CEAS Space Journal
https://doi.org/10.1007/s12567-024-00560-0

ORIGINAL PAPER

Satellite and robotic arm combined control for spacecraft close-proximity operations

Federico Basana[1] · Zeno Pavanello[2] · Francesco Branz[3] · Alessandro Francesconi[3] · Giacomo Borelli[4] · Davide Invernizzi[4] · Mauro Massari[4] · Marco Lovera[4] · Alessia Nocerino[5] · Roberto Opromolla[5] · Irene Huertas Garcia[6] · Pedro Simplicio[6]

Received: 22 February 2024 / Revised: 19 July 2024 / Accepted: 22 July 2024

Abstract

This work presents the development of the navigation and control subsystems of a guidance, navigation and control (GNC) system for controlling an autonomous satellite, called chaser, equipped with a redundant manipulator. In the study, the capture operations of a target spacecraft and its stabilization are considered. The control function employs a combined control strategy: the chaser satellite actuators and the robotic arm joint motors are degrees of freedom of the same control plant. Robust control methods are used to cope with uncertain, nonlinear dynamics of the chaser and of the complete chaser–target stack after capture. The developed navigation function is based on active or passive electro-optical sensors (i.e., passive cameras and/or LIDARs) for the relative pose determination. A numerical simulator capable of representing the dynamics of the system (chaser, manipulator, target) in orbit is developed in the MATLAB/Simulink environment to validate the developed GNC algorithms. The simulation tool allows to thoroughly assess the GNC system performance, accounting for all the relevant external disturbances and error sources. The simulator interfaces with a synthetic image generator (i.e., PANGU) for relative navigation performance assessment. This paper presents the testing of the GNC system in two relevant In-Orbit Servicing (IOS) scenarios: (1) the servicing of a large GEO platform and (2) the servicing of a small satellite in Low Earth Orbit (LEO).

Keywords Close proximity operations · In-orbit servicing · Combined control · Robotic arm · Visual-based relative navigation

✉ Federico Basana
federico.basana@phd.unipd.it

Zeno Pavanello
zpav176@aucklanduni.ac.nz

Francesco Branz
francesco.branz@unipd.it

Alessandro Francesconi
alessandro.francesconi@unipd.it

Giacomo Borelli
giacomo.borelli@polimi.it

Davide Invernizzi
davide.invernizzi@polimi.it

Mauro Massari
mauro.massari@polimi.it

Marco Lovera
marco.lovera@polimi.it

Alessia Nocerino
alessia.nocerino@unina.it

Roberto Opromolla
roberto.opromolla@unina.it

Irene Huertas Garcia
Irene.Huertas@esa.int

Pedro Simplicio
Pedro.Simplicio@ext.esa.int

[1] Centre of Studies and Activities for Space “Giuseppe Colombo” (CISAS), University of Padova, via Venezia 15, 35131 Padova, Italy

[2] Te Pūnaha Ātea - Space Institute, The University of Auckland, Auckland 1010, New Zealand

[3] Dip. di Ingegneria Industriale, University of Padova, via Venezia 1, 35131 Padua, Italy

[4] Dip. di Scienze e Tecnologie Aerospaziali, Politecnico di Milano, via La Masa 34, 20156 Milano, Italy

[5] Dip. di Ingegneria Industriale, Università degli Studi di Napoli “Federico II”, via Claudio 21, 80125 Naples, Italy

[6] ESA–ESTEC, European Space Agency, Keplerlaan 1, Noordwijk NL-2200, The Netherlands

Published online: 03 August 2024

1 Introduction

In-Orbit Servicing (IOS) refers to the in-orbit maintenance of space systems after their deployment, including repair, assembly, refueling and/or upgrade. An IOS mission typically involves two space vehicles, namely a chaser and a target. To perform servicing operations, a mechanical connection is established between the two, usually through a robotic arm. IOS missions usually consist of multiple phases, for example: (i) orbit transfer and phasing, (ii) far- and mid-range rendezvous, (iii) inspection, (iv) close range rendezvous, (v) reach and capture, (vi) manipulator stiffening and target stabilization, and (vii) servicing/deorbiting [1].

The interest in IOS missions has grown in the past years since they offer the possibility of extending the operational life of spacecraft or of disposing of debris. As a consequence of a servicing mission, the revenue from existing space assets is increased and the risk of generating space debris after a satellite failure is reduced. The key technologies developed in the framework of IOS can be applied to different scenarios. For instance, they can be tailored to the mission extension and servicing of large geostationary communication platforms. In Ref. [2], the cost of replacing failed geosynchronous equatorial orbit (GEO) satellites is traded with the returns that can be achieved through an IOS mission. These returns can take the form of revenue, scientific data, or continued in-orbit capability. The study demonstrates the cost effectiveness of servicing missions. In this framework, MEV-1 and MEV-2 missions proved the feasibility of IOS by relocating and keeping the orbit of GEO satellites [3]. Several studies across the global space community have been dedicated to IOS mission architectures, e.g., DLR DEOS mission [4] and IOS missions have been proposed such as e.Deorbit [5].

Considering a different orbital application, technologies developed for IOS missions can be tailored to operate on a large constellation of small platforms. In the last years, the population of active objects in LEO has increased due to the deployment of large constellations of small satellites. To ensure the sustainability of the future crowded orbital regions, the interest among constellation operators is towards a service able to maintain the health of the constellation and guarantee the compliance with debris mitigation guidelines [6]. The objective is the development of a servicer platform used to maintain an operative constellation, providing services such as repair, refueling, and deorbiting in case of failure of a satellite [7].

Coupling a satellite with a manipulator creates challenging conditions on the GNC system. The traditional approach is to provide the satellite with a dedicated GNC system which works separately from the robotic arm control. The robotic arm is kept in a dormant state until the chaser is close to the target, at which point the chaser GNC is deactivated, and the robotic arm is activated. The motion of the robotic arm generates forces/torques at its base that act as perturbations on the satellite base. Thus, the satellite changes its attitude and the robotic arm must compensate the base movements to perform correctly the desired motion. Although the chaser may have access to additional actuation systems such as thrusters, the robotic arm GNC does not or cannot make use of them. This is referred to as *decoupled control*. An alternative is the *combined control* strategy, where the chaser satellite actuators and the robotic arm joint motors are seen as multiple degrees of freedom of the same control plant. The combined controller can then make use of the chaser thrusters at the same time as the robotic arm is extending to grab the target.

1.1 Objectives of the study

This study focuses on the development of enabling GNC technologies suitable for IOS missions conducted by a chaser spacecraft equipped with a seven Degrees of Freedom (DoF) robotic arm. Specifically, it deals with electro-optical (EO)-based relative navigation and combined control aspects. The key aspects of the developed GNC system are: (1) the adoption of a multi-variable combined control approach that represents a step forward with respect to the state of the art and proves advantageous in terms of performances, resources management and safety and (2) a fully autonomous navigation architecture based on active or passive EO sensors, which pushes forward the state of the art of technologies for final approach and docking operations, especially when dealing with non-cooperative targets.

The developed algorithms are designed to carry out the operations in the chronological proximity of the capture event, considering highly relevant mission scenarios spanning from partial- to non-cooperative and partial- to non-collaborative target objects (Table 1). Operations are organized in a *pre-capture phase* and a *post-capture phase*: the first includes reach and capture operations, the second deals with robotic arm stiffening and target detumbling. The presented simulations focus mainly on the capture and stiffening subphases, which involve the actuation of both the robotic arm and the chaser GNC system. Both the relative navigation and combined control approaches are accurately tested by means of a complete custom simulation tool. The study is conducted under the European Space Agency (ESA) contract by a consortium of Italian universities composed by teams from the Università degli Studi di Padova, the Politecnico di Milano and the Università degli Studi di Napoli “Federico II”.

Table 1 Definition of *collaborativeness* and *cooperativeness* of a target for close-proximity operations

Target type	Description
Cooperative	The target provides direct information about its relative states to the servicer in real time
Semi-cooperative	The target provides indirect information about its relative states to the servicer by means of active/passive markers
Non-cooperative	The target does not offer any support to the relative navigation
Collaborative	The target actively and accurately maintains an attitude profile that support the approach and docking/capture process
Semi-collaborative	The target actively keeps an attitude profile to aid the approach but not accurately enough for the docking/capture process (coarse attitude control)
Non-collaborative	The target attitude is uncontrolled and it cannot aid the capture operation

In Ref. [8], the preliminary results of unit-level tests regarding a preliminary version of the GNC algorithm presented in this article are shown. Specifically, in Ref. [8], the tests were carried out in open-loop to validate a first version of the GNC algorithm. On the other hand, the development of a simplified simulator for validating GNC algorithm is presented in Ref. [9]. This paper reports the detailed description of the matured guidance, navigation, and control approaches. In particular, the navigation algorithm and the guidance and control algorithm architectures are the same as the ones used in this article. However, in Ref. [8], they are not integrated in a single GNC algorithm. In this paper, the two algorithms are: (1) integrated in a single GNC algorithm, (2) optimized to make the navigation and the guidance and control work together and (3) tuned as a closed-loop control system. In addition, the GNC algorithm presented in this work is tested within a complete functional engineering simulator developed starting from the one introduced in Ref. [9]. The test campaign aims at verifying the functionality of the GNC algorithm. The test campaign includes: (1) closed-loop simulations of the integrated GNC pipeline under nominal conditions in terms of uncertainties, (2) an error budget analysis to evaluate the relative importance of the different contributions to the overall control error, and (3) a preliminary Monte Carlo analysis to assess the architecture's robustness against increased level of uncertainties.

This paper presents a combined control approach to the space capture problem and is organized as follows: Sect. 2 presents the two scenarios considered in this study; Sects. 3 and 4 describe the developed control and navigation functions, respectively; Sect. 5 provides details on the simulation environment implementation and testing; Sect. 6 introduces the numerical simulations executed and presents their results; in Sect. 7, conclusions are drawn.

2 Scenarios definition

Two different scenarios are considered: the IOS of a large platform in GEO (SC1) and the IOS of the member of a constellation in LEO (SC2). For both the considered scenarios, the study focuses on the capture of the target spacecraft and the manipulator stiffening after the capture. In the former, the chaser is maintained at a very short distance from the target by the GNC system, while the robotic arm reaches and grasps the target on the selected capture point. In the latter, the chaser actuates the robotic arm to damp the residual angular rates between chaser and target. In both phases and scenarios, the design of the navigation and control subsystems is strongly affected by the degree of cooperation and collaboration of the target. The two representative IOS scenarios have been defined focusing on the challenges posed by the lack of cooperativeness and collaborativeness in space robotics operations. Table 2 summarizes the two scenarios which are presented in detail in Sects. 2.3 and 2.4, respectively.

2.1 Definition of the main reference frames

A set of reference frames are used to describe the GNC architecture and to present the results. The following list reports the complete set of reference frames considered (see Fig. 1) and gives their definitions, while Table 3 reports the notation used for describing the multibody mechanics. The origin of each reference frame is attached

Table 2 Scenarios definition. SC1 considers the servicing of a large GEO platform; SC2 takes into account servicing a small platform of a large constellation in LEO

Scenario	Mission type	Cooperativeness	Collaborativeness	Target
GEO	IOS	Semi-cooperative	Semi-collaborative	SSL-1300 GEO platform
LEO	IOS/ADR	Semi-cooperative	Non-collaborative	Arrow platform (OneWeb)

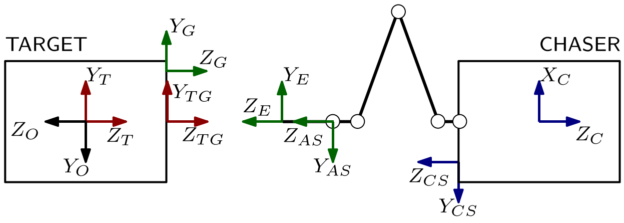


Fig. 1 Reference frames used by the GNC algorithm. All the frames are right-handed

to a body of the system. The frame rotates rigidly with the body geometry, except for the Orbital and Hill frames (O, H) that are orbital frames.

- Chaser rotating orbital frame (C): it is centered in the geometric center of the chaser bus.
- Chaser Sensor fixed frame (CS): it defines the position and orientation of the camera that is attached to the bus.
- End-effector frame (E): it is centered on the manipulator end-effector and it is useful to represent the pose of the end-effector with respect to frame C.
- Arm Sensor fixed frame (AS): it defines the position and orientation of the camera attached to the robotic arm. It is a fixed frame with respect to the E frame.
- Target Geometric fixed frame (TG): it is a fixed frame centered on one of the faces of the target spacecraft.
- Target rotating orbital frame (T): it is centered in the geometric center of the target bus.
- Grasping point frame (G): it is centered on the designed point for the robotic grasping operation.
- Orbital frame (O): it is a frame coincident with the standard LVLH orbital frame in which $\hat{\mathbf{z}}_O$ is towards the direction of the Nadir, $\hat{\mathbf{y}}_O$ is opposite to the direction of the orbital angular momentum and $\hat{\mathbf{x}}_O = \hat{\mathbf{y}}_O \times \hat{\mathbf{z}}_O$.
- Hill frame (H): it is a frame in which $\hat{\mathbf{x}}_H$ points outward in the direction that connects the center of the planet with the spacecraft, $\hat{\mathbf{z}}_H$ points in the direction of the angular momentum and $\hat{\mathbf{y}}_H$ completes the right-handed frame.
- Earth Centered Inertial (ECI) frame: it has the origin at the center of mass of the Earth. The X-axis points at the vernal equinox, the Z-axis points in the direction of the geographical North Pole and the Y-axis completes the right-handed frame.

2.2 Chaser architecture common to the two scenarios

The general structure of the chaser spacecraft is common to the two scenarios. The chaser mass is proportional to that of the target. In the GEO scenario, the chaser also comprises two large solar arrays (SA). To perform the attitude control maneuvers, four reaction wheels (RWs) are adopted. A reaction control system thruster assembly is used to execute orbital maneuvers and to desaturate the RWs when necessary. The minimum number of thrusters to achieve a comprehensive decoupled control of both translational and rotational motion of the chaser is twelve, two for each direction. Nonetheless, a high level of redundancy is needed during the proximity operations, where the whole control authority capability shall be maintained even in the case of a thruster failure along the mission, thus a total of 24 thrusters are considered. Three thrusters are located at each solid angle of the chaser in an XYZ configuration.

The robotic arm is mounted on one side of the spacecraft and it is composed of seven revolute joints, thus having seven DoF, as shown in Fig. 2. The brushless Direct Current (DC) motors on the joints can reach a maximum torque of 10 N m. The length of the links varies according to the scenario.

2.3 Scenario 1: servicing of a large GEO platform

The first scenario considers the refueling mission of a SSL/1300 GEO platform. The serviced GEO satellite is considered operational, so its attitude is nadir-pointing with an accuracy below 1 deg. During the maneuver, the

Table 3 Notation used for the multibody mechanics

Notation	Meaning
$\vec{r}^K_{I/J}$	Position of frame J with respect to (w.r.t) I expressed in the frame K
$\vec{v}^K_{I/J}$	Linear velocity of frame J w.r.t I expressed in frame K
$\vec{\omega}^K_{I/J}$	Angular velocity of frame J w.r.t I expressed in frame K
$\vec{\eta}_{I/J}$	Unit quaternion describing the rotation from frame J to I. The convention $\vec{\eta} = [\eta_0\ \vec{\eta}^\top]^\top$ is adopted, where $\eta_0 \in \mathbb{R}$ and $\vec{\eta} \in \mathbb{R}^3$ are the scalar and vector part of the quaternion, respectively
R^B_A	Rotation matrix that transforms the coordinates of a vector from frame A to frame B
$\vec{q}$	Joint angles
$\dot{\vec{q}}$	Joint velocities

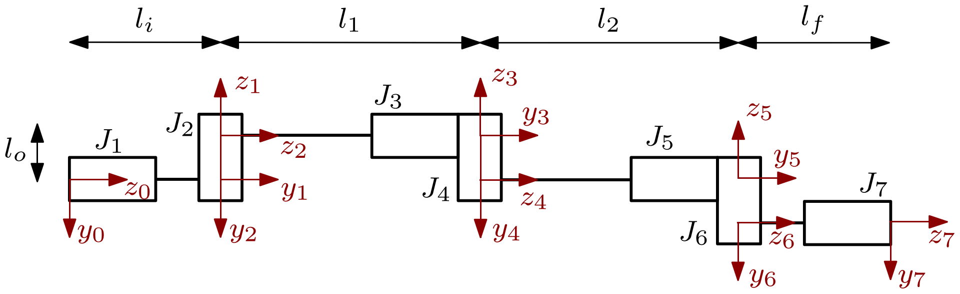


Fig. 2 Robotic arm configuration. It is composed by 7 links connected by 7 revolute joints. The length of the links depends on the scenario considered

Table 4 Chaser (C), target (T) and arm (A) characteristics of the two considered scenarios. The table reports the wet mass of C and T at the beginning of the capture phase

	Parameter	SC1	SC2
C	Mass [kg]	1900	372
	Bus dim. [m]	[2;2.3;3.1]	[1.3;1.3;1.7]
	SA dim. [m]	[11;2.1;0.01]	/
	MoI [kg m^2]	[13.5;2;14] × 10^3	[200;200;140]
	PoI [kg m^2]	[0;0;0]	[0;0;0]
T	Mass [kg]	2000	150
	Bus dim. [m]	[2.5;2.8;3.5]	[1;1;1]
	SA dim. [m]	[14;2.5;0.01]	[1;1;0.01]
A	l_i[m]	0.3	0.19
	l_1[m]	1.15	0.8
	l_2[m]	1.15	0.8
	l_f[m]	0.4	0.21
	l_o[m]	0.3	0.16

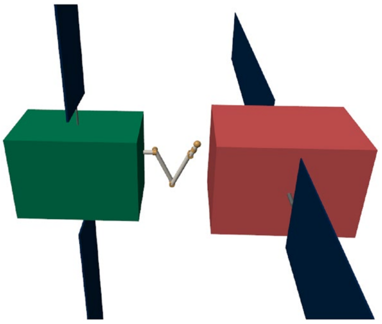


Fig. 3 Representation of the SC1 at the beginning of the capture phase. The chaser is represented in green, while the target is represented in red. The solar panels are cropped due to their dimensions

inter-satellite distance, i.e., the distance between the C and the T frames, is maintained at 5.1 m. The satellite is semi-cooperative as ten retroreflectors that can be illuminated by an infrared laser source on the chaser and three passive markers form two sets of fiducial markers that aid the capture maneuver. Table 4 reports the physical and geometrical characteristics of the spacecraft.

The chaser platform design for the GEO scenario is derived from literature [10]. The spacecraft total launch mass is 2500 kg. Since the servicer is assumed to be performing refueling of the GEO platforms, its mass is estimated considering 400 kg of bipropellant for refueling purposes. This value accounts for two refueling operations in GEO, which would extend the life of two GEO platforms by at least 5 years [11]. The bipropellant is stored in two spherical tanks sized for 200 kg. In addition, 400 kg of propellant is considered necessary to transfer the servicer from the geostationary transfer orbit to the GEO orbit. Therefore, the chaser mass is considered at the beginning of the scenario, i.e., when it starts the reach and capture phase of the second service, with a total mass at service equal to 1900 kg. The choice to study the second service is motivated by the fact that, from a combined control point of view, it is more critical than the first one. In fact, the refueling tanks at the beginning of the second service are at half capacity (100 kg of the total 200 kg for each tank), which is a worst-case scenario for the sloshing attitude disturbance [12]. At the beginning of the simulation, the end-effector is distant 0.54 m from the grasping point. The maneuver to grasp the target last 75 s. The robotic arm length, when fully extended, is 3 m; the breakdown of the length of each link is provided in Table 4. Figure 3 provides a representation of system at the beginning of the capture phase using simplified geometries.

2.4 Scenario 2: servicing a small platform of a large constellation in LEO

The second scenario represents a servicing mission in LEO, particularly tailored to a small platform of a large constellation (i.e., the Airbus Arrow platform, OneWeb constellation) which is on a nominally circular orbit at 1210 kg of altitude and with an inclination of 88 deg. The selection is considered representative of the constellation satellites configurations (IRIDIUM, GLOBALSAR, Starlink). The target is represented as a 1x1x1 m box with two deployed solar panels (Table 4) attached to it. These are connected to the target by two beams of length 0.5 m. Hence, the total length of the appendages is 1.5 m. In this scenario the target is prepared for servicing, featuring a grapple interface and fiducial markers across the spacecraft body as navigation aid. During the mission phase, the target is considered non-collaborative. As nominal conditions for the scenario, the target satellite rotates with angular rates up to 2.5 deg/s along the Y-axis of the target rotating orbital frame (T). The target is not controlled. Therefore, the GNC design and architecture can be generalized for both an IOS and Active Debris Removal (ADR) mission. The simulations performed do not consider the synchronization maneuvers as the capture phase starts after them. At its beginning, the C frame is 2.5 m far from the T frame; the distance is maintained during the maneuver.

The chaser platform is designed based on a literature review process (Table 4). In the considered mission, the chaser services multiple satellites of the constellation. From Ref. [13], the baseline value of 500 kg wet mass is considered, which allows the servicing of two targets (assuming a dry mass of 245 kg). The nominal baseline value for the chaser mass is 372 kg, which corresponds to the mass left after the propellant expenditure for reaching the first satellite, servicing it and reaching the second one. Similar to the GEO scenario, this selection implies that the propellant mass at the beginning of the scenario will be half of the capacity of the tanks, which poses a more challenging sloshing disturbance to the GNC system. Body-fixed solar panels are employed for power generation. They offer advantages in terms of safety during the approach and capture operations, as well as in reducing the influence of flexibility on the control system. The distance between the end-effector and the grasping point is 0.3 m, which is traveled in 20 s. The robotic arm length, when fully extended, is 2 m; the breakdown of the length of each link is provided in Table 4. Figure 4 provides a representation of system at the beginning of the capture phase using simplified geometries.

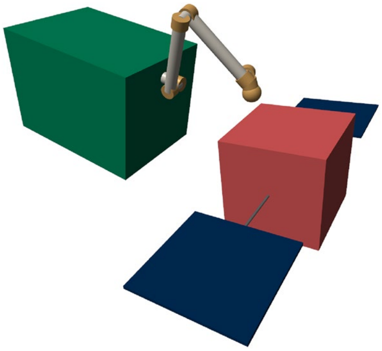


Fig. 4 Representation of the SC2 at the beginning of the capture phase. The chaser is represented in green, while the target is represented in red

3 Guidance and control

This section is devoted to presenting the design and synthesis of the control laws. First, the derivation of suitable control-oriented models is discussed, then the control strategies that have been developed to address the two different phases of the mission are presented. The proposed design is based on a combined approach wherein base and manipulator states are controlled together, following ideas recently proposed in the literature, and exploiting a robust synthesis procedure to compute the control law gains. Using a combined architecture has several advantages over decoupled control strategies, from fuel efficiency to performance improvement, as shown by previous works [14], where a task-space formulation is considered with emphasis on control coordination of the different actuators. In Ref. [15], structured H_∞ synthesis has been proposed to design independent controllers for base spacecraft and its six DoFs manipulator, while μ-analysis has been used to evaluate stability and performance properties according to different configurations of the manipulator (see Sect. 3.4 and [16] for details on H_∞ and μ-analysis). Reference [17] is one of the most important study on combined control using robust control theory. The main goal of Ref. [17] was to design and test combined controllers for active debris removal; both linear H_∞ and nonlinear Lyapunov-based control strategies have been considered. In Ref. [18], a complete methodology for modeling for on-orbit servicing missions is developed, accommodating uncertainties and changes in system dynamics in an efficient full analytical Linear Fractional Representation (LFR) model. In Ref. [19], the capturing problem is solved with a coordinate-free task-space controller that directly tracks the outputs provided by an EKF that estimates the target and grasping point pose.

3.1 Combined control system design

The specific approach developed in this work consists in using nonlinear control laws, based on extensions of the well-known computed torque controller to space robot [20–23], together with a systematic tuning procedure based on the H_∞ framework. Indeed, while computed torque controllers deliver good tracking performance in a large domain of operating conditions, they suffer from modeling uncertainty (they are based feedback linearization) and no rule is given to tune the gains of the feedback component of the control law, which is typically based on a (nonlinear) Proportional Derivative (PD) law. Hence, trial and error procedures are often employed in practice to select the gains and achieve satisfactory performance.

Such an approach is made more challenging by the large number of states of space robots. Lyapunov-based controllers provide energy-saving solutions by avoiding compensation of non-harmful nonlinearities, yet they require knowledge of system inertial parameters for trajectory tracking tasks and careful tuning in practice, when unmodeled dynamics are considered.

Therefore, the following systematic tuning approach has been considered: first, both the plant and the control law are linearized about a nominal operating point and a linear uncertain description of the closed-loop system is derived; then, the gains of the control law are tuned by leveraging the structured H_∞ framework. In this manner, the control law handles by design the rigid body nonlinearities, while performance requirements can be imposed in the neighborhood of the desired configurations when tuning the gains. The proposed approach allows accounting for dynamics effects at synthesis time, such as sloshing, actuator dynamics, flexibility, orbital dynamics, which are neglected when deriving the nonlinear control law.

3.2 Control-oriented modeling

Given the complexity of the multibody system under investigation, a simplified mathematical model has been developed to obtain sufficiently accurate representations of the system dynamics and a tractable control problem. The floating version of the well-known Recursive Newton–Euler Algorithm (RNEA) has been chosen, following the approach in Ref. [24] that exploits the Composite Rigid-Body Algorithm (CRBA) [25], since it allows dealing with arbitrarily complex configurations while ensuring computational efficiency (see Fig. 5). Leveraging the flexibility of the recursive formulation, an orbital disturbance term has been considered to evaluate possible undesired effects due to the coupling between multibody and orbital dynamics that are typically neglected in previous studies on space robots.

The nonlinear equations of motion of the system can be cast in the following general form:

$$\boldsymbol{H}(\vec{q})\begin{bmatrix}\dot{\vec{\omega}}^{C}_{O/C}\\ \dot{\vec{v}}^{C}_{O/C}\\ \ddot{\vec{q}}\end{bmatrix}+\boldsymbol{C}(\vec{q},\ \vec{\omega}^{C}_{O/C},\ \vec{v}^{C}_{O/C},\ \dot{\vec{q}},\ \bar{\eta}_{O/C},\ \vec{r}^{C}_{T/C})=\begin{bmatrix}\vec{M}_C\\ \vec{F}_C\\ \vec{\tau}\end{bmatrix}$$

where $\vec{M}_C, \vec{F}_C \in \mathbb{R}^3$ and $\vec{\tau} \in \mathbb{R}^7$ are base torque, force and joint motor torque, respectively. The matrix $\boldsymbol{H} \in \mathbb{R}^{13\times13}$ is the joint-space inertia matrix, while $\boldsymbol{C} \in \mathbb{R}^{13}$ is the Coriolis/centrifugal term which accounts for terms related to the relative orbital dynamics.

Notably, the considered model allows describing the space robot dynamics both for the capture and the post-capture phase, in which the target spacecraft is rigidly attached to the last link of the manipulator.

3.3 Nonlinear control law design and control architectures

The design of the control law depends on the phase of the mission but also on the available control inputs, measurements, and state estimates. Figure 6 shows the basic architecture of computed torque controllers. A joint-space architecture has been chosen for both the pre- and post-capture phase. Specifically, given a desired pose trajectory $p_{O/C_d}(t) = (\vec{r}^{O}_{O/C_d}(t), \bar{\eta}_{O/C_d}(t))$, the desired trajectory for the joint angles $\vec{q}_d(t)$, and the corresponding velocity, the following computed torque control law is employed for the pre-capture phase:

$$\begin{bmatrix}\vec{M}_C\\ \vec{F}_C\\ \vec{\tau}\end{bmatrix}=\hat{\boldsymbol{C}}(\vec{q},\ \vec{\omega}^{C}_{O/C},\ \vec{v}^{C}_{O/C},\ \dot{\vec{q}})+\hat{\boldsymbol{H}}(\vec{q})\begin{bmatrix}\vec{a}_\omega\\ \vec{a}_v\\ \vec{a}_q\end{bmatrix} \tag{1}$$

where $\hat{\boldsymbol{C}}$ is an estimate of the Coriolis term of the robot without including orbital effects, $\hat{\boldsymbol{H}}(\vec{q})$ is the estimate of the robot mass matrix and $\vec{a}_\omega, \vec{a}_v, \vec{a}_q$ are virtual inputs to be designed. As mentioned above, orbital effects will be accounted for at synthesis time, when tuning the control gains. The virtual attitude input is designed as

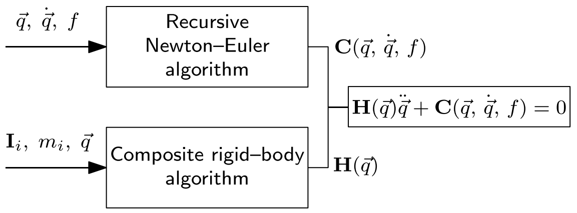

$\vec{q},\ \dot{\vec{q}},\ f$
Recursive Newton–Euler algorithm
$\mathbf{C}(\vec{q},\ \dot{\vec{q}},\ f)$
$\mathbf{H}(\vec{q})\ddot{\vec{q}}+\mathbf{C}(\vec{q},\ \dot{\vec{q}},\ f)=0$
$\mathbf{I}_i,\ m_i,\ \vec{q}$
Composite rigid–body algorithm
$\mathbf{H}(\vec{q})$

Fig. 5 Recursive model formulation according to [24]

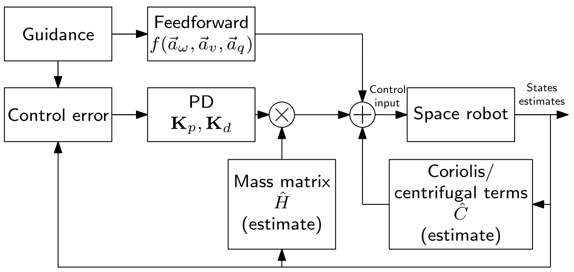


Fig. 6 Computed torque controller

$$\begin{aligned}\vec{a}_\omega =&\boldsymbol{K}_{d,att}\vec{\omega}^{C}_{C/C_d} + 2\boldsymbol{K}_{p,att}\eta_{C/C_d}\vec{\eta}_{C/C_d}\\ &+\boldsymbol{R}^{C}_{C_d}\dot{\vec{\omega}}^{C_d}_{O/C_d} + \vec{\omega}^{C}_{C/C_d} \times \boldsymbol{R}^{C}_{C_d}\vec{\omega}^{C_d}_{O/C_d}\end{aligned}$$

where

$$\vec{\omega}^{C}_{C/C_d} = \vec{\omega}^{C}_{O/C} - \boldsymbol{R}^{C}_{C_d}\vec{\omega}^{C_d}_{O/C_d}$$

$$\bar{\eta}_{C/C_d} = \begin{bmatrix}\eta_{C/C_d}\\ \vec{\eta}_{C/C_d}\end{bmatrix} = \bar{\eta}^{*}_{O/C} \otimes \bar{\eta}_{O/C_d}$$

are the velocity and attitude errors, respectively, and $\boldsymbol{K}_{d,att}$, $\boldsymbol{K}_{p,att} \in \mathbb{R}^{3\times3}$ are the proportional and derivative gains for the attitude, respectively, $\otimes$ is the quaternion product and $\bar{\eta}^*$ is the quaternion inverse, i.e., $\bar{\eta}^{*}_{O/C} = \bar{\eta}_{C/O}$.

The position virtual command can be computed as

$$\begin{aligned}\vec{a}_v =&\boldsymbol{R}^{C}_{O}\Big[\boldsymbol{K}_{d,pos}\big(\vec{v}^{O}_{O/C_d} - \boldsymbol{R}^{O}_{C}\vec{v}^{C}_{O/C}\big)\\ &+\boldsymbol{K}_{p,pos}\big(\vec{r}^{O}_{O/C_d} - \vec{r}^{C}_{O/C}\big) + \dot{\vec{v}}^{O}_{O/C_d}\Big]\\ &- \vec{\omega}^{C}_{O/C} \times \vec{v}^{C}_{O/C},\end{aligned}$$

with $\boldsymbol{K}_{d,pos}$, $\boldsymbol{K}_{p,pos} \in \mathbb{R}^{3\times3}$ and the joints virtual command as

$$\vec{a}_q = \ddot{\vec{q}}_d + \boldsymbol{K}_{d,joint}(\dot{\vec{q}}_d - \dot{\vec{q}}) + \boldsymbol{K}_{p,joint}(\vec{q}_d - \vec{q})$$

where $\boldsymbol{K}_{d,joint}$, $\boldsymbol{K}_{p,joint} \in \mathbb{R}^{7\times7}$. The free parameters of this controller can be collected as

$$\boldsymbol{K}_p = diag\Big(\vec{K}_{p,att}, \vec{K}_{p,pos}, \vec{K}_{p,joint}\Big)$$

$$\boldsymbol{K}_d = diag\Big(\vec{K}_{d,att}, \vec{K}_{d,pos}, \vec{K}_{d,joint}\Big)$$

The desired acceleration of the joint angles, i.e., $\ddot{\vec{q}}_d$, is also needed to compute feed-forward control actions (without performing numerical differentiation) to achieve good tracking performance. The states of the base and of the manipulator are needed for feedback control.

The considered law ensures asymptotic tracking of the error for time-varying reference signals under the assumption of perfect knowledge of the system parameters. It is worth noting that for small control errors and low speeds typical of close-proximity operations (Coriolis/centrifugal terms are negligible), the controller approximately behaves like a PD controller with no constraints on the gains. This fact will be leveraged for the systematic tuning of the control gains using the structured H_∞ synthesis technique, as described in the following section.

The post-capture phase of the mission requires a different solution since the control of the position and linear velocity of the stack is not needed in such a case: the most critical part consists in stopping the relative motion between the chaser base and the target such that a rigid stack is obtained. Addressing post-capture in a combined fashion means simultaneous regulation of both base and manipulator (connected to the target) as in, e.g., [26]. Notably, the control law for post-capture is the same as in Eq. 1, without the rows corresponding to the control force $\vec{F}_C$.

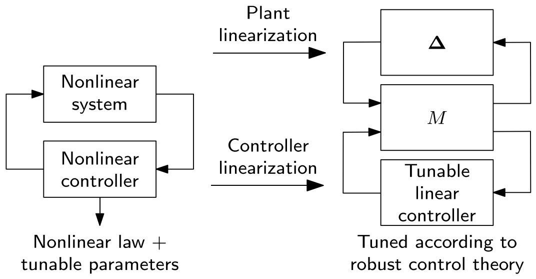

Plant linearization
Nonlinear system
Nonlinear controller
Controller linearization
Δ
M
Tunable linear controller
Nonlinear law + tunable parameters
Tuned according to robust control theory

Fig. 7 Robust tuning procedure

3.4 Control law tuning

This section presents the approach developed in this work to systematically tune the controller gains. The starting point of the procedure is a nonlinear system interconnected via feedback with a nonlinear controller which contains some tunable gains that can be freely assigned. Then, both the plant and the controller are linearized about a desired operating point: as shown in the right part of Fig. 7, the closed loop will result in a linearized plant, possibly accounting for uncertainties in Linear fractional transformation (LFT) form, and a linearized control law containing the tunable gains. At this point, robust control methods and structured routines are applied to find optimal values for the gains. This approach allows tuning the controller while accounting for parametric uncertainties and unmodeled dynamics, not considered when deriving the nonlinear control law. It is also worth noting that orbital dynamics effects (neglected in previous works on space robots) can be considered at synthesis time.

In the following, the linearization of the equations of motion and the controller described in Sect. 3.3 are briefly reported, with emphasis on the dependency of the model on uncertain parameters using the LFT form. Since the expected maneuvers coming from the guidance are slow, a constant reference for the linearization will be considered; this assumption is consistent with existing approaches in space robotics (see, e.g., [15, 27] and references therein). While the resulting robustness and performance properties will be valid in a neighborhood of the considered linearization point, it is worth noting that the inclusion of uncertainties in mass, moments of inertia and center of mass position compensates indirectly for the variations in the system configuration as well. Moreover, selecting as linearization point the capturing condition provides the desired level of robustness and performance in the most critical point of the capture.

3.4.1 Plant linearization

To derive the linearized plant dynamics, consider first the following definitions:

$$\vec{x} = [\delta\vec{\eta}_{C/C_d}, \delta\vec{r}^{C}_{C/C_d}, \delta\vec{q}, \delta\vec{\omega}^{C}_{C/C_d}, \delta\vec{v}^{C}_{C/C_d}, \delta\dot{\vec{q}}\,]^\top$$

$$\boldsymbol{A}_{kin} = blkdiag\left(\frac{1}{2}\boldsymbol{I}_3, \boldsymbol{R}^{O}_{C_d}, \boldsymbol{I}_7\right)$$

$$\vec{u} = \left[\delta\vec{M}_C, \delta\vec{F}_C, \delta\vec{\tau}\right]^\top$$

where $\delta\vec{\eta}_{C/C_d}$ denotes the vector part of the quaternion perturbation, i.e., $\bar{\eta}_{O/C} = \bar{\eta}_{O/C_d} \otimes \delta\bar{\eta}_{C/C_d}$, and $\boldsymbol{I}_i$ is the identity matrix of dimension i. The function *blkdiag* returns a block diagonal matrix where the input matrices are positioned along the main diagonal, with all off-diagonal blocks being zero. When the relative orbital dynamics is considered, the linearized equations of motion about a desired configuration are given by

$$\dot{\vec{x}} = \begin{bmatrix} 0 & \boldsymbol{A}_{kin} \\ -\boldsymbol{H}^{-1}\boldsymbol{C}_{\vec{x}_{kin}} & -\boldsymbol{H}^{-1}\boldsymbol{C}_{\vec{x}_{dyn}} \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ \boldsymbol{H}^{-1} \end{bmatrix} \vec{u} \tag{2}$$

where the terms $\boldsymbol{C}_{\vec{x}_{kin}}, \boldsymbol{C}_{\vec{x}_{dyn}}$ denote the partial derivatives of the Coriolis-centrifugal term $\boldsymbol{C}$ with respect to the sets of variables $\left[\delta\vec{\eta}_{C/C_d}, \delta\vec{r}_{C/C_d}, \delta\vec{q}\right]$ and $\left[\delta\vec{\omega}^{C}_{C/C_d}, \delta\vec{v}^{C}_{C/C_d}, \delta\dot{\vec{q}}\right]$, respectively. In complete analogy with point mass systems, whose relative orbital motion can be described by the Yamanaka–Ankersen equations [28], the presence of the orbital effects moves the eigenvalues from the origin. To the best of our knowledge, there exist no analytical methods to compute the terms $\boldsymbol{C}_{\vec{x}_{kin}}, \boldsymbol{C}_{\vec{x}_{dyn}}$ when the orbital perturbations are considered; as a consequence of that, the part of the dynamic matrix $\boldsymbol{A}$ in Eq. 2 containing $\boldsymbol{C}_{\vec{x}_{kin}}, \boldsymbol{C}_{\vec{x}_{dyn}}$ will be computed numerically. Once linear models have been obtained, uncertainties are included in the model in LFT form.

Concerning the *pre-capture phase* of the mission, the uncertainties listed in Table 12 have been included in the model. Uncertainties related to the robotic arm are not considered, as it can be assumed that accurate calibrations can be performed before operations; this assumption is standard in the robotics literature. The uncertain parameters can be included in Eq. 2 using routines implemented in the MATLAB robust control toolbox. In this case, the uncertain dynamics (note that matrices are now function of the uncertain parameters δ) can be written as

$$\dot{\vec{x}} = \boldsymbol{A}(\delta)\vec{x} + \boldsymbol{B}(\delta)\vec{u}$$

A procedure based on the evaluation of the Vinnicombe metric has been considered to show that one can use the nominal matrix $\boldsymbol{A}$ at synthesis time, thereby simplifying the complexity of the uncertain model.

In the *post-capture phase*, a different approach must be followed since the position of the base of the stack is not controlled. As a consequence, no external forces act on the system (we neglect the motion along the orbit [29]), and the CoM position of the stack will not vary. From a mathematical point of view, the complete system has uncontrollable modes at the origin, and it is not stabilizable; this means that one cannot find solutions to the control synthesis problem. To overcome this issue, the linear state variables of the base has been removed assuming $\vec{F}_C = 0$, solving for $\dot{\vec{v}}^{C}_{O/C}$ in the translational dynamics and then substituting it in the equations for the manipulator and attitude dynamics. In turn, the constraint of conservation of linear momentum increases the complexity of the LFT. In this case, the number of uncertain parameters (including repetitions) raises to 123.

So far, the linear systems have been modeled considering only rigid bodies: this modeling paradigm is widely adopted for control design of space robot as it accurately reflects the main behavior of the system. However, the actual behavior of the system is affected by additional second order effects like sloshing and flexibility, which can have a significant impact on the stability and performance of the system, especially can result in a wrong selection of the control gains. These effects can be included in the linear models so that one can perform at least a robust stability analysis of the system about the reference motion, since performing robust synthesis is extremely challenging given the complexity of the resulting uncertain models.

Specifically, the spring–mass–damper equivalent model presented in Ref. [28] has been included in the linearized equations of motion to account for sloshing. This sloshing model requires few parameters to be implemented, such as the sloshing frequency, the damping coefficient, the mass and positions of the tanks. In this case, the state space formulation for the reach and capture phase depends on the scenario, as in SC1 there are two tanks, while in SC2, there is only one tank. The presence of sloshing increases the complexity of the LFTs; indeed, for SC1, the LFT model contains 125 parameters, while in SC2, the model contains 103 parameters: there is a great gap in complexity with respect to the rigid model (27 parameters). For post-capture models, the sloshing masses of both the chaser and the target should be considered. However, since the sloshing masses of the latter are negligible, they have not been included.

Similarly, flexibility is another effect which may influence the dynamics of a spacecraft. The major source of flexibility in the chaser and target is given by flexible appendages, like solar panels. Specifically, the chaser in SC1 and the target in SC1 and SC2 have flexible body-mounted panels. Typically, flexible effects are accounted for in a linearized fashion considering detailed data coming from detailed modal analysis. Given the lack of modal data in this study, a simple lumped modeling approach has been pursued and the linearized plant has been augmented with suitable terms accounting for the corresponding dynamics.

3.4.2 Nonlinear controller linearization

This section shows the linearized version of the control law presented in Sect. 3.3. In the pre-capture phase, the controller linearization yields

$$\begin{bmatrix} \delta\vec{M}_C \\ \delta\vec{F}_C \\ \delta\vec{\tau} \end{bmatrix} = \hat{\boldsymbol{H}} \begin{bmatrix} \boldsymbol{I}_3 & 0 & 0 \\ 0 & \boldsymbol{R}^O_{C_d} & 0 \\ 0 & 0 & \boldsymbol{I}_7 \end{bmatrix} \begin{bmatrix} \boldsymbol{K}_p & \boldsymbol{K}_d \end{bmatrix} \begin{bmatrix} 2\delta\vec{\eta}_{C/C_d} \\ \delta\vec{r}^{\,C}_{C/C_d} \\ \delta\vec{q} \\ \delta\vec{\omega}^{\,C}_{C/C_d} \\ \delta\vec{v}^{\,C}_{C/C_d} \\ \delta\dot{\vec{q}} \end{bmatrix} \tag{3}$$

in which the rotation matrix maps virtual commands expressed in the Orbital frame (O) into the corresponding control actions generated in base frame. By inspecting Eq. 3, one immediately sees that locally the controller behaves as a PD control law, with tunable gains contained in the matrices $\boldsymbol{K}_p$, $\boldsymbol{K}_d$. The post-capture architecture can be linearized in a similar fashion.

3.5 Control synthesis results

In the following, the control synthesis will be performed considering the rigid system for all the scenarios and the corresponding sub-cases. When necessary, a re-tuning considering the additional effects described above (sloshing and flexibility) has been executed. Before performing the synthesis, suitable control requirements have been defined following the mixed-sensitivity H_∞ approach (see [16]). Specifically, given the fixed structure of the controller, structured controller synthesis has been performed leveraging the MATLAB routine *systune*, which allows imposing multiple objectives at the same time.

3.5.1 Pre-capture phase

The following objectives have been considered for the synthesis:

- Obj. 1: Tracking performance requirement from base pose setpoint to base pose error.
- Obj. 2: Tracking performance requirement from joint angles setpoint to joint angles.
- Obj. 3: Control effort moderation from base and joint setpoints to base control wrench.
- Obj. 4: Control effort moderation from base and joint setpoints to arm torque commands.

Table 5 Achieved worst-case tuning goals for both SC1 and SC2

	Obj. 1	Obj. 2	Obj. 3	Obj. 4
SC1	1.074	1.074	1.074	0.0186
SC2	1.56	1.56	1.56	0.007

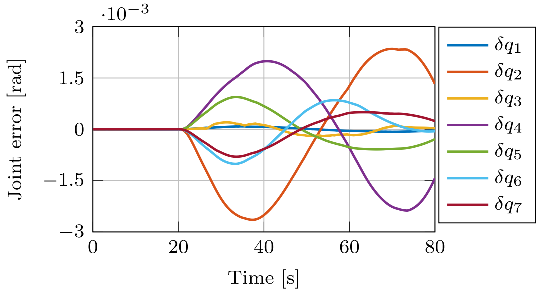


Fig. 8 Joint error during tracking in SC1. Structured controller

In the optimization, all the goals were imposed as soft requirements. Concerning SC1, in the pre-capture phase the target is stabilized by its own Attitude and Orbit Control System (AOCS) and the control task consists in grasping a target which is still. Data provided by the experiments made by the NAtional Space Development Agency (NASDA) in the ETS VII [30] are used to specify the bandwidth of the manipulator controller, while the ones provided by the e.Deorbit project are employed for the position and attitude control bandwidths [31], given the similarities with SC1. Concerning control moderation requirements, the main limitation is here given by the control authority of the base (Obj. 3). As a matter of fact, the base mass is large, it has large moments of inertia and it is equipped with RWs whose maximum torque is 0.5 N m, while the manipulator motors have a greater control authority (10 N m). Using the *systune* routine in MATLAB it is also possible to tune control systems for uncertain plants: considering the uncertainties modeled in Table 12, satisfactory results are obtained, as shown in Table 5 (a value smaller than one implies that the objective is satisfied, values close to one are still acceptable).

The controller has been tested on a nonlinear simulator with the uncertain parameters randomly sampled in the uncertainty set, and the effects of thrusters discretization have been included as well. The commanded trajectory is a point-to-point motion for which at the final time instant the end-effector coincides with the grasping point of the target;

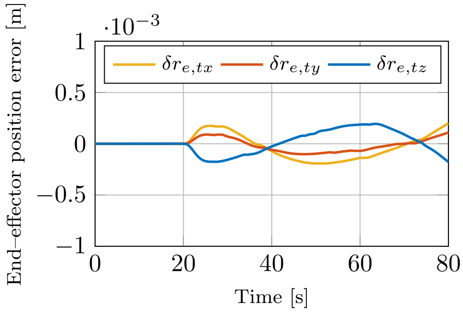


Fig. 9 End-effector position error during tracking in SC1. Structured controller

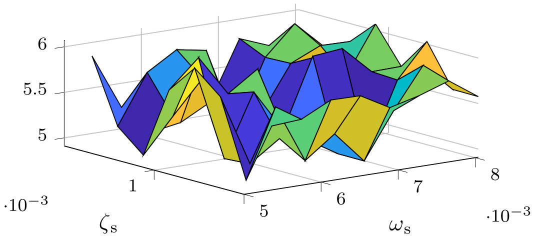


Fig. 10 Lower bound of the stability margin for the selected grid points, SC1 (ζ_s = damping, ω_s = frequency)

the base performs station keeping at a fixed distance from the target, while the end-effector moves along a polynomial trajectory from the initial point to the target grasping point. The results, in terms of joints and end-effector position errors, are shown in Figs. 8 and 9, respectively.

As the controller gains have been tuned accounting only for inertial uncertainties, the robustness of the linearized controller has been evaluated considering the model including the sloshing dynamics (because of the lack of reliable data concerning flexible components properties and mass budget, flexible analysis has not been performed). Computing the stability margins with the *robstab* routine in MATLAB, a lower bound of approximately 0.3 and an upper bound of 2.5 have been obtained. Indeed, given these values, nothing about robust stability can be concluded.

To overcome such a numerical, the sloshing frequency and damping parameters have been discretized within the uncertainty set and then, for any point on the grid, a robust analysis has been performed considering the remaining parameters as uncertain. For the specific case under investigation, an 8 × 8 grid in which the points are equally spaced has been selected. The robust analysis problem translates into computing 64 stability margins (i.e., 64 computations of the structured singular value). The analysis is really demanding in terms of computations, and it is feasible only for a reduced number of parameters to grid, as the number of evaluations increases exponentially. The results are shown in Fig. 10, in which the lower bound of the stability margin of any point of the grid are reported: robust stability is confirmed for any point on the grid.

For the reach and capture phase in SC2, the synthesis follows the approach outlined for the same phase in SC1 with differences only related to the selection of the control requirements since in this case the object to track has a higher maximum tumbling rate (up to 2.5 deg/s). The high tumbling rate of the target translates into imposing higher control bandwidths with respect to the previous scenario. The same requirements as in Ref. [31] have been imposed for the base attitude and position closed loops, namely 0.3 rad/s and 0.2 rad/s, respectively. The bandwidth for the joint angles closed loops is set to 3 rad/s, almost one decade faster than the one used for tracking cooperative targets as in SC1. Satisfactory synthesis results in line with SC1 have been obtained (see Table 5). As for the robustness analysis in presence of sloshing effects, even in this scenario issues have been encountered in the computation of robustness margins when sloshing uncertainties are considered. This issue has been overcome using the gridding approach described above, thanks to which robust stability could be assessed. Table 6 reports the controller gains of the two scenarios during Phase A.

3.5.2 Post-capture phase

Leveraging the considerations made in Sect. 3.3, the controller synthesis has been formulated considering the controller

Table 6 Controller gains of the two scenarios (Phase A)

	SC1	SC2
$\vec{K}_{p,att}$	[0.01, 0.04, 0.01]	[0.04, 0.04, 0.04]
$\vec{K}_{p,pos}$	[0.03, 0.06, 0.04]	[0.05, 0.05, 0.05]
$\vec{K}_{p,joint}$	[0.27, 0.26, 0.27, 0.27, 0.27, 0.27, 0.26]	[1.94, 1.93, 1.93, 1.94, 2.66, 2.12, 39.0]
$\vec{K}_{d,att}$	[0.31, 0.66, 0.31]	[0.33, 0.33, 0.33]
$\vec{K}_{d,pos}$	[0.75, 1.16, 1.03]	[0.24, 0.24, 0.24]
$\vec{K}_{d,joint}$	[1.32, 1.32, 1.27, 1.32, 1.31, 1.31, 13.09]	[2.15, 2.20, 2.21, 2.29, 3.32, 2.60, 7.58]

architecture for the post-capture phase with the following control objectives:

- Obj 5: Tracking requirement from base attitude setpoint to base attitude error.
- Obj 6: Tracking requirement from joint angles setpoint to joint angles error.
- Obj 7: Control moderation requirement from base and manipulator setpoint to base torque command.
- Obj 8: Control moderation requirement from base and manipulator setpoint to arm torque command.

In the post-capture phase, since the target has larger uncertainties with respect to the chaser, robust synthesis accounting for inertial uncertainties is mandatory.

The post-capture phase of SC1 consists in stabilizing the attitude of the target, which is in turn maintained by controlling the attitude of the chaser and the relative chaser/target attitude (i.e., the joints of the manipulator) via the inverse kinematics. Typical control requirements for attitude control systems have been imposed, considering the orders of magnitude detailed in Ref. [28]. In particular, the bandwidth of both base attitude and manipulator to the value 0.08 rad/s. The cost function value for the different control objectives is shown in Table 7, confirming that all the objectives are met.

The robust analysis in the presence of sloshing effects is characterized by numerical limitations due to the large LFT size for the scenario (401 uncertain parameters): once again the robust stability analysis was inconclusive. Hence, the gridding approach has been implemented again: differently from the reach and capture case, for this scenario a grid of 25 (5 × 5) points is analyzed as the arising LFTs have higher complexity. Once again, gridding helps in the verification of robust stability, and the lower bound of the stability margin is shown in Fig. 11.

For SC2, the post-capture phase consists in the detumbling the target and stopping the relative chaser–target motion (i.e., stiffening of the robotic arm). Therefore, the problem has been formulated to control the base angular velocity and the manipulator configuration, as there is no interest in controlling the attitude of the bodies of the stack. Concerning the controller structure, Eq. 1 has been modified by eliminating the terms corresponding to the attitude errors. As such, control requirements on the performance of the closed-loop sensitivity of base angular rate and joint angles configuration have been imposed. To set the bandwidth of the base angular rate closed loop, considerations regarding the settling time of the response have followed (the stabilization should not last more than 5 min). Considering a safety factor of 2.5 for robustness reasons, the corresponding cross-over frequency has been set to 0.04 rad/s, which is compatible with the one imposed for a similar case study (see Ref. [32]). As for the manipulator, a higher bandwidth requirement (0.08 rad/s) has been imposed as it is desirable to stop the manipulator motion before the base. The control objectives considered for the synthesis are the same as in SC1 except for the first one, in which the tracking requirement from base angular rate setpoint to base angular rate is considered instead of the attitude setpoint. The results of the robust synthesis are summarized (in terms of achieved optimization costs) in Table 7.

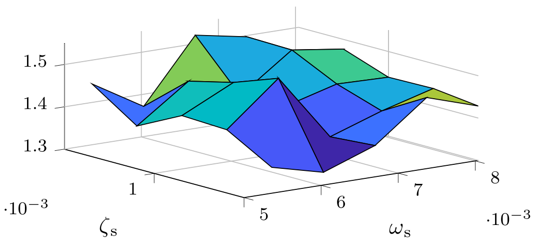


Fig. 11 Lower bound of the stability margin for the selected grid points, SC1 post-capture (ζ_s = damping, ω_s = frequency)

Table 7 Cost function (worst-case) for the two scenarios, post-capture phase

	Obj. 5	Obj. 6	Obj. 7	Obj. 8
SC1	0.93	0.93	0.93	0.19
SC2	0.91	0.91	0.91	0.03

The design based on the rigid model has then been verified considering sloshing and flexibility effects. Concerning sloshing, only the sloshing mass of the chaser has been considered (as the target tanks are almost empty). In the absence of a detailed structural analysis providing vibration modes and modal participation factors, a lumped parameter model has been used to evaluate flexibility effects. Only the panels of the target have been considered as flexible elements, as the chaser is equipped with body-mounted panels. The length of these appendages is 1.5 m, while the panels have a mass of 7.56 kg. Stiffness and damping coefficients of the flexible mode have been computed referring to guess data with a wide uncertainty (frequency 0.5 Hz ± 20%, damping 0.01 ±40%). In this case, the LFT considering chaser and target inertial uncertainties, chaser sloshing and target flexible uncertainties has 385 uncertain parameters. Luckily, robust stability margins are computed with a good level of accuracy, allowing the assessment of robust stability as shown in Table 8. Table 9 reports the controller gains of the two scenarios during Phase B.

Table 8 Robust stability margins for the post-capture model with sloshing and flexibility effects, SC2

Lower bound	Upper bound	Critical frequency [rad/s]
1.01	1.74	∞

Table 9 Controller gains of the two scenarios (Phase B)

	SC1	SC2
$\vec{K}_{p,att}$	[0.08, 0.1, 0.08]	[0, 0, 0]
$\vec{K}_{p,pos}$	[0, 0, 0]	[0, 0, 0]
$\vec{K}_{p,joint}$	[0.27, 0.19, 0.18, 0.21, 0.16, 0.20, 0.20]	[0.46, 0.47, 0.59, 0.68, 1.06, 0.54, 0.91]
$\vec{K}_{d,att}$	[2.22, 2.79, 2.28]	[0.02, 0.02, 0.04]
$\vec{K}_{d,pos}$	[0, 0, 0]	[0, 0, 0]
$\vec{K}_{d,joint}$	[4.91, 5.30, 5.04, 5.84, 4.53, 5.42, 5.08]	[12.69, 12.80, 16.21, 18.75, 28.60, 14.70, 24.78]

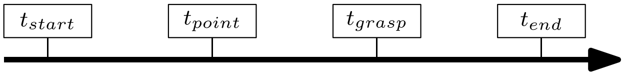


Fig. 12 Relevant time instants for trajectory design

3.6 Guidance strategy

A guidance strategy has been developed with the aim of generating suitable trajectories for the validation of the combined controller. Existing works either use direct estimates from the navigation function to specify the desired combined trajectory [19] or employ optimization algorithms to compute optimal and safe trajectories [29, 33, 34]. Regarding the latter approaches, this work proposes a simpler yet effective guidance design based on classic approaches for ground-fixed robots. Specifically, polynomial trajectories are generated based on the propagated motion of the target and are continuously updated to take advantage of the improved accuracy of the measurements coming from the navigation filter as the chaser gets closer to the target.

As mentioned at the end of Sect. 3.3, the control objective for the post-capture phase is just to stop the relative motion between the manipulator and the chaser body. In this case, the guidance function provides a desired constant attitude $\bar{\eta}_{O/C_d}$ for the chaser body given by the attitude at the end of the pre-capture and a desired constant configuration of the joints $\vec{q}_d$, which ensures a safe final configuration for the overall stack (chaser+target). The position is left free as this degree of freedom is not controlled in this phase.

Instead, as shown in Fig. 12, the pre-capture maneuver is built considering four main instants:

1. t_{start}: the manipulator starts moving;
2. from t_{point} to t_{grasp}: the end-effector camera is directed towards the grasping point thanks to the guidance design;
3. t_{grasp}: the end-effector is brought onto the grasping point;
4. t_{end}: the capture maneuver is completed.

The desired final pose of the end-effector, denoted as $p_{O/E_d}(t) = (\vec{r}^{\,O}_{O/E_d}(t), \bar{\eta}_{O/E_d}(t))$, is specified in terms of the pose of the grasping point on the target at a specific time instant in the future, which is predicted by propagating forward the target motion until the desired time. Then, the initial and final pose of the end-effector are interpolated using a 5th-order polynomial. As the motion prediction of the target is accurate only in very short time horizon, the coefficients of the 5th-order polynomial interpolation are updated at each step using the current measurements of both the chaser and target states coming from the navigation and doing a new propagation of the target motion to performing a new prediction of the future grasping point location. In this way, the guidance algorithm is compensating for modeling errors.

Due to the choice of implementing the controller according to the joint-space formulation, the guidance strategy needs to generate appropriate reference trajectories in the joint space, while guaranteeing the fulfillment of end-effector trajectory requirements. Therefore, the reference trajectory for the joint angles $\vec{q}_d(t)$ is computed starting from the desired pose trajectory of the end-effector, computed as described above, by solving the so-called inverse kinematic problem [35]. The chosen approach consists in solving the inverse kinematics at the joints velocity level using a closed-loop inverse kinematic approach. Using a closed-loop kinematic solution prevents the unavoidable drifting of the desired joint profiles due to the accumulation of numerical integration errors. Specifically, the angular rate setpoint for the joint angles is computed as

$$\dot{\vec{q}}_d = \boldsymbol{J}^{\#}_{E}(q)\left(\vec{v}^{\,E}_{C/E_d} + \boldsymbol{K}\left(\vec{p}^{\,E}_{E/E_d}\right)\right) + \left(\boldsymbol{I}_{N_B} - \boldsymbol{J}^{\#}_{E}(q)\boldsymbol{J}_E(q)\right)\dot{\vec{q}}_0,$$

where $\boldsymbol{J}^{\#}_{E}$ is the right (Moore–Penrose) pseudo-inverse of the end-effector Jacobian,

$$\vec{v}^{\,E}_{C/E_d} = (\boldsymbol{R}^{E}_{O}(\vec{v}^{\,O}_{O/E_d} - \vec{v}^{\,O}_{O/C}), \vec{\omega}^{\,E}_{O/E_d} - \boldsymbol{R}^{E}_{O}\vec{\omega}^{\,O}_{O/C})$$

is the desired end-effector velocity relative to the base, while $\boldsymbol{K}\left(\vec{p}^{\,E}_{E/E_d}\right)$ is a feedback term to account for the end-effector pose error $\vec{p}^{\,E}_{E/E_d} = (\boldsymbol{R}^{E}_{O}(\vec{r}^{\,O}_{O/E_d} - \vec{r}^{\,O}_{O/E}), \bar{\eta}_{E/O} \otimes \bar{\eta}_{O/E_d})$. Note that $\left(\boldsymbol{I}_{N_B} - \boldsymbol{J}^{\#}_{E}(q)\boldsymbol{J}_E(q)\right)$ is a projection matrix onto the null space of the Jacobian, which allows exploiting the manipulator redundancy to improve maneuverability and/or to respect joint angle constraints, by selecting $\dot{\vec{q}}_0 = -\frac{\partial w(q)}{\partial q}$, where $w(q)$ is a suitably defined objective function to be minimized. It must be noted that the this procedure does not allow imposing directly hard constraints on the input or state variables, such as joint and workspace limits. Yet, the approach provides satisfactory results in practice.

The proposed combined controller requires specifying the desired pose trajectory for the chaser base as well, denoted as $p_{O/C_d}(t) = (\vec{r}^{\,O}_{O/C_d}(t), \bar{\eta}_{O/C_d}(t))$, which is specified as a suitable relative position and attitude of the chaser base with respect to the grasping point (different in each scenario). Following this approach, the guidance strategy for the chaser base is obtained either by selecting a proper fixed pose in the target frame or by a proper fixed pose in the orbital frame, depending on whether the target is tumbling or not. For the specific scenarios considered in this work, the spacecraft attitude and position are kept fixed in the orbital frame, given that the manipulator capabilities are compatible with the target tumbling state so that the capturing maneuver can be completed without synchronizing the chaser base with the one of the target spacecraft motion. Figures 13 and 14 show the planned trajectory of the end-effector for SC1 and SC2 respectively.

4 EO-based relative navigation system

Active and passive EO sensors represent the best technological solution to conduct relative navigation with respect to semi- or non-cooperative targets in close-proximity [36]. Given the focus on the pre-capture phase, the relative navigation function must be able to estimate the chaser relative state with respect to the target (T/C), as well as the robotic arm end-effector relative state with respect to the selected grasping point (G/E). To this aim, several architectural strategies have been investigated in the literature mainly differing in terms of the role played by sensors potentially installed on the robotic arm to support the grasping operations.

In the framework of the COMRADE project [17], two different options have been proposed. In the former case, relative navigation is entrusted to a single sensor mounted on the chaser body, whose measurements are used to estimate the T/C relative state; the G/E relative state is then retrieved using the robotic arm joints measurements in its forward kinematic model and exploiting the knowledge of the T/C pose as well as of the target geometry. In the latter case, relative navigation is entrusted to two different EO sensors, rigidly attached to the chaser body and to the end-effector, whose measurements are processed by two separate navigation

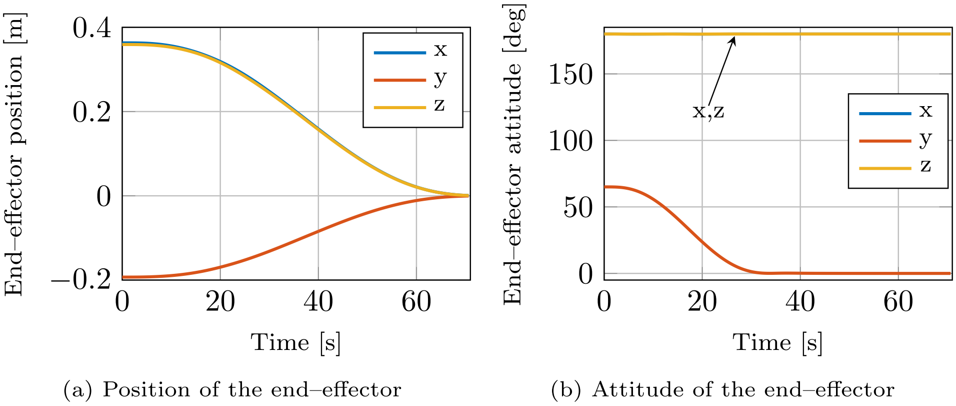


Fig. 13 Guidance of the end-effector during phase A of SC1

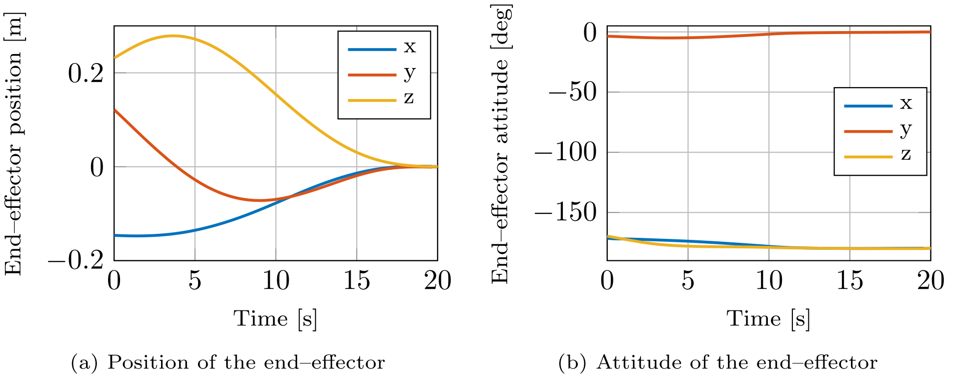


Fig. 14 Guidance of the end-effector during phase A of SC2

filters for T/C and G/E relative state estimation, respectively. The use of two EO sensors, one body-fixed and the other mounted on the end-effector, has also been proposed in [37], where two cameras share the feature recognition task in a collaborative behavior to estimate the T/C pose. Finally, the measurements of a single sensor mounted on the robotic arm end-effector are used to estimate both the T/C and G/E relative state within a unique filtering architecture in [38].

A relative navigation solution based on two sensors and two separate filtering schemes, inspired by the second option investigated in the frame of the COMRADE project, has been proposed in this study. The main advantage with respect to the first COMRADE option [17] and to the solution based on collaborative cameras [37] is that direct pose measurements from an end-effector sensor allow compensating the robotic arm ego motion uncertainty, introduced when applying the forward kinematic model, which would affect the G/E relative state estimation. This phenomenon, which can have a significant impact when dealing with tumbling targets, is mainly caused by the error sources affecting the joints sensors measurements, as well as by the uncertainty in the knowledge of the robotic arm geometry [39]. Moreover, the proposed approach has advantages in terms of flexibility and modularity with respect to designing a single filter for both the states (as in Ref. [38]) since it allows continuously updating the G/E relative state independently of the availability of measurements from the robotic arm sensor, and, more importantly, the absence of such measurements (e.g., when the robotic arm is being deployed from its stowed configuration at the beginning of the reach and capture maneuver) does not affect the T/C relative state estimation.

A high-level block diagram of the proposed architecture is provided in Fig. 15. The configuration is loosely coupled meaning that the raw data produced by the EO sensors are processed within a separate block (i.e., outside of the filter) to get relative position and attitude measurements to be used in the correction step of the filtering schemes [40, 41]. For T/C relative state estimation, the filter only receives T/C pose observations in input. For G/E relative state estimation, instead, the observation vector includes not only the direct G/E pose measurements (when available), but also end-effector pose measurements with respect to base of the robotic arm (C/E) obtained applying its forward kinematics model. Clearly, these latter measurements are available at a much higher frequency than the former ones. Additional input data for the two filters are target, chaser and robotic arm geometric information (including the positions in target coordinates of fiducial markers), the chaser absolute state estimates (as provided by its absolute navigation system), and the robotic arm joint sensors measurements. Finally, it is worth highlighting that while the T/C relative state estimation process is decoupled from the G/E one, the inverse relation does not hold, since the G/E filter operation is also supported by the output of the T/C filter. The T/C and G/E relative state estimates are used to feed the combined control function which provides feedback chaser commands (e.g., thrusts) that can be used by the T/C relative navigation filter to compute the resulting accelerations acting on the chaser.

For the scenarios under study, a monocular camera has been selected as both body-mounted and robotic arm sensor. This choice is justified by the lower size, weight, and power constraints characterizing passive sensors (compared to active LIDARs), thus ensuring lower complexity to the relative navigation subsystem design. Also, the possibility to get precise line-of-sight measurements corresponding to the fiducial markers installed on the target ensures to attain highly accurate pose and relative state estimates, especially at very close range [42, 43].

Concerning the selection of fiducial markers, passive/semi-passive solutions have been adopted. In SC1, ten retroreflectors

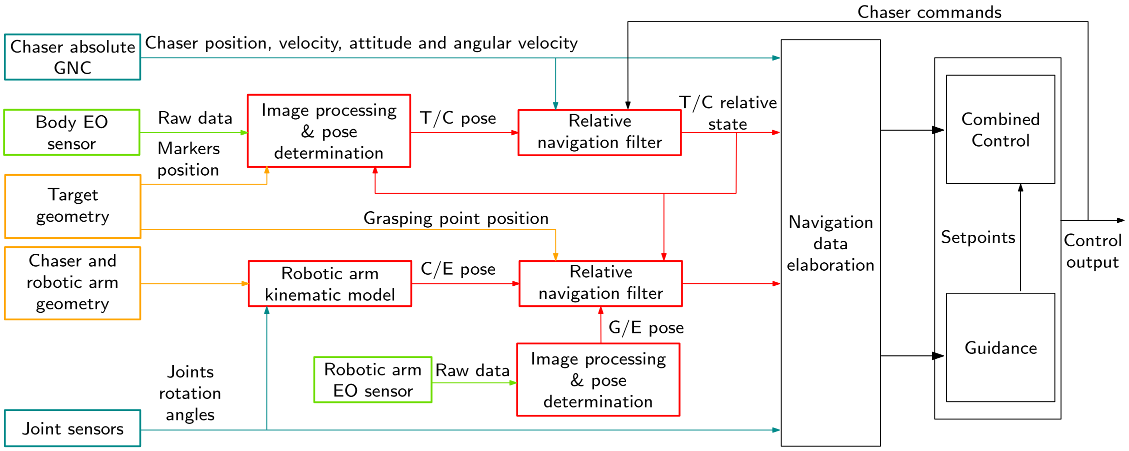

Chaser commands
Chaser absolute GNC
Chaser position, velocity, attitude and angular velocity
Body EO sensor
Raw data
Markers position
Image processing & pose determination
T/C pose
Relative navigation filter
T/C relative state
Target geometry
Grasping point position
Chaser and robotic arm geometry
Robotic arm kinematic model
C/E pose
Relative navigation filter
G/E pose
Robotic arm EO sensor
Raw data
Image processing & pose determination
Joints rotation angles
Joint sensors
Navigation data elaboration
Combined Control
Setpoints
Guidance
Control output

Fig. 15 High-level block diagram of the proposed loosely coupled relative navigation architecture. The processing blocks developed in this study are highlighted in red

(nine coplanar and one mounted with an offset in the surface outer normal direction), illuminated by an infrared laser source on the chaser, form the set of fiducial markers for the body-mounted camera. Indeed, this is a robust and safe solution for the servicing of a high-value GEO asset. The number of retroreflectors ensures an adequate redundancy (in case the target is not approached with nominal attitude) and avoids sudden performance degradation when single retroreflectors fall outside from the Field of View (FoV). Instead, the robotic arm camera is designed to detect a set of three white circular markers on a black background (similarly to the concept of concentric contrasting circles [42]). In this latter case, the possibility to minimize the number of correspondences is motivated by the fact that an accurate initial guess of the G/E pose is obtained by combining the T/C and C/E relative state estimates in the prediction step of the G/E filter. In SC2, code-based visual markers are selected to support pose estimation of both the cameras since they represent a promising solution for standard-designed targets like the elements of a large LEO constellation. Specifically, one 20 cm-by-20 cm AruCo marker [44] is placed at the center of each target face (excluding those hosting the solar panels along which the approach is not feasible), while one 5 cm-by-5 cm AruCo marker is placed close to the grappling interface.

Details about the pose determination processing pipeline, including image processing and image-to-model matching algorithms (for markers detection and identification) followed by a solution to the Perspective-n-Points problem, are provided in [45] and [46] for SC1 and SC2, respectively. The following subsections describe instead the adopted filtering schemes.

4.1 T/C relative state estimation filter

State-of-the-art relative navigation architectures for close-proximity scenarios adopt different filtering schemes to estimate the full target chaser-navigation state, including Kalman filters, minimum energy filters and H_∞ filters [47, 48]. In this work, an Extended Kalman Filter (EKF) has been selected for the real-time filtering of the camera-based pose measurements and to fully characterize the target–chaser relative motion in terms of relative position, velocity, attitude and angular velocity according to the scheme provided in Fig. 15. The EKF selection is the result of a trade-off between achievable accuracy and computational cost. Indeed, the EKF can handle nonlinear systems by applying the simple equations of the classical Kalman filter to the first order Taylor approximation of both prediction and observation models. The state vector is defined as follows:

$$\vec{x} = \left[\vec{r}^{\,H}_{T/C},\ \vec{v}^{\,H}_{T/C},\ \bar{\eta}_{T/C},\ \vec{\omega}^{\,T}_{ECI/T}\right]^{\mathsf{T}}$$

If quaternions are used to represent both attitude state and covariance, singularity problem may arise in the covariance matrix: indeed, the unit norm constraint introduces a linear dependence among the rows of the covariance matrix. To overcome this issue, the multiplicative EKF (MEKF) is introduced [49]: it adopts a three-parameter representation of the covariance of the relative attitude (under the assumption of small angular errors), while a globally and non-singular attitude state is represented by a unit quaternion [50]. The main difference between the classical EKF and the MEKF is that the state update is computed by exploiting the linearized error model instead of the linearized state model.

The dynamic and observation models adopted in this work are reported in the following. The translational dynamic has been modeled by the CW equations, reported in Eq. 4, i.e., a set of linear differential equations derived from the two-body mechanics by applying two further assumptions [51]:

- the target moves on a circular orbit
- the spacecraft fly in close-proximity

$$\begin{aligned} \ddot{\vec{x}} - 2n\dot{\vec{y}} - 3n^2\vec{x} &= \vec{a}_{T,x} \\ \ddot{\vec{y}} + 2n\dot{\vec{x}} &= \vec{a}_{T,y} \\ \ddot{\vec{z}} + n^2\vec{z} &= \vec{a}_{T,z} \end{aligned} \tag{4}$$

where $\vec{x}$, $\vec{y}$ and $\vec{z}$ are the Cartesian components of $\vec{r}^{\,H}_{T/C}$, n is the mean motion of the target spacecraft and $\vec{a}_T$ is the chaser specific force.

As regards the rotational motion model, the temporal evolution of the relative attitude quaternion is described by Eq. 5, while the absolute rotational dynamic is propagated through Eq. 6 if the target is three-axis stabilized during the pre-capture phase (as in SC1) or through Eq. 7 if the target is freely tumbling (as in SC2):

$$\dot{\bar{\eta}}_{T/C} = \frac{1}{2}\left[0,\ \vec{\omega}^{\,T}_{C/T}\right]^{\mathsf{T}} \otimes \bar{\eta}_{T/C} \tag{5}$$

$$\dot{\vec{\omega}}^{\,T}_{ECI/T} = 0 \tag{6}$$

$$\dot{\vec{\omega}}^{\,T}_{ECI/T} = -\boldsymbol{I}_T^{-1}\left(\vec{\omega}^{\,T}_{ECI/T} \times \boldsymbol{I}_T\vec{\omega}^{\,T}_{ECI/T}\right) \tag{7}$$

where $\boldsymbol{I}_T$ is the target inertia matrix.

Within the filtering scheme, the prediction of the state vector obtained from the integration of the dynamic model are corrected by exploiting visual-based pose measurements. Thus, an observation model, i.e., a system of equation relating the measurement vector, $\vec{z}$ (see Eq. 8) to the state vector $\vec{x}$ must be defined as shown in Eq. 9:

$$\vec{z} = \left[\vec{r}^{\,CS}_{CS/TG},\ \bar{\eta}_{CS/TG}\right]^{\mathsf{T}} \tag{8}$$

$$\vec{z} = \begin{cases} \boldsymbol{R}_{C}^{CS}\left(\boldsymbol{R}_{T}^{C}\,\vec{r}_{T/TG}^{\,T} - \vec{r}_{C/CS}^{\,C} - \boldsymbol{R}_{H}^{C}\vec{r}_{T/C}^{\,H}\right) \\ \bar{\eta}_{CS/C} \otimes \bar{\eta}_{C/T} \end{cases} \tag{9}$$

4.2 G/E relative state estimation filter

The MEKF scheme is also employed to estimate the gripper-grasping point relative motion. As shown in Fig. 15, it exploits as measurements (i) the base-gripper relative motion parameters provided by the forward kinematics model and (ii) the grasping point-end-effector pose obtained by processing the images coming from the hand-in-eye camera (when available). Summarizing, the state vector is defined as follows:

$$\vec{x} = \left[\vec{r}_{E/G}^{\,E},\ \vec{v}_{E/G}^{\,E},\ \bar{\eta}_{E/G},\ \vec{\omega}_{G/E}^{\,E}\right]^{\mathsf{T}} \tag{10}$$

Due to the slow motion of the robotic arm and the frequent update provided by the measurements of the encoders, constant (both linear and angular) velocity models are adopted in the prediction step of the G/E filter, as shown in the following:

$$\begin{cases} \dot{\vec{v}}_{E/G}^{\,E} = 0 \\ \dot{\bar{\eta}}_{E/G} = \frac{1}{2}\left[0,\ \vec{\omega}_{G/E}^{\,E}\right]^{\mathsf{T}} \otimes \bar{\eta}_{E/G} \\ \dot{\vec{\omega}}_{G/E}^{\,E} = 0 \end{cases}$$

Regarding the correction step of the filter, the measurements listed in Eq. 11 are linked to the state vector of Eq. 10 through the following observation model:

$$\vec{z} = \left[\vec{r}_{C/E}^{\,C},\ \vec{v}_{C/E}^{\,C},\ \bar{\eta}_{E/C},\ \vec{\omega}_{C/E}^{\,E},\ \vec{r}_{AS/G}^{\,AS},\ \bar{\eta}_{AS/G}\right]^{\mathsf{T}} \tag{11}$$

where

$$\begin{aligned} \vec{r}_{C/E}^{\,C} =& -\boldsymbol{R}_{H}^{C}\vec{r}_{T/C}^{\,H} + \boldsymbol{R}_{T}^{C}\left(\vec{r}_{T/G}^{\,T} - \boldsymbol{R}_{G}^{E}\vec{r}_{E/G}^{\,E}\right) \\ \vec{v}_{C/E}^{\,C} =& \frac{1}{2}\Bigg(-\boldsymbol{R}_{T}^{C}\Big(\boldsymbol{R}_{E}^{G}\big(\vec{v}_{E/G}^{\,E} + \vec{\omega}_{G/E}^{\,E} \times \vec{r}_{E/G}^{\,E}\big) + \\ & +\vec{\omega}_{C/T}^{\,T} \times \boldsymbol{R}_{E}^{G}\vec{r}_{E/G}^{\,E} - \vec{\omega}_{C/T}^{\,T} \times \vec{r}_{T/G}^{\,T}\Big) + \\ & -2\boldsymbol{R}_{H}^{C}\big(\vec{v}_{T/C}^{\,H} + \vec{\omega}_{C/H}^{\,H} \times \vec{r}_{T/C}^{\,H}\big)\Bigg) \\ \bar{\eta}_{E/C} =& \bar{\eta}_{E/G} \otimes \bar{\eta}_{T/C} \\ \vec{\omega}_{C/E}^{\,E} =& \vec{\omega}_{G/E}^{\,E} + \boldsymbol{R}_{G}^{E}\vec{\omega}_{C/T}^{\,T} \\ \vec{r}_{AS/G}^{\,AS} =& -\boldsymbol{R}_{E}^{AS}\vec{r}_{E/AS}^{\,E} + \boldsymbol{R}_{E}^{AS}\vec{r}_{E/G}^{\,E} \\ \bar{\eta}_{AS/G} =& \bar{\eta}_{AS/E} \otimes \bar{\eta}_{E/G} \end{aligned}$$

It is worth noting that, depending on weather or not the markers close to the grasping point are imaged by the camera on the robotic arm, the observation model can include the equations relating image-based pose parameters, namely $\vec{r}_{AS/G}^{\,AS}$ and $\bar{\eta}_{AS/G}$, to the state vector.

5 Simulation environment

A numerical simulation tool called Functional Engineering Simulator (FES) has been developed in the MATLAB/Simulink environment to test and validate the control and navigation functions. Models of the chaser and target satellites are included in the FES. In addition, the effects of the environmental disturbances are considered as well as internal disturbances, e.g., the sloshing of propellant in the tanks. Moreover, the simulator is complete of models of actuators and sensors. The control and navigation operations are simulated by implementing the algorithms presented in Sects. 3 and 4 respectively.

The realistic representation of the scenarios (see Sect. 2) is a challenging task due to: (1) the 13 DoFs of the chaser-manipulator system (six related with the rotation and translation of the chaser body and seven degrees of freedom of the robotic arm) and (2) the kinematics and the dynamics of the bodies that compose the chaser–target system are governed by complex system of differential equations. The chaser and the target spacecrafts are modeled using the Simscape Multibody package [52]. This allows to model complex systems as a chain of bodies connected by a variety of different joints without the need to implement complex mathematical equations. The manipulator is modeled using a chain of cylindrical elements connected to each other by revolute joints, which allow only the rotation around one axis.

In the FES, a full set of models of the relevant environmental aspects is implemented, namely: gravity field [53], geomagnetic field [54], gravity gradient [11], third body attraction [55], atmospheric drag [56] and solar pressure [57]. The forces/torques generated by the external perturbations are applied to every component of the system. Thus, the differential effects of the perturbation on the satellite attitude are considered in the simulations. For what concerns the internal disturbance, the sloshing of the propellant in the satellites tanks is considered and is modeled with an equivalent three-dimensional rigid pendulum [58].

The FES features different actuators and sensors models that reproduce the real behavior of the hardware. In particular, the actuators include thrusters, RWs, and DC motors for the robotic arm joints. The sensors include inertial measurements units (IMU), GNSS receivers, star trackers and optical encoders for the manipulator joints. The navigation function relies mainly on active and passive EO sensors to provide estimates of the relative state of the chaser with respect to the target. For that purpose, the FES interfaces with the ESA Planet and Asteroid Natural Scene Generation Utility (PANGU) tool that allows to generate synthetic images in real time during the simulation. In this fashion, the data obtained by simulated monocular cameras are provided to the navigation in real time.

The FES can simulate two different phases of an IOS mission: the pre-capture phase and the post-capture phase. In the transition between these two phases, a contact happens between the end-effector of the manipulator and the target. The contact generates forces and torques that are complex to simulate. They can cause large deviations in the position and velocity of the joints and of the spacecraft between the two mission phases [59]. The contact dynamics is not modeled in the FES since the scope of the study focuses is the development and test of a GNC algorithms. Nonetheless, it is fundamental that the model of the transition between the two phases respects the physical boundaries of a contact event. Hence, starting from Refs. [60] and [61], a synthetic method based on the conservation of the total angular and linear momenta of the system is employed to compute the state of all the bodies of the system after the contact event as a function of their state prior to the contact.

Before using the FES to validate the developed GNC solutions, several tests have been conducted to verify the simulator functionalities. During the FES implementation the subsystems that compose the FES have been validated at subsystem level. For instance, the environmental perturbations (gravity, drag, geomagnetic field, etc.) have been validated against the Simulink Aerospace Toolbox and the orbit propagation has been validated against the NASA General Mission Analysis Tool (GMAT) software. Moreover, it has been checked that nor energy neither linear/angular momentum is generated during the simulation if no external forces or torques act on the system.

Other tests have been conducted for accuracy assessment. First, the effects of the step size on the integration accuracy have been studied allowing to determine the maximum threshold step size below which the round-off error dominates. Second, a trade-off study on the solver type has been performed. The study considered that the accuracy of the solver increases with the integrator order that, conversely, increases the computational cost. The trade-off suggested to employ the Bogacki–Shampine algorithm (*ode3* in MATLAB).

In addition, specifically tailored tests have been conducted to verify the FES features that help the user in the system performance analysis. These features include: (1) a contact detection tool that provides a warning if the spacecraft bodies come into contact, (2) an on-line requirements check tool that rises warnings when requirements are not satisfied during the simulation and (3) a simulation report generator.

6 Numerical simulations and results

The validation of the developed GNC system is of key importance to prove its capability to reach the IOS mission goals. The GNC system presented in Sects. 3 and 4 is tested within the FES environment to achieve a deep understanding of the system behavior and a clear assessment of its performance. With this aim, three different types of tests have been conducted, namely: (1) nominal simulation (NS), (2) error budget (EB) analysis and (3) preliminary Monte Carlo (MC) analysis. These three tests are executed for both the considered scenarios.

6.1 Nominal simulation

The NSs aim at estimating the nominal performance of the system. During a NS the environmental disturbances and the sloshing of the propellant in the tanks are considered together with the realistic behavior of the sensors and actuators. On the other hand, the uncertainties that may affect the system, e.g., mass, inertia and CoM position uncertainties, and flexibility of elongated bodies are not considered. The sensors and the actuators have a realistic behavior since they are modeled in the FES to accurately represent a real hardware which introduces errors in the simulation, e.g., delays, overshoots, nonlinearities. For each scenario, two mission phases are simulated: the capture of the target (Phase A) and the stiffening of the robotic arm (Phase B).

The performance of the GNC system during Phase A is evaluated by means of performance metrics computed by comparing the trajectory defined by the guidance with the real state of the system. The equation for the evaluation of the control metrics is

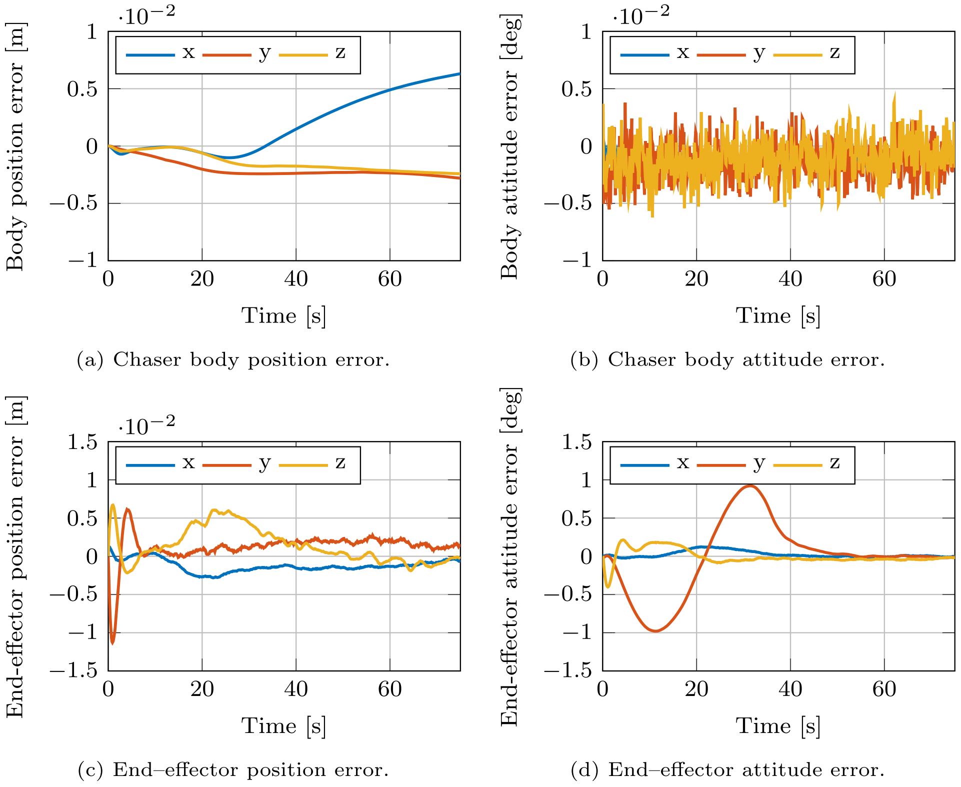
$$\varepsilon = X_{real} - X_{guid}$$

where X_{real} is the state of the system simulated by the FES, and X_{guid} is the trajectory defined by the guidance. In this study, the considered performance metrics are: (1) chaser body position error, (2) chaser body attitude error, (3) end-effector position error and (4) end-effector attitude error.

6.1.1 Scenario 1: servicing a large GEO platform

For what concerns SC1, Fig. 16a and Fig. 16b show the evolution of the performance metrics of the chaser body position and attitude, respectively. The GNC system proves good performance keeping the chaser body position error well below 0.01 m during the maneuver. The chaser body attitude error is very stable and close to zero. This is possible thanks to the highly accurate relative attitude

Fig. 16 Results of the nominal simulation of SC1—Phase A



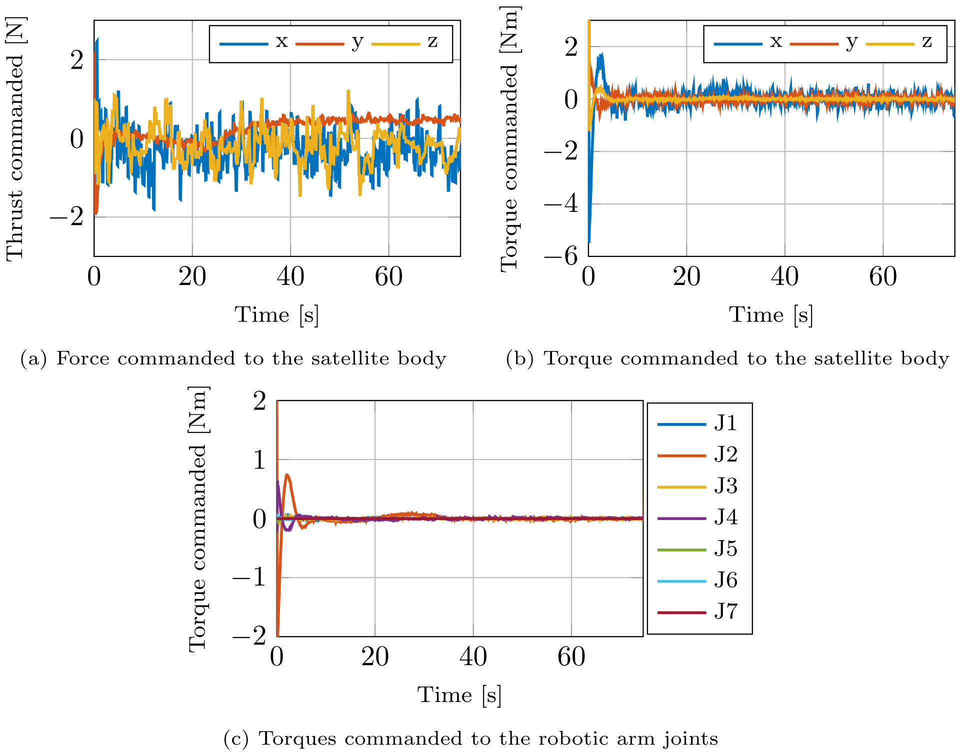
(a) Chaser body position error.

(b) Chaser body attitude error.

(c) End–effector position error.

(d) End–effector attitude error.

Fig. 17 Forces and torques acting on the system during phase A of SC1



(a) Force commanded to the satellite body

(b) Torque commanded to the satellite body

(c) Torques commanded to the robotic arm joints estimate provided by the navigation subsystem (in the order of 10^{-2} deg) and to the accuracy of the RWs.

Figure 16c shows the position error of the end-effector during the maneuver. After a transient due to the low-level PID controllers of the robotic arm joints, the end-effector has a stable trend and the target is grasped with an error lower than 1 cm in norm. Figure 16d reports the attitude error of the end-effector. Its behavior is due to the not-ideal Brush-Less DC (BLDC) motor models implemented in the simulator and to their low-level control (i.e., PID controllers), which introduces delays and overshoots in the response of the actuators. It is also worth pointing out that the angular control error below 1 deg corresponds to a very small translational error considering the short distance between the end-effector and the grasping point on the target during the capture phase. Figure 17 shows the forces and torques commanded by the GNC algorithm to the satellite body and to the robotic arm joints during Phase A. The peak at the start of the simulation is the starting torque of the motor. This is due to the fact that the robotic arm starts the maneuver from a static initial condition, hence high torques are generated at the beginning of the maneuver. As a consequence, high torques are commanded to the motors when they start to move in order to balance the starting torques. The reaction torque generated by the robotic arm motion is transmitted to the satellite body, so the controller activates the thrusters to compensate. However, the torques commanded to the DC motors do not reach the saturation of the motors that are equal to 10 N m [4].

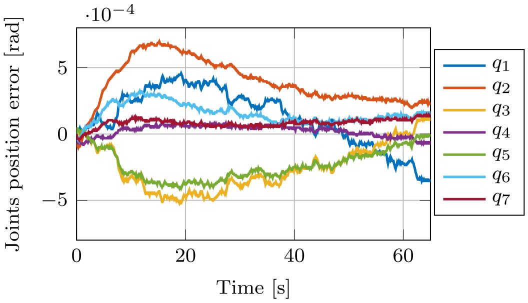


Fig. 18 Joint position error in SC1—Phase B

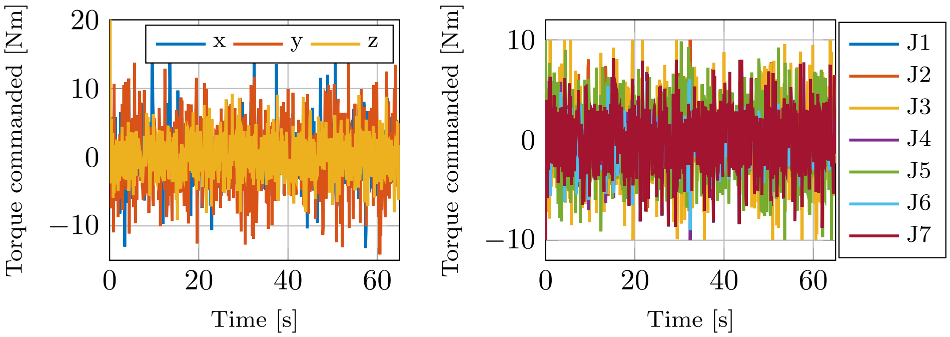
Phase B lasts for 60 s. The simulation runs for five additional seconds to check if the robotic arm is stable. Figure 18 shows the error of the robotic arm joints position during Phase B. The error have an order of magnitude of 10^{-4} rad and it stabilizes to a constant value after 60 s. The maximum torque that the joints provide during the maneuver is 10 N m. Figure 19 shows the torques commanded by the GNC algorithm to the satellite body and to the robotic arm joints during Phase B. During Phase B, only the RWs are used to stabilize the stack. Hence, the force command coming from the GNC algorithm, i.e., the force that the thrusters must apply, is zero for the whole maneuver and is not represented in Fig. 19. The torques commanded to the satellite RWs and to the robotic arm joints are compatible with the response time of the actuators that are able of performing successfully the stabilization maneuver as shown in Fig. 18.

6.1.2 Scenario 2: servicing a small satellite in LEO

Taking SC2 into consideration, Fig. 20a and 20b shows the position error and the attitude error of the chaser body, respectively. During the maneuver, the GNC system shall keep the chaser body position constant. However, as shown in Fig. 20a, an error of about 0.01 m is reached at the end of the maneuver. This is caused by the thrusters actuation and by the disturbances due to propellant sloshing, which degrade the performance of the GNC system. The thrusters are not able to keep the desired position of the chaser body; the sloshing causes the oscillatory behavior of the errors. The EB analysis confirms that these are the main sources of error. On the other hand, the chaser body attitude error is virtually zero only disturbed by some noise (Fig. 20b).

The end-effector position error is reported in Fig. 20c: its oscillatory behavior is due to the sloshing combined to the error brought by the thrusters. The thrusters introduce an error due to the delay caused by the opening and closing

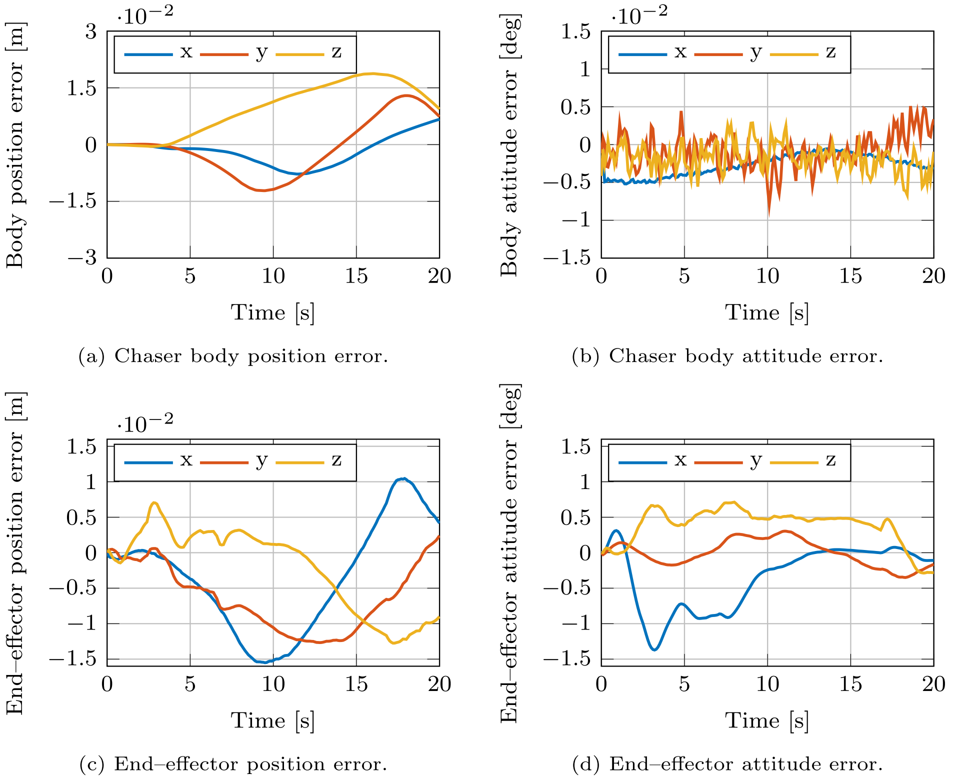
Fig. 19 Forces and torques acting on the system during phase B of SC1



(a) Torque commanded to the satellite body

(b) Torque commanded to the robotic arm joints

Fig. 20 Results of the nominal simulation of SC2—Phase A



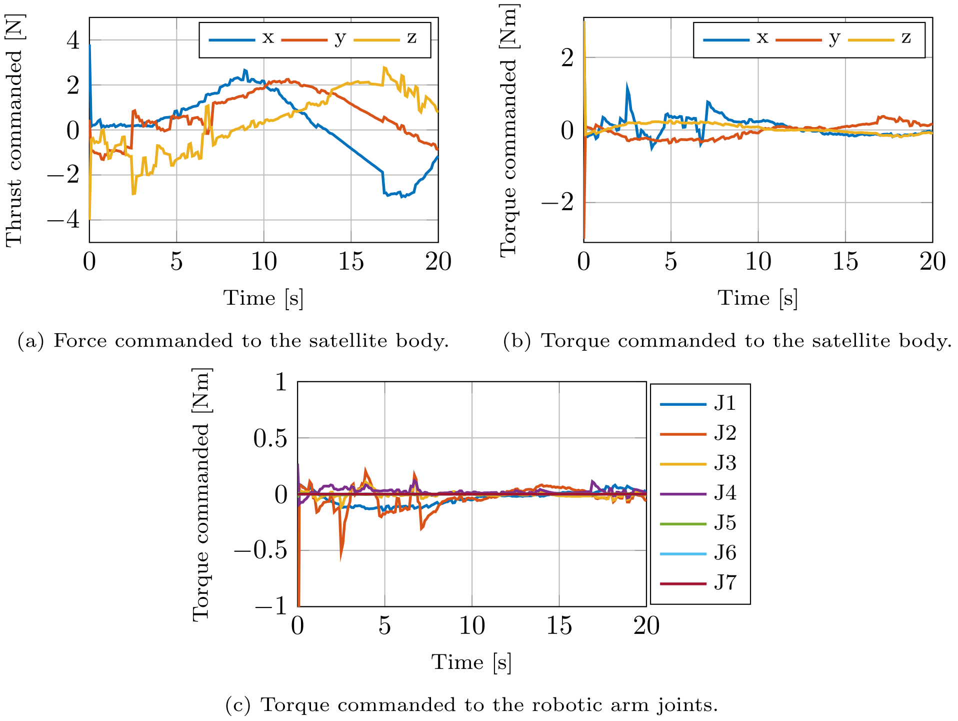
(a) Chaser body position error.

(b) Chaser body attitude error.

(c) End–effector position error.

(d) End–effector attitude error.

Fig. 21 Forces and torques acting on the system during phase A of SC2



(a) Force commanded to the satellite body.

(b) Torque commanded to the satellite body.

(c) Torque commanded to the robotic arm joints.

of the nozzle valve and the modulation of the thrust commanded according to the PWPFM (Pulse Width Pulse Frequency Modulation) scheme. Both these phenomena are implemented in the FES. Despite all these sources of error, the norm of the final position error is less than 0.015 m. Figure 20d shows the effect of different disturbances on the end-effector attitude control. In addition to the delays and overshoots introduced by the low-level control of the joints

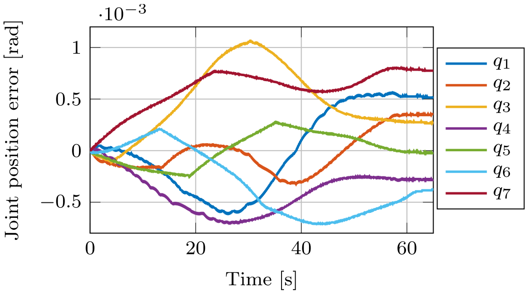


Fig. 22 Joint position error in SC2—Phase B

motors, the sloshing amplifies the oscillations by disturbing the chaser body, i.e., the base of the robotic arm. Nevertheless, the GNC system is able to compensate for the disturbances and the attitude error norm of the end-effector at the end of the maneuver is lower than 0.5 deg. Figure 21 shows the forces and torques commanded by the GNC algorithm to the satellite body and to the robotic arm joints during Phase A. The same considerations made for the starting torque of the joints and actuators in SC1 can be done for this scenario.

Phase B lasts for 60 s with the addition of five seconds similarly to what was done for SC1. Figure 22 shows that the robotic arm joint position error stabilizes to a constant value in the order of 10^{-4} rad in the desired time. The noise of the error is due to the optical encoders, which behave as real sensors thus introducing non-ideal measurements in the system. During the maneuver, the joints provide torques lower than 1 N m and, after the manipulator stiffening, the torques generated are stable and close to zero. Figure 23 shows the torques commanded by the GNC algorithm to the satellite body and to the robotic arm joints during Phase B. As for SC1, during Phase B, only the RWs are used to stabilize the stack. Hence, the force command coming from the GNC algorithm, i.e., the force that the thrusters must apply, is zero for the whole maneuver and is not represented in Fig. 23. The torques commanded to the satellite RWs and to the robotic arm joints are compatible with the response time of the actuators that are able of performing successfully the stabilization maneuver as shown in Fig. 22.

6.1.3 Scenarios results comparison

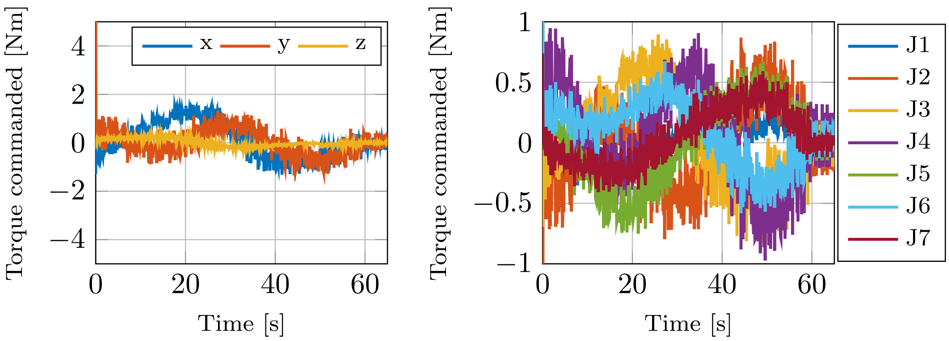
As presented in Sects. 2.3 and 2.4, the developed GNC algorithm shows good performance in both the considered scenarios. The developed GNC system has proven versatile since in both the scenarios the performance metrics are well within the requirements. However, the errors obtained in SC1 are lower than those of SC2. This is due to the fact that in SC1 the target is three axes stabilized and maintains its attitude during the Close Proximity Operation maneuver, while the target is rotating with an angular velocity of 2.5 deg/s in SC2. The latter is a more challenging scenario for the GNC system. The rotation of the target poses serious challenges to the navigation function that has to deal with the relative motion of the marker used for the pose determination and prediction of the target.

6.2 Error budget analysis

In real systems, a large number of sources contributes to the overall control error, e.g., external disturbances, model uncertainties, and nonlinearities. The EB analysis responds to the need to quantify the portion of the total error due to each source. The analysis is based on a number of simulations, singularly used to determine the influence of each identified error source. These simulations refer to ideal cases where all the parameters are set at their nominal value, except from the one to be examined, which is set considering the maximum expected deviation from nominal. The considered sources of errors are the following:

- real actuators behavior (thrusters, RW, DC motor);
- real sensors behavior (GNSS, Star Tracker, IMU, Optical Encoder);
- propellant sloshing dynamics;

Fig. 23 Forces and torques acting on the system during phase B of SC2



(a) Torque commanded to the satellite body.

(b) Torque commanded to the robotic arm joint

- environmental disturbances (gravitational harmonics, geomagnetic field, solar radiation pressure, single body gravity gradient, third body attraction);
- navigation function;
- mass, inertia, CoM position, sloshing mode uncertainties of chaser and target.

The obtained results are compared with a benchmark simulation executed considering ideal parameters only. The resulting control error (i.e., the difference between performance metrics in the benchmark case and those of the executed simulation) is therefore a function of the considered error source. In the conservative hypothesis that the error sources are not correlated and that they can be considered small perturbations around the current working point (so that linearization can be performed), they can be linearly summed, leading to the complete error estimation.

The EB analysis shows that, for both the scenarios, the main sources of control error affecting the performance of the chaser body are different from those affecting the robotic arm. The BLDC joint motors and the optical encoders contribute to the end-effector pose error but their effects are negligible for what concerns the metrics related to the chaser body.

The common factor for the two scenarios is that the state estimates from navigation contribute in a major way to the overall control error. This is an expected result considering how the benchmark simulation is executed. Specifically, this simulation is run providing the real state of the system to the guidance and control functions, which then operate as if all the sensors and estimation algorithms performed ideally. Differently, the error generated by the navigation function is obtained by executing a nominal closed-loop simulation relying on the navigation function to provide the state estimates to the system. The state estimates error comes from the combined operation of the relative and the absolute navigation algorithms.

Table 10 Major sources of error that influence the performance metrics in SC1

Performance metric	Error source	Order of magnitude
Chaser body pos. error [m]	Thruster	10^{-3}
	Navigation fun	10^{-3}
End-effector pos. error [m]	Thruster	10^{-3}
	Navigation fun	10^{-3}
	Chaser CoM uncert	10^{-3}
	Target CoM uncert	10^{-3}
End-effector att. error [deg]	Thruster	10^{-2}
	BLDC motors	10^{-2}
	Optical encoders	10^{-2}
	Navigation fun	10^{-2}
	Chaser CoM uncert	10^{-2}
	Target CoM uncert	10^{-2}

6.2.1 Scenario 1

The major contributions to the control error of the chaser base are due to the state estimates from the navigation function and the thrusters model (see Table 10). The error generated by the simulation of thrusters is explained considering that they have a pulsed actuation affected by delays and uncertainties: the continuous control input is converted in a PWM logic introducing an inaccuracy in the simulation.

Taking the end-effector metrics into account, different actors contribute to the control error: the BLDC joint motors, the optical encoders and the uncertainties of the center of mass of the chaser and target are the most relevant

Table 11 Major sources of error that influence the performance metrics in SC2

Performance metric	Error source	Order of magnitude
Chaser body pos. error [m]	Thruster	10^{-2}
	Navigation fun	10^{-3}
Chaser body att. error [deg]	Star Tracker	10^{-3}
End-effector pos. error [m]	Thruster	10^{-2}
	Navigation fun	10^{-3}
	Chaser CoM uncert	10^{-3}
	Target CoM uncert	10^{-3}
	Propellant sloshing	10^{-3}
End-effector att. error [deg]	Thruster	10^{-2}
	Navigation fun	10^{-1}
	Chaser CoM uncert	10^{-1}
	Target CoM uncert	10^{-1}

Table 12 Variability ranges of the parameters in the MC analysis

Parameter uncertainties	SC1	SC2
Chaser mass [%]	5	5
Target mass [%]	10	10
Chaser MoI [%]	10	10
Target MoI [%]	20	20
Chaser PoI [kg m^2]	100	1
Target PoI [kg m^2]	100	1
Chaser CoM position [m]	0.2	0.05
Target CoM position [m]	0.2	0.05
Sloshing frequency [%]	20	20
Sloshing damping [%]	40	40

(see Table 10). The BLDC joint motors introduce an error mainly due to their low-level control (PID). The optical encoders generate a quantization error and a constant bias. The error generated by the uncertainty on the CoM position is explained considering that the guidance is designed starting from the expected center of mass of the satellites. If the actual CoM does not correspond to the one expected by the guidance, the trajectory is generated with an error that affects the performance.

6.2.2 Scenario 2

As for SC1, the predominant error sources for the chaser body performance metrics are the thrusters and the error generated by the navigation input. In this scenario the sloshing loads come into play in the EB. The error generated by them has the same magnitude of the error generated by the navigation input for almost every performance metrics of the chaser body (see Table 11).

Taking the end-effector performance into consideration, the major sources of error that influence its performance are the ones identified for the chaser body (thrusters, error generated by the navigation input and propellant sloshing). In addition, the analysis shows that, although they play a role in the error budget, real models of BLDC motors and optical encoders do not bring a relevant contribution to the overall error. The CoM position uncertainties of the chaser and the target are not negligible. The error that they generate is on average one order of magnitude greater than the same error estimated for SC1.

6.3 Preliminary Monte Carlo analysis

In a real IOS mission, some parameters are known with uncertainty, e.g., the mass of the target spacecraft. The variability in the scenario parameters affects the control performance. The simulation in nominal conditions is not sufficient to accurately determine the system capabilities. In order to assess the GNC system robustness, a preliminary MC analysis has been executed. During the analysis 100 simulation are executed for each scenario varying with Gaussian probability distribution the spacecraft mass, inertia, CoM position and propellant sloshing mode within the ranges reported in Table 12. The number of executed simulations is not considered a statistically relevant sample, for which a much larger number of simulations would be required. Nevertheless, preliminary results can be obtained also with this limited number of simulations, while proving the capabilities of the MC tool implemented in the FES.

For each simulation, the performance metrics are considered and their deviation from the zero-error condition is calculated. Then, the mean value of each performance metric is computed; this value is averaged over the 100 simulations, allowing to compute a mean and a standard deviation (1σ). In this fashion, it is possible to compare the results with the values obtained with the nominal simulation and determine the error growths due to a variation of the previously defined parameters.

Table 13 SC1 Monte Carlo analysis results. The mean of the NS and the mean and standard deviation (1σ) of the Monte Carlo simulations for the considered performance metrics are presented

Parameter	NS mean	MC mean + STD (1σ)
Chaser body position error [m]	0.0036	0.06 ± 0.03
Chaser body attitude error [deg]	0.0034	0.0035 ± 0.0002
End-effector position error [m]	0.005	0.028 ± 0.015
End-effector attitude error [deg]	0.39	0.43 ± 0.09

Table 14 SC2 Monte Carlo analysis results. The mean of the NS and the mean and standard deviation (1σ) of the Monte Carlo simulations for the considered performance metrics are presented

Parameter	NS mean	MC mean + STD (1σ)
Chaser body position error [m]	0.012	0.025 ± 0.008
Chaser body attitude error [deg]	0.0049	0.0053 ± 0.0005
End-effector position error [m]	0.012	0.013 ± 0.002
End-effector attitude error [deg]	0.67	0.69 ± 0.08

Considering SC1, the GNC system proves good performance keeping the chaser body and the end-effector attitude errors comparable with those obtained from the NSs (Table 13). The position errors of the chaser body and of the end-effector have an order of magnitude of 10^{-2} m. Considering the wide uncertainty levels of the chaser and target parameters selected in this study, these results show the robustness of the GNC algorithm.

For what concerns SC2, the results of the preliminary MC analysis (Table 14) show that the errors of the considered performance metrics have the same order of magnitude of those of the NSs. The algorithm handles the levels of uncertainty considered in the MC analysis well. Given the differences between the two scenarios, the preliminary MC analysis indicates that the developed algorithm is robust and versatile.

7 Conclusions

This paper presents the development and numerical validation of GNC algorithms for the combined control of an autonomous spacecraft equipped with a robotic arm to perform the capture of a target object.

Three different analyses have been performed to assess the GNC system performance. First, nominal simulations of the scenarios were conducted to estimate the performance of the system in the expected orbital condition. Second, an Error Budget analysis has allowed to determine the contribution of each error source to the overall control error; finally, a preliminary Monte Carlo analysis assessed the robustness to uncertain parameters of the developed GNC system.

The results proved the good performance of the developed algorithms despite the important differences between the considered scenarios. The system is capable of performing the capture maneuvers and the after-capture robotic arm stiffening of the multibody system in scenarios with challenging dynamic conditions. The adoption of a multi-variable combined control approach and of a fully autonomous navigation architecture based on active or passive EO sensors allows to successfully cope with the intrinsic uncertainties of the close proximity operations problem. Future development steps include Software-In-the-Loop tests, Hardware-In-the-Loop tests and downscaled dynamic tests in a relevant environment on ground.

Acknowledgements The authors wish to thank the following researchers for their contribution, (alphabetically): Camilla Colombo, Giancarmine Fasano, Pietro Ghignoni, Michele Grassi, Pietro Iob, Lorenzo Maria Ticozzi, Claudio Vela, and Davide Vertuani.

Author contributions FBa, ZP, and FBr developed the simulation tool and performed the numerical simulations; GB, DI, MM, and ML developed the guidance and control algorithms; AN and RO developed the navigation algorithms, and IHG and PS supervised the research activity. FBa, ZP, GB, DI, AN, and RO wrote the manuscript; FBr, AF, MM, ML, IHG and PS reviewed the manuscript.

Funding Open access funding provided by Università degli Studi di Padova within the CRUI-CARE Agreement. The research activity has been funded by the European Space Agency through the General Support Technology Programme, contract n. 4000132611/20/NL/CRS.

Availability of data and materials Not applicable.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval Not applicable.

References

1. Flores-Abad, A., Ma, O., Pham, K., Ulrich, S.: A review of space robotics technologies for on-orbit servicing. Prog. Aerosp. Sci. **68**, 1–26 (2014). https://doi.org/10.1016/j.paerosci.2014.03.002
2. Akin, D., Sullivan, B.: A survey of serviceable spacecraft failures. In: AIAA Space 2001 Conference and Exposition (2001). https://doi.org/10.2514/6.2001-4540
3. Redd, N.T.: Bringing satellites back from the dead: Mission extension vehicles give defunct spacecraft a new lease on life - [news]. IEEE Spectr. **57**(8), 6–7 (2020). https://doi.org/10.1109/MSPEC.2020.9150540
4. Reintsema, D., Thaeter, J., Rathke, A., Naumann, W., Rank, P., Sommer, J.: Deos - the german robotics approach to secure and de-orbit malfunctioned satellites from low earth orbits. In: I-SAIRAS, Sopporo, Japan (2010)
5. Estable, S., Pruvost, C., Ferreira, E., Telaar, J., Fruhnert, M., Imhof, C., Rybus, T., Peckover, G., Lucas, R., Ahmed, R., Oki, T., Wygachiewicz, M., Kicman, P., Lukasik, A., Santos, N., Milhano, T., Arroz, P., Biesbroek, R., Wolahan, A.: Capturing and deorbiting envisat with an airbus spacetug. results from the esa e.deorbit consolidation phase study. J. Space Saf. Eng. **7**(1), 52–66 (2020). https://doi.org/10.1016/j.jsse.2020.01.003
6. ADC Steering Group and Working Group 4: IADC space debris mitigation guidelines. In: Inter-Agency Space Debris Coordination Committee (IADC) (2020). Rev.3
7. Maclay, T., Goff, J., Sheehan, J.P., Han, E.: The development of commercially viable adr services: introduction of a small-satellite grappling interface. J. Space Saf. Eng. **7**(3), 364–368 (2020). https://doi.org/10.1016/j.jsse.2020.08.002
8. Pavanello, Z., et al.: Combined control and navigation approach to the robotic capture of space vehicles. In: 72 Nd International Astronautical Congress (IAC) (2021)
9. Basana, F., Branz, F.: Simulation of robotic space operations with minimum base reaction manipulator. J. Space Saf. Eng. (2022). https://doi.org/10.1016/j.jsse.2022.06.005
10. Reed, B.B., Smith, R.C., Naasz, B., Pellegrino, J., Bacon, C.: The restore-l servicing mission. AIAA Space and Astronautics Forum and Exposition, SPACE 2016 (2016). https://doi.org/10.2514/6.2016-5478
11. Wertz, J.R.: Spacecraft attitude determination and control, pp. 113–149. Kluwer Academic (1987)
12. Silverman, S., Abramson, H.N.: The dynamic behavior of liquids in moving containers, pp. 13–68. National Aeronautics and Space Administration (NASA) (1966). Chap. 2
13. Huang, S., Colombo, C., Gonzalo, J.L., Masserini, A., Nugnes, M., Vallini, L., Petit, M.: Preliminary mission analysis of active debris removal service for large constellations. In: 71st International Astronautical Congress (IAC) (2020)
14. Giordano, A.M., Ott, C., Albu-Schäffer, A.: Coordinated control of spacecraft's attitude and end-effector for space robots. IEEE Robot. Autom. Lett. **4**, 2108–2115 (2019). https://doi.org/10.1109/LRA.2019.2899433
15. Dubanchet, V.: Modeling and control of a flexible space robot to capture a tumbling debris. Phd thesis, Ecole Polytechnique de Montreal, Canada (2016)
16. Skogestad, S., Postlethwaite, I.: Multivariable Feedback Control - Analisys and Design **1**, 355–404 (2001)
17. Colmenarejo, P., et al.: Methods and outcomes of the COMRADE project - Design of robust Combined control for robotic spacecraft and manipulator in servicing missions: comparison between between H_∞ and nonlinear Lyapunov-based approaches. In: 69th International Astronautical Congress (IAC) (2018)

18. Rodrigues, R., Preda, V., Sanfedino, F., Alazard, D.: Modeling, robust control synthesis and worst-case analysis for an on-orbit servicing mission with large flexible spacecraft. Aerosp. Sci. Technol. **129**, 107865 (2022). https://doi.org/10.1016/j.ast.2022.107865
19. Mishra, H., De Stefano, M., Giordano, A.M., Lampariello, R., Ott, C.: A geometric controller for fully-actuated robotic capture of a tumbling target. In: 2020 American Control Conference (ACC), pp. 2150–2157 (2020). https://doi.org/10.23919/ACC45564.2020.9147294
20. Siciliano, B., Khatib, O.: Handbook of robotics. Springer (2008). https://doi.org/10.1007/978-3-319-32552-1
21. Dubowsky, S., Papadopoulos, E.: The kinematics, dynamics, and control of free-flying and free-floating space robotic systems. IEEE Trans. Robot. Autom. **9**, 531–543 (1993). https://doi.org/10.1109/70.258046
22. Papadopoulos, E., Dubowsky, S.: Coordinated manipulator/spacecraft motion control for space robotic systems. Proceed.-IEEE Int. Conf. Robot. Autom. **2**, 1696–1701 (1991). https://doi.org/10.1109/ROBOT.1991.131864
23. Papadopoulos, S.E., Moosavian, A.A.: Dynamics & control of multi-arm space robots during chase & capture operations. IEEE/RSJ/GI Int. Conf. Intell. Robots Syst. **3**, 1554–1561 (1994). https://doi.org/10.1109/IROS.1994.407648
24. Featherstone, R.: Rigid body dynamics algorithms. Rigid Body Dyn. Algorithms (2008). https://doi.org/10.1007/978-1-4899-7560-7
25. Walker, M.W., Orin, D.E.: Efficient dynamic computer simulation of robotic mechanisms. J. Dyn. Syst. Meas. Contr. **104**(3), 205–211 (1982). https://doi.org/10.1115/1.3139699
26. Farhad, A.: Optimal control of a space manipulator for detumbling of a target satellite. Proceedings - IEEE International Conference on Robotics and Automation, 3019–3024 (2009). https://doi.org/10.1109/ROBOT.2009.5152235
27. Dubanchet, V., Saussié, D., Alazard, D., Bérard, C., Peuvédic, C.L.: Modeling and control of a space robot for active debris removal. CEAS Space J. **7**, 203–218 (2015). https://doi.org/10.1007/S12567-015-0082-4/TABLES/5
28. Ankersen, F.: Guidance, navigation, control and relative dynamics for spacecraft proximity maneuvers. Phd thesis, Aalborg Universitet, Denmark (2010)
29. Aghili, F.: Coordination control of a free–flying manipulator and its base attitude to capture and detumble a noncooperative satellite. 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009, 2365–2372 (2009). https://doi.org/10.1109/IROS.2009.5353968
30. Inaba, N., Oda, M.: Autonomous satellite capture by a space robot - world first on-orbit experiment on a japanese robot satellite ets-vii. Proceed. - IEEE Int. Conf. Robot. Autom. **2**, 1169–1174 (2000). https://doi.org/10.1109/ROBOT.2000.844757
31. Telaar, J., Estable, S., De Stefano, M., Rackl, W., Lampariello, R., Ankersen, F., Gil-Fernandez, J.: Coupled control of chaser platform and robot arm for the e.deorbit mission. In: 10th International ESA Conference on Guidance Navigation andControl Systems (GNC), p. 4 (2017)
32. GMV Innovating Solutions: Investigation of Active Detumbling Solutions for Debris Removal - Final Presentation, ESA Industry Days, ESA, 18/07/2017
33. Dubanchet, V., Saussié, D., Alazard, D., Bérard, C., Le Peuvédic, C.: Motion planning and control of a space robot to capture a tumbling debris. In: Bordeneuve-Guibé, J., Drouin, A., Roos, C. (eds.) Advances in Aerospace Guidance, Navigation and Control, pp. 699–717. Springer, Cham (2015)
34. Virgili-Llop, J., Zagaris, C., Richard Zappulla, I., Bradstreet, A., Romano, M.: A convex-programming-based guidance algorithm to capture a tumbling object on orbit using a spacecraft equipped with a robotic manipulator. Int. J. Robot. Res. **38**(1), 40–72 (2019). https://doi.org/10.1177/0278364918804660
35. Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G.: Robotics Modelling. Planning and Control. Springer (2009)
36. Opromolla, R., Fasano, G., Rufino, G., Grassi, M.: A review of cooperative and uncooperative spacecraft pose determination techniques for close-proximity operations. Prog. Aerosp. Sci. **93**, 53–72 (2017). https://doi.org/10.1016/j.paerosci.2017.07.001
37. Du, X., He, Y., Chen, L., Gao, S.: Pose estimation of large noncooperative spacecraft based on extended PnP model. 2016 IEEE International Conference on Robotics and Biomimetics, ROBIO 2016, 413–418 (2016). https://doi.org/10.1109/ROBIO.2016.7866357
38. Mishra, H., Schmidt, P.: Motion and parameter estimation for the robotic capture of a non-cooperative space target considering egomotion uncertainty. In: 14th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA) (2017)
39. Wen, X., He, S., Qiao, G.F., Wang, D., Song, A., Kang, C.S., Lv, Z.: Uncertainty estimation of robot geometric parameters and end-effecter position based on new generation GPS. Math. Probl. Eng. (2019). https://doi.org/10.1155/2019/7830489
40. Opromolla, R., Nocerino, A.: Uncooperative spacecraft relative navigation with LIDAR-based unscented Kalman filter. IEEE Access **7**, 180012–180026 (2019). https://doi.org/10.1109/ACCESS.2019.2959438
41. Pasqualetto Cassinis, L., Fonod, R., Gill, E., Ahrns, I., Gil-Fernández, J.: Evaluation of tightly- and loosely-coupled approaches in CNN-based pose estimation systems for uncooperative spacecraft. Acta Astronaut. **182**, 189–202 (2021). https://doi.org/10.1016/j.actaastro.2021.01.035
42. Tweddle, B.E., Saenz-Otero, A.: Relative computer vision-based navigation for small inspection spacecraft. J. Guid. Control. Dyn. **38**, 969–978 (2014). https://doi.org/10.2514/1.G000687
43. Wen, Z., Wang, Y., Luo, J., Kuijper, A., Di, N., Jin, M.: Robust, fast and accurate vision-based localization of a cooperative target used for space robotic arm. Acta Astronaut. **136**, 101–114 (2017). https://doi.org/10.1016/j.actaastro.2017.03.008
44. Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F.J., Marín-Jiménez, M.J.: Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recogn. **47**(6), 2280–2292 (2014). https://doi.org/10.1016/j.patcog.2014.01.005
45. Opromolla, R., Vela, C., Nocerino, A., Lombardi, C.: Monocular-based pose estimation based on fiducial markers for space robotic capture operations in GEO. Remote Sens. (2022). https://doi.org/10.3390/rs14184483
46. Vela, C., Fasano, G., Opromolla, R.: Pose determination of passively cooperative spacecraft in close proximity using a monocular camera and AruCo markers. Acta Astronaut. **201**, 22–38 (2022). https://doi.org/10.1016/J.ACTAASTRO.2022.08.024
47. Zhang, L., Zhang, S., Yang, H., Cai, H., Qian, S.: Relative attitude and position estimation for a tumbling spacecraft. Aerosp. Sci. Technol. **42**, 97–105 (2015). https://doi.org/10.1016/J.AST.2014.12.025
48. Pesce, V., Haydar, M.F., Lavagna, M., Lovera, M.: Comparison of filtering techniques for relative attitude estimation of uncooperative space objects. Aerosp. Sci. Technol. **84**, 318–328 (2019). https://doi.org/10.1016/j.ast.2018.10.031
49. Leffens, E.J., Markley, F.L., Shuster, M.D.: Kalman filtering for spacecraft attitude estimation. J. Guid. Control. Dyn. **5**, 417–429 (2012). https://doi.org/10.2514/3.56190
50. Markley, F.L.: Attitude error representations for Kalman filtering. J. Guid. Control. Dyn. **26**(2), 311–317 (2003). https://doi.org/10.2514/2.5048

51. Clohessy, W., Wiltshire, R.: Terminal guidance system for satellite rendezvous. J. Aerosp. Sci. **27**, 653–658 (1960). https://doi.org/10.2514/8.8704
52. MATLAB: Simscape multibody 2020b. The MathWorks Inc., Natick, Massachusetts (2020)
53. Pavlis, N.K., Holmes, S.A., Kenyon, S.C., Factor, J.K.: The development and evaluation of the Earth Gravitational Model 2008 (EGM2008). J. Geophys. Res.: Solid Earth (2012). https://doi.org/10.1029/2011JB008916
54. Alken, P., et al.: International geomagnetic reference field: the thirteenth generation. Earth Planets Space **73**, 1–25 (2021). https://doi.org/10.1186/S40623-020-01288-X/FIGURES/5
55. Government Publishing Office: Astronomical Almanac for the Year 2020. Government Printing Office, Astronomical Almanac For the Year. U.S (2019)
56. Picone, J.M., Hedin, A.E., Drob, D.P., Aikin, A.C.: NRLMSISE-00 empirical model of the atmosphere: statistical comparisons and scientific issues. J. Geophys. Res. Space Phys. (2002). https://doi.org/10.1029/2002JA009430
57. List, M., Bremer, S., Rievers, B., Selig, H.: Modelling of solar radiation pressure effects: parameter analysis for the MICROSCOPE mission. Int. J. Aerosp. Eng. (2015). https://doi.org/10.1155/2015/928206
58. Dodge, F.T.: The New “dynamic Behavior of Liquids in Moving Containers’’. Southwest Research Inst, San Antonio, Texas (2000)
59. Zhang, X.-Y., Liu, X.-F., Wang, M.-M., Cai, G.-P., Liu, X.-F., Zhang, X.-Y.: A new strategy for capturing a noncooperative spacecraft by a robotic arm. Multibody Sys.Dyn. (2022). https://doi.org/10.1007/s11044-022-09868-3
60. Yoshida, K., Sashida, N.: Modeling of impact dynamics and impulse minimization for space robots. In: 1993 International Conference on Intelligent Robots and Systems, 2064–2069 (1993). https://doi.org/10.1109/IROS.1993.583915
61. Yoshida, K., Sashida, N., Kurazume, R., Umetani, Y.: Modeling of collision dynamics for space free-floating links with extended generalized inertia tensor. Proceed. - IEEE Int. Conf. Robot. Autom. **1**, 899–904 (1992). https://doi.org/10.1109/ROBOT.1992.220182