POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Propulsion System Modelling for Guidance and Control Realistic Simulation

Laurea Magistrale in Space Engineering - Ingegneria Spaziale

Author: Jacopo Ferro, Chiara Pasquariello

Advisor: Prof. Gabriella Vittoria Maria Gaias

Co-advisor: Prof. Camilla Colombo

Academic year: 2023 - 2024

1. Introduction

Space missions are becoming increasingly complex, demanding higher precision and reliability in spacecraft operations. At the heart of these operations is the Guidance and Control (G&C) system, which ensures that spacecraft can follow predetermined paths, maintain stability and achieve mission objectives. Given the importance of the accuracy with which the G&C is able to accomplish its requirements, it is fundamental to model more realistically the overall behaviour of the system. It is then of crucial importance that the configuration and type of actuators selected are able to realise the desired command while maintaining high accuracy on the control. This thesis considers orbit control systems that rely on thrusters as actuators, and it aims at developing a comprehensive tool that enhances the performances of G&C systems by optimising the use of thrusters for spacecraft control. To realise this objective, a high-fidelity model of the propulsion system shall be implemented, and a control allocation problem shall be solved. The novelty of the tool stands in the high generality with which it is developed, being able to handle a variety of different mission scenarios. To deal with these generality requirements, the thrusters were modelled considering as many characteristics as possible, and the method chosen to solve the control allocation problem is the Optimal Thrusters Combinations Table (OTCT) [1, 2].

The tool's application is demonstrated on the ESA's TriHex formation flying mission [3, 4], where a novel control law based on Relative Orbital Elements (ROE) is introduced to efficiently manage the J_2 perturbation, specifically targeting the differential effects on each spacecraft. The three-satellite of the formation fly on a General Circular Orbit (GCO) at a distance of 5 m: one of the main challenges of the G&C system is indeed represented by the high accuracy required by this close-proximity.

The next section presents the OTCT method, while Sec. 3 introduces the implementation of the tool. Sec. 4 highlights the performances and capabilities of the tool exploiting some simple application cases, and Sec. 5 presents the TriHex mission concept and discusses the results of the two simulations conducted on it.

2. OTCT method

The *control allocation problem* is the process of determining how to distribute control efforts

among thrusters to achieve the desired force and torque commands from the spacecraft's control system. Since the number of thrusters on a satellite is typically higher than the number of degrees of freedom to control, the problem has an infinite number of solutions. This allows the introduction of an objective function to minimise, in this case, fuel consumption: the selected thrusters will be the ones that guarantee the less expensive manoeuvres. The problem is formulated as in [1] and it is stated as:

$$\text{find}\min f \text{ s.t.} \begin{cases} f = \boldsymbol{c}^T \boldsymbol{t} \\ \mathrm{A}\boldsymbol{t} = \boldsymbol{u} \\ 0 < t_i < t_i^{max} \quad (i = 1, ..., n) \end{cases}$$

where A is the configuration matrix, whose column contains the forces and torques delivered by each thruster; $\boldsymbol{t}$ is the unknown vector of firing times; $\boldsymbol{u}$ is the ideal control command to match; $\boldsymbol{c}$ is a vector containing the efficiency of each thruster (inversely proportional to the specific impulse); and the constraints $t_i < t_i^{max}$ represent the physical limitation of the thrusters.
The idea of the OTCT method is to solve the optimisation on-ground, where the limits on the computational effort are not stringent. The aim of this ground optimisation - typically performed through Linear Programming (LP) techniques - is to pre-compute all the possible optimal thrusters combinations, saving them as tables that will be loaded on the on-board computer. Hence, the on-board algorithm will be based on simple matrix operations and it will result much quicker while maintaining the accuracy of the LP method used.

2.1. OTCT algorithms

The on-ground table construction is typically performed through a modified Simplex Method, an LP technique extensively explained in [5]. The main difference in the application of this method for the OTCT construction lies in the fact that the unknown of the problem is not the vector of firing times, but the so-called *optimal bases*, composed by the identifiers of the thrusters that have a nonzero firing duration value in the optimal solution. Indeed, the aim is to find and store all the possible optimal bases, that will fill the columns of the OTCT. Additionally, the command $\boldsymbol{u}$ is not a known variable and is subject to uncertainty. This uncertainty affects the firing duration but does not alter the optimal basis, thus it does not cause any difference in the results. On the contrary, it is this uncertainty itself that allows to find all the possible solutions, as explained in [1]: by properly managing the branching of the Simplex algorithm, the complete set of optimal thrusters combinations can be found. These bases will be stored both as vectors containing the indexes of the engines to activate - constituting the Optimal Thruster Combinations Table (OTCT) - and as inverses of the corresponding columns of the configuration matrix A, which will constitute the Inverse Matrices Table (IMT).
Having stored on-board the OTCT and the IMT, the real-time control allocation algorithm only performs two operations. First, it computes the firing times of all the optimal bases as $\boldsymbol{t} = \mathrm{B}\boldsymbol{u}$, where B is the inverse matrix considered. Then, it selects the optimal combination as the only one giving non-negative firing times.

3. OTCT implementation

To properly adapt the OTCT method to the framework of this thesis, the main requirements that the developed tool shall satisfy were identified. These include the following:

1. the tool shall effectively solve the control allocation problem, accounting for the thrusters' locations and orientations on the spacecraft;
2. it shall focus on the particular case in which only the orbit shall be controlled, without generating spurious torques on the platform;
3. it shall be able to serve a variety of different scenarios and to account for as many properties of the thrusters as possible;
4. it shall be adaptable to formation-flying missions;
5. it shall support also the use of electric thrusters.

Hence, the reason for choosing the OTCT method stands also in its adaptability. The on-board selection, in particular, can be easily modified and integrated with additional computations that allow to satisfy all of the above requirements.
The characteristics used to model the propulsion system are reported below.

- **Thrusters' position and orientation**: allows to represent the geometry of the propul-

sion system in the spacecraft's body frame.

- **Thrust type**: both fixed and variable thrust level engines are considered.
- **Thrust value(s)**: in the fixed thrust engine case, only one value of force is deliverable. In the variable thrust case, the minimum and maximum values of thrust achievable are modelling parameters.
- **Minimum thrust level variation**: most variable thrust engines can provide force in a discrete way, meaning that not any value of thrust can be realised.
- **Minimum Impulse Bit** (MIB): expressed in seconds as the minimum time for which the thruster must be activated.
- **Maximum firing time**: longest duration for which the thruster can operate continuously.
- **Efficiency**: linked to the specific impulse.

Additional inputs required by the tool are:

- **Spacecraft mass**: considered fixed in this work, but easily adaptable to variable mass cases.
- **Reaction wheel tolerance**: acceptance threshold on the spurious torque generated during the manoeuvre.
- **Rotation RTN to body frame**: needed to handle input commands written in the RTN frame.
- **Manoeuvre type**: for variable thrust cases, it can be *impulsive* - if firing times shall be minimised - or *continuous* - if the thrust level shall be minimised.
- **Minimum thrusters number**: needed to deal with contingency case operations, in which one or more motors are unavailable. It is typically set to $n_{min} = m + 1$, where m is the number of DOFs to control.

3.1. Algorithms

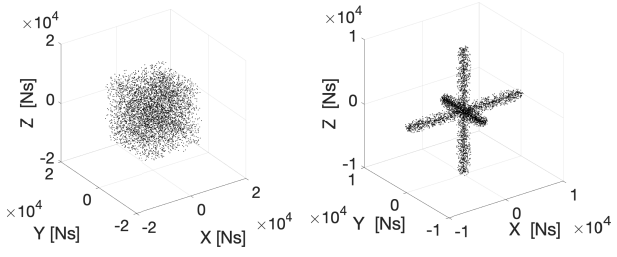
The following sections present the algorithms included in the tool, focusing on the aspects that differ from the standard OTCT method.

3.1.1 Configuration matrix construction

First, in order to properly apply the method, it is needed to know the configuration matrix A. An algorithm that allows to build it was therefore implemented taking [6] as reference. This takes as input the thrusters' positions and orientations along with the reference thrust value, which will be the minimum level possible in the variable thrust engine case. The output is the matrix containing the forces and torques delivered in the body frame along with a plot of the resulting configuration.

3.1.2 On-ground OTCT construction

Starting from the configuration matrix, the optimisation is performed with MATLAB's '`linprog`' function, with the '`dual-simplex`' algorithm option. This requires a definite value of $\boldsymbol{u}$ vector, thus, to be sure to find all the possible combinations, a search space is defined according to the mission requirements. This is composed of randomly generated points in the 6D space defined by all possible thrust and moment directions. In this work, the moments components are always set to 0, since no torque shall be generated by the optimal combinations (from requirement 3). As for the thrust components, these can be defined as points inside a square, a sphere, or along three envelopes aligned with radial, tangential and normal axes of the RTN frame. This last option is useful only when the rotation between RTN and the body frame is constant. Fig. 1 represents two of these options.



(a) Squared shape (b) RTN-aligned shape

Figure 1: Search space defined by all possible command vectors in body frame

Each of these possible commands is passed to the LP algorithm, and it will output an optimal combination. The combinations are stored in the OTCT and IMT table corresponding to the whole space spanning (Fig. 1a) or to the RTN-aligned envelopes (Fig. 1b).

A similar procedure is performed for the contingency case tables. Starting from n (total number of thrusters) and n_{min}, all possible combinations of off-motors are found. For each of these combinations, the columns of the corresponding off-motor in the configuration matrix A are sub-

stituted with columns of zeros. These matrices are then passed as inputs to the optimisation algorithm, which outputs the complete set of contingency case tables.
All the results are stored into structures, that will be loaded onto the Simulink® model.

3.1.3 On-board algorithm

The very simple procedure of Chapter 2 is expanded to account for other requirements and modelling parameters, such as variable thrust and instantaneous torque generation management. The on-board algorithms are developed in the Simulink® environment. They require as inputs: the characteristics of the platform and the thrusters, the whole set of tables previously computed, and the ideal $\Delta \boldsymbol{v}$ command (if in RTN frame it will be rotated to the body frame) and multiplied by the mass before the beginning of the selection algorithms. The onboard selection algorithm follows some predefined steps:

1. Select the correct table discerning between all the tables loaded, in this work: standard, RTN aligned, and contingency tables. This selection is mainly based on the direction of the ideal control command, the output of the G&C algorithm, and the eventual presence of non-usable motors.
2. Compute all the firing times for each optimal combination contained in the selected table. The bounds on the minimum and maximum firing times are imposed at this level for fixed-thrust engines.
3. Save the combinations with only positive firing times and confront them to choose the one that minimises the actuation error.
4. When dealing with variable-thrust engines, adjust the thrust level and the firing times to match the command enforcing: the technological bounds associated with each motor as well as the type of manoeuvre required (a more 'impulsive' one to minimise the manoeuvre duration or a 'continuous' one, that matches a specific manoeuvring time).
5. When the moment produced by the actuation either reaches instantaneous values, too high to be handled by the reaction wheels, or is unbalanced within the manoeuvre (the total moment produced is not null, very rare) the algorithm computes some motor's ignition delay with respect to the initial manoeuvre time. When a symmetric thruster configuration is mounted, this procedure enables keeping the moment generated null for the whole manoeuvre duration in almost all cases; while if the motors are mounted in a non-symmetric configuration it can only mitigate these effects.
6. The command, specifying the activation status and thrust level, is sent to each motor at the precise time instant.

4. Tool Applications

The proposed tool's capabilities are demonstrated through four simulated scenarios, which highlight its unique features for future users. These scenarios do not represent realistic situations but rather illustrate the tool's performance under different conditions.
The first three scenarios use a symmetric configuration matrix with fixed-thrust engines (7N) in the first two cases and variable-thrust engines in the third. The fourth scenario employs a Tri-Hex configuration matrix with variable-thrust engines, where all firing directions pass through the centre of mass, eliminating moment production. All simulations share the same initial spacecraft state and open-loop logic, with a randomly generated ideal command vector at a random time instant. Notably, in open-loop simulations, the `dt_command` input can be set to NaN, whereas it is essential in closed-loop simulations, representing the time between successive ideal command computations. In the on-board selection algorithm, `t_fir_max` can be substituted with `dt_command` when provided.

4.1. Moment balancing capability

This simulation consists of three distinct manoeuvres. Although the total generated moment is zero over each manoeuvre (balanced manoeuvre), instantaneous peaks occur when all motors are activated simultaneously. Two scenarios are compared: in the first, the reaction wheel tolerance is set to a very high value, allowing it to absorb all instantaneous torques generated by actuation. In the second case, the reaction wheel is insufficient, and the algorithm automatically computes ignition delays for some motors.
The results show a similar actuation error for thrust, approximately 0.1% of the ideal commanded value. In contrast, the error in moment varies significantly between the two cases: it is

in the order of 0.5% of the reaction wheel tolerance in the first case and 1e-12% in the second case, despite a lower reaction wheel tolerance. This demonstrates that the delay computation effectively maintains a null moment at each manoeuvre instant. Notably, in the first case, the reaction wheel saturates and de-saturates within each manoeuvre due to the balanced nature of the manoeuvres.

4.2. Firing times bounding

This simulation demonstrates the algorithm's performance when firing times exceed technological bounds. A single manoeuvre is simulated, aligned with the 5th thruster's direction. The required firing times to match the ideal command are 100s for motor 5 and 50s for motors 4 and 8. Two scenarios are compared:

1. Firing duration lower than MIB_0. With a MIB_0 of 51s, motors 4 and 8 need to fire for at least 51s, resulting in an unbalanced moment during the final second of the manoeuvre. The reaction wheel absorbs this moment, preventing desaturation. The error is approximately 1% of the ideal commanded thrust on the y and z axes, and 9% of the reaction wheel's capacity on the x-axis. This case usually corresponds to small manoeuvres, resulting in low errors, which can be adjusted in the next manoeuvre. The algorithm evaluates whether firing is convenient or not, in the presented case it does not fire for MIB_0 values above 55s.

2. Firing duration higher than engine limit. With a `t_fir_max` of 90s, the manoeuvre duration is bounded, and the error is unavoidable. The algorithm adjust the firing duration of motors 4 and 8 to match motor 5, maintaining a balanced configuration. The error committed is 20% of the ideal command z component. This situation occurs when a large manoeuvre is required, and the spacecraft is far from its nominal position. Not firing can create a hazardous situation, leading to an unmanageable cycle of actuator demands. Therefore, the algorithm always decides to perform the actuation, minimising the unavoidable actuation error.

4.3. Impulsive and continuous manoeuvres

Only one manoeuvre is programmed and the simulation is firstly run envisioning an 'impulsive' manoeuvre. Then, a second simulation is performed with a more 'continuous' one, imposed to last around 1000s, coinciding with the maximum firing time of the motors. From the result obtained, we can see that the manoeuvre passes from a duration of some seconds, when the thrusters fire at their highest thrust level, to a duration of more than 500s in the second case when they fire at their lowest level.
The percentage of thrust error committed varies significantly between the 'impulsive' and 'continuous' cases. Specifically, the error decreases from 13% of the ideal command y component in the 'impulsive' case to 1% in the 'continuous' case. This reduction is attributed to the truncation error, which is larger when firing at high thrust levels.

4.4. Contingency case

The last simulation aims to show that the algorithm can deal with contingency cases, in which one or more engines are not working, highlighting both the effectiveness of the tables constructed and the configuration used. In the first simulation, a nominal situation with all thrusters available, the fuel-optimising set of firing motors are 1, 2 and 3. In the second and third simulations motors 3 and 2 are set off respectively and substituted by 6 and 5. The results of the three simulations are similar in terms of actuation error enabling the mission to continue also in off-nominal situations.

5. Application to Tri-Hex mission

The developed tool is applied to the TriHex mission concept, which is being developed by ESA with the aim of advancing the capabilities of passive L-band observations from space. It envisions three platforms orbiting on a GCO around a virtual chief, located at the centre of the RTN frame. The chief follows a circular trajectory in the LEO region, while the three deputies move on a circle of radius $\rho = 2.8868$ m. In the unperturbed motion case, the platforms remain at a distance of 5 m, but the perturbations cause the satellites to deviate from the nominal trajectory: to accomplish mission objectives, the G&C shall maintain the nominal distances with high accuracy.
In this framework, the tool is applied in two

different simulations. The first one adopts an open-loop control strategy, with no feedback on the real position of the platforms and no control logic; the second one, instead, is based on a closed-loop G&C framework, in which a novel control law is designed. In both cases:

- a symmetric configuration of the propulsion system is adopted
- all thrusters point towards the centre of mass, hence no torque can be generated
- the reference for the engine model is the the RIT-μX, from AGG-Germany[1]
- the results of the on-ground algorithms yields 18 optimal combinations in the standard whole-space spanning case, and a total of 83 contingency combinations. The configuration is shown in Fig. 2.

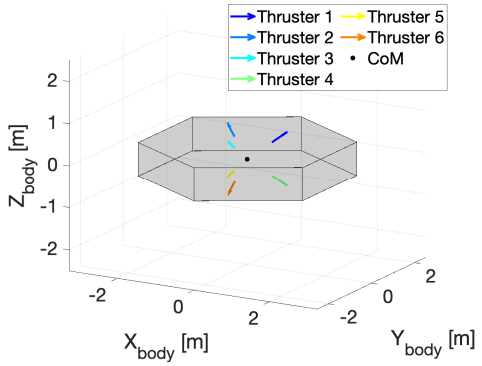


Figure 2: TriHex's thrusters configuration

5.1. Open-Loop Simulation

In the open-loop case, the commands are computed separately from the simulation environment, and given directly as input to the model. These ideal commands are based on a dynamics model that is much more precise than the one here adopted. The aim of the simulation is thus to evaluate the feasibility of this realistic control when the propulsion system is more accurately modeled, along with the assessment of the tool's performances. The actuated thrust profile is shown in Fig. 3, while Fig. 4 represents the error of the actuated command with respect to the reference ideal one.

The mismatch on the thrust profile is believed to be caused by the rotation matrix from RTN to body frame. Indeed, this matrix was built assuming that all the spacecraft remain perfectly on the GCO. Due to the different dynamical models used for the command computation and the present simulation, the control is not able to perfectly match the GCO, causing the rotation to introduce some errors on the thrust pointing. Note that in the x-axis the ideal thrust is null.

[1] ArianeGroup_RIT-μX_Datasheet.pdf, last access on 26/08/2024

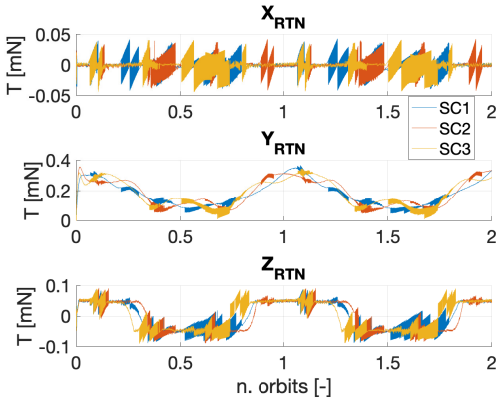


Figure 3: Actuated thrust

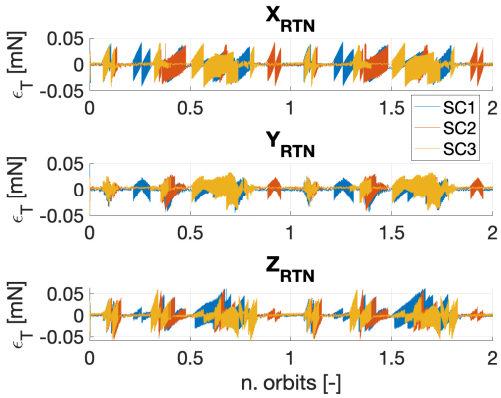


Figure 4: Error on thrust command

5.2. Closed-Loop Simulation

The second simulation performed is based on a closed-loop control strategy. The loop is closed with a newly designed control law, aiming at counteracting only the relative J_2 effects on the GCO. The controller is based on an analytical prediction of these effects and exploits the natural dynamics of the relative orbit. The idea of the simulation is to verify the performance of the controller.

5.2.1 Controller design

The proposed control law is built to focus only on the contributions that deviate the relative

motion, causing the distance between the spacecraft to change. Hence it does not cancel out the whole J_2 perturbation. The analytical model of these effects, based on ROE, is presented in [7]: the main issue for the GCO is represented by relative eccentricity ($\delta\boldsymbol{e}$) and inclination ($\delta\boldsymbol{i}$) vectors' change. Indeed, $\delta\boldsymbol{e}$ vector tends to move along a circle of radius δe, centred in the origin of $(\delta e_x, \delta e_y)$ plane, with a velocity of $\dot{\varphi}$, which depends on the characteristics of the absolute reference orbit as in Eq. (1), where $\gamma = \frac{J_2}{2}\frac{R_E^2}{a_c^2\eta^4}$ and $\eta = \sqrt{1-e_c^2}$ - subscript c indicating the chief's quantities [8].

$$\dot{\varphi} = \frac{3}{2}\gamma(5\cos^2 i_c - 1.0) \tag{1}$$

As for the relative inclination vector, the y-component of $\delta\boldsymbol{i}$ vector tends to increase or decrease according to the initial value of the x-component, which instead remains constant.
The conditions for the deputies to follow a GCO can be written in terms of phase angles and amplitudes of the relative motion. Translating them into ROE description, the result in Eq. (2) is obtained, with $c_1 = \rho$ and $c_2 = (\sqrt{3}/2)\cdot\rho$. The phase angles φ and θ derive from the polar representation of $\delta\boldsymbol{e}$ and $\delta\boldsymbol{i}$.

$$\begin{cases} a_c\delta a = 0 & a_c\delta\lambda = 0 & a_c\delta e = c_1/2 \\ a_c\delta i = c_2 & \varphi = 3\pi/2 - \alpha & \theta = -\alpha \end{cases} \tag{2}$$

Hence, the idea of the controller is to let the phase of $\delta\boldsymbol{e}$ vector change due to J_2, since the variation is the same for the three deputies and it is not causing the relative distances to change. Then, this phase change is used to find the new phase of $\delta\boldsymbol{i}$ vector that satisfies the conditions to have a GCO. These conditions are then transformed into RTN state and used as a reference to build a simple PD controller with feed-forward on the velocity. The steps are summarised in Algorithm 1.

5.2.2 Simulation

To verify the effectiveness of the control law, 25 revolutions were simulated. The results are appreciable in the following plots. Fig. 5 shows the evolution of the real ROE (evaluated after the actuation block which includes the developed tool): only small oscillations are introduced by the tool. Fig. 6 presents the resulting distance between the platforms, showing also the admissible ±15mm variation from the nominal 5m (the dashed black lines): it can be noticed how the requirement is always respected.

Algorithm 1 Control law design

1: Compute the new phase of $\delta\boldsymbol{e}$ as $\varphi_{new} = \varphi_0 + \dot{\varphi}\cdot n_c(t - t_0)$
2: Compute the new phase of $\delta\boldsymbol{i}$ as $\theta_{new} = \varphi_{new} - 3\pi/2$
3: Compute the new $\delta\boldsymbol{e}$ and $\delta\boldsymbol{i}$ vectors keeping constant the magnitudes
4: Use the so-defined reference ROE to get the deputies' absolute orbital elements
5: Transform the orbital elements into inertial state, then into RTN state
6: Evaluate the error between real and reference state as $\boldsymbol{\varepsilon} = \boldsymbol{x}_{RTN}^{ref} - \boldsymbol{x}_{RTN}$
7: Compute the ideal command with a PD
8: Create the output $\Delta\boldsymbol{v}_{RTN}^{ideal}$

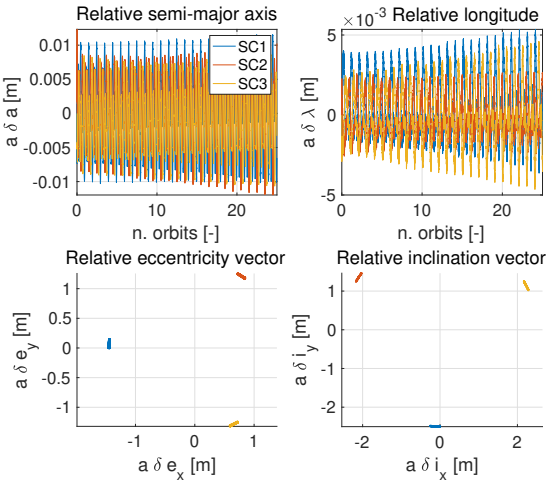


Figure 5: Actuated ROE evolution

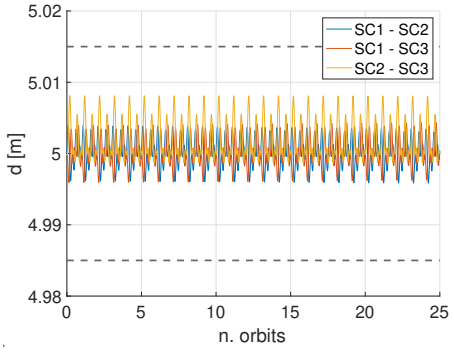


Figure 6: Actuated distance between platforms

From these results, it can be stated that the controller can effectively maintain the distances between the platforms well inside the admissible range, with short-term oscillations and very small long-term drifts. Due to its focus on the relative effect of J_2 only, the fuel consumption for formation maintenance results lower than with a control that cancels out the complete perturbation, constituting hence an interesting starting point for a future complete controller development.

6. Conclusions

This work presents a comprehensive tool for optimising the performance of generic G&C systems for spacecraft control, focusing on the efficient use of thrusters and realistic emulation of the thrust command. The proposed tool, based on the Optimal Thrusters Combinations Table (OTCT) method, has been designed to be adaptable to various space missions and to represent the behaviours of different propulsion systems, accounting for many properties of the thrusters. Key novelties introduced include:

- the ease of implementation through various missions and the high level of generality of the tool, which is able to serve a variety of different scenarios by only adjusting the input parameters;
- the extension of the standard OTCT method to more complete models of the propulsion systems, included both in the on-ground and onboard algorithms;
- the newly designed controller, which allows to counteract only the relative effects of J_2 perturbation by including its analytical model in the control law and by exploiting the natural motion of the GCO trajectory.

Future improvements and expansions of the proposed tool could explore alternative optimisation techniques (in particular for non-linear objective function and constraints), implement moment control, consider variable mass cases, include more thrusters' properties for the propulsion system model (such as pointing errors, delays, etc.), develop an optimal configuration selection algorithm that relies on a faster onground optimisation, or investigate the inclusion of an interface with the navigation subsystem of GNC simulations.

In general, the tool developed is believed to represent a good starting point for future guidance and control realistic simulations.

References

[1] M. Wang et al. "Design of the Optimal Thruster Combinations Table for the Real Time Control Allocation of Spacecraft Thrusters". In: *Proceedings of the IEEE Conference on Decision and Control*. Dec. 2009, pp. 5063–5068. DOI: 10.1109/CDC.2009.5400664.

[2] M. Wang et al. "Control capability analysis for complex spacecraft thruster configurations". In: *Science China Technological Sciences* 53 (Aug. 2010), pp. 2089–2096. DOI: 10.1007/s11431-010-3139-9.

[3] M. Martin-Neira, F. Scala, et al. "TriHex: Combining Formation Flying, General Circular Orbits, and Alias-Free Imaging, for High-Resolution L-Band Aperture Synthesis". In: *IEEE Transactions on Geoscience and Remote Sensing* 61 (2023). ISSN: 15580644. DOI: 10.1109/TGRS.2023.3268560.

[4] F. Scala. "Multiple Satellites Formation Flying for Earth Observation Applications in Low Earth Orbit". PhD thesis. Politecnico di Milano, Department of Aerospace Science and Technology, 2022.

[5] H. A. Taha. *Operations Research: An Introduction*. Ed. by J. Partridge. 10th ed. Pearson Education Limited, 2017, pp. 99–191. ISBN: 9780134444017.

[6] M. Wang et al. "A Thruster Configuration Method for Spacecraft Six-Dimensional Control". In: *2023 6th International Symposium on Autonomous Systems, ISAS 2023*. Institute of Electrical and Electronics Engineers Inc., 2023. ISBN: 9798350316155. DOI: 10.1109/ISAS59543.2023.10164488.

[7] S. D'Amico. "Autonomous Formation Flying in Low Earth Orbit". PhD thesis. TU Delft, 2010.

[8] G. Gaias et al. "Model of J2 perturbed satellite relative motion with time-varying differential drag". In: *Celestial Mechanics and Dynamical Astronomy* 123 (4 Aug. 2015), pp. 411–433. ISSN: 15729478. DOI: 10.1007/s10569-015-9643-2.