Toward Autonomous Guidance and Control: A Robust AI-Based Solution for Low-Thrust Orbit Transfers

Matteo Stoisa [1] Luca Romanelli [2] Federica Paganelli Azza [3] Pietro De Marchi [4] Mattia Varile [5]

[1]matteo@aikospace.com [2]luca.romanelli@aikospace.com [3]federica@aikospace.com [4]pietro@aikospace.com [5]mattia@aikospace.com

AIKO
INFINITE WAYS TO AUTONOMY

Autonomous on-board maneuvering

The advancement of conventional approaches towards methods with enhanced **autonomy** is a continuous and ongoing process, which ensures superior **performance**, **reliability**, and **scalability**. Autonomous Guidance and Control (G&C) systems are set to become a crucial technology in applications such as low-thrust orbit transfers, station keeping, rendezvous, and deep space maneuvers. In this context, our primary objective is to develop a comprehensive framework for **autonomous on-board G&C** that is adaptable and applicable across diverse scenarios.

Overview

The focus of our initial application scenario centers around a **low-thrust** orbit **transfer** in Low-Earth Orbit (**LEO**). This specific use-case has been chosen due to its inherent challenges, including the requirements for **robustness** and **real-time computation**.

We propose an **AI-based** solution capable of autonomous and robust on-board G&C. The core of our approach leverages a **Deep Neural Network** (DNN) trained through **Reinforcement Learning** (RL) techniques. Our method aims at enhancing a **traditional guidance** approach by managing environmental perturbations, it processes the on-board **navigation coordinates** and provides the **thrust** to be imposed by the propulsion subsystem.

Our approach demonstrates effectiveness in performing maneuvers changing **semi-major axis** (SMA), **eccentricity** (ECC), and **inclination** (INC), operating **continuously** with a control horizon of several **days**. Robustness is tested by using physical model **uncertainties**, introducing **disturbances** in the mission coordinates, and injecting **perturbations** in subsystems.

Architecture

The **Preprocessing** module is in charge of transforming the **Cartesian measurements** coming from the Navigation Subsystem into a **processed** form that is meaningful for the **Maneuver Computation** block, which encloses the **DNN**. It produces as output the **force vector** to be imposed in the center of mass. The **Spacecraft Simulator** propagates the orbit while applying the thrust, this **control loop** is iterated until the target orbit is reached. The period of action currently used is **10 minutes**.

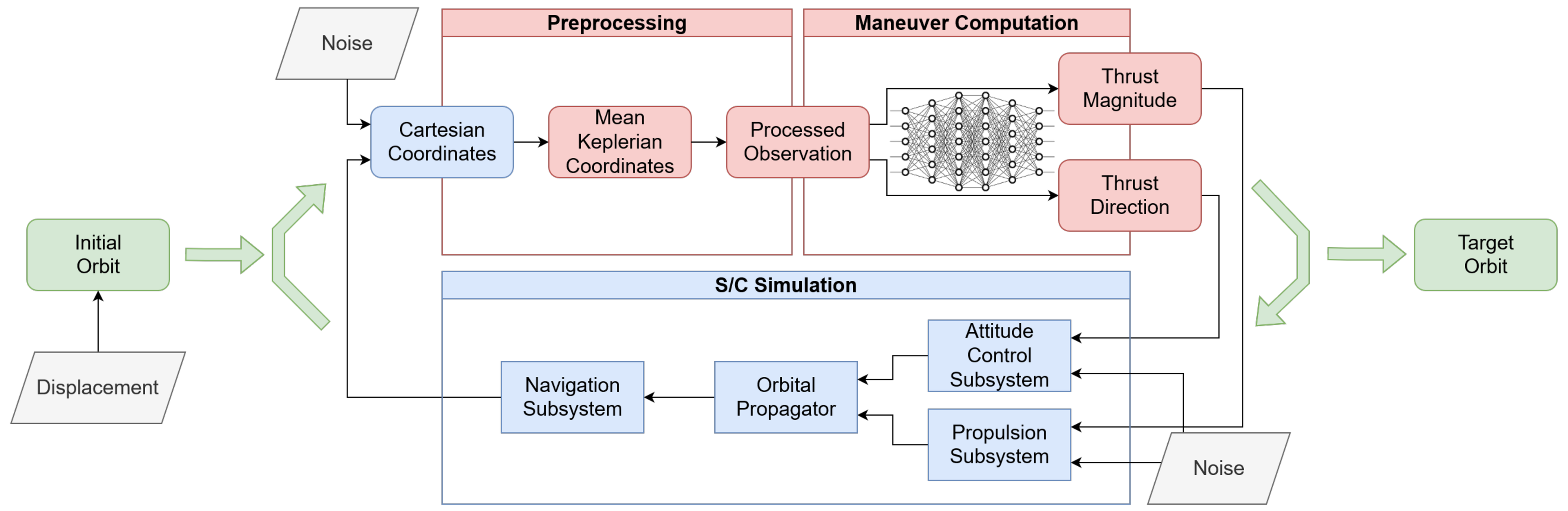


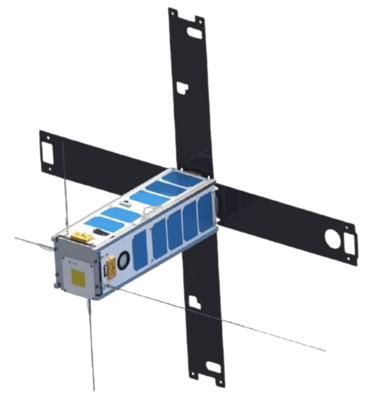
Figure 1. Maneuvering loop structure.

The mission is identified through the **starting** orbital configuration, the **target** orbital configuration, the **epoch**, and the **tolerance ranges** of acceptance for the maneuver to be considered complete. These ranges have to be properly set according to the specific maneuvers. In our scenario, the followings are chosen: 2 km for SMA, 0.001 for ECC, and 0.005 deg for INC.

Simulation scenario

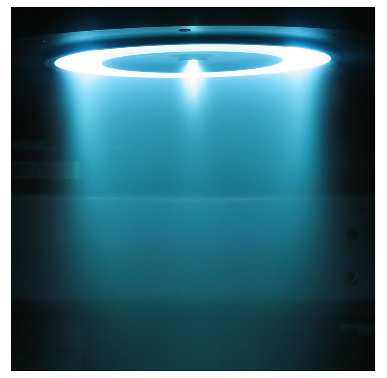
Platform

12U CubeSat with $15kg$ mass, $14.5kg$ dry mass. Attitude control is considered decoupled.



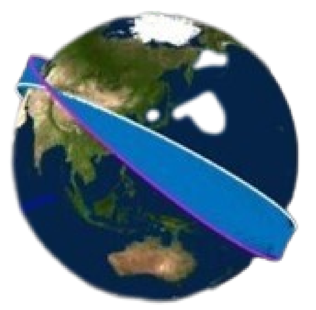
Thruster

Low-thrust engine with continuous throttle. Maximum thrust is $2.5mN$ with $1200s$ specific impulse.



Use-case

LEO maneuvers to change the semi-major axis (hundreds of km), the eccentricity, and the inclination (up to 1° corrections).



Validation

To push our framework towards the **real-world**, it is crucial to validate the robustness of the **closed-loop** maneuvering interaction. Simulations must be physically accurate and address non-nominalities of the models that could influence the potential **deployment**.

We employ **Basilisk** [3] simulator to validate our solutions. We use ***Runge-Kutta*** 4^{th} ***order*** propagation with $60s$ of integration step, including **J70** gravity perturbation, ***msis*** atmospheric drag and ***CannonBall*** solar radiation pressure models, and **third-body** gravity perturbation by sun and moon. Since the actual initial orbital coordinates of the mission may vary from the planned ones, we introduce an **orbital displacement** in the initial conditions of the maneuvers, using normal distribution bounded within standard station-keeping ranges.

Sensors and actuators uncertainties

Data coming from the on-board navigation cannot be considered exact since hardware subsystems are subject to **measurement errors**. To make sure that our solution is robust to such non-nominalities, we provide distorted Cartesian coordinates instead of the real **position** and **velocity**. Distorted values are sampled with normal distribution within an **ellipsoid** lying along the tangential direction.

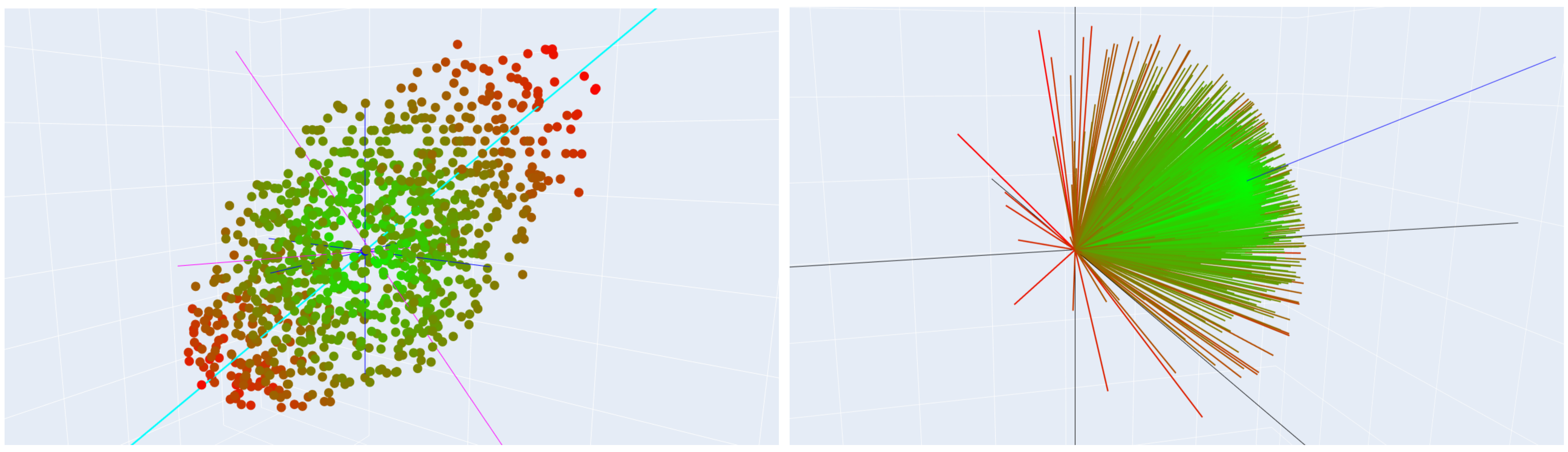


Figure 2. Visualization of the **probabilistic distribution** for the perturbed position and for the thrust direction. Green values represent variations close to **nominal states**, while red values represent more **critical discrepancies**.

Our algorithm requests the imposition of a continuous force vector to the thrust subsystem, but we cannot assure that this value would be exactly produced. In order to verify the robustness to non-nominalities in thrust production, we inject normal perturbations in the actual force provided within the simulation, both in **direction** and **magnitude**.

Reinforcement Learning

The core of our Autonomous Maneuvering System relies on **Multilayer Perceptrons** (MLP) trained through RL techniques [5]. In particular, for this application, we employed the **Soft Actor-Critic algorithm** [2], which represents the state-of-the-art in terms of performance and accuracy for trainings in the **control field**.

The Markov Decision Process

The maneuvering problem is formalized as a **state-action loop**, along with a **reward signal** assigned during the training phases.

The observation signal is composed starting from the **Cartesian coordinates**, which are transformed into **mean Keplerian** values. Those are **enriched** with data of the pre-computed traditional trajectory reported for multiple instants, then **normalized** and **scaled** with respect to the target orbit. The action signal is composed of **three continual** values, that are translated into the **force vector** requested to the thrust subsystem. The value function has a ”**shaped**” design based on the ”guess” trajectory, refined with scaling factors that allow faster convergence to proper behaviors.

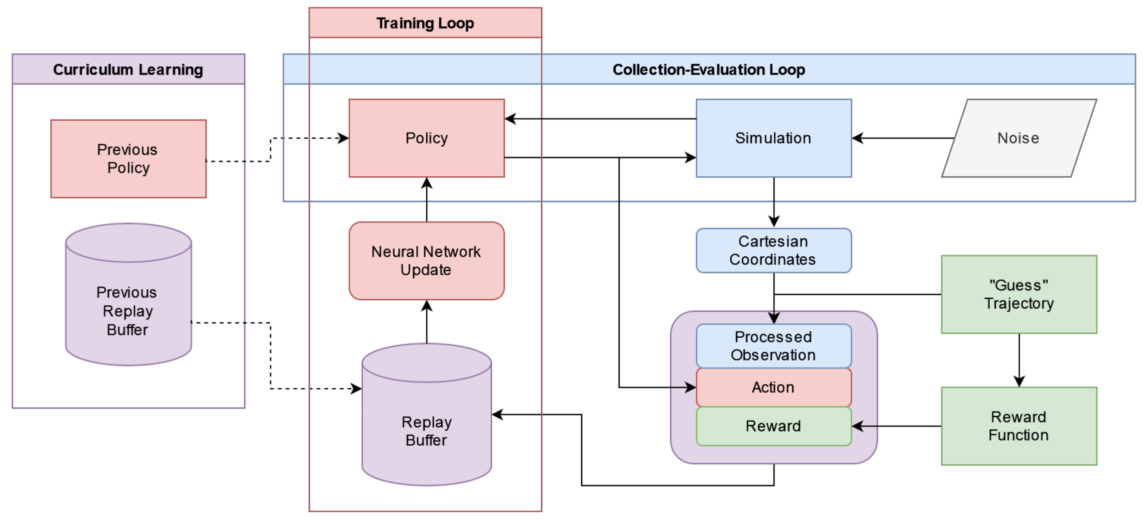


Figure 3. Training loop schema.

The ”guess” trajectory

During training procedures, the possible exploration of state-action-reward tuples is extremely vast. In order to limit the waste of computational and time effort, we exploit a trajectory pre-computed through **Edelbaum's** control [1] in nominal conditions. It allows defining **exploration boundaries** to focus the learning process in a reasonably smaller space region, managing to converge on a satisfactory solution.

Curriculum Learning

We employ Curriculum Learning [4] procedures in order to **refine** inference performance and robustness to disturbances. In addition, the usage of already-trained MLP allows significant **savings** in the training pipeline when addressing new maneuvers.

References

[1] Lorenzo Casalino. Approximate optimization of low-thrust transfers between low-eccentricity close orbits. *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, May-June 2014, 2014.

[2] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *CoRR*, abs/1801.01290, 2018.

[3] Patrick Kenneally, Scott Piggott, and Hanspeter Schaub. Basilisk: A flexible, scalable and modular astrodynamics simulation framework. *Journal of Aerospace Information Systems*, 17:1–13, 05 2020. doi:10.2514/1.I010762.

[4] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey. *CoRR*, abs/2101.10382, 2021.

[5] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.