqrt{x^2 + \varepsilon}}{2} \tag{2a}$$

$$\phi'(x) = \frac{1}{2}\left(1 + \frac{x}{\sqrt{x^2 + \varepsilon}}\right) \tag{2b}$$

For each mapping point P_i, based on the current state x in the target body-fixed reference frame, the correspondent set of imaging conditions $\left(\vartheta_i(x), \varphi_i(x), d_i(x)\right)$ can be found and the following four cost functions can be defined:

$$\phi_{i,1}(x) = \phi\left(\cos(\vartheta_{max}) - \cos\left(\vartheta_i(x)\right)\right) \tag{3a}$$

$$\phi_{i,2}(x) = \phi\left(\cos(\varphi_{max}) - \cos\left(\varphi_i(x)\right)\right) \tag{3b}$$

$$\phi_{i,3}(x) = \phi\left(d_i(x) - D_{max}\right) \tag{3c}$$

$$\phi_{i,4}(x) = \phi\left(D_{min} - d_i(x)\right) \tag{3d}$$

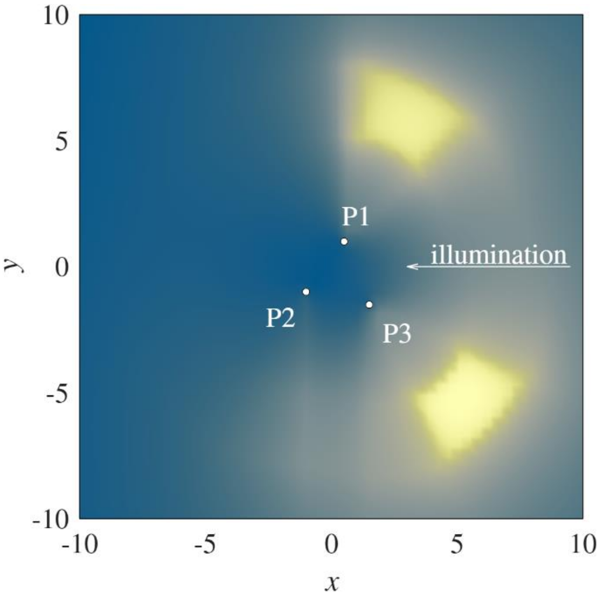

Fig. 1. Simplified 2D representation of the imaging cost function in logarithmic scale.

Where $\vartheta_i(x)$ is the off-nadir angle with respect to the local surface, $\varphi_i(x)$ is the Sun phase angle, and $d_i(x)$ is the distance of observation.

Then, for each surface point i, the local penalty can be defined as a weighted sum of each imaging constraint:

$$g_i(x) = w_1\phi_{i,1}(x) + w_2\phi_{i,2}(x) + w_3\phi_{i,3}(x) + w_4\phi_{i,4}(x) \tag{4}$$

The multiplicative total imaging cost is then computed as the product of the contribution of each surface point P as:

$$J_{map} = \prod_{i=1}^{P} g_i(x) \tag{5}$$

A multiplicative approach is preferred with respect to an additive one, in order to have $J_{map} = 0$ at a given timestep if the spacecraft is located in any desired imaging region. A simplified 2D representation of the mapping cost function is shown in Fig. 1 in logarithmic scale, for a clearer visualization.

The mapping cost function is then added to a quadratic term for the control effort to obtain the full objective of the Optimal Control Problem (OCP) as:

$$J = \frac{1}{2}u^T u + J_{map} \tag{6}$$

C. Collision Avoidance and Keep-Out Zones

For a swarm of N agents, inter-agent collisions must be strictly avoided. For any pair of agents (i, j), the spatial separation must be bounded by a minimum safety distance d_{min}:

$$\left\|r_i(t) - r_j(t)\right\|_2 \geq d_{min}, \quad \forall t, \forall i \neq j \tag{7}$$

Additionally, agents must not violate the primary Keep-Out Zone (KOZ) enclosing the asteroid surface plus a desired safety threshold. This constraint is formalized as $\|r_i(t)\| \geq R_{KOZ}$. Clearly, in both cases the collision avoidance constraint represents a nonconvex region in the state space. This is handled via linearization in the optimization scheme described in the following section.

III. Distributed MPC Implementation

To embed the optimization into the flight software of low-power computational architectures, the nonlinear optimal control problem is discretized and transcribed into a sequence of convex subproblems. The core of this framework is based on the Guaranteed Sequential Trajectory Optimization (GuSTO) algorithm [9], which utilizes a dynamic trust region to guarantee convergence to local optima.

A. *Local Linearization and Trust Region*

At each SCP guidance iteration k, the nonlinear dynamics and non-convex mapping costs are linearized around a reference trajectory $\bar{x}_i^{(k)}$ via first-order Taylor expansion. In GuSTO, the collision avoidance constraint is handled via augmenting the cost function, similarly to what is done in Augmented Lagrangian methods [10], by using a penalty function $h_\lambda(z) = \mathrm{ReLU}(z)$ that penalises constraint violation while being agnostic to nonpositive values. To ensure the artificial linearization remains physically valid, an $\mathcal{L}_\infty$ trust region bounds the maximum deviation. This term is also added to the cost function as done for the nonconvex path constraints. The full objective then reads:

$$J = \frac{1}{2}u^T u + J_{map} + g_{tr}(x) + g_{pc}(x) \tag{8}$$

The Jacobians evaluated at the reference trajectory are defined as $A_k^g = \nabla_x J_{map}(\bar{x}_k)$, which is computed analytically via the chain rule as in [8], and the geometric constraint Jacobian $\hat{C}_k$. For any obstacle, whether the central asteroid fixed at the origin, or a neighbouring agent j at predicted position $\bar{r}_{j,k}$, the generalized geometric constraint Jacobian with respect to agent i evaluates to $\hat{C}_k = \left[-2(\bar{r}_{i,k} - \bar{r}_{obst,k}) \mid 0_{1\times 3}\right]$. By discretizing the computation over the trajectory with a zero-hold approximation over K nodes with a step size of Δt, the fully assembled, convex subproblem cost function is formulated as:

$$L(x,u) = \Delta t \sum_{k=1}^{K} \frac{1}{2} u_k^T u_k + J_{map}(\bar{x}_k) + A_k^g\,\delta x_k + h_\lambda(\|x_k\|_\infty - \eta) + s_{koz}(\bar{x}_k) + \hat{C}_k \delta x_k \tag{9}$$

The trust-region size η is dynamically adjusted based on the convexification accuracy ratio between the predicted linear cost reduction and the actual nonlinear cost reduction [9]. Then, after an initial exploration, the trust region is exponentially reduced to ensure convergence. On the other hand, the linearized dynamics evolution constraint to obtain feasible trajectories is modelled via State Transition Matrix propagation as:

$$x_{k+1} = \bar{x}_{k+1} + \Phi_k \delta x_k + \Gamma_k \delta u_k \tag{10}$$

Where $\delta x_k = x_k - \bar{x}_k$ and $\delta u_k = u_k - \bar{u}_k$ represent the differences between the new solution and the previous guess.

B. *Strategies for Distributed Information Exchange*

While the inner SCP loop efficiently solves the trajectory for a single agent, the swarm must coordinate to ensure optimal global coverage and avoid inter-agent collisions. In a distributed architecture, agents treat the predicted trajectories of their neighbours as dynamic obstacles. The method by which these predicted trajectories are shared and updated dictates the stability and convergence speed of the entire swarm. This paper analyses two primary information exchange topologies: Jacobi and Gauss-Seidel.

Jacobi (Parallel) Update

In the Jacobi update scheme, all agents compute their optimal trajectories simultaneously. During the outer distributed iteration p, agent i optimizes its current trajectory r_i^p by assuming the broadcasted trajectories of all other agents from the previous iteration $r_j^{(p-1)}$.

The primary advantage of the Jacobi topology is the possibility of full computational parallelization; the wall-clock time for a swarm iteration update is equivalent to a single agent's computation time. However, because all agents react to outdated information simultaneously, this topology is highly susceptible to oscillatory behaviour. If two agents are on a collision course, they may both symmetrically alter their trajectories to dodge into the same adjacent space (the *flip-flop* effect), requiring many iterations to settle or failing to converge entirely.

Gauss-Seidel (Sequential) Update

To guarantee mathematical stability and eliminate symmetric evasion deadlocks, the Gauss-Seidel topology enforces a strict hierarchical sequence. Agents are assigned an execution order (e.g., based on ID or available fuel).

Agent i computes its update at iteration p using the already calculated states of agents $j < i$ from the current iteration p, and the older states of agents $m > i$ from the previous iteration $p-1$. The dynamic collision avoidance constraint for Agent i is thus updated asymmetrically:

- Against Agent j (where $j < i$): Uses r_j^p
- Against Agent m (where $m > i$): Uses r_m^{p-1}

By immediately incorporating new trajectory information, the Gauss-Seidel approach breaks the evasion symmetry. Once Agent 1 decides to dodge *left*, Agent 2 sees that decision immediately and plans to dodge *right*. This inherently damps oscillations and provides a mathematically stable convergence path toward a collision-free swarm trajectory, albeit at the cost of sequential execution delays.
In both topologies, numerical convergence can be improved by warm-starting the distributed optimization with the trajectory guess from the preceding iteration. Additionally, the initial trust region of the internal SCP loop is progressively tightened after a predefined number of iterations, effectively restricting the search space and forcing the algorithm to converge.

IV. Numerical Simulations and Results

A. Simulation Setup

To validate the operational advantages of the proposed distributed guidance algorithms, extensive numerical simulations were performed in the highly perturbed environment of asteroid Eros and the smaller asteroid Bennu. The gravity fields were derived from the standard Gaskell shape models [11, 12]. The minimum inter-satellite distance was fixed to 1 km around Eros and 100 m around Bennu. The simulations assume CubeSat-like platforms, reported in Tab. 1.

Tab. 1. Summary of spacecraft physical properties used in the simulations.

	Eros	Bennu
Mass	20 kg	20 kg
Cross-section area	1 m^2	1 m^2
Reflectivity	0.7	0.7
Max thrust	100 mN	100 mN

A Monte Carlo campaign of 500 independent runs was executed. For each run, 5 geological landmarks were randomly distributed across the asteroid surface. A swarm of $N = 3$ agents was initialized to cooperatively

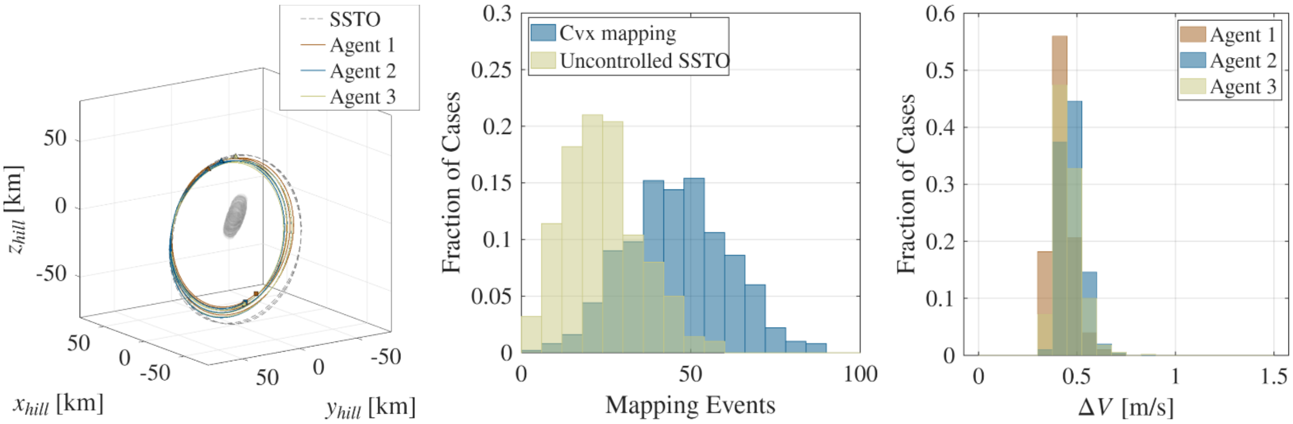

Fig. 2. Results of Monte Carlo simulations for Eros asteroid with Gauss-Seidel updates: Cooperative mapping trajectories of N=3 agents (left), and improvement of number of mapping events (centre) and ΔV distribution across all runs (right).

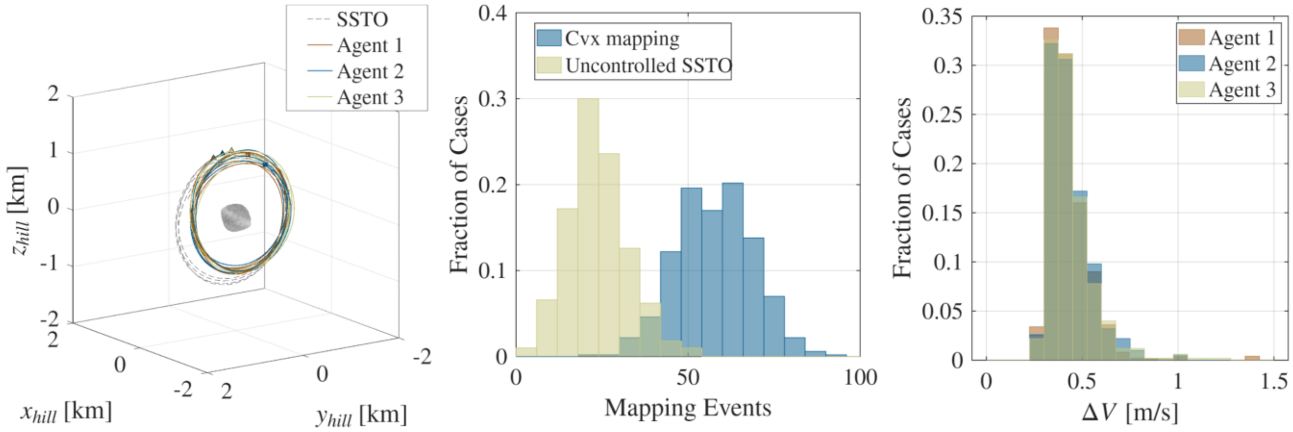

Fig. 3. Results of Monte Carlo simulations for Bennu asteroid with Gauss-Seidel updates: Cooperative mapping trajectories of N=3 agents (left), and improvement of number of mapping events (centre) and ΔV distribution across all runs (right).

observe these features. The distributed SCP optimization was benchmarked against the standard Sun-Stabilized Terminator Orbit (SSTO) over a five-day mapping horizon. The two different distribution topologies show very similar mapping performance. To obtain the following results, Gauss-Seidel topology was utilized in the Monte Carlo runs to ensure collision avoidance.

B. *Mapping Performance*

The results of the simulations are depicted in Fig. 2 and Fig. 3 for the two targets. As expected, the distributed optimization framework demonstrates a drastic improvement in mapping efficiency with respect to the baseline SSTO. Driven by the gradient of the mapping cost function, the agents successfully diverged from stable generic orbits to actively seek viewing angles over the randomized landmarks. The improvement is amplified for asteroid Bennu, for which the less perturbed environment allowed a higher degree of exploration of the agents, while maintaining a stable orbital motion around the target. The mean results obtained from the simulations are also summarized in Tab. 2, demonstrating a doubled occurrence of imaging events at a moderate requirement of ΔV.

Tab. 2. Summary of mapping results across 500 Monte Carlo simulations using Gauss-Seidel distribution.

	Eros		Bennu	
	SSTO	Cvx	SSTO	Cvx
Mean map events	23.37	**45.44**	23.29	**57.20**
Mean ΔV (per agent)	-	0.45 m/s	-	0.44 m/s

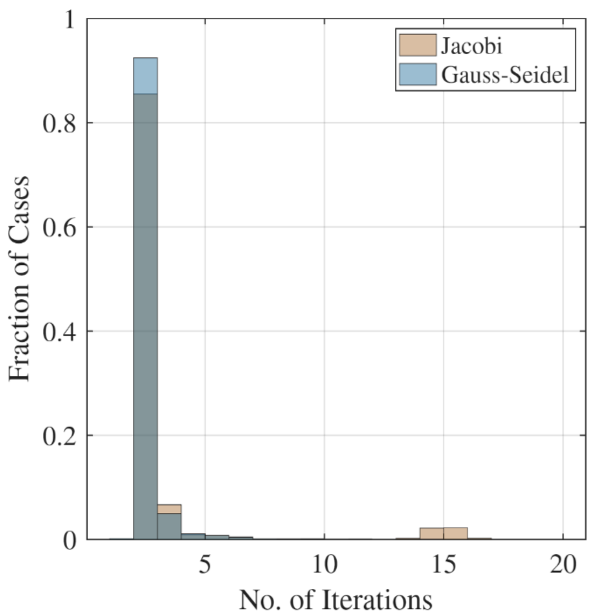

Fig. 4. Number of external distributed iterations across all Eros Monte Carlo runs for N=3 agents.

C. *Comparison of Distributed Strategies*

A sub-study was then conducted to explicitly compare the Jacobi and Gauss-Seidel solvers during close-proximity manoeuvres. As predicted by the theory, the Jacobi solver exhibits chattering behaviour in which two agents of the swarm were continuously bouncing between two solutions, failing to reach a consensus. Due to the exponential forced numerical convergence mechanism, the maximum number of Jacobi iterations was still limited to about 15, whereas Gauss-Seidel always remained below six, as shown in Fig. 4. By treating higher-priority agents as absolute constraints, lower-priority agents smoothly adapted their trajectories without mathematical windup.

The min-max envelops of inter-agent distance are instead represented in Fig. 5, showing that, with a close release, the constraint comes into play very frequently throughout the simulations, but both strategies manage to force the desired keep-out-distance up to a given tolerance given by the distributed convergence threshold, which in this case was set to 100 meters.

A custom CVXGEN C-code implementation executed the sequential optimal control updates in milliseconds in SIL simulations on a laptop with an Intel(R) Core(TM) Ultra 7 265U CPU, proving that despite the sequential nature of Gauss-Seidel, the total wall-clock time should remain well within the bounds required for real-time onboard execution even considering a 50x wall-clock time increase on SmallSat hardware.

D. *Numerical Behaviour*

As typical in Sequential Convex Programming (SCP) architectures, the proposed framework relies on numerical conditioning to ensure rapid and deterministic convergence. First of all, variable scaling is paramount,

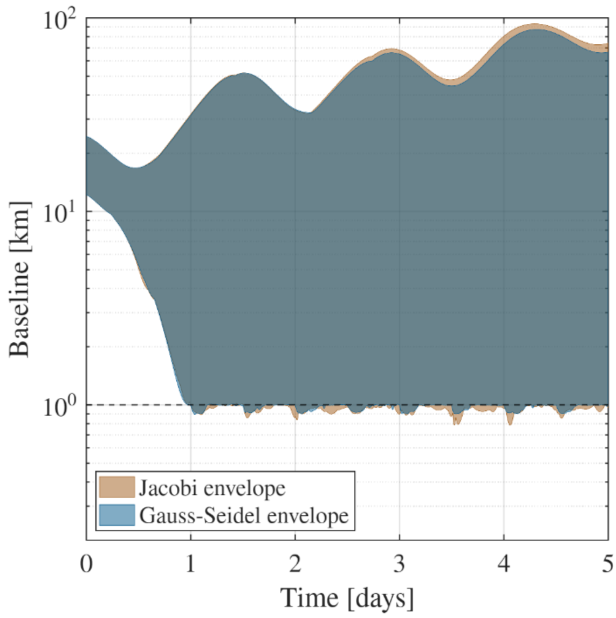

Fig. 5. Min-max envelopes of intersatellite distance evolution in time across all Eros Monte Carlo runs.

as interior-point solvers are inherently scale-dependent. Without proper normalization, the vastly disparate magnitudes of absolute Cartesian position vectors (e.g., $O(10^4)$ meters) and continuous low-thrust control accelerations (e.g., $O(10^{-3})$ m/s^2) severely degrade the condition number of the underlying Karush-Kuhn-Tucker (KKT) matrix, risking solver failure or numerical infeasibility.
Furthermore, SCP convergence usually requires careful hyperparameter tuning, particularly for trust-region update parameters and penalty weights, although empirical evaluations demonstrate that once the problem is adequately scaled, a single, well-tuned parameter set generalizes remarkably well across diverse dynamical environments.
Finally, the nature of the cost function creates a multi-objective optimization problem, placing scientific mapping returns in direct contrast with propellant saving. The resulting trajectory geometry is highly sensitive to the relative weighting between these terms. Penalizing control effort too heavily anchors the spacecraft to its initial guess (e.g., the SSTO), completely neutralizing the autonomous exploration capability. Conversely, over-prioritizing the mapping objective commands aggressive manoeuvres that could exceed the limited ΔV typical of SmallSat platforms. A Pareto front analysis was therefore conducted by sweeping the relative weighting between the imaging objective and control effort. The results of this analysis are displayed in Fig. 6. Also in this case, Gauss-Seidel distribution strategy was used to obtain the results.
As characteristic of highly non-convex problems solved via Sequential Convex Programming, the resulting front exhibits scatter. This variance is expected, as shifting the cost weights dynamically reshapes the local basins of attraction, occasionally causing the solver to converge upon fundamentally different local manoeuvre geometries. Despite this noise, a clear macroscopic trend emerges, confirming the algorithm's ability to navigate the strict trade-off space between propellant expenditure and scientific return, with diminishing returns for excessive imaging weight increase.

E. Applicability to Larger Swarms

To assess the scalability of the proposed distributed guidance framework, a dedicated simulation campaign was conducted around asteroid Bennu with a swarm of $N = 10$ agents. This configuration represents a significant increase in swarm density with respect to the three-agent case, directly stressing both the collision avoidance and the distributed convergence properties of the algorithm. The simulation setup mirrors the parameters of Section IV-A, with the minimum inter-satellite distance maintained at 100 m and the same CubeSat physical properties reported in Tab. 1.
In this case, only a simulation is run to qualitatively assess the algorithm resilience to an increasing number of agents, while a full statistical characterisation across larger swarm sizes is left as future work.
As expected, the mapping score scales favourably with the number of agents. More relevant to the scalability assessment is the convergence behaviour of the distributed system in the denser environment, which is examined through the pairwise inter-agent distance evolution represented in Fig. 7. Despite the significantly increased number of agent pairs, the algorithm maintains the 100 m keep-out constraint across all runs within the established convergence tolerance, with no appreciable increase in the number of outer Gauss-Seidel iterations with respect to the three-agent case.

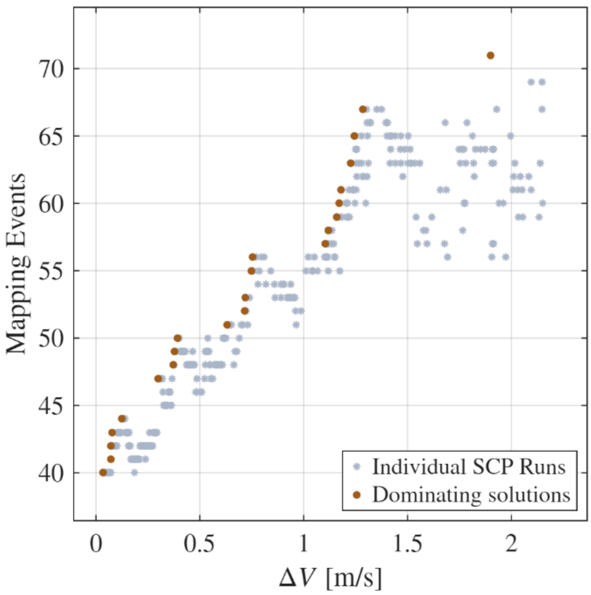

Fig. 6. Pareto front of mapping events vs. ΔV obtained varying the weight of the imaging cost in the objective.

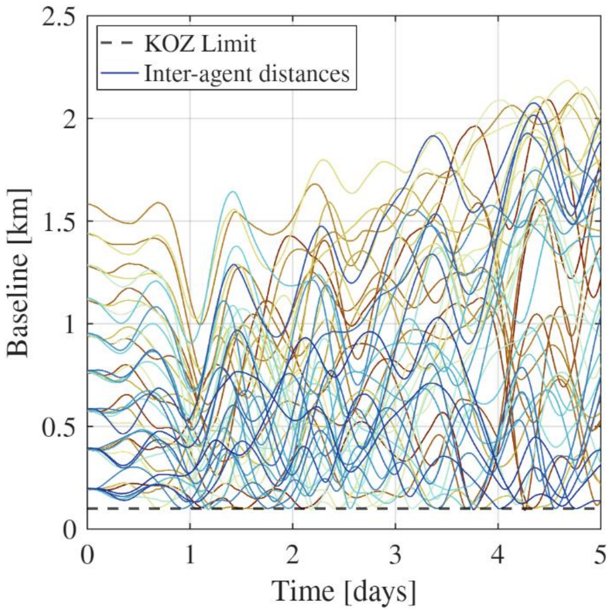

Fig. 7. Inter-satellite distance evolution around Bennu for N=10 agents.

This provides confidence that the proposed distributed framework scales gracefully to larger formations, supporting its applicability to future high-density mapping swarms.

V. CONCLUSIONS

This paper presented a distributed Model Predictive Control framework based on Sequential Convex Programming for the autonomous mapping of small celestial bodies by multi-agent spacecraft swarms. By including inter-agent collision avoidance in a cost-coupled distributed optimization framework, the proposed architecture allows spacecraft to aggressively and safely optimize their observation geometries. A structural comparison between Jacobi and Gauss-Seidel information exchange topologies demonstrated that while Jacobi offers parallelization, the sequential Gauss-Seidel approach is more reliable to guarantee stable, collision avoidance in dense swarms. Monte Carlo simulations against a standard Sun-Stabilized Terminator Orbit baseline demonstrated robust convergence and a highly significant increase in scientific mapping return, all while remaining computationally feasible for real-time onboard execution. Future developments will focus on testing the scalability of this strategy to even larger swarms, together with its resilience to inter-satellite communication failures or interruptions.

VI. REFERENCES

[1] Williams, B., Antreasian, P., Carranza, E. et al. "OSIRIS-REx Flight Dynamics and Navigation Design". *Space Sci Rev* 214, 69 (2018). https://doi.org/10.1007/s11214-018-0501-x

[2] Ono, G., Terui, F., Ogawa, N., et al. "GNC strategies and flight results of Hayabusa2 first touchdown operation". *Acta Astronautica*, 2020, 131–147. https://doi.org/10.1016/j.actaastro.2020.04.029

[3] Brandonisio, A., Capra, L., and Lavagna, M., "Deep reinforcement learning spacecraft guidance with state uncertainty for autonomous shape reconstruction of uncooperative target," *Advances in Space Research*, Vol. 73, No. 11, 2024, pp. 5741–5755. https://doi.org/10.1016/j.asr.2023.07.007

[4] Capolupo, F., Simeon, T., and Berges, J.-C., "Heuristic Guidance Techniques for the Exploration of Small Celestial Bodies", *20th IFAC World Congress*, Vol. 50, No. 1, 2017, pp. 8279–8284. https://doi.org/10.1016/j.ifacol.2017.08.1401

[5] S. Boyd and L. Vandenberghe, "Convex Optimization". Cambridge University Press, 2004.

[6] R. Werner and D. Scheeres, "Exterior Gravitation of a Polyhedron Derived and Compared with Harmonic and Mascon Gravitation Representations of Asteroid 4769 Castalia", *Celestial Mechanics and Dynamical Astronomy*, vol. 65, pp. 313–344, Sep. 1996. https://doi.org/10.1007/BF00053511

[7] D. Scheeres, "Orbital mechanics about small bodies", *Acta Astronautica*, vol. 72, pp. 1–14, 2012, issn: 0094-5765. https://doi.org/10.1016/j.actaastro.2011.10.021

[8] Belloni, E., and Lavagna, M. "Convex Guidance Generation for Target Inspection Trajectories", AIAA 2026-2445. *AIAA SciTech 2026 Forum*. January 2026. https://doi.org/10.2514/6.2026-2445

[9] Malyuta, D., Reynolds, T. P., Szmuk, M., Lew, T., Bonalli, R., Pavone, M., and Açıkmeşe, B., "Convex Optimization for Trajectory Generation: A Tutorial on Generating Dynamically Feasible Trajectories Reliably and Efficiently", *IEEE Control Systems Magazine*, Vol. 42, No. 5, 2022, pp. 40–113. https://doi.org/10.1109/MCS.2022.3187542

[10] Birgin, E. and Martínez, J. M. "Practical augmented Lagrangian methods for constrained optimization", Society for Industrial and Applied Mathematics, 2007

[11] Gaskell, R. W., "Gaskell Eros Shape Model V1.1", urn:nasa:pds:gaskell.ast-eros.shape-model::1.1, 2021. https://doi.org/10.26033/d0gq-9427

[12] Nolan, M. C., Magri, C., Howell, E. S., et al., "Asteroid (101955) Bennu Shape Model V1.1", urn:nasa:pds:ast-bennu.radar.shape-model::1.1, 2021. https://doi.org/10.26033/33ba-s419