Space it Up! Days, 26 – 28 January 2026

DART Group

Learning-Augmented Sequential Convex Optimization for MPC-Based Deep-Space Autonomous Guidance

Alessandro Costa*, Francesco Topputo

* Corresponding author: alessandro2.costa@polimi.it

POLITECNICO MILANO 1863

Introduction Over the past two decades, CubeSats have revolutionized space exploration by democratizing access to Low-Earth Orbit. However, extending CubeSat utility to deep-space missions is currently hindered by the intensive **Human-in-the-Loop** resources required for ground operations. With communication windows becoming increasingly scarce and over-saturated, a paradigm shift toward spacecraft autonomy is essential. Nonetheless, current trajectory planning and correction maneuvers rely on ground-based computation due to limited onboard processing power[1]. This research proposes a robust, lightweight, **closed-loop low-thrust guidance algorithm** designed for complete onboard autonomy. By integrating **Deep Reinforcement Learning (DRL)** with **Sequential Convex Programming (SCP)**, a high-reliability **Model Predictive Control (MPC)** framework is developed. The DRL agent provides precise, computationally inexpensive initial guesses to warm-start the SCP algorithm, significantly reducing optimization time and increasing convergence success rates. The algorithm is validated through an Earth—Mars interplanetary transfer culminating in Ballistic Capture (BC).

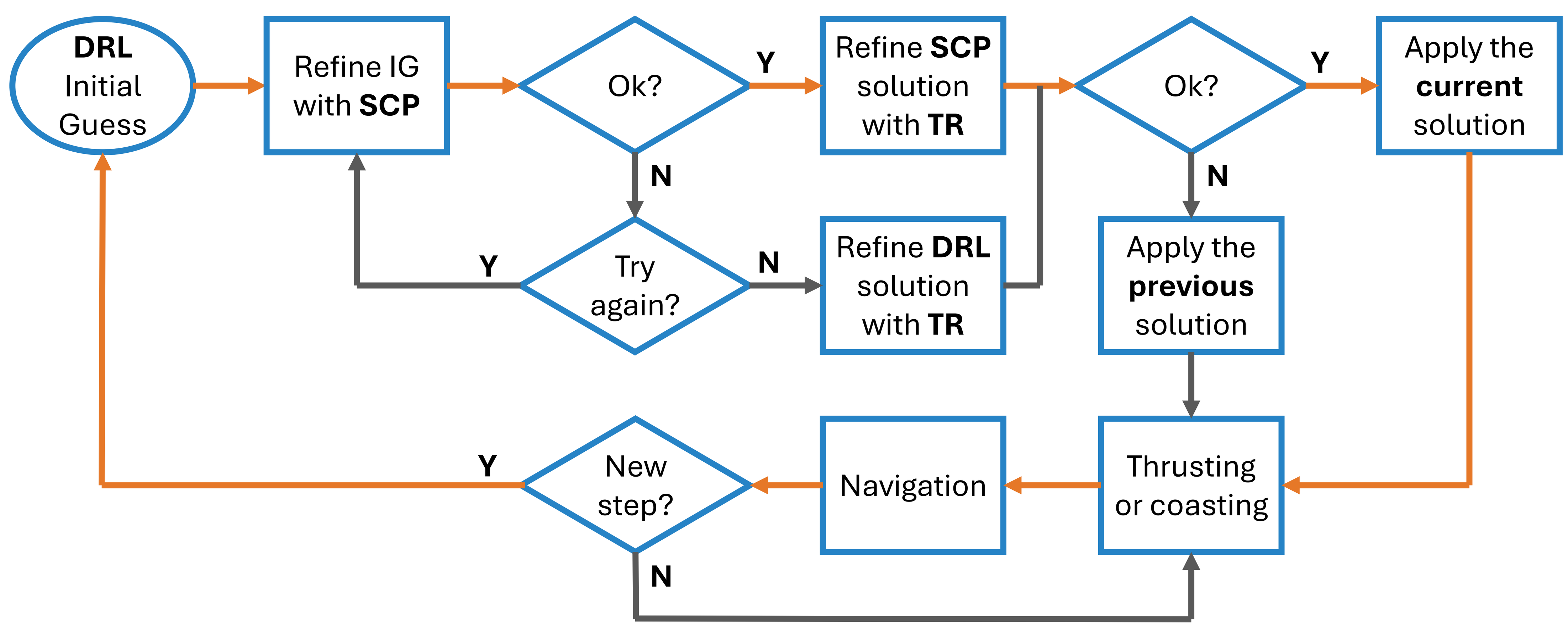

Fig. 1: MPC closed-loop guidance logic

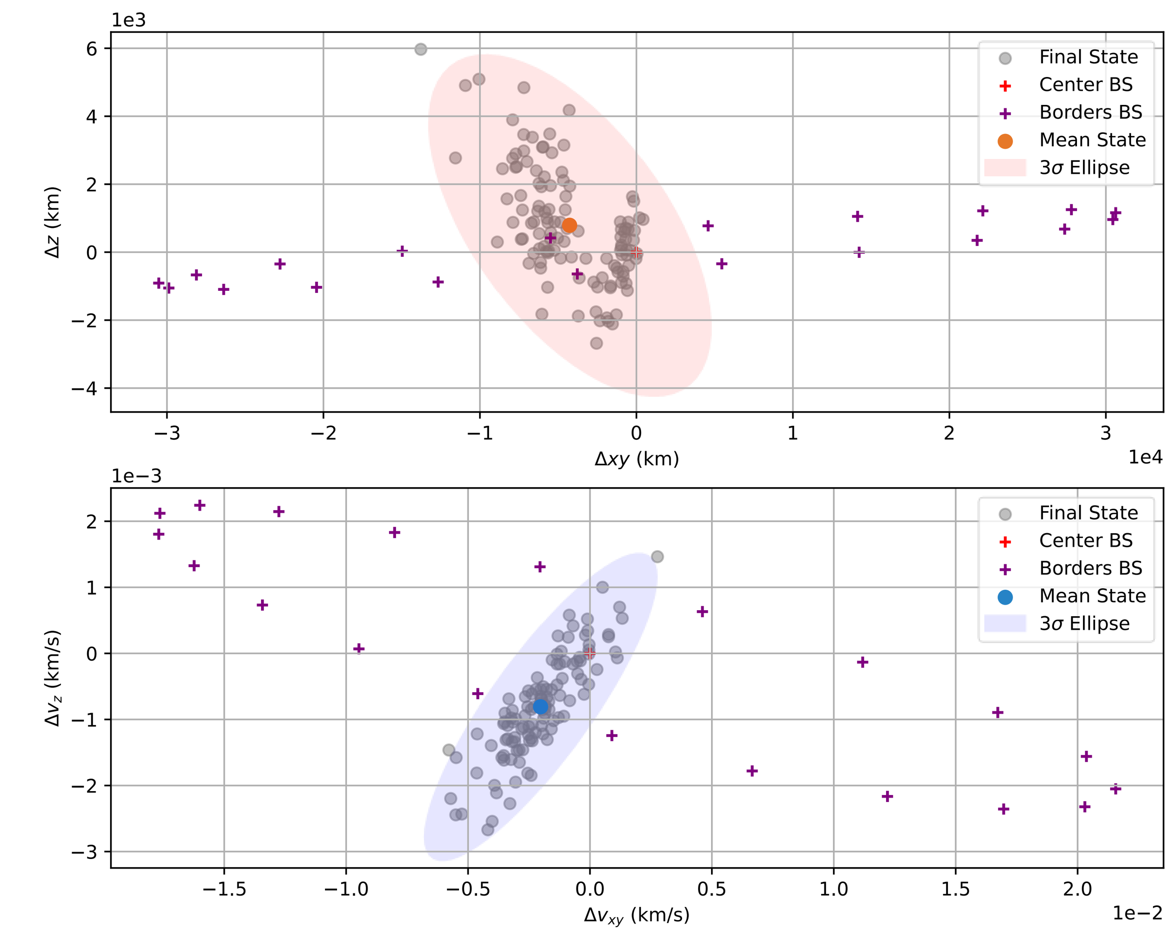

Fig. 2: Final states distribution on ballistic sub-corridor section

Deep Reinforcement Learning To generate high-precision initial guesses, a DRL agent is trained to mimic fuel-optimal low-thrust transfers. DRL has been selected for its inherent compatibility with closed-loop guidance, enabling the agent to navigate optimal solutions while accounting for navigation, actuation, and modeling uncertainties[2]. The training utilizes a **model-free, policy-based framework** powered by **Proximal Policy Optimization (PPO)**. Over 50 million training steps, the agent learns to produce near-optimal solutions in real-time, specifically tuned to minimize state displacement relative to the target capture sub-corridor. Once deployed, the agent acts as a computationally efficient warm-start mechanism, mapping current states to high-fidelity trajectories instantaneously.

Thrust Regularization The SCP algorithm outputs a time-discretized trajectory, thus, the solution must be regularized to generate a continuous control profile suitable for onboard execution[4]. The **Thrust Regularization (TR)** process transforms the discretized results into a true **bang-bang control** profile. The thrust magnitude is fixed at the maximum available value, while the thrust angles are parameterized using a sequence of **third-order polynomials**. The regularization is executed in two stages: first, a series of **single-shooting subproblems** is solved for each identified thrust arc to determine optimal switching times and polynomial coefficients. Subsequently, a **multiple-shooting problem** is solved to link all thrust and ballistic arcs. By leveraging the subproblems solutions as initial guesses, the algorithm efficiently enforces boundary constraints and trajectory continuity to obtain a flyable trajectory.

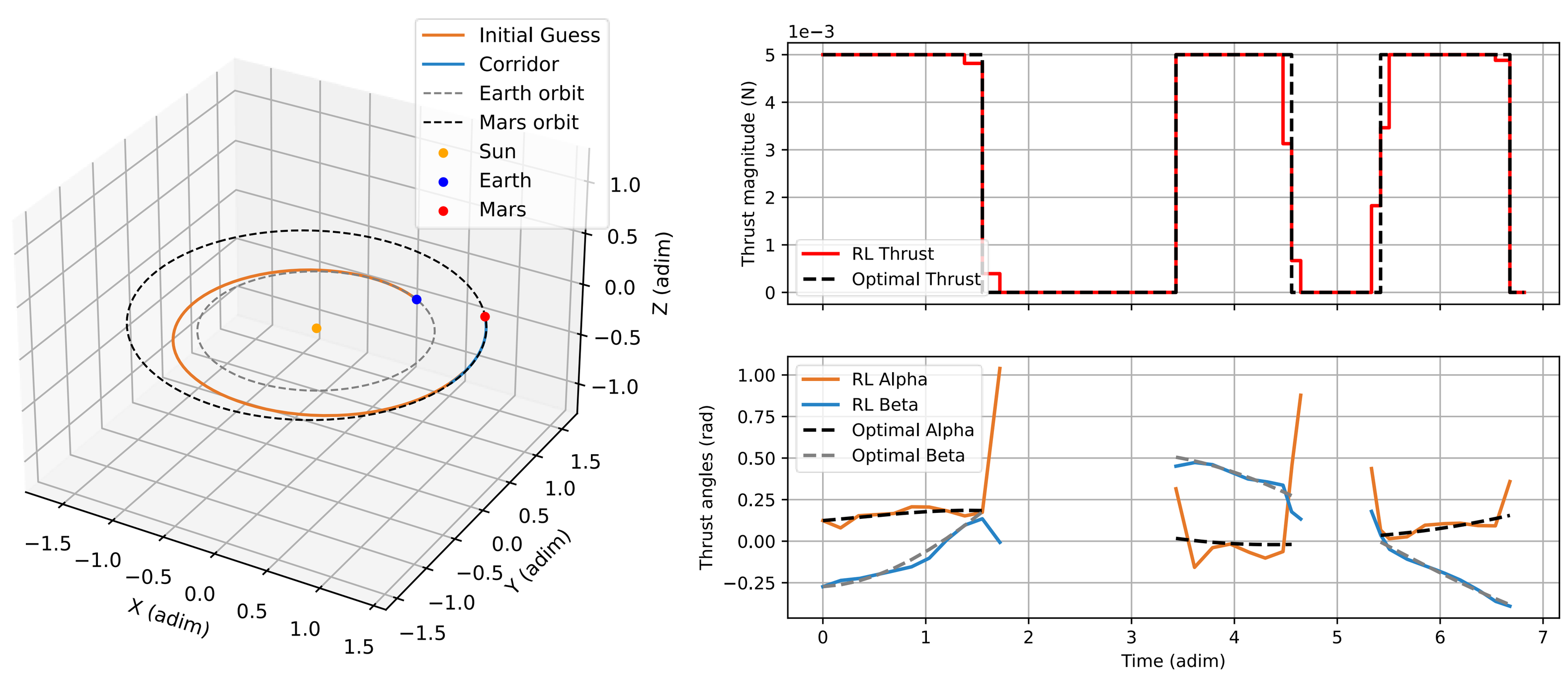

Fig. 3: DRL initial guess transfer to ballistic sub-corridor

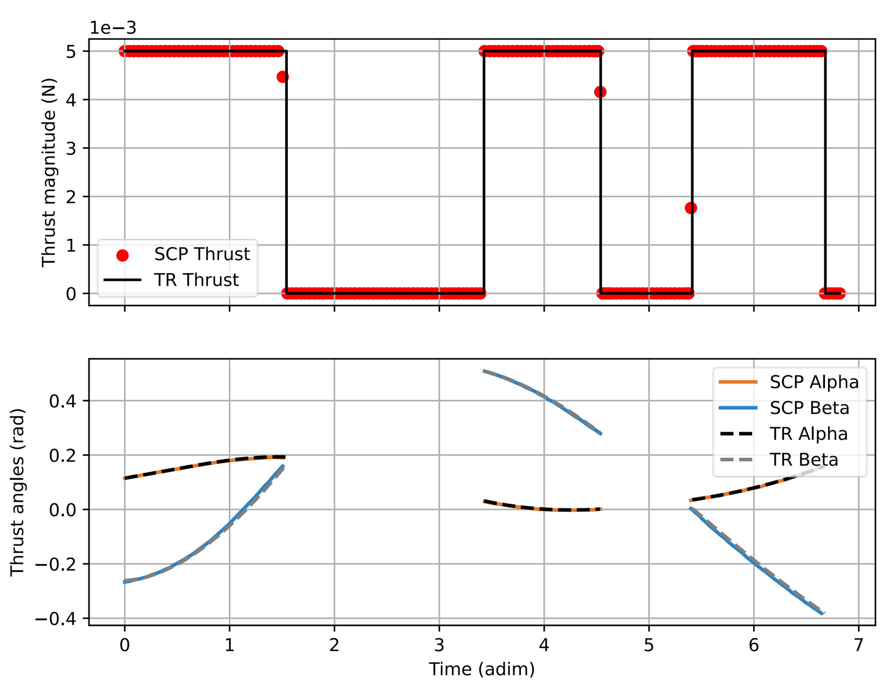

Fig. 4: SCP and TR solution comparison

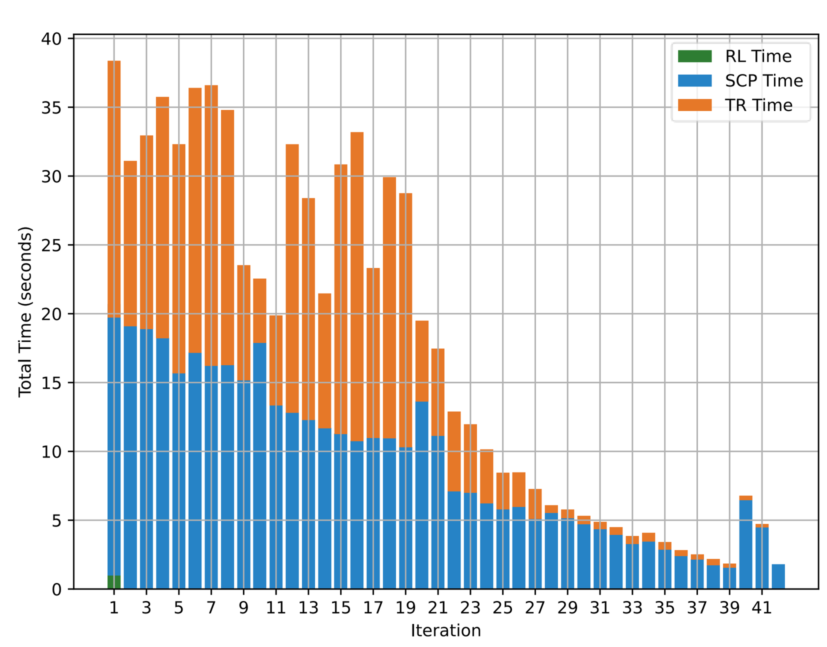

Fig. 5: CPU time of MPC process iterations

Sequential Convex Programming SCP is an iterative technique that solves nonconvex optimization problems by considering a sequence of convex subproblems whose solutions eventually converge to the solution of the original problem. SCP has been selected due to its computational efficiency, being able to provide optimal solutions in a fraction of the time required by traditional direct or indirect methods.

To perform SCP, the continuous-time non-linear dynamics are discretized over a temporal grid through a **Zero-Order Hold**. The linearized problem is then fed to the convex problem solver **Clarabel** to minimize **a multi-objective cost function** subject to **boundary conditions**, **trust region** and **physical bounds**. The cost function aims at minimizing the fuel consumption and the control gradients while driving the slack variables to zero to ensure solution feasibility.

$$J = \sum T_k \Delta t_k + w_r \sum \|u_{k+1} - u_k\|_2^2 + w_s \sum \|s_k\|_1$$

The algorithm manages the convergence of the sequence of convex problems to the nonlinear optimum using a **Trust Region** framework[3]. After solving the subproblem, the algorithm calculates the ratio of actual improvement to predicted improvement: **the convex accuracy ρ**. If ρ meets the minimum threshold, the step is accepted and the trust region is dynamically resized, expanding to accelerate convergence or contracting to ensure stability. If the threshold is not met, the step is rejected and recomputed with a reduced trust region to prevent solution degradation and ensure a robust path to the non-linear optimum.

Early Results The closed-loop autonomous guidance algorithm has been validated via a **120-sample Monte Carlo simulation**, modeling a 400-day Earth—Mars transfer culminating in Ballistic Capture. Guidance commands were recomputed at 10-day intervals to account for accumulated errors. As illustrated in **Figure 2**, the MPC framework successfully navigated the spacecraft despite significant actuation and navigation uncertainties. The majority of samples achieved a final state displacement within 10^4 km in position and 10 m/s in velocity relative to the center of the BC sub-corridor. Computational performance for a reference sample, shown in **Figure 5**, indicates that the highest processing demands occur during initial iterations, where both the SCP node density and the number of TR variables are at their peak. Conversely, DRL initial guess generation remains negligible, consistently requiring only a few milliseconds. Despite the preliminary nature of these results, the hybrid DRL-SCP approach proves to be a robust candidate for autonomous onboard guidance. Future work will focus on further validating these results within **Processor-in-the-Loop** and **Hardware-in-the-Loop** environments to confirm real-time flight readiness.

Acknowledgments This research was supported by the **Space It Up!** project. The author gratefully acknowledges the funding and support provided by the **Italian Space Agency** and the **Ministero dell'Università e della Ricerca**.

References

[1] A. Mannocchi, D. Perico, G. O. Pinho, G. Di Domenico, A. Morselli, F. Topputo. *An overview of EXTREMA project Pillar II: validating autonomous guidance algorithms via HIL testing.* International Astronautical Congress, 2024.

[2] L. Federici, R. Furfaro. *Improving reinforcement learning performance in spacecraft guidance and control through meta-learning: a comparison on planetary landing.* Neural Computing and Applications, 2025.

[3] Z. Whang, Y. Lu. *Improved Sequential Convex Programming Algorithms for Entry Trajectory Optimization.* Journal of Spacecraft and Rockets, 2020.

[4] A. C. Morelli, A. Morselli, C. Giordano, F. Topputo. *Convex Trajectory Optimization Using Thrust Regularization.* Journal of Guidance, Control, and Dynamics, 2024.

ASI Agenzia Spaziale Italiana

SPACE IT UP!

MUR