A modular approach for rendezvous and proximity operations missions: from simulations to operations.

Thibault Lebeke[1], Andrea Fiorentino[2]
Exotrail, Toulouse, 31000, France

ABSTRACT

Rendezvous and proximity operations once limited to the servicing of the International Space Station might be on the point of becoming a key technological enabler for several commercial applications such as in-orbit servicing and tugging. Many potential customers, from commercial operators to national defense agencies are eager to find a provider for this kind of space services. In the present work we present a modular simulation environment that allows to easily go from high level mission objectives to operational maneuver plans. We start by discussing the context of proximity operations in the New Space era. In the following section some hypotheses are introduced, and the theory of linearized relative motion is recalled. Then the safety guidelines for proximity operations are summarized. After that the proposed modular approach is introduced with a focus on the solution of the resulting convex problem. Finally, two simulated test cases are presented, detailing a formation flying and relative proximity operations (RPO) scenarios by using a low-thrust propulsion system.

1. INTRODUCTION

Formation Flying is defined by NASA's Goddard Space Flight Center as "*the tracking or maintenance of a desired relative separation between or among spacecraft"*. Spacecraft flying in formation constitute thus an example of a *distributed system* as opposed to a *monolithic* one. Several scientific, military, and commercial applications are enabled using spacecraft' formations: interferometry, passive aperture radar, and ground track observations, just to cite a few.

Here we present a modular approach that has been developed at Exotrail to design simulate and operate formation flying and relative proximity operations (RPO) missions. The schematic in Figure 1 shows how the formation flying maneuver computation block can be plugged into an operational product. In the present paper we will focus on the maneuver computation block allowing to output a maneuver plan (sequence of controls) to satisfy given control objectives. In what follows, we will focus on a realistic low-thrust propulsion system for small satellites.

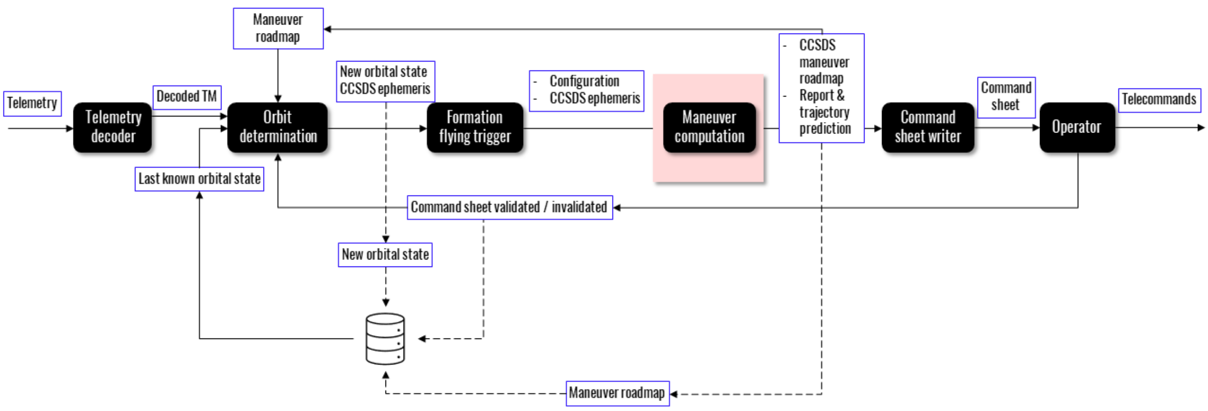

Figure 1 Modular mission operations block

The simulator that has been developed leverages the flexibility of the Java object-oriented programming language. This makes it easily extensible with respect to the solvers used for solving the optimization problem or the dynamical model used to represent the relative motion of the spacecraft. The version presented here computes the control by solving a constrained (convex) minimization problem with linear or quadratic objective. A linearized dynamical model is used at this stage. However, nonlinear optimization methods could easily be added in the future if more stringent requirements are to be met.

[1] Space dynamics engineer, thibault.lebeke@exotrail.com
[2] Space dynamics expert, andrea.fiorentino@exotrail.com

For all the basics astrodynamics computation the present work relies upon the state-of-the-art astrodynamics library Orekit [1].

2. METHODS AND MODELS

As it has been mentioned in the introduction, the focus here is on the simulation of a formation flying or RPO mission scenario. We will describe: the dynamical models used for the relative motion, the type of constraints which are supported and the control problem formulation. We will then outline the chosen solution approach. We will also discuss thrust quantization, which is important to output solutions that respect the constraint that the propulsion system be firing at a fixed thrust level for a specific functioning point. In the following we focus for simplicity on scenarios involving two spacecraft.

We start by defining the scope of the formation flying and RPO scenarios: the former consists in maintaining the relative separation between the spacecraft, which is translated in applying controls such that the maneuvering spacecraft stays in a box surrounding its nominal state. The RPO mission is intended here as an inspection mission with a servicer tug flying around a target spacecraft. In this case safety constraints need to be enforced to avoid collisions and ensuring safety even in case of propulsion system failure.

2.1 SIMULATOR AND CONTROLLER WORKFLOW

Figure 2 shows the simulator which is used to simulate missions where the relative dynamics of 2 spacecraft needs to be controlled. Because of its modular nature the simulator can be used for cases involving larger formations.

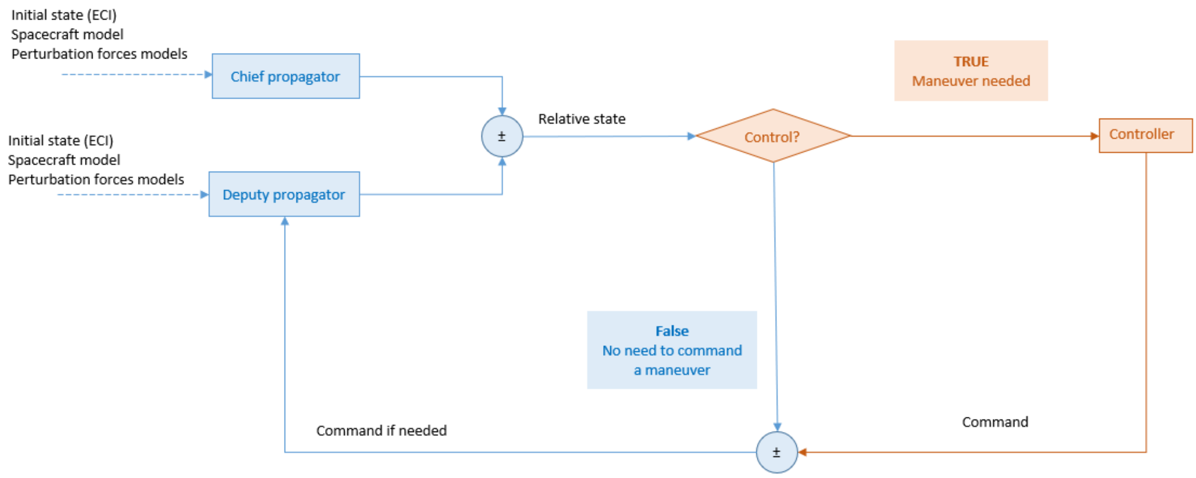

Figure 2: Simulator general workflow

First, depending on the mission scenario, a suitable set of constraints will be specified. These can be constraints on the state variables or on the control variables (thrust magnitude and direction).

Each spacecraft trajectory is propagated using a state-of-the-art numerical propagator with all the relevant perturbations included. Complex spacecraft geometries can be simulated as well if needed, to enhance the realism of the simulation. A generic propagation block is shown in Figure 3. The relative state is sampled at each time step and used to evaluate a control triggering function. When this function activates, for example by reaching a threshold value, a maneuver plan is computed, over a user-specified duration or *control horizon*, by solving a convex optimization problem. This consists in minimizing a convex objective function while satisfying the user level constraints imposed for the given mission.

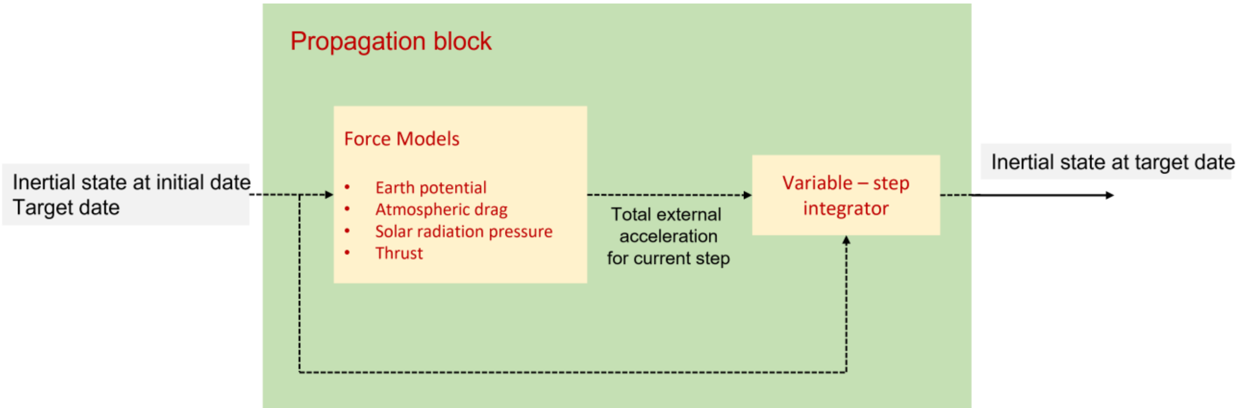

Figure 3: Propagation block

2.2 DESCRIPTION OF THE CONTROLLER

A model-based controller has been designed which allows to keep the nominal formation or perform the given RPO mission. This control methodology is made up by three main components:

1. An objective function, in our case a linear or quadratic criterion to be minimized.
2. A linear dynamical model as those described in 2.3.
3. An optimizer that allows to solve the convex minimization problem while satisfying a set of constraints.

The control is computed over the control horizon T_c which can be specified by the user. We will now describe in more detail each of these elements.

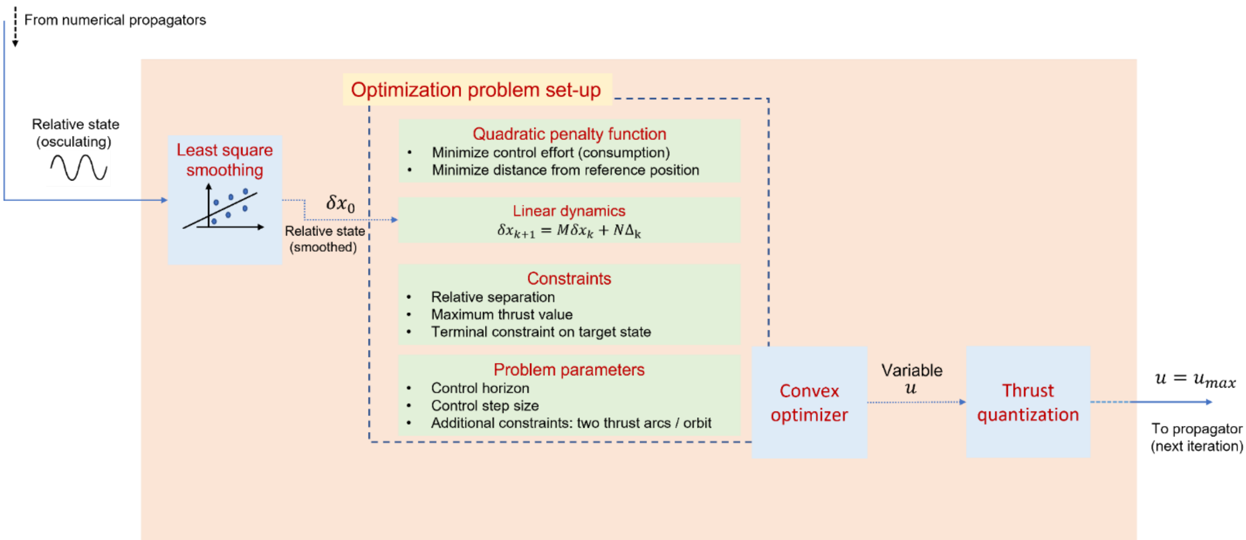

Figure 4: Controller workflow

2.3 RELATIVE MOTION MODELS

In this section the two relative motion models currently implemented in our modular simulator when computing the control are described. These are the Clohessy-Wiltshire equations and linearized Gauss variational equations. Because of the reduced distance between the spacecraft in the missions considered, the differential perturbations (due to the atmospheric drag and higher order gravity) remain small and linearized models work well for control purposes. However, one of the advantages of our modular approach is to make it easy to use more complex dynamical models if necessary.

2.3.1 CLOHESSY-WILTHSIRE EQUATIONS

The Clohessy-Wiltshire equations are the simplest model of relative motion between two spacecraft. They are easily obtained by linearizing the relative acceleration under the hypothesis that the separation between the leader and the follower is small compared to the leader's orbit radius, see [2]. The corresponding equations in the LVLH frame (x directed along track, z radially inward and y completing the right-hand triad), see [3], are:

$$\begin{cases} \ddot{x} = -2n\dot{z} + a_x \\ \ddot{y} = -n^2 y + a_y \\ \ddot{z} = 2n\dot{z} + 3n^2 z + a_z \end{cases}$$

We see that the out of plane dynamics (along the y-direction) is that of a simple harmonic oscillator and decoupled from the in-plane motion.

Although it is a simple system, the Clohessy Wiltshire equations work fine for proximity operations and the computed ephemerides are close enough to a high-fidelity propagator that could be used in spacecraft operations.

To account for linearization errors a curvilinear coordinate system is used to correctly compute the relative state between the target and the servicer. These curvilinear coordinates correspond to the one described in [4] and are suited to extend the Clohessy Wiltshire model validity along the orbit, especially for some formation flying missions where the spacecraft can be several kilometers away.

2.3.2 GAUSS VARIATIONAL EQUATIONS

Gauss variational equations give the derivatives of the osculating orbital elements in presence of a perturbing acceleration. They can be simplified when the orbit is almost circular and expressed in terms of nonsingular elements. We recall their expression:

$$X = \left[a, e_x, e_y, i, \Omega, \alpha\right]^T$$

With $e_x = e\cos\omega$, $e_y = e\sin\omega$ the component of the eccentricity vector and $\alpha_M = \omega + M$ the mean argument of latitude.

Gauss variational equations then take the form:

$$\dot{X} = N + Bu$$

Considering a perturbative acceleration $u = \left[u_x, u_y, u_z\right]^T$ vector expressed in the LVLH frame as defined by the CCSDS standards [3] we have for nearly circular orbits:

$$N = [0,0,0,0,0,n]^T$$

Describing the Keplerian motion, and the matrix B mapping the perturbative acceleration to an effect on the derivatives of the orbital elements

$$B = \frac{1}{na}\begin{bmatrix} 2a & 0 & 0 \\ 2\cos\alpha & 0 & -\sin\alpha \\ 2\sin\alpha & 0 & \cos\alpha \\ 0 & \cos\alpha & 0 \\ 0 & \dfrac{-\sin\alpha}{\sin i} & 0 \\ 0 & \dfrac{\sin\alpha}{\tan i} & 2 \end{bmatrix}$$

The relative equations are obtained by differencing those for the two spacecraft. A linearization can then be done about the reference orbit following [5].

2.3.3 DISCRETIZATION OF THE DYNAMICS

Both dynamical models presented so far can be discretized in the form:

$$X_{k+1} = \Phi_{k+1,k} X_k + \Gamma_k u_k + \Lambda_k w_k$$

Where:

- Subscripts indicate time steps.
- $\Phi_{k,k+1}$: is the state transition matrix corresponding to the natural motion.
- Γ_k: maps the control at time step k to an effect on the state at time step $k+1$.
- Λ_k: maps the disturbance at time step k to an effect on the state at time step $k+1$.

For the details on the computation of the matrices Φ, Γ, Λ the reader can consult [6]. For simplicity, in the following we will neglect the effects of the disturbance w_k. As already mentioned, this can be done quite safely for tight formations and proximity operations, because of almost vanishing differential perturbation effects.

2.4 OBJECTIVE FUNCTIONS AND CONSTRAINTS

For a linear objective function with linear constraints, we want to minimize the 1-norm of the controls. In this case, since this objective is not convex, slack variables are introduced. [7] slack variables $U_k^+ > 0, U_k^- > 0$ allow writing $U_k = U_k^+ - U_k^-$. Then stacking up the whole control sequence we define the vector:

$$U = \begin{bmatrix} U_k^+ \\ U_k^- \end{bmatrix}$$

And the optimization problem to be solved takes the form:

$$\min_{U_k} c^t U$$
$$\text{s.t.} \begin{matrix} SU \leq E \\ CU = D \end{matrix}$$

Where:

- c is a vector of coefficients.
- S, C are (inequality or equality) constraints influence matrix.
- E, D are constraints vector.

We also consider the following program with a quadratic criterion:

$$\min_{u} \frac{1}{2} \mathrm{u}^t Q \mathrm{u} + p^t \mathrm{u}$$
$$\text{s.t.} \begin{matrix} SU \leq E \\ CU = D \end{matrix}$$

Where:

- $Q = A.A$ with A a diagonal matrix with the vector c described above on its diagonal. Other elements can be added if we want to minimize the distance from the target in addition to the minimization of propellant consumption.
- $p = -A^t.P$ with P a vector containing the target for each element, 0 for control minimization or a specified state for distance minimization.

The presence of the quadratic term allows to enforce at each time step to minimize the distance to some prescribed state in addition to the minimization of the control.

2.5 CONSTRAINTS

The constraints available in our simulator allow to:

- Reaching a target by setting an equality constraint.
- Achieve the security requirements by inequality on relative position and velocity.

- Enforce the maximum-thrust constraint by inequality norm constraint on the computed command.

The different constraints are described hereafter.

2.5.1 SAFETY CONSTRAINTS

The safety constraints ensure that there is no collision risk between the servicer and target. Some of them are provided by space agency guidelines, as in [8]. Among all the recommendations in the guidelines only the most important ones concerning the servicer trajectory for proximity operations where implemented.

ALL-0013	The servicer can follow any trajectory within the approach zone if such trajectory does not enter the *keep out zone* (except for an Approach Corridor) at any point during its execution, including in the event of failure preventing further orbit and attitude control.
ALL-0085	In case of loss of major functions which cannot be recovered by the on-board failure detection, isolation and recovery function, the servicer has the capability to ensure safety for an extended period. Once in this mode any further actions others than associated to maintaining the mode is initiated from ground.

Table 1 Safety requirement

The requirements corresponding to safety constraints rely on the proximity operations concept defined in [9] and is illustrated by different zones in the following figure:

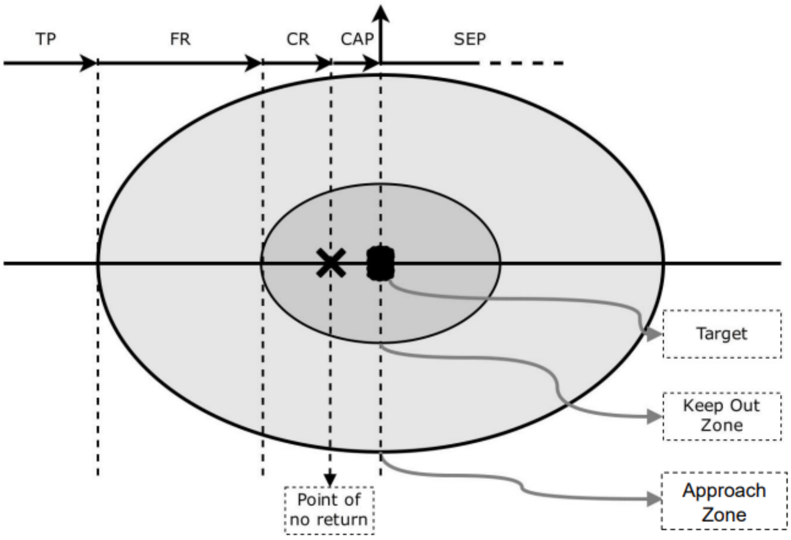

Figure 5 Definition of zones for safety constraints (from [8])

All these constraints can be modelled using inequality constraints on the state of the servicer, that is as:

$$SU \leq E$$

If the servicer trajectory remains in this convex space, its trajectory does not reach keep out zone. To model it, the relative linear dynamics can be used and build the inequality system by:

- $E = YX_{target} - MN_{\Delta t}YX_0$ which corresponds to the difference between initial state propagated to target date without control and target state.
- $S = MN_{\Delta t_1}NB_{\Delta t_1^{man}}$ with $NM_{\Delta t_1}$ corresponding to the effect of the extended time of free motion after the command and $NB_{\Delta t_1^{man}}$ to the effect of command at end of maneuver.

2.5.2 NORM CONSTRAINTS AND THRUST QUANTIZATION

To respect thruster capabilities every command must be under the maximum acceleration provided by the propulsion system of the spacecraft. However, real low-thrust propulsion system only provides a fixed level of thrust.

At solver level an inequality constraint on the norm of the command was implemented. The goal being to build a polyhedron around a sphere with the maximum acceleration as radius ($\|u\|_2 \leq u_{max}$), as depicted in the following figure:

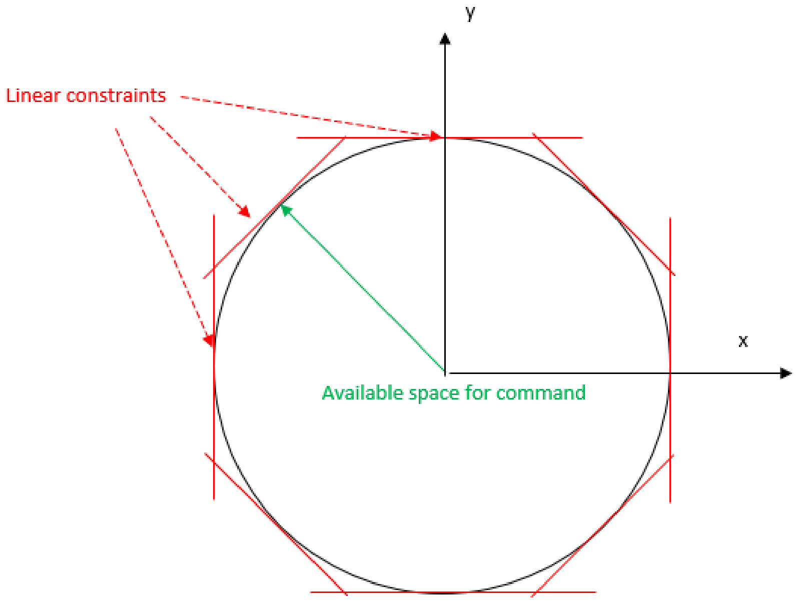

Figure 6 Illustration of constraints on command norm

Again, all constraints can be defined as a set of multiple planes which are tangent to the maximum acceleration sphere and can be written as:

$$SU \leq E$$

Where:

- E correspond to the effective maximum command.
- S correspond to the plane normal direction.

The command computed by the solver is not directly applicable to a spacecraft, the thrust need to be quantized at the Hall effect propulsion thrust.

At first approach and thanks to the linear model an averaging of the thrust is available, for each thrust:

$$\vec{U}_q = \frac{\vec{U}}{\|\vec{U}\|} . U_{max}$$

With:

- $\vec{U}_q$ is the command after quantization.
- $\vec{U}$ the command computed by the solver.
- U_{max} the fixed-control level achievable by the Hall effect thruster.

The quantization process does not alter the control direction computed by the solver but only its magnitude. Regarding the execution time of the command and its duration the following approach is proposed:

$$T_{man}^{q} = T_{man} + \frac{\Delta t_{man}}{2}\left(1 - \frac{\|\vec{U}\|}{U_{max}}\right)$$

$$\Delta t_{man}^{q} = \Delta t_{man} \frac{\|\vec{U}\|}{U_{max}}$$

Where:

- T_{man}^{q} is the start time of the quantized thrust.
- T_{man} is the start time of the thrust arc as computed by the solver.

- Δt_{man} is the thrust duration for the command computed by the solver.
- Δt_{man}^{q} is the thrust duration once the maneuver is quantized.

The quantized thrust is thus placed at the middle of the initial thrust arc.

As it will be shown from the results presented in the test cases part of this paper, this simple thrust quantization scheme works well for the formation flying and inspection mission considered in this paper.

3. TEST CASES

In this section two test cases are presented, for a formation flying and inspection mission around a target spacecraft. These two studies use the algorithms and methods describe above.

3.1 FORMATION FLYING TEST CASE

The formation flying case consists of relative station keeping between two spacecraft. In this mission the relative distance is a crucial point for the success of the mission. The studied case corresponds to a formation flying mission with two spacecraft flying in tandem, this design enables the possibility to measure radar signal from the ground and located the signal source.

The leader spacecraft is designated as the reference, its orbit is a circular SSO at 650km of altitude with a frozen eccentricity and the follower spacecraft performs the maneuvers. The formation center is 10 km following along the orbit path. The follower must be in a 2km box surrounding the formation center as shown in Figure 7.

Parameter	Value
Semi major axis [km]	7028.1363
Eccentricity [-]	1.05e-3
Inclination [deg]	97.98
RAAN [deg]	-19.59
Argument of perigee [deg]	0
True anomaly [deg]	0

Table 2 Keplerian parameters of the leader spacecraft

The spacecraft parameters are reported in the following table:

Parameter	Value
Volume [U]	6
Dry mass [kg]	9.5
Drag area (in-plane thrust) [m²]	0.02
Drag area (out of-plane thrust) [m²]	0.09
Drag coefficient	2.2
Thrust [N]	2.5e-3
Duty cycle (% of orbital period)	40
Specific impulse [s]	800

Table 3 Spacecraft parameters

The initial condition (cartesian coordinates) for this test case in the leader's LVLH frame (already introduced):

$$X_0 = [10001 \quad 1 \quad 10 \quad 0 \quad 0 \quad 0]^T$$

The maneuver target, expressed in the same frame:

$$X_{ref} = [10000 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T$$

Other test parameters are:

- Control box around the reference position (formation center) is ±2 km along the axes of the LVLH frame.

- Control trigger threshold is 1.5 km. This allows considering unmodelled dynamics and error introduced by thrust quantization.
- The simulation duration for this test case is 30 days.
- Perturbations: Earth potential, solar radiation pressure, third body effects from Sun and Moon.

We consider two scenarios differing in the dynamics used in the controller as well as the objective function:

Scenario 1
- The Clohessy-Wiltshire model is used to compute the control.
- Objective function: quadratic. A tradeoff is made between minimizing the time to target and propellant consumption at each time step.

Scenario 2
- The Gauss variational model is used to compute the control.
- Objective function: quadratic minimizing only propellant consumption. Target state will be reached, if possible, at final time.

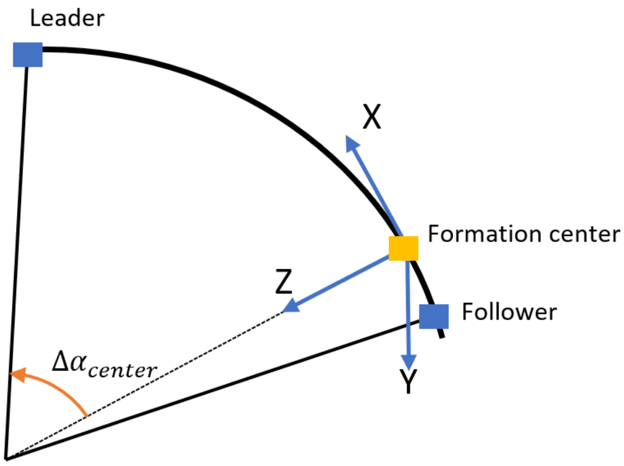

Figure 7 Formation flying illustration with the LVLH axis.

3.1.1 RESULTS FOR SCENARIO 1

The following tables summarize the main results for this test case (Scenario 1):

Maneuver nb.	1st	2nd	3rd	4th	Total
Thruster on time [s]	2620	2476	2482	2492	10070
ΔV [m/s]	0.7	0.66	0.66	0.66	2.68
Propellant consumption [g]	0.8	0.8	0.8	0.8	3.2

Table 4

Maneuver	1st	2nd	3rd	4th
X [m]	804	169	112	15
Y [m]	1.11	-7.4e-1	-1.7	-8.0e-1
Z [m]	34	-5.3	6.2	14
VX [m/s]	6.2e-2	4.5e-2	4.4e-2	2.4e-2
VY [m/s]	5.2e-5	9.7e-4	4.1e-4	-3.2e-3
VZ [m/s]	-2.5e-2	2.3e-2	5.4e-3	2.0e-3

Table 5

Figure 9 shows that the relative trajectory in orbital plane, the along track position respects the constraint of 2km. The four and an-half loops in the figure correspond to the maneuvers needed to bring the follower at the reference position. Figure 10

Figure 10 shows that after each maneuver, the along-track velocity is decreasing, which gives a longer waiting time before a new maneuver is needed.

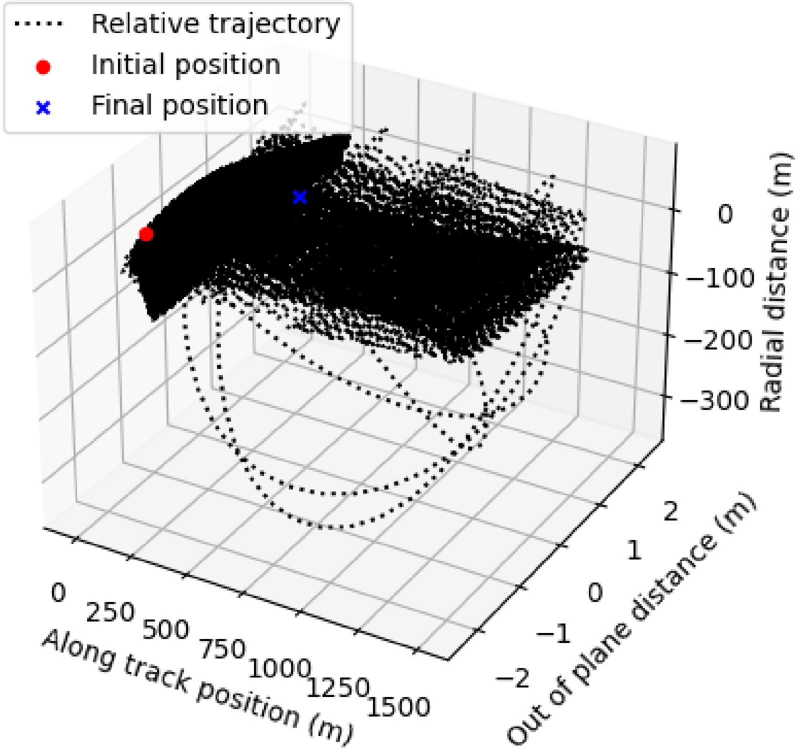

Figure 8

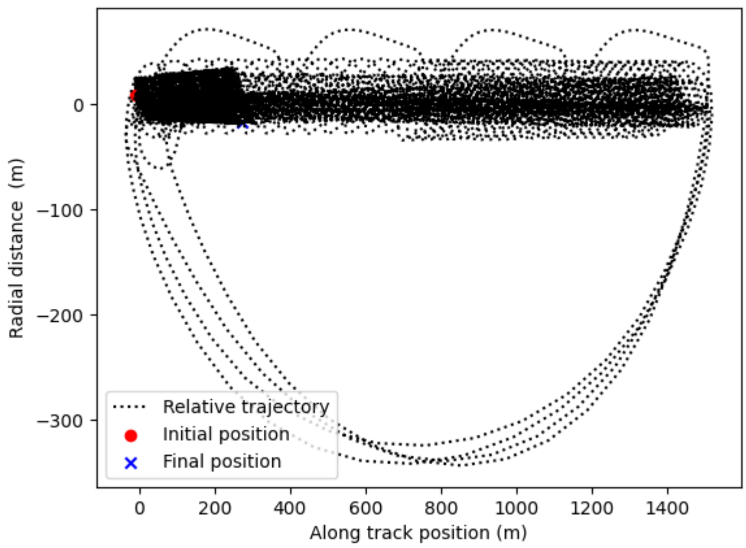

Figure 9 Relative motion in the X-Z plane

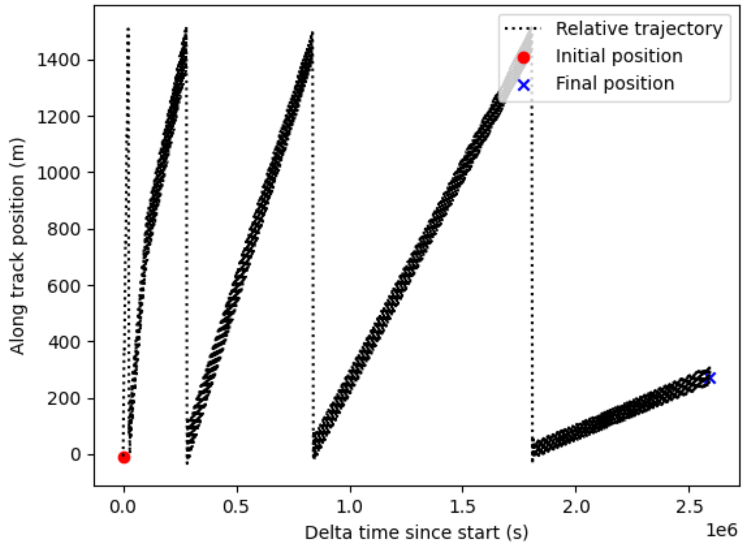

Figure 10 Evolution of the along-track position in time

3.1.2 RESULTS FOR SCENARIO 2

The following tables summarize the main results for Scenario 2:

Maneuver	1st
Thruster on time (s)	190
ΔV(m/s)	5e-2
Propellant consumption (g)	6e-2

Table 6

Maneuver	1st
X (m)	50
Y (m)	1.12
Z (m)	25
VX (m/s)	4.3e-2
VY (m/s)	-2.0e-4
VZ (m/s)	-6.6e-3

Table 7

In this scenario the behavior is very different and more efficient because of the solver parametrization that give better precision on target. This leads to only one maneuver in 30 days instead of four.
Regarding the relative trajectory in orbital plane shown in Figure 12, the along track position complies with the mission constraint of 2km. Contrary to the Clohessy Wiltshire model, in this case we better exploit the control box arriving close to the limit distance of 2km but without exceeding it.

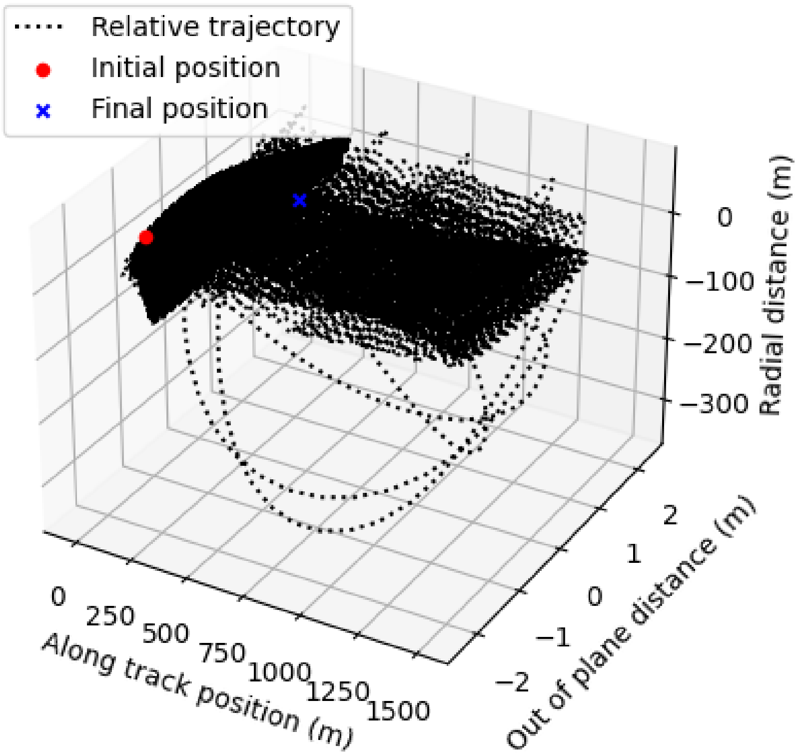

Figure 11

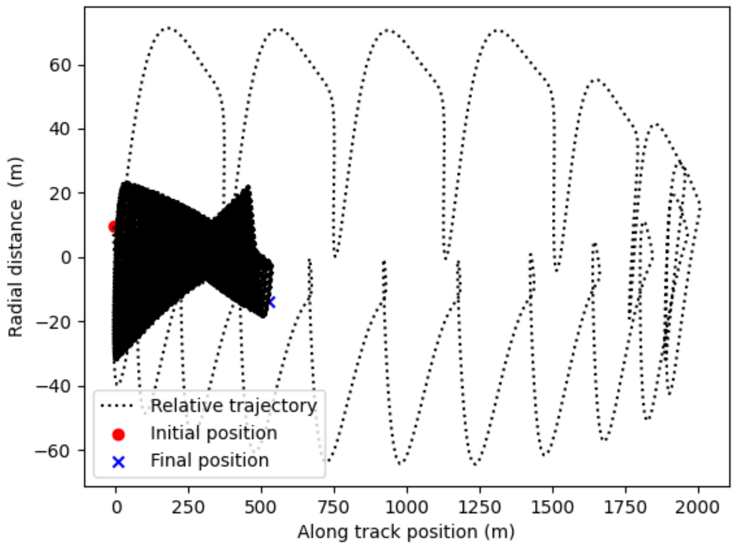

Figure 12

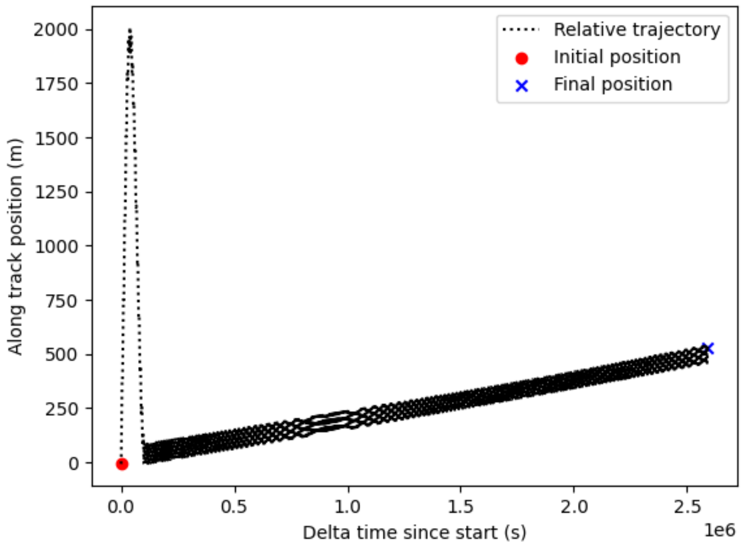

Figure 13

3.2 INSPECTION MISSION TEST CASE

For this test scenario we consider the same orbital parameters for the target spacecraft than before, while the servicer starts 1km away from it, along the along-track axis of the LVLH frame centered on the target. The spacecraft parameters (thruster and platform) are those in Table 3.

The test consists of the following steps:

- Start from the initial condition
$$X = [1001, 1, 1, 0, 0, 0]^T$$
- Acquisition of a stable inspection orbit by aiming for the following target:
$$X = [2r_0, 0, 0, 0, 0, -nr_0]^T$$
- Stay 12 h on the inspection orbit.
- Return to the safe initial position.

Where:
- r_0 = 100m is the minimum distance from target.
- n is the target mean motion.

The value of $2r_0$ on the X axis in LVLH local orbital frame corresponds to the semi minor axis of the stable circumnavigation ellipse. The $-nr_0$ on Z axis is the relative velocity of the spacecraft at this point of the circumnavigation relative orbit.

The Clohessy-Wiltshire dynamics is used to compute the control. The same perturbations used for the previous test are considered in this scenario.

4. RESULTS

The following tables summarize the main results for the inspection mission:

Maneuver	1st	2nd	Total
Thruster on time [s]	631	267	898
ΔV [m/s]	0.17	7.1e-2	0.24
Propellant consumption [g]	0.2	0.1	0.3

Table 8

Maneuver	1st	2nd
X (m)	12	-45
Y (m)	-5e-1	-1.19
Z (m)	-9.7e-1	-2.8
VX (m/s)	-2.4e-3	-2.6e-3
VY (m/s)	-1.5e-4	-5.0e-4
VZ (m/s)	-2.8e-2	-1.6e-3

Table 9

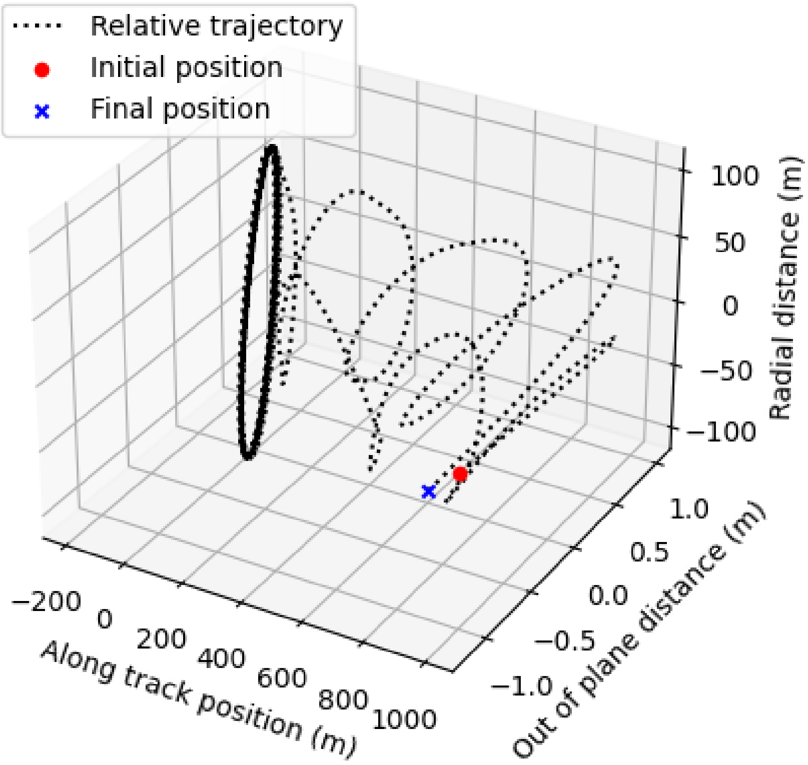

Figure 14

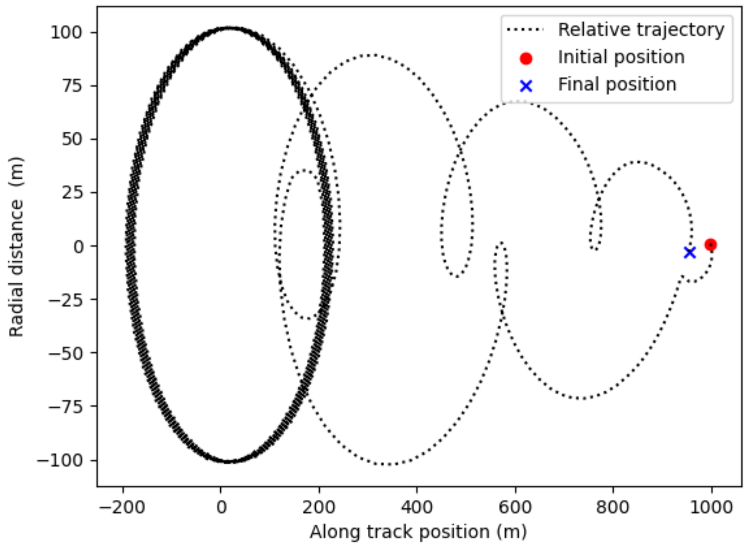

Figure 15 Maneuvers to and from inspection orbit (target is at the origin)

CONCLUSIONS AND FUTURE WORK

Exotrail formation flying and RPO simulator, capable of computing maneuver plans respecting realistic thruster and platform constraints typical of small satellites equipped with low-thrust propulsion systems has been presented in this paper. This simulator is part of Exotrail astrodynamics software and leveraging the genericity of object-oriented programming can be easily plugged into an operational flight dynamics system.

We have presented simulation cases for both a simple formation-keeping scenario and target inspection by a servicer spacecraft. The novelty of this work is not on the theoretical part, which uses well-known concepts from astrodynamics, linear system theory and optimization but on the modularity of the approach. This modularity is introduced early into the design of the simulator and allows to easily connect it to other astrodynamics components such as a mission control center. Modularity also makes it easy to maintain its code and update its core elements: the solver library can be extended to include new and more efficient ones while the dynamical models can be enhanced to better account for orbital perturbations when computing the control. Also, we could use these same algorithms to perform missions around other bodies of the solar system.

Despite all the work done, several improvements can be envisaged for this simulator:

- Develop a user interface taking great care to the user experience. This is a key point if we want this work to translate into value for the user designing or operating their innovative mission concepts.
- Implement a more robust approach based on second order cone programming, making the enforcement of the norm-constraint more precise and computationally more efficient.
- Simulate a rendezvous mission enforcing all the proximity operations guidelines [8].

These will bring us closer to enabling rendezvous and proximity operations mission concepts which will be critical for commercial applications such as in-orbit servicing and tugging.

REFERENCES

[1] L. Maisonobe, V. Pommier-Maurussane and P. Parraud, "Orekit : an Open-source Library for Operational Flight Dynamics Applications," in *International Conference on Astrodynamics Tools and Techniques*, 2010.

[2] K. Alfriend, S. Vadali, P. Gurfil, J. How and L. Breger, Spacecraft Formation Flying, Elsevier, 2010.

[3] CCSDS, "Navigation Data Definitions and Conventions CCSDS 500.0-G-4," 2019.

[4] F. deBruijn, E. Gill and J. How, "Comparative analysis of Cartesian and curvilinear Clohessy-Wiltshire equations," *Journal of Aerospace Engineering, Sciences and Applications,* vol. 3, p. 1–15, May 2011.

[5] L. Berger and J. P. How, "Gauss's Variational Equation-Based Dynamics and Control for Formation Flying Spacecraft," *Journal of Guidance, Control, and Dynamics*, pp. 437-448, 2007.

[6] R. Armellin, M. Massari and A. E. Finzi, "Optimal formation flying reconfiguration and station keeping maneuvers using low thrust propulsion," 2004.

[7] M. Tillerson and J. P. How, "Advanced guidance algorithms for spacecraft formation-keeping," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, 2002.

[8] E. C. P. O. W. Group, "Guidelines on Safe Proximity Operations," ESA, 2021.

[9] W. Fehse, Automated Rendezvous and Docking of Spacecraft, Cambridge University Press, 2003.