Acta Astronautica 192 (2022) 368–378

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Direct visual servoing and interaction control for a two-arms on-orbit servicing spacecraft

José L. Ramón [a], Jorge Pomares [a,*], Leonard Felicetti [b]

[a] University of Alicante, Department of Physics, Systems Engineering and Signal Theory, San Vicente del Raspeig, 03690, Alicante, Spain
[b] Cranfield University, School of Aerospace, Transport and Manufacturing, Cranfield, MK43 0AL, United Kingdom

ARTICLE INFO

Keywords:
Visual servoing
Space manipulator
Interaction control

ABSTRACT

A direct visual-servoing algorithm for control of a space-based two-arm manipulator is proposed in this paper. The algorithm can be utilized in a two-arm manipulators configuration, where one of the arms performs the manipulation and the second arm is dedicated to the observation of the target zone of manipulation. The algorithm utilizes both visual features extracted from onboard cameras and force and torque measured at the manipulator's end-effector to control the movements of the manipulator during on-orbit servicing operations. The algorithm takes into account the relative dynamics of the bodies involved, it relies on images taken independently from de-localized cameras, e.g. at the end-effector of a second manipulator, and it integrates an impedance control for the compensation of eventual contact reactions when the end effector touches and operates the target body. The analytical derivations demonstrate the stability of the algorithm and incorporate an impedance compliance strategy into an optimal framework formulation. Simulations results in two different scenarios have been presented to show the adequate behavior of the presented approach in on-orbit-servicing operations.

1. Introduction

On-orbit manipulation plays an essential role in current and future space debris removal, on-orbit servicing, assembly, rescue, repair or refueling missions [1,2]. On-going active debris removal missions, such as the Astroscale Elsa-D [3] and Clearspace-1 [4], are starting to demonstrate technologies needed to approach and manipulate debris. Vehicles such as the Northrop Grumman's Mission Extension Vehicle-1 (MEV-1) have also demonstrated the feasibility of commercial on-orbit servicing and life extension missions [5]. In the near future, robotic servicing spacecraft may perform a wider range of operations which will involve even more complex tasks with higher levels of autonomy, reliability, efficiency, and safety. Under development programs, such as the DARPA's Robotic Servicing of Geosynchronous Satellites (RSGS) [6], NASA's On-orbit Servicing, Assembly and Manufacturing – 1 (OSAM-1) [7], previously known as Restore-L [8], will rely on several robotic arms and autonomous systems for performing complex operations, such as in-orbit repairing, assemble and manufacturing of satellites.

In this flourishing context, the development of efficient and reliable on-orbit manipulation techniques represents one of the challenging aspects and the key enabler for the success of such missions. Robotic operations performed in very close proximity represent one of the most dangerous phases of the overall mission, where two spacecraft (namely, the servicing spacecraft and the target spacecraft) might collide and be damaged in case of errors or unforeseen occurrences [9]. Such operations also impose very strict requirements in terms of accuracy of the robotic maneuvers, which need to be performed in free-floating conditions: with the attitude of the servicing spacecraft moving as a consequence of the motion of the robotic arms [10]. In addition, the contact dynamics in free-floating condition and the eventual manipulation of components of the target satellite might also represent a challenge for the robotic operations, especially when the target satellite is uncontrolled non-cooperative conditions [11]. Torque and force sensors located on end effectors of the robotic manipulators can be generally used to sense and evaluate the contacts among the bodies and eventually compensate for eventual errors in the positions of specific tools and interfaces located at the manipulator end to perform a given task such as a cutting or unscrewing tool [12].

The relative position and attitude awareness between the two spacecraft represent another critical aspect of the operations. The servicing spacecraft shall continuously monitor the position of the devices and elements that need to be manipulated and avoid eventual

* Corresponding author.
E-mail address: jpomares@ua.es (J. Pomares).

https://doi.org/10.1016/j.actaastro.2021.12.045
Received 15 June 2021; Received in revised form 23 November 2021; Accepted 23 December 2021
Available online 29 December 2021

collision with other elements of the target spacecraft [13]. Such a task is often performed by onboard cameras in monocular [13,14] or stereo-vision [15] configurations or by more complex systems such as lidars [16], radars [17] or infrared sensors [18]. In general, camera-based approaches are preferred in such tasks as they have higher technology readiness levels, a higher degree of reliability and better versatility than other solutions [19]. When it comes to robotic manipulation, where specific operations require fine adjustments of the position of the end-effector of the robotic manipulator, partial obstructions or shadows might arise. These issues can strongly affect the end-effector positioning and become critical if the camera cannot move and focus on the specific target zone where the operation needs to be performed. For this reason, eye-in-hand architectures or dual manipulators can represent suitable solutions for overcoming these problems [20,21]. In this paper a spacecraft is equipped with two robotics arms for doing on-orbit-servicing operations. One of the robotics arms has a camera at its end-effector and is employed to extract the required visual features to perform the guidance of the other manipulator arm. This last arm has at its end effector the required tool for a given task, and a force sensor employed to measure the contact forces. Specifically, this paper proposes a new direct image-based visual servoing control for tracking manipulator's trajectories defined directly on the camera image plane by taking into account contact forces when the manipulator interacts with the target spacecraft. Visual servoing control can indeed be subdivided into two main categories [22]: position-based visual servoing controls are those where the reference trajectory is defined via a tridimensional position and orientation of the end effector. In contrast, image-based visual servoing systems utilize a reference trajectory directly expressed in the camera image plane. This second typology of control is adopted and extended in this paper.

A survey of visual servoing strategies and applications to ground-based and space-based systems can be found in Refs. [22,23], respectively. In Ref. [24], a generalized Jacobian matrix-based visual servoing approach is proposed to control a satellite-mounted multiple arms robotic system. An alternative visual servoing controller is developed in Ref. [25] to reduce the disturbances produced by the manipulator motion on the main spacecraft in free-floating conditions during the capture of non-cooperative targets. A vision-based incremental kinematic control strategy that utilizes an extended Kalman filter based on image photograms is proposed in Ref. [26] for a robotic manipulator performing autonomous capture. All the aforementioned approaches are considered indirect visual servoing systems. The control action is specified in terms of velocity to be applied to the end effector and does not take the system dynamics into account. On the other hand, in direct controllers, the control actions are directly computed in terms of forces and torques applied to the robot joints. Hardware in the loop [26] and ground verification systems [27,28] are proposed to experimentally verify and test the reliability of visual servoing control systems and path planning of space robots. These studies show the necessity to integrate the robot dynamics in the visual servoing systems using a direct approach. In Ref. [29] a direct image-based visual control system is proposed to guide a free-floating manipulator using a camera located at the robot end. Previous works by the authors, such as [23,30], proposed direct visual servoing without interaction control to guide free-floating robotic manipulators and performing spacecraft rendezvous manoeuvres, respectively. In contrast with these previous approaches, in this paper a new visual-impedance control law has been designed based on a first-order impedance control law. Additionally, in the approach presented in this paper, a servicing spacecraft equipped with two robotic arms is proposed (robotic camera and manipulator). Therefore, the tracking controller formulation also includes the necessity to guide the manipulator considering the free-floating conditions and the image information obtained from the robotic camera.

The visual servoing controller drives both the camera and the manipulation systems in a regime of free-floating dynamics. As it is shown throughout the paper, the concept of virtual features is employed to allow the guidance of the manipulator arm independently of the camera position. The objective of the visual controller is to let such virtual features match the ones attached to the target driving the robotic arm to perform a tool insertion task on the body of the target satellite. Such a visual-servoing controller uses the images captured from a camera located to a second manipulator to avoid possible occlusions of the field of view. In this way, as it is more detailed explained throughout the paper, the robotic manipulator can be guided independently of the unknown camera position, which can be moved to guarantee the visibility of the visual features. Only a set of visual features needs to be extracted from the observed spacecraft and then tracked in the image plane during the maneuvers, allowing for performing relative navigation without any estimation of both relative position and attitude of the target. Different methods can be used for extracting the visual features [31,32] and their space application has been verified by taking into account different simulated lighting conditions [33,34]. In this paper, it is assumed that all the image and pixels signal processing operations have been already performed by an image-processing unit, and a set of well-defined visual features in the image plane can be utilized by the controller to perform the desired maneuvers.

In addition, the proposed visual servoing algorithm uses force and torque measurements at the end effector to increase the system robustness when the manipulator interacts with the target body. This represents a challenging problem indeed, especially for space robotics operations, as demonstrated by some researches found in the literature. Several previous works have been presented to combine visual and force information in on-orbit operations [35,36]. These last previous works employ an impedance-based approach during on-orbit docking operations. Additionally, an impedance controller and several control strategies are proposed in Ref. [37] to regulate the motion of a robot-based spacecraft docking system. In Ref. [38] the concept of a servicing spacecraft with a force-controlled manipulation system along with an experimental model and a facility is designed to test and tune the proposed control system. The free-floating robot dynamics is taken into account in Ref. [39], where an impedance control is used to perform a grasping manoeuvre of a tumbling target with model uncertainty.

The algorithm presented in this paper proposes image-based visual impedance control law, which combines both the inputs from the camera and the force sensor simultaneously. In this way, the controller can guarantee the correct pose and insertion of specific tools on the body of the target spacecraft by adopting compliant behaviour that can correct for eventual misalignments of the two platforms due to the relative free-floating dynamics. Specifically, the optimization of a cost index, as already done by the authors in previous studies [40], will lead to the definition of new direct visual servoing impedance controllers, which take the robot kinematics and dynamics into account while performing the task of tracking trajectories on the image plane. The tracking performance is compared with respect previous acceleration-based approaches [41] to highlight the tracking precision of the presented approach. When an insertion task is performed, the controllers can compensate for the tool's eventual misalignments that need to be inserted into the target satellite. The approach can be extended to different space robotics applications such as docking, assembly and on-orbit maintenance or refueling.

The different sections of the paper are organized as follows: Section 2 describes the proposed scenario. The kinematics and dynamics of both manipulators and spacecraft are detailed in this section. The visual-based kinematics and the servo control in the presence of contact dynamics are described in Section 3 and Section 4, respectively. Section 5 presents the optimal framework for developing the final image-based visual servoing controllers based on the achievements obtained in the previous two sections. Section 6 presents the simulation results obtained in two on-orbit servicing scenarios to validate the proposed approach. Finally, Section 7 summarizes the main concluding remarks associated with this study.

2. System architecture and dynamics

This section describes the main kinematic and dynamic properties of the satellite and robotic system studied in this paper. Fig. 1 shows the main components of the considered on-orbit servicing scenario. A servicing spacecraft is represented with a coordinate frame B located at its center of mass. This servicing spacecraft is equipped with two different robotic arms called as *manipulator arm* and *robotic camera* throughout the paper.

The manipulator arm will perform some manipulation tasks over the target spacecraft. This manipulator arm is an anthropomorphic robotic arm with *ne* degrees of freedom. The manipulator presents a force sensor at its end-effector which will be employed to measure the contact forces and moments when the manipulator touches the target satellite. The frame E is located at the end of the robotic manipulator.

The main objective of the robotic camera is to extract the required visual information to guide the manipulator arm. This robotic camera is also an anthropomorphic robotic arm with *nc* degrees of freedom. Additionally, this arm presents a camera at its end effector. The frame C is located at the end of the robotic camera.

Fig. 1 also shows the target spacecraft with several point features (the frame T is located at the target spacecraft center of mass). It is assumed that at least four coplanar, but not collinear, image features can be extracted using the robotic camera. More specifically, a set of m visual features are extracted by the camera at the end of the robotic camera manipulator. Finally, an Earth-Centered Inertial coordinate frame, called I, is considered as represented in Fig. 1.

With respect the system kinematics, the servicing spacecraft configuration can be defined by the position and attitude of the base spacecraft, $\boldsymbol{t}_b^I$ and $\boldsymbol{\varphi}_b^I$ (both with respect the inertia frame), and by the joint configuration of both the manipulator arm, $\boldsymbol{q}_e^T$, and the robotic camera, $\boldsymbol{q}_c^T$. The full kinematics of the spacecraft kinematics is defined by the vector $\boldsymbol{\varepsilon} = \left[\boldsymbol{t}_b^{I\,T}, \boldsymbol{\varphi}_b^{I\,T}, \boldsymbol{q}_e^T, \boldsymbol{q}_c^T\right]^T$. With respect the system dynamics, this information provides a relationship between the acceleration and forces and torques in both the base satellite and manipulators. More specifically, the system dynamics relates the linear and angular accelerations of the base spacecraft $\dot{\boldsymbol{v}}_b^I = [\dot{\boldsymbol{t}}_b^{I\,T}, \dot{\boldsymbol{\omega}}_b^{I\,T}]^T \in \Re^6$ expressed in the Inertial coordinate frame, the joint accelerations of the manipulator arm, $\ddot{\boldsymbol{q}}_e$, and robotic camera, $\ddot{\boldsymbol{q}}_c$, with the forces and torques exerted on the base of the servicing spacecraft, $\boldsymbol{h}_b^I \in \Re^6$, the torques applied on the robot manipulator joints, $\boldsymbol{\tau}_e \in \Re^{ne}$, and the torques applied on the robotic camera joints, $\boldsymbol{\tau}_c \in \Re^{nc}$. This relationship can be defined as [1]:

$$\begin{bmatrix} \boldsymbol{h}_b^I \\ \boldsymbol{\tau}_e \\ \boldsymbol{\tau}_c \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}_{bb} & \boldsymbol{M}_{be} & \boldsymbol{M}_{bc} \\ \boldsymbol{M}_{be}^T & \boldsymbol{M}_{ee} & 0 \\ \boldsymbol{M}_{bc}^T & 0 & \boldsymbol{M}_{cc} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{v}}_b^I \\ \ddot{\boldsymbol{q}}_e \\ \ddot{\boldsymbol{q}}_c \end{bmatrix} + \begin{bmatrix} \boldsymbol{c}_b \\ \boldsymbol{c}_e \\ \boldsymbol{c}_c \end{bmatrix} + \boldsymbol{J}^T \begin{bmatrix} 0_6 \\ \boldsymbol{h}_e^I \\ 0_6 \end{bmatrix} \tag{1}$$

where $\boldsymbol{M}_{bb} \in \Re^{6\times6}$ is the inertia matrix of the base spacecraft, $\boldsymbol{M}_{be} \in \Re^{6\times ne}$ is the coupled inertia matrix of the base spacecraft and the manipulator arm, $\boldsymbol{M}_{ee} \in \Re^{ne\times ne}$ is the inertia matrix of the manipulator arm, $\boldsymbol{M}_{bc} \in \Re^{6\times nc}$ is the coupled inertia matrix of the base spacecraft and the robotic camera, $\boldsymbol{M}_{cc} \in \Re^{nc\times nc}$ is the inertia matrix of the robotic camera. Additionally, $\boldsymbol{c}_b$, $\boldsymbol{c}_e$, and $\boldsymbol{c}_c \in \Re^6$ are velocity/displacement-dependent, non-linear terms for the base spacecraft, manipulator arm and robotic camera, respectively. It is also worth noting that eventual external wrenches $\boldsymbol{h}_e^I$ on the end effector can be projected into the joint space by using the Jacobian $\boldsymbol{J}^T$ and therefore can be included in the robot dynamics.

The first member of Eq. (1) includes the major forces and torques acting on the system. This paper assumes that the gravity gradient provides the major perturbing action acting on the system. Each of the gravitational forces and torques acting on the system can be computed by considering the multibody representation indicated in Fig. 2. In this figure, a reference frame b_i is associated to each body and the position of the origin of b_i with respect the origin of the Inertial frame is $\boldsymbol{t}_{b,i}^I$. Fig. 2 also shows as $\boldsymbol{t}_{cm,i}^I$ the position of the center of mass of each body, being m_i the corresponding mass of the body and $\boldsymbol{I}_{b,i}^I$ the moment of inertia of the body, respectively.

For each *i-th* body of the robotic systems, the forces $\boldsymbol{f}_{grav,i}^I$ and torques $\boldsymbol{\tau}_{grav,i}^I$ due to the gravity read as follows [42]:

$$\boldsymbol{f}_{grav,i}^I = -m_i \frac{\mu_\oplus}{t_{b,i}^{I\,3}} \boldsymbol{t}_{b,i}^I \tag{2}$$

$$\boldsymbol{\tau}_{grav,i}^I = -m_i \frac{\mu_\oplus}{t_{b,i}^{I\,3}} sk\left(\boldsymbol{t}_{cm,i}^I\right) \boldsymbol{t}_{b,i}^I + \frac{3\mu_\oplus}{t_{b,i}^{I\,3}} sk\left(\boldsymbol{t}_{b,i}^I\right) \boldsymbol{I}_{b,i}^I \boldsymbol{t}_{b,i}^I \tag{3}$$

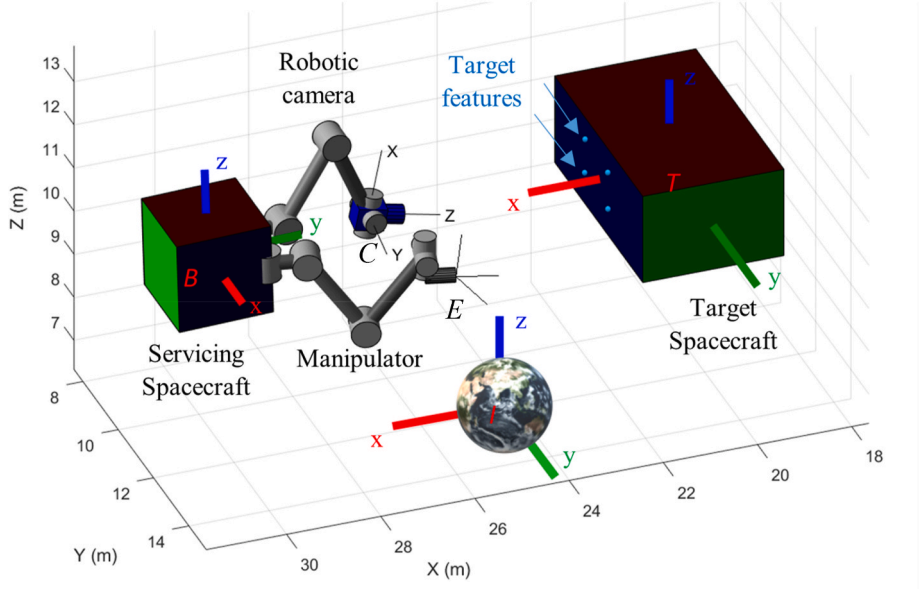

Fig. 1. On-orbit servicing scenario representation and definition of the adopted coordinate frames.

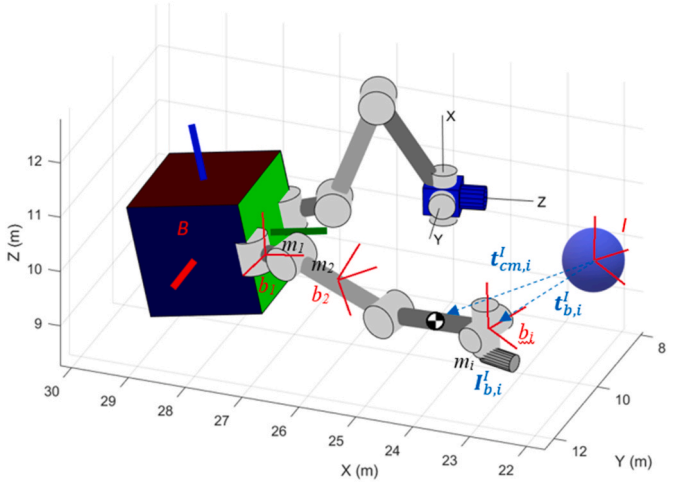

Fig. 2. Multibody representation of the robotic system.

where $\mu_{\oplus}$ is the Earth gravitational constant and $sk() \in \Re^{3x3}$ is the skew-symmetric matrix.

The contribution of such actions to the multibody dynamics of the system in Eq. (1) can be computed by projecting both forces and torques acting to each of the bodies to the space defined by the degrees of freedom of the system [43]. This operation can indeed be performed by determining the generalized Jacobian matrix $\boldsymbol{J}_\chi$ for the system which formalizes the dependencies between the base's and joints' velocities ($\dot{\boldsymbol{\epsilon}}$) and the full set of linear and angular velocities that characterize the state of all the bodies in the systems ($\dot{\boldsymbol{\chi}}$), as follows:

$$\dot{\boldsymbol{\chi}} = \boldsymbol{J}_\chi \dot{\boldsymbol{\epsilon}} \tag{4}$$

where the vector $\dot{\boldsymbol{\epsilon}}$ is defined as follows:

$$\dot{\boldsymbol{\epsilon}} = \left[{\dot{\boldsymbol{t}}^I_{b,b}}^T, {\boldsymbol{\omega}^I_{b,b}}^T, \dot{\boldsymbol{q}}_e^T, \dot{\boldsymbol{q}}_c^T \right]^T \tag{5}$$

and the time derivative of extended state space vector $\dot{\boldsymbol{\chi}}$ reads as:

$$\dot{\boldsymbol{\chi}} = \begin{bmatrix} {\dot{\boldsymbol{t}}^I_{b,b}}^T, {\boldsymbol{\omega}^I_{b,b}}^T, {\dot{\boldsymbol{t}}^I_{b,1}}^T, {\boldsymbol{\omega}^I_{b,1}}^T, \dots \\ \dots, {\dot{\boldsymbol{t}}^I_{b,ne}}^T, {\boldsymbol{\omega}^I_{b,ne}}^T, {\dot{\boldsymbol{t}}^I_{b,ne+1}}^T, {\boldsymbol{\omega}^I_{b,ne+1}}^T, \dots, {\dot{\boldsymbol{t}}^I_{b,ne+nc+1}}^T, {\boldsymbol{\omega}^I_{b,ne+nc+1}}^T \end{bmatrix}^T \tag{6}$$

Thus, the inclusion of such forcing terms within the system in Eq. (1) is done by defining the full set of the forcing actions to the systems as:

$$\widetilde{\boldsymbol{F}}_{grav} = \begin{bmatrix} {\boldsymbol{f}^I_{grav,b}}^T, {\boldsymbol{\tau}^I_{grav,b}}^T, {\boldsymbol{f}^I_{grav,1}}^T, {\boldsymbol{\tau}^I_{grav,1}}^T, \dots, {\boldsymbol{f}^I_{grav,ne}}^T, {\boldsymbol{\tau}^I_{grav,ne}}^T, \dots \\ {\boldsymbol{f}^I_{grav,ne+1}}^T, {\boldsymbol{\tau}^I_{grav,ne+1}}^T, \dots, {\boldsymbol{f}^I_{grav,ne+nc+1}}^T, {\boldsymbol{\tau}^I_{grav,ne+nc+1}}^T \end{bmatrix}^T \tag{7}$$

which can be then be projected in the joint space by using the Jacobian matrix $\boldsymbol{J}_\chi$ as follows:

$$\widetilde{\boldsymbol{\tau}}_{grav} = \boldsymbol{J}_\chi^T \widetilde{\boldsymbol{F}}_{grav} \tag{8}$$

with $\widetilde{\boldsymbol{\tau}}_{grav} = \left[{\boldsymbol{h}^I_{grav,b}}^T \quad \boldsymbol{\tau}^T_{grav,e} \quad \boldsymbol{\tau}^T_{grav,c} \right]^T$. Thus, Eq. (1) can be rewritten in a compact form taking the gravity forces into account, as follows:

$$\widetilde{\boldsymbol{\tau}} - \boldsymbol{J}^T \widetilde{\boldsymbol{F}}_e - \widetilde{\boldsymbol{\tau}}_{grav} = \widetilde{\boldsymbol{M}} \ddot{\boldsymbol{\epsilon}} + \widetilde{\boldsymbol{C}} \tag{9}$$

where $\widetilde{\boldsymbol{\tau}} = \left[{\boldsymbol{h}^I_b}^T \quad \boldsymbol{\tau}_e^T \quad \boldsymbol{\tau}_c^T \right]^T$, $\widetilde{\boldsymbol{M}}$ is the mass matrix of the entire system, $\widetilde{\boldsymbol{C}} = \left[\boldsymbol{c}_b^T \quad \boldsymbol{c}_e^T \quad \boldsymbol{c}_c^T \right]^T$ includes all the velocity/displacements dependent non-linear terms of the system, $\widetilde{\boldsymbol{F}}_e = \left[\boldsymbol{0}_6 \quad {\boldsymbol{h}^I_e}^T \quad \boldsymbol{0}_6 \right]^T$ the external wrenches on the end effector.

3. Visual-based kinematics and servo control

Fig. 1 shows a pattern of m points (target features) attached on the body of the target satellite that represents possible visual features that can be observed by the robotic camera. A set of m points are extracted from both, the target satellite, and the manipulator end. With respect the m points in the target satellite, these points have fixed positions with respect to, T, i.e. the target coordinate frame ($\boldsymbol{p}^T_{t,i} = \left[x^T_{t,i} \quad y^T_{t,i} \quad z^T_{t,i} \right]$, with $i = 1 \dots m$). When these points are measured in the robotic camera image space, the corresponding points will be $\boldsymbol{s}_{t,i} = [X_{t,i}, Y_{t,i}]^T \in \Re^2$. A pin-hole camera can be used to model this projection [44]:

$$\boldsymbol{s}_{t,i} = \begin{bmatrix} X_{t,i} \\ Y_{t,i} \end{bmatrix} = \frac{1}{z^C_{t,i}} \begin{bmatrix} x^C_{t,i} \\ y^C_{t,i} \end{bmatrix} \tag{10}$$

where $x^C_{t,i}$, $y^C_{t,i}$ and $z^C_{t,i}$ are the components of the vector $\boldsymbol{p}^C_{t,i}$ that can be obtained from $\boldsymbol{p}^T_{t,i}$ through the following rotation:

$$\boldsymbol{p}^C_{t,i} = \boldsymbol{R}^C_t \boldsymbol{p}^T_{t,i} + \boldsymbol{p}^C_t \tag{11}$$

where $\boldsymbol{R}^C_t$ is the rotation matrix between the target satellite coordinate frame and the robotic camera reference frame and $\boldsymbol{p}^C_t$ is the relative distance between the target and the camera reference frames.

A set of m features can also be extracted from the manipulator end. The controller is built upon the concept that the task is achieved when $\boldsymbol{s}_t = \boldsymbol{s}_e$, i.e., the target features are equal to the end-effector features. The visual-servoing controller aims to let the manipulator features, $\boldsymbol{s}_e$, match the ones attached to the target by driving the manipulator's arm to follow a specified trajectory by using the information retrieved with the robotic camera manipulator. Thus, the position of each of the features, $\boldsymbol{p}^E_{e,i}$, with $i = 1 \dots m$, will be considered constant with respect to the coordinate frame of the end-effector. The corresponding image features, $\boldsymbol{s}_{e,i} = [X_{e,i}, Y_{e,i}]^T \in \Re^2$, are obtained by taking into account the manipulator kinematics. These point features in the image space can be obtained either by computing their positions considering the manipulator kinematics or by measuring them directly in the image plane of the camera. This last approach is generally more robust with respect to calibration errors of the kinematics. However, it becomes difficult to guarantee the image features visibility during the considered tasks, e.g. when the robot manipulator hides the visual features and the visual servoing task cannot be accomplished. To avoid this problem, we have adopted the first approach: using the manipulator kinematics for computing the positions of the end-effector image features (virtual features).

The position of the point features with respect the camera coordinate frame, $\boldsymbol{p}^C_{e,i} = [x^C_{e,i}, y^C_{e,i}, z^C_{e,i}]^T \in \Re^3$, can be projected in the image space by using the relationship:

$$\boldsymbol{s}_{e,i} = \frac{1}{z^C_{e,i}} \begin{bmatrix} x^C_{e,i} \\ y^C_{e,i} \end{bmatrix} \tag{12}$$

where $\boldsymbol{s}_{e,i} = [X_{e,i}, Y_{e,i}]^T$ is the image coordinates of the i-th point. The variation of these features with respect the time can be obtained using the following equation:

$$\dot{\boldsymbol{s}}_{e,i} = \left[\dot{X}_{e,i}, \dot{Y}_{e,i} \right]^T = \boldsymbol{L}_{e,i} \dot{\boldsymbol{p}}^C_{e,i} \tag{13}$$

where the matrix $\boldsymbol{L}_{e,i}$ is equal to:

$$\boldsymbol{L}_{e,i} = \frac{1}{z^C_{e,i}} \begin{bmatrix} 1 & 0 & -X_{e,i} \\ 0 & 1 & -Y_{e,i} \end{bmatrix} \tag{14}$$

The following equation can be used to relate the time derivative of the feature points with respect the camera frame, $\dot{\boldsymbol{p}}^C_{e,i}$, and the positions of the same points with respect the manipulator end-effector, $\boldsymbol{p}^E_{e,i}$:

$$\dot{\boldsymbol{p}}_{e,i}^{C} = \begin{bmatrix} \boldsymbol{E}_3 & -sk\left(\boldsymbol{R}_E^C \boldsymbol{p}_{e,i}^E\right) \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_C^I & \boldsymbol{0}_{3x3} \\ \boldsymbol{0}_{3x3} & \boldsymbol{R}_C^I \end{bmatrix}^T \left(\boldsymbol{v}_e^I - \boldsymbol{v}_c^I\right) \tag{15}$$

where the rotation matrix $\boldsymbol{R}_C^I$ can be computed from the relative orientation between the inertial, I, and camera frame, C. Additionally, the velocities of both end-effector and robotic camera $\boldsymbol{v}_e^I$ and $\boldsymbol{v}_c^I$ are used in this last equation (both with respect the inertial frame). Note that with $\boldsymbol{E}_3$ is represented an identity matrix of size 3×3. The terms of Equation (15) can be used to define a new Jacobian matrix, $\boldsymbol{J}_{e,i}$, that defines the relationship between the time derivatives of the image features, $\dot{\boldsymbol{s}}_{e,i}$, and the previous indicated twists $\boldsymbol{v}_e^I$ and $\boldsymbol{v}_c^I$:

$$\dot{\boldsymbol{s}}_{e,i} = \boldsymbol{L}_{e,i} \begin{bmatrix} \boldsymbol{E}_3 & -sk\left(\boldsymbol{R}_E^C \boldsymbol{p}_{e,i}^E\right) \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_C^I & \boldsymbol{0}_{3x3} \\ \boldsymbol{0}_{3x3} & \boldsymbol{R}_C^I \end{bmatrix}^T \left(\boldsymbol{v}_e^I - \boldsymbol{v}_c^I\right) = \boldsymbol{J}_{e,i}\left(\boldsymbol{v}_e^I - \boldsymbol{v}_c^I\right) \tag{16}$$

The target features indicated in Equation (10) are now studied. In image-based visual servoing systems it is required to define the relationship between the image coordinates of a point and the position of the same coordinates in the 3D Cartesian space. This relationship can be defined by:

$$\dot{\boldsymbol{s}}_{t,i} = \left[\dot{X}_{t,i}, \dot{Y}_{t,i}\right]^T = \boldsymbol{L}_{t,i} \begin{bmatrix} \boldsymbol{R}_C^I & \boldsymbol{0}_{3x3} \\ \boldsymbol{0}_{3x3} & \boldsymbol{R}_C^I \end{bmatrix}^T \boldsymbol{v}_c^I + \frac{\partial \boldsymbol{e}_s}{\partial t} = \boldsymbol{J}_{t,i} \boldsymbol{v}_c^I + \frac{\partial \boldsymbol{e}_s}{\partial t} \tag{17}$$

The term $\frac{\partial \boldsymbol{e}_s}{\partial t}$ represents the time variation of $\boldsymbol{e}_s$ due to the generally unknown target motion. A Kalman filter [45] or more elaborate filtering methods [46] can then be used to improve the estimation of this term. The integration of this term and the use of an estimator allow to compensate for abrupt changes in the target motion and reduce the noise introduced throughout the extraction of the visual data. The interaction matrix $\boldsymbol{L}_{t,i}$ is the one defined in previous works such as [44]:

$$\boldsymbol{L}_{t,i} = \begin{bmatrix} -\frac{1}{z_{t,i}^C} & 0 & \frac{X_{t,i}}{z_{t,i}^C} & X_{t,i}Y_{t,i} & -1 - X_{t,i}^2 & Y_{t,i} \\ 0 & -\frac{1}{z_{t,i}^C} & \frac{Y_{t,i}}{z_{t,i}^C} & 1 + Y_{t,i}^2 & -X_{t,i}Y_{t,i} & -X_{t,i} \end{bmatrix} \tag{18}$$

The control objective will be the nullification of an error defined in the image space, $\boldsymbol{e}_s$. As previously indicated, at least four coplanar but not collinear points are extracted in both the target satellite and the manipulator end. In both cases, the corresponding image features extracted by the robotic camera are defined as $\boldsymbol{s}_t = [\boldsymbol{s}_{t1}, \boldsymbol{s}_{t2}, \ldots, \boldsymbol{s}_{tm}]^T$ and $\boldsymbol{s}_e = [\boldsymbol{s}_{e1}, \boldsymbol{s}_{e2}, \ldots, \boldsymbol{s}_{em}]^T$, respectively. Therefore, the task is achieved when $\boldsymbol{s}_t = \boldsymbol{s}_e$, i.e., the target features are equal to the end-effector features. Using the procedure presented in Eqs. (12)–(16) we can "virtualize" the value of the image point features $\boldsymbol{s}_e$, relaxing the field of view constraint. Therefore, to guide the manipulator to achieve the required location the image error is defined as $\boldsymbol{e}_s = \boldsymbol{s}_e - \boldsymbol{s}_t$.

4. Visual-based interaction control in the presence of contact dynamics

When the manipulator's end effect reaches the desired target position, it touches the target's surfaces. The proposed control scheme takes the contact dynamics between the two bodies into account to compensate for eventual reactions and disturbances produced during the contact. This study assumes that the target spacecraft has a greater mass than the servicing spacecraft. The target's motion does not change significantly due to the interaction with the servicer's end-effector. On the other hand, reaction forces produced by the contact dynamics can produce significant changes in both the position and attitude dynamics of the servicing spacecraft.

A damper-spring model is used in this paper to characterize this kind of interaction [47]:

$$\boldsymbol{D}\boldsymbol{v}_{ed}^I + \boldsymbol{\alpha} = \boldsymbol{h}_e^I \tag{19}$$

where $\boldsymbol{D}$ is the damping matrix, $\boldsymbol{v}_{ed}^I$ is the desired linear and angular velocity of the manipulator-end, $\boldsymbol{h}_e^I \in \Re^6$ is the external wrench action on the manipulator and $\boldsymbol{\alpha} \in \Re^6$ is the visual control approach described in the next paragraphs. To do this, the following Lyapunov function is first considered:

$$\boldsymbol{V}(\boldsymbol{e}_s) = \frac{1}{2}\boldsymbol{e}_s^T \boldsymbol{Q} \boldsymbol{e}_s \tag{20}$$

where $\boldsymbol{Q}$ is a diagonal positive definite matrix. The time derivative of the previous Lyapunov function is:

$$\dot{\boldsymbol{V}}(\boldsymbol{e}_s) = \boldsymbol{e}_s^T \boldsymbol{Q} \dot{\boldsymbol{e}}_s \tag{21}$$

Considering Eqs. (16) and (17), the image error time derivative is equal to:

$$\dot{\boldsymbol{e}}_s = \dot{\boldsymbol{s}}_e - \dot{\boldsymbol{s}}_t = \boldsymbol{J}_e\left(\boldsymbol{v}_{ed}^I - \boldsymbol{v}_c^I\right) - \boldsymbol{J}_t \boldsymbol{v}_c^I + \frac{\partial \boldsymbol{e}_s}{\partial t} = \boldsymbol{J}_e \boldsymbol{v}_{ed}^I - (\boldsymbol{J}_t + \boldsymbol{J}_e)\boldsymbol{v}_c^I + \frac{\partial \boldsymbol{e}_s}{\partial t} = \boldsymbol{J}_e \boldsymbol{v}_{ed}^I - \boldsymbol{J}_c \boldsymbol{v}_c^I + \frac{\partial \boldsymbol{e}_s}{\partial t} \tag{22}$$

where the Jacobians $\boldsymbol{J}_e$ and, $\boldsymbol{J}_t$ are defined as $\boldsymbol{J}_e = [\boldsymbol{J}_{e,1}^T, \boldsymbol{J}_{e,2}^T, \ldots, \boldsymbol{J}_{e,m}^T]^T \in \Re^{2mx6}$, $\boldsymbol{J}_t = [\boldsymbol{J}_{t,1}^T, \boldsymbol{J}_{t,2}^T, \ldots, \boldsymbol{J}_{t,m}^T]^T \in \Re^{2mx6}$ and $\boldsymbol{J}_c = \boldsymbol{J}_t + \boldsymbol{J}_e$.

The value of the time derivative of the Lyapunov function presented in Equation (20) can be obtained as:

$$\dot{\boldsymbol{V}}(\boldsymbol{e}_s) = \boldsymbol{e}_s^T \boldsymbol{Q}\left(\boldsymbol{J}_e \boldsymbol{v}_{ed}^I - \boldsymbol{J}_c \boldsymbol{v}_c^I + \frac{\partial \boldsymbol{e}_s}{\partial t}\right) = \boldsymbol{e}_s^T \boldsymbol{Q}\left(\boldsymbol{J}_e \boldsymbol{D}^{-1}\left(\boldsymbol{h}_e^I - \boldsymbol{\alpha}\right) - \boldsymbol{J}_c \boldsymbol{v}_c^I + \frac{\partial \boldsymbol{e}_s}{\partial t}\right) \tag{23}$$

Additionally, the visual servoing control law is defined as:

$$\boldsymbol{\alpha} = \boldsymbol{D}\left(\boldsymbol{K}\boldsymbol{J}_e^+ \boldsymbol{Q}\boldsymbol{e}_s - \boldsymbol{J}_e^+ \boldsymbol{J}_c \boldsymbol{v}_c^I + \boldsymbol{J}_e^+ \frac{\partial \boldsymbol{e}_s}{\partial t}\right) \tag{24}$$

where $\boldsymbol{K}$ a proportional positive matrix. Considering the previous visual servoing controller, the time derivative of the Lyapunov function results:

$$\dot{\boldsymbol{V}}(\boldsymbol{e}_s) = \boldsymbol{e}_s^T \boldsymbol{Q} \boldsymbol{J}_e \boldsymbol{D}^{-1} \boldsymbol{h}_e^I - \boldsymbol{e}_s^T \boldsymbol{Q} \boldsymbol{J}_e \boldsymbol{K} \boldsymbol{J}_e^+ \boldsymbol{Q} \boldsymbol{e}_s \tag{25}$$

This last equation depends on the external wrenches, $\boldsymbol{h}_e^I$, and the image error to be annulled. When the manipulator arm does not touch the target spacecraft, the time derivative of the Lapunov function is equal to $\dot{\boldsymbol{V}}(\boldsymbol{e}_s) = -\boldsymbol{e}_s^T \boldsymbol{Q} \boldsymbol{J}_e \boldsymbol{K} \boldsymbol{J}_e^+ \boldsymbol{Q} \boldsymbol{e}_s$. Therefore, system (19) is asymptotically stable and the systems achieves a state where the image error $\boldsymbol{e}_s$ is null when $\boldsymbol{J}_e$ is nonsingular. In the more general case, the external wrenches are not zero and the manipulator arm is touching the target spacecraft. In this case, from Eq. (24) and Eq. (19) we can obtain:

$$\boldsymbol{D}\boldsymbol{v}_{ed}^I + \boldsymbol{D}\boldsymbol{K}\left(\boldsymbol{J}_e^+ \boldsymbol{Q}\boldsymbol{e}_s - \boldsymbol{K}^{-1}\boldsymbol{J}_e^+ \boldsymbol{J}_c \boldsymbol{v}_c^I + \boldsymbol{K}^{-1}\boldsymbol{J}_e^+ \frac{\partial \boldsymbol{e}_s}{\partial t}\right) = \boldsymbol{h}_e^I \tag{26}$$

The desired interaction compliance can be defined in the Cartesian space using the following matrices:

- The proportional positive definite matrix $\boldsymbol{K}$ used in the visual servoing controller (Eq. (24))
- The damping matrix $\boldsymbol{D}$, defined in Eq. (19).
- The matrix $\boldsymbol{Q}$. This matrix can be employed to regulate the image convergence. Specifically, this is done in the next section by formulating an optimal control strategy for tracking the reference trajectory obtained from the interaction wrench and the image error.

5. Optimal control

The Cartesian task for tracking the reference trajectory obtained from the interaction wrench and the image errors is defined as follows:

$$\boldsymbol{A}(\boldsymbol{\varepsilon}, \dot{\boldsymbol{\varepsilon}}, t)\ddot{\boldsymbol{\varepsilon}} = \boldsymbol{b}(\boldsymbol{\varepsilon}, \dot{\boldsymbol{\varepsilon}}, t) \tag{27}$$

where the value $\boldsymbol{A}(\boldsymbol{\varepsilon},\ \dot{\boldsymbol{\varepsilon}},\ t)$ and $\boldsymbol{b}(\boldsymbol{\varepsilon},\ \dot{\boldsymbol{\varepsilon}},\ t)$ will be then calculated in the following paragraphs. For the tracking of the desired trajectory, we consider both the achievement of the desired trajectory and the minimization of a cost function defined using the following metric:

$$\Omega = \widetilde{\boldsymbol{\tau}}^T \boldsymbol{W} \widetilde{\boldsymbol{\tau}} \tag{28}$$

where $\boldsymbol{W}$ is a positive semi-definite matrix. For the tracking of the reference trajectory, the system dynamics is defined by:

$$\widetilde{\boldsymbol{\tau}} = \widetilde{\boldsymbol{M}}\ddot{\boldsymbol{\varepsilon}} + \widetilde{\boldsymbol{C}} + \widetilde{\boldsymbol{\tau}}_{grav} \tag{29}$$

The controller which minimizes the cost function in Eq. (28) for the robotic system defined in Equation (29) while the task defined as indicated in (27) is fulfilled is the following [23]:

$$\widetilde{\boldsymbol{\tau}} = \boldsymbol{W}^{-1/2}\left(\boldsymbol{A}\widetilde{\boldsymbol{M}}^{-1}\boldsymbol{W}^{-1/2}\right)^{+}\cdot\left(\boldsymbol{b} - \boldsymbol{A}\widetilde{\boldsymbol{M}}^{-1}\widetilde{\boldsymbol{C}}\right) + \widetilde{\boldsymbol{\tau}}_{grav} \tag{30}$$

where the symbol + denotes the pseudo-inverse of a general matrix. As it will be discussed in the next paragraphs, the definition of the matrix $\boldsymbol{W}$ plays an essential role in the resulting controller.

The goal of the Cartesian controller is to reduce the error $\boldsymbol{e} = \boldsymbol{x}^I - \boldsymbol{x}_d^I$ progressively while the cost function (28) is minimized. In this error, the quantity $\boldsymbol{x}^I = [\boldsymbol{t}_e^{I\,T}, \mathcal{Q}_e^{I\,T}, \boldsymbol{t}_c^{I\,T}, \mathcal{Q}_c^{I\,T},]^T \in \Re^{14}$ contains the current position vector and quaternion of the manipulator $(\boldsymbol{t}_e^I, \mathcal{Q}_e^I)$ and camera $(\boldsymbol{t}_c^I, \mathcal{Q}_c^I)$, respectively. The desired poses of the manipulator end-effector and robotic camera are defined by the desired state vector $\boldsymbol{x}_d^I = [\boldsymbol{t}_{ed}^{I\,T}, \mathcal{Q}_{ed}^{I\,T}, \boldsymbol{t}_{cd}^{I\,T}, \mathcal{Q}_{cd}^{I\,T}]^T \in \Re^{14}$, where $\boldsymbol{t}_{ed}^I$ and $\boldsymbol{t}_{cd}^I$ are the desired positions of the manipulator end-effector and camera, and $\mathcal{Q}_{ed}^I$ and $\mathcal{Q}_{cd}^I$ are the quaternions corresponding at the desired orientation of the manipulator end effector and the camera with respect to the Inertial frame. Both the current and desired robot poses are defined with respect to the Inertial frame. Similarly, $\boldsymbol{v}^I = [\dot{\boldsymbol{t}}_e^{I\,T}, \boldsymbol{\omega}_e^{I\,T}, \dot{\boldsymbol{t}}_c^{I\,T}, \boldsymbol{\omega}_c^{I\,T}]^T \in \Re^{12}$ and $\boldsymbol{v}_d^I = [\dot{\boldsymbol{t}}_{ed}^{I\,T}, \boldsymbol{\omega}_{ed}^{I\,T}, \dot{\boldsymbol{t}}_{cd}^{I\,T}, \boldsymbol{\omega}_{cd}^{I\,T}]^T \in \Re^{12}$ are the current and the desired velocity of both the manipulator end-effector and camera, where $\boldsymbol{\omega}_e^I$ and $\boldsymbol{\omega}_{ed}^I$ are the current and desired angular velocities of the end effector and, $\boldsymbol{\omega}_c^I$ and $\boldsymbol{\omega}_{cd}^I$ are the current and desired angular velocities of the camera, respectively.

Therefore, the task constraint can be defined as:

$$\left(\dot{\boldsymbol{v}}^I - \dot{\boldsymbol{v}}_d^I\right) + \boldsymbol{K}_d\left(\boldsymbol{v}^I - \boldsymbol{v}_d^I\right) + \boldsymbol{K}_p\left(\boldsymbol{x}^I - \boldsymbol{x}_d^I\right) = 0 \tag{31}$$

where $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$ are proportional and derivative gain matrices, respectively. Equation (31) can be reformulated as follows:

$$\dot{\boldsymbol{v}}^I = \dot{\boldsymbol{v}}_d^I + \boldsymbol{K}_d\left(\boldsymbol{v}_d^I - \boldsymbol{v}^I\right) + \boldsymbol{K}_p\left(\boldsymbol{x}_d^I - \boldsymbol{x}^I\right) \tag{32}$$

The twist of the camera frame with respect to the Inertial frame is $\boldsymbol{v}_c^I = [\dot{\boldsymbol{t}}_c^{I\,T}, \boldsymbol{\omega}_c^{I\,T}]^T$. The geometric Jacobian of the camera frame relates the joint velocities of the robotic camera and $\boldsymbol{v}_c^I$:

$$\boldsymbol{v}_c^I = \boldsymbol{J}_{rc}(\boldsymbol{q}_c)\dot{\boldsymbol{q}}_c + \boldsymbol{J}_b\,\boldsymbol{v}_b^I \tag{33}$$

where $\boldsymbol{J}_{rc} \in \Re^{6\times nc}$ is the robotic camera Jacobian matrix, and $\boldsymbol{J}_b \in \Re^{6\times 6}$ is the Jacobian matrix for the base satellite. The same relationship can be obtained for the manipulator. In this case, $\boldsymbol{v}_e^I = [\dot{\boldsymbol{t}}_e^{I\,T}, \boldsymbol{\omega}_e^{I\,T}]^T$ is the twist of the manipulator end-effector frame with respect to the Inertial frame and:

$$\boldsymbol{v}_e^I = \boldsymbol{J}_{re}(\boldsymbol{q}_e)\dot{\boldsymbol{q}}_e + \boldsymbol{J}_b\,\boldsymbol{v}_b^I \tag{34}$$

where $\boldsymbol{J}_{re} \in \Re^{6\times ne}$ is the robotic manipulator Jacobian matrix. The linear and angular momenta of the system $\left(\boldsymbol{\mathcal{P}}^T, \boldsymbol{\Psi}^T\right)^T \in \Re^6$ read as:

$$\begin{bmatrix}\boldsymbol{\mathcal{P}}\\ \boldsymbol{\Psi}\end{bmatrix} = \boldsymbol{M}_{bb}\boldsymbol{v}_b^I + \boldsymbol{M}_{be}\dot{\boldsymbol{q}}_e + \boldsymbol{M}_{bc}\dot{\boldsymbol{q}}_c \tag{35}$$

By combining Eq. (34) and Eq. (35), we obtain:

$$\boldsymbol{v}_e^I = \left(\boldsymbol{J}_{re}(\boldsymbol{q}_e) - \boldsymbol{J}_b\boldsymbol{M}_{bb}^{-1}\boldsymbol{M}_{be}\right)\dot{\boldsymbol{q}}_e - \boldsymbol{J}_b\boldsymbol{M}_{bb}^{-1}\boldsymbol{M}_{bc}\dot{\boldsymbol{q}}_c + \boldsymbol{J}_b\,\boldsymbol{M}_{bb}^{-1}\begin{bmatrix}\boldsymbol{\mathcal{P}}\\ \boldsymbol{\Psi}\end{bmatrix} \tag{36}$$

$$\boldsymbol{v}_e^I = \boldsymbol{J}_{ge}\dot{\boldsymbol{q}}_e - \boldsymbol{J}_b\boldsymbol{M}_{bb}^{-1}\boldsymbol{M}_{bc}\dot{\boldsymbol{q}}_c + \boldsymbol{v}_{ge} \tag{37}$$

where $\boldsymbol{J}_{ge}$ is the Generalized Jacobian Matrix for the manipulator, and $\boldsymbol{v}_{ge}$ is an offset velocity due to the non-zero momentum. A similar relationship can be obtained for the robotic camera by using Eqs. (33) and (35):

$$\boldsymbol{v}_c^I = \left(\boldsymbol{J}_{rc}(\boldsymbol{q}_c) - \boldsymbol{J}_b\boldsymbol{M}_{bb}^{-1}\boldsymbol{M}_{bc}\right)\dot{\boldsymbol{q}}_c - \boldsymbol{J}_b\boldsymbol{M}_{bb}^{-1}\boldsymbol{M}_{be}\dot{\boldsymbol{q}}_e + \boldsymbol{J}_b\,\boldsymbol{M}_{bb}^{-1}\begin{bmatrix}\boldsymbol{\mathcal{P}}\\ \boldsymbol{\Psi}\end{bmatrix} \tag{38}$$

$$\boldsymbol{v}_c^I = \boldsymbol{J}_{gc}\dot{\boldsymbol{q}}_c - \boldsymbol{J}_b\boldsymbol{M}_{bb}^{-1}\boldsymbol{M}_{be}\dot{\boldsymbol{q}}_e + \boldsymbol{v}_{ge} \tag{39}$$

where $\boldsymbol{J}_{gc}$ is the Generalized Jacobian Matrix for the robotic camera. Thus, the relationship between $\boldsymbol{v}^I$ and the time derivative of $\boldsymbol{\varepsilon}$ reads as:

$$\boldsymbol{v}^I = \begin{bmatrix}\boldsymbol{v}_e^I\\ \boldsymbol{v}_c^I\end{bmatrix} = \begin{bmatrix}\boldsymbol{0}_{3x3} & \boldsymbol{0}_{3x3} & \boldsymbol{J}_{ge} & -\boldsymbol{J}_b\boldsymbol{M}_{bb}^{-1}\boldsymbol{M}_{bc}\\ \boldsymbol{0}_{3x3} & \boldsymbol{0}_{3x3} & -\boldsymbol{J}_b\boldsymbol{M}_{bb}^{-1}\boldsymbol{M}_{be} & \boldsymbol{J}_{gc}\end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{t}}_b^I\\ \boldsymbol{\omega}_b^I\\ \dot{\boldsymbol{q}}_e\\ \dot{\boldsymbol{q}}_c\end{bmatrix} + \begin{bmatrix}\boldsymbol{v}_{ge}\\ \boldsymbol{v}_{ge}\end{bmatrix} = \boldsymbol{J}(\boldsymbol{\varepsilon})\dot{\boldsymbol{\varepsilon}} + \boldsymbol{v}_{ge2} \tag{40}$$

and, its time derivative is:

$$\dot{\boldsymbol{v}}^I = \boldsymbol{J}(\boldsymbol{\varepsilon})\ddot{\boldsymbol{\varepsilon}} + \dot{\boldsymbol{J}}(\boldsymbol{\varepsilon})\dot{\boldsymbol{\varepsilon}} + \dot{\boldsymbol{v}}_{ge2} \tag{41}$$

Therefore, by substituting Eq. (41) into Eq. (31), the following relation can be obtained:

$$\dot{\boldsymbol{v}}_d^I + \boldsymbol{K}_d\left(\boldsymbol{v}_d^I - \boldsymbol{v}^I\right) + \boldsymbol{K}_p\left(\boldsymbol{x}_d^I - \boldsymbol{x}^I\right) = \boldsymbol{J}(\boldsymbol{\varepsilon})\ddot{\boldsymbol{\varepsilon}} + \dot{\boldsymbol{J}}(\boldsymbol{\varepsilon})\dot{\boldsymbol{\varepsilon}} + \dot{\boldsymbol{v}}_{ge2} \tag{42}$$

which shows that the task constraint in Eq. (32) that can be represented in the form indicated in Eq. (27), with:

$$\begin{aligned}\boldsymbol{A}(\boldsymbol{\varepsilon},\ \dot{\boldsymbol{\varepsilon}},\ t) &= \boldsymbol{J}(\boldsymbol{\varepsilon})\\ \boldsymbol{b}(\boldsymbol{\varepsilon},\ \dot{\boldsymbol{\varepsilon}},\ t) &= \dot{\boldsymbol{v}}_d^I + \boldsymbol{K}_d\left(\boldsymbol{v}_d^I - \boldsymbol{v}^I\right) + \boldsymbol{K}_p\left(\boldsymbol{x}_d^I - \boldsymbol{x}^I\right) - \dot{\boldsymbol{J}}(\boldsymbol{\varepsilon})\dot{\boldsymbol{\varepsilon}} - \dot{\boldsymbol{v}}_{ge2}\end{aligned} \tag{43}$$

The asymptotic stability of the system, under the application of Eq. (30), can be demonstrated by analyzing the closed-loop system. Specifically, by using Eqs. (29), (30) and (43), the following closed-loop system can be obtained:

$$\begin{aligned}&\widetilde{\boldsymbol{M}}\ddot{\boldsymbol{\varepsilon}} + \widetilde{\boldsymbol{C}} + \widetilde{\boldsymbol{\tau}}_{grav}\\ &= \boldsymbol{W}^{-1/2}\left(\boldsymbol{J}\widetilde{\boldsymbol{M}}^{-1}\boldsymbol{W}^{-1/2}\right)^{+}\cdot\left(\dot{\boldsymbol{v}}_d^I + \boldsymbol{K}_d\left(\boldsymbol{v}_d^I - \boldsymbol{v}^I\right) + \boldsymbol{K}_p\left(\boldsymbol{x}_d^I - \boldsymbol{x}^I\right) - \dot{\boldsymbol{J}}\dot{\boldsymbol{\varepsilon}} - \dot{\boldsymbol{v}}_{ge2} - \boldsymbol{A}\widetilde{\boldsymbol{M}}^{-1}\widetilde{\boldsymbol{C}}\right) + \widetilde{\boldsymbol{\tau}}_{grav}\end{aligned} \tag{44}$$

By multiplying both sides of Eq. (44) by $\left(\boldsymbol{J}\widetilde{\boldsymbol{M}}^{-1}\boldsymbol{W}^{-1/2}\right)\boldsymbol{W}^{1/2}$, the

system becomes

$$\boldsymbol{J}\ddot{\boldsymbol{\varepsilon}} + \dot{\boldsymbol{J}}\dot{\boldsymbol{\varepsilon}} = \dot{\boldsymbol{v}}_d^I + \boldsymbol{K}_d\left(\boldsymbol{v}_d^I - \boldsymbol{v}^I\right) + \boldsymbol{K}_p\left(\boldsymbol{x}_d^I - \boldsymbol{x}^I\right) - \dot{\boldsymbol{v}}_{ge2} \tag{45}$$

and, if Eq. (41) is substituted, the system follows the task indeed in Eq. (32):

$$\dot{\boldsymbol{v}}^I = \dot{\boldsymbol{v}}_d^I + \boldsymbol{K}_d\left(\boldsymbol{v}_d^I - \boldsymbol{v}^I\right) + \boldsymbol{K}_p\left(\boldsymbol{x}_d^I - \boldsymbol{x}^I\right) \tag{46}$$

It is also worth noting that the second derivative of the error function is equal to:

$$\ddot{\boldsymbol{e}} = \dot{\boldsymbol{v}}_d - \boldsymbol{K}_d\dot{\boldsymbol{e}} - \boldsymbol{K}_p\boldsymbol{e} \tag{47}$$

where $\dot{\boldsymbol{e}} = \boldsymbol{v}^I - \boldsymbol{v}_d^I$ and, therefore, asymptotic tracking is achieved.

6. Simulation results

This section describes simulations results obtained during the tracking of different trajectories using the proposed robotic system. The geometric and mass properties of the manipulator, of the robotic camera and the base satellite are listed in Table 1. The simulations consider only the phase of close approach when the target satellite falls within the robotic manipulator workspace and camera field of view. The reference orbit used to compute the gravity gradient perturbing forces and torques is supposed to be circular and with a radius of $r_{orb} = 6878\,km$. The gravity gradient is considered as the main perturbing action. The study assumes that both chaser and target spacecraft have similar Area-to-Mass ratio, and therefore differential drag effects are not considered in the simulation.

Two different scenarios are considered. In the first scenario, the manipulator is supposed to track a specific trajectory of the visual features without touching the target satellite. The second scenario analyzes the case of physical interaction between the two spacecraft, with the manipulator performing an insertion task of a tool into the body of the target satellite. For both the simulations, a predefined trajectory is followed by the robotic camera. This trajectory is pre-planned offline, so the visual features remain within the field of view during the task. The control of the robot camera trajectory is out of the scope of this study, but approaches like the ones shown in Refs. [48,49] can guarantee that the areas of interest for performing the operations can be correctly observed by the camera, with the scene always within the field of view of the camera. Although these approaches have been proved to be robust for generating trajectories to guarantee visibility, if the visual features are outside the camera field of view, the visual servoing task cannot be accomplished because the controller does not have the required information to generate the control action.

The proportional and derivative matrices used in these simulations are $\boldsymbol{K}_p = diag(0.1)$ and $\boldsymbol{K}_d = diag(0.5)$, where $diag() \in \Re^{6\times 6}$ is a matrix with the diagonal elements equal to the argument of the function. Additionally, for the visual and interaction guidance, $\boldsymbol{Q} = diag(0.01) \in \Re^{2m\times 2m}$, $\boldsymbol{K} = diag(0.1,0.1,0.4,10,10,15)$, $\boldsymbol{D} = diag(100,100,400,10,10,20)$. The gains in the aforementioned matrices have been selected by using a trial-and-error approach, but always complying with the hypothesis of the adopted visual servoing controllers. A camera that acquires 20 images per second with a resolution of 640 × 480 pixels is used. The camera is supposed to be previously calibrated: the camera intrinsic parameters are $(u_0, v_0) = (298, 225)$ px, and $(f_u, f_v) = (1082.3, 1073.7)$ px, where u_0 and v_0 are the positions of the optical centre and f_u and f_v are the focal lengths in the x and y directions, respectively.

One of the main limitations in the application and implementation of visual servoing systems is the relatively low sampling rate. This aspect has been clearly improved in the last years. However, the utilization of time-consuming image processing algorithms habitually can cause time delays in the control system. In this case, the delays can cause the required sampling time of the motion controller is not achieved. The most common approach to get the information at the required rate is to upsample the visual feedbacks by repeating the latest data between the slow sampling instances. This approach has proved to be adequate in applications like the one presented in this paper. However, if the target has a fast-changing speed the camera can lose the visual features in the field of view. In this case, three different approaches can be applied for solving the time-delay problem:

- Reduction of the time delay by using embedded image processing, by simplifying the image processing algorithms of even by grabbing only sub-regions of interest.
- Estimation of image features or the target position to compensate the time-delay [50,51]. To do this, Kalman filter [45] or estimation of image Jacobians are used to compensate for the delays during the visual feedback.
- Use of cloud-base networked visual servo control [52]. In this case, to speed up the image processing, a distributed computation is proposed to do image processing in parallel.

Fig. 1 represents a global view of the typical rendezvous scenario, where both satellites are represented as boxes for sake of simplicity. The size of box representing the service spacecraft is of $2 \times 2 \times 2\,m^3$ and centered in the base frame, B. The size for the target spacecraft is of $3 \times 5 \times 2\,m^3$ centered in the target frame, T. Fig. 3 provides additional information about the initial conditions of the test case maneuvers. In this figure, the target and servicing spacecraft have been removed to show more clearly the manipulator size and the target features extracted from the target spacecraft. The four points captured as visual features by the camera are in the front surface of the target satellite, and they are visible during the entire task. In the first test case maneuver, in Section 6.1, the position of the target spacecraft, T, is $(0,\ 10,\ 0)\,m$ with respect to the servicing spacecraft coordinate frame, B. In the second simulation, in Section 6.2, the displacement between the two bodies is $(0,\ 5,\ 0)\,m$.

Table 1
Mass and inertia parameters of the base and robotic arms.

Base	Mass (kg)	Inertia (kg•m^2) I_x	I_y	I_z
	2550	6200	3540	7090
Arms	**Mass (kg)**	**Inertia (kg•m^2) I_x**	**I_y**	**I_z**
Link 1	35	2	0.2	2
Link 2	22	3	0.2	3
Link 3	22	3	0.2	3
Link 4	10	0.15	0.2	0.4
Link 5	10	0.15	0.2	0.3
Link 6	10	0.2	0.25	0.3

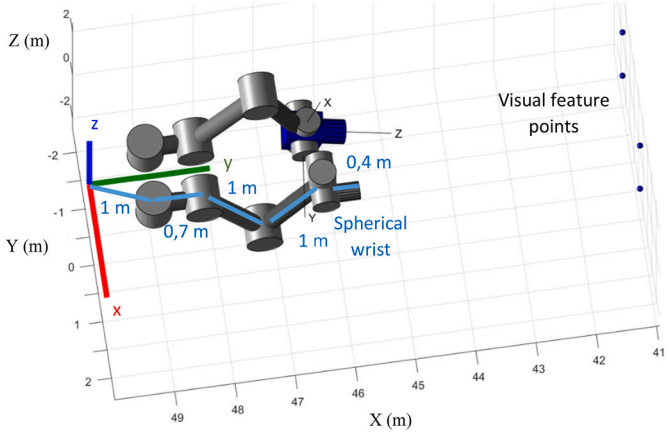

Fig. 3. Detail of the robot kinematic properties and extracted visual features.

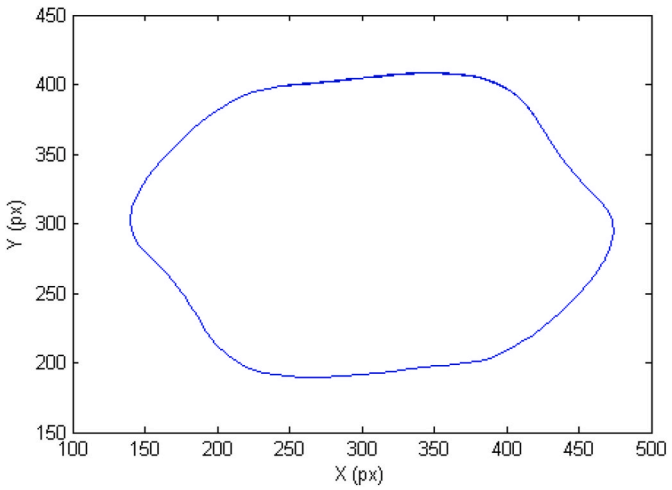

Fig. 4. Desired image trajectory for one of the features considered in the tracking experiment (without interaction control).

6.1. Tracking trajectories without interaction control

The aim of the first simulation is to evaluate the performance of the visual servoing controller on tracking a repetitive trajectory in the 3D space when the manipulator does not touch the target satellite. Specifically, four image feature points, displaced in a square configuration in one of the sides of the target satellite as in Fig. 1, are captured by the robotic camera. A similar configuration is then virtually generated, assuming that the four virtual image features are rigidly connected to the end-effector of the robotic manipulator and laying on a plane parallel to the camera image plane. The pattern to be followed by each of the virtual features is shown in Fig. 4. As previously indicated, the image error to be annulled is defined as the difference between the end-effector image features, s_e, and the target features, s_t. In on-orbit servicing applications these last features can be obtained from the target satellite while, s_e are the features extracted from the robot-end by the robotic camera using the steps described in Eqs. (12)–(16). In this simulation, we consider the target features following the trajectory indicated in Fig. 4. For the sake of clarity, this figure represents the desired repetitive image trajectory of only one of the features that need to be tracked during the maneuver. These trajectories are defined by:

$$s_t = \begin{bmatrix} s_{tix} + 160cos(\omega t + \pi/4) + 8sin(\omega 5t) \\ s_{tiy} + 110sin(\omega t + \pi/4) + 6sin(\omega 5t) \end{bmatrix}^T \quad (48)$$

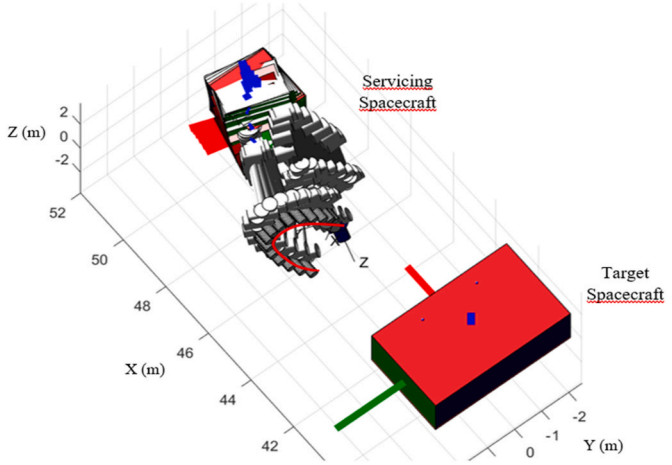

Fig. 5. 3D trajectory during the tracking.

where ω is the rate factor. Please note that in Eq. (48) s_{tix} and s_{tiy} are the x and y components of the *i-th* (with *i=1 … 4*) visual feature of $s_{ti} = \begin{bmatrix} s_{tix} \\ s_{tiy} \end{bmatrix}^T = \begin{bmatrix} 446 & 464 & 578 & 560 \\ 560 & 446 & 464 & 578 \end{bmatrix}^T$. Thus, Eq. (48) can be applied to all the four given visual features in the example.

Fig. 5 shows the resulting 3D trajectory drawn by the end-effector of the robotic manipulator (red line) during the maneuver when $\omega = 0.5\ rad/s$ and the weight matrix of the controller is $\boldsymbol{W} = \widetilde{\boldsymbol{M}}^{-2}$. It is also worth noting that the tracking operation is accomplished although the base satellite is floating, and therefore moving under the actions due to the robotic arm movement, as illustrated by the overlapping frames shown in Fig. 5. This also changes the actual scene observed by the robotic camera, but this does not compromise the accomplishment of the tracking task. Indeed, Fig. 6 shows that the module error, $|\boldsymbol{e}|$ converges to values close to 0, when the tracking rate ω is equal to 0.5 rad/s, 1 rad/s and 2 rad/s, respectively. The mean error during the tracking is equal to $8.4{\cdot}10^{-4}\ m$ when the tracking rate is $\omega = 0.5\ rad/s$. It is worth noting that, the mean error remains still limited to average values of $1.1{\cdot}10^{-3}\ m$ and $1.7{\cdot}10^{-3}\ m$ for tracking rates of $\omega = 1\ rad/s$ and $\omega = 2\ rad/s$, respectively. Therefore, the tracking error remains low considering the free-floating conditions when different tracking rates are considered, and, consequently, the robotic manipulator end-effector is able to track the desired 3D trajectory.

The proposed Cartesian controller has been defined depending on the weighting matrix $\boldsymbol{W}$. The choice of $\boldsymbol{W}$ plays an important role in the controller because it determines how the torques are distributed over the joints. Table 2 shows the controller's performance with different tracking velocities when different values for the weighting matrix are used. It is worth noting that the tracking error is lower for the controllers based on either $W = \widetilde{\boldsymbol{M}}^{-2}$ *or* $W = \widetilde{\boldsymbol{M}}^{-1}$. In order to compare the tracking behavior when the optimal controller approach is not used, Table 2 also

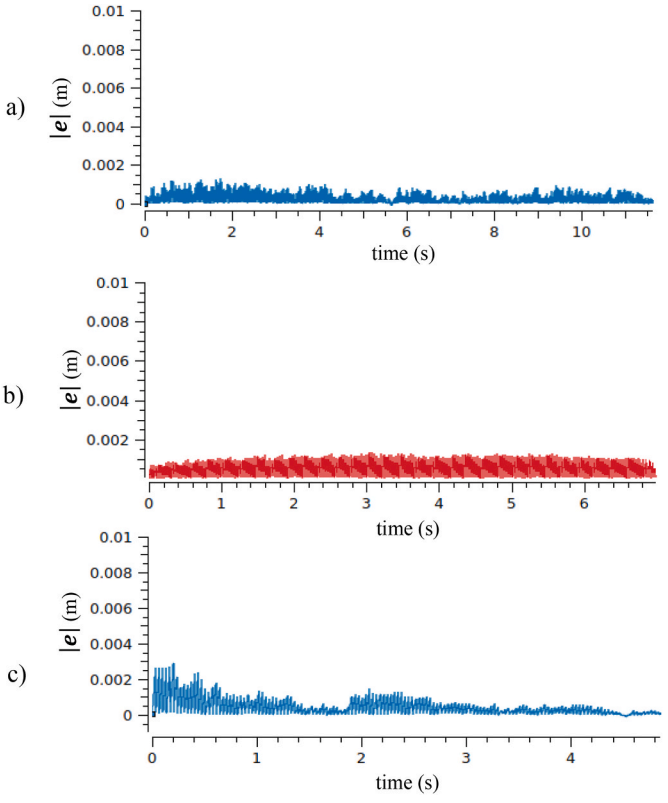

Fig. 6. Error module obtained when: a) $\omega = 0.5\ rad/s$, b) $\omega = 1\ rad/s$, c) $\omega = 2\ rad/s$.

Table 2
Mean error (m) obtained for different tracking velocities for different values of $\boldsymbol{W}$.

$\boldsymbol{W}$	$\omega = 0.5\ rad/s$	$\omega = 1\ rad/s$	$\omega = 2\ rad/s$
$\widetilde{\boldsymbol{M}}^{-2}$	8.4E-4	1.1E-3	1.7E-3
$\widetilde{\boldsymbol{M}}^{-1}$	7.3E-4	9.8E-4	2.1E-3
$\boldsymbol{I}$	1.2E-3	1.9E-3	3.3E-3
$\boldsymbol{RAC}$	9.7E-4	1.5E-3	2.8E-3

shows the controller's performance when a resolved acceleration-based controller (RAC) [41] is used. The equation of this controller for the system kinematics and dynamics considered is indicated in Eq. (49). Although the mean error remains low during the tracking, the optimal approach improves the tracking with respect the RAC based controller.

$$\widetilde{\boldsymbol{\tau}} = \widetilde{\boldsymbol{M}}\boldsymbol{J}^{+}\left(\dot{\boldsymbol{v}}_d^I + \boldsymbol{K}_d\left(\boldsymbol{v}_d^I - \boldsymbol{v}^I\right) + \boldsymbol{K}_p\left(\boldsymbol{x}_d^I - \boldsymbol{x}^I\right) - \dot{\boldsymbol{J}}\dot{\boldsymbol{\varepsilon}} - \dot{\boldsymbol{v}}_{ge2}\right) + \widetilde{\boldsymbol{C}} \tag{49}$$

6.2. Tracking trajectories with interaction control

In this second experiment, the end of the robot manipulator presents a force sensor and a tool. As in the previous case, the eye-in-hand camera extracts four visual features from the target satellite. In this experiment, the parameters of a Gigabit Ethernet TM6740GEV camera also employed in Ref. [23] is considered. This camera allows a resolution of 1280 × 1024 pixels. These point features are obtained from the pattern represented in Fig. 1. The virtual image features of the manipulator end-effector are generated in a way to be coincident with the extracted visual features when the tool at the manipulator end is inserted 2 cm into the target spacecraft. To simulate real conditions and to evaluate the effect of the force sensor and the proposed impedance control scheme, an error is introduced in the target features. More specifically, a deviation of 3 mm is included these target features with respect the ideal ones. Indeed, having shifted the virtual features means that the image error cannot be completely deleted using the information provided by the camera images, but it is necessary to use the interaction control terms in Eq. (19) to guarantee the correct insertion of the tool. It is worth noting that tool insertion task necessarily requires an exchange of momentum between servicer and target, which could generate unpredictable changes of relative position and attitude of the target and compromise the success of the maneuver, if the target is not controlled. To perform the simulations, the model assumes that the target bigger mass of the servicing spacecraft so that the effects of momentum exchange are affecting only the relative dynamics of the servicer and therefore can be compensated by the proposed controller. An equivalent simulation and results could be obtained under the assumption of having a cooperative target that is able to compensate with its actuators the eventual effects of the contact reactions induced by the tool.

Figures from 7 to 10 show the obtained results. The scheme represented in Fig. 7 shows the motion of the manipulator arm (trajectory represent in red), and robotic camera (trajectory represented in blue) during the experiment. The different frames in that figure show that the manipulator tries to reach the target insertion position while the floating base satellite moves due to the motion of both manipulator and robotic camera. A detail of the trajectory described by the manipulator end-effector with respect to the chaser satellite is represented in Fig. 8. The grey circles represent the positions of the hole on the target satellite where the tool needs to be inserted. On the other hand, the colored circles represent the positions of the end effector of the robotic manipulator during the maneuver. It is worth noting that the colored circles converge into the grey ones, meaning that the tool insertion task is performed correctly by the manipulator.

Fig. 9 represents in blue the image features extracted from the target spacecraft. These features are indicated in the figure as the "real features". This figure also shows in red the features extracted from the robot tool. These last features are denoted in the figure as "virtual features" because they are obtained by the procedure described in Equations (12) – (16). As an ideal situation, the controller tries to nullify the image error between both features. However, as previously indicated, an additional error of 3 mm is included that has to be compensated by the force control term. As it can be seen in Fig. 9 the value of the final features is close but is not completely annulled (the final position of the features are indicated by solid circles). However, this error is in any case compensated by the controller, that uses the interactions terms in Eq. (19). The interaction forces at the end effector are represented in Fig. 10. The x, y and z components are represented in green, red and blue, respectively. In this last figure we can clearly appreciate the point where the manipulator collides with the target satellite (after 7 s). The impedance based proposed approach compensates this first collision and the forces are reduced. At this point the insertion is performed and the hole located at the manipulator is inserted in the target spacecraft.

7. Conclusions

A visual-force control approach suitable for on-orbit servicing and manipulation was presented in this paper. The algorithm assumed a scenario with a servicing spacecraft with two-arm manipulators: one of the arms performs the manipulation task, and the second arm (robotic camera) holds a camera to observe the target zone of manipulation. Such configuration was selected as it allows for greater versatility of the

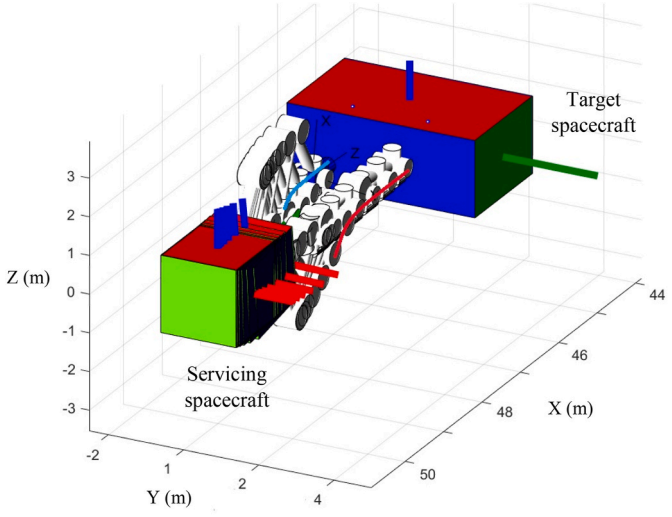

Fig. 7. 3D trajectory during the insertion task.

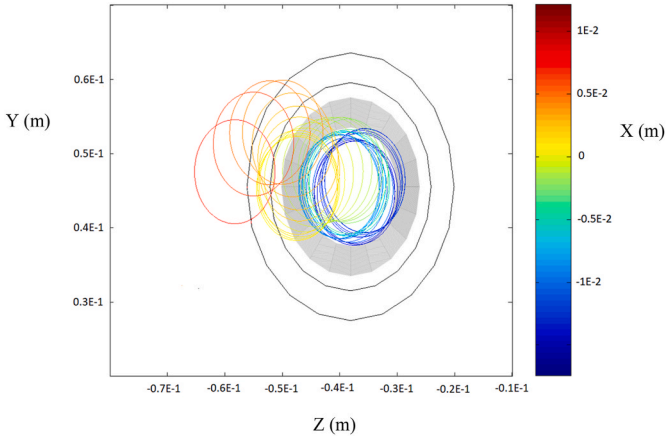

Fig. 8. 2D representation of the insertion task.

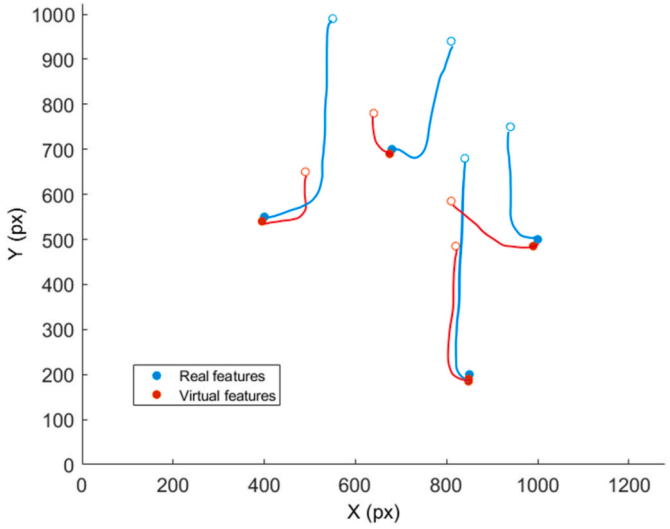

Fig. 9. Image trajectory of the target visual features during the insertion task.

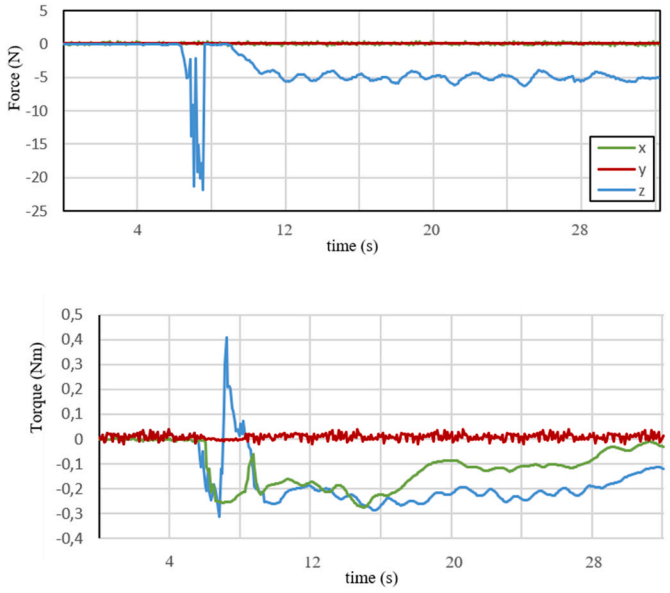

Fig. 10. Interaction forces and torques during the task.

operations. A visual-impedance based approach is proposed for the guidance of the manipulator arm when it touches or interacts with the target satellite. A damper-spring model is considered to characterize this interaction.

Additionally, the robotic camera can eventually move to avoid obstructing objects that do not allow for a full view of the scene. The visual control objective is to reduce an error defined in the image space progressively. To compute this error, a set of visual features are extracted from the target satellite and the manipulator end. The concept of virtual features is presented to define the features at the manipulator end virtually reconstructed from the robot kinematics. Then a visual servoing approach is present to guide the manipulator arm by reducing the error between the features extracted from the target satellite and the virtual ones.

Two different scenarios are considered in the simulation results. In the first one, the manipulator is supposed to track a specific trajectory of the visual features without touching the target satellite. The visual servoing approach proposed for the guidance of the manipulator arm is validated by using this first experiment. The second scenario analyzes the case of physical interaction between the two spacecraft, with the manipulator performing an insertion task of a tool into the body of the target satellite. Further future studies will integrate estimators based on Kalman filter to improve the visual features reconstruction and will assess the robustness of the improved controllers against differential drag effects.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Flores-Abad, O. Ma, K. Pham, S. Ulrich, A review of space robotics technologies for on-orbit servicing, Prog. Aero. Sci. 68 (2014) 1–26, https://doi.org/10.1016/j.paerosci.2014.03.002.

[2] M. Wang, J. Luo, J. Yuan, W. Ulrich, Detumbling strategy and coordination control of kinematically redundant space robot after capturing a tumbling target, Nonlinear Dynam. 92 (2018) 1023–1043, https://doi.org/10.1007/s11071-018-4106-4.

[3] C. Blackerby, A. Okamoto, S. Iizuka, Y. Kobayashi, K. Fujimoto, Y. Seto, S. Fujita, T. Iwai, N. Okada, J. Forshaw, A. Bradford, The ELSA-d end-of-life debris removal mission: preparing for launch, in: Proceedings of the 70th International Astronautical Congress (IAC), Washington DC, US, 21-25 October 2019. IAC-19, A6,5,2.

[4] R. Biesbroek, S. Aziz, A. Wolahan, S. Cipolla, M. Richard-Noca, L. Piguet, The clearspace-1 mission: ESA and Clearspace team up to remove debris, in: Proceedings of the 8th European Conference on Space Debris, 2021, 20–23 April Darmstadt, Germany.

[5] N.T. Redd, Bringing satellites back from the dead: mission extension vehicles give defunct spacecraft a new lease on life, IEEE Spectrum 57 (8) (2020) 6–7, https://doi.org/10.1109/MSPEC.2020.9150540.

[6] B.R. Sullivan, J.C. Parrish, G. Roesler, Upgrading in-service spacecraft with on-orbit attachable capabilities, in: 2018 AIAA SPACE and Astronautics Forum and Exposition, AIAA Paper 2018-5223, 2018, pp. 1–17, https://doi.org/10.2514/6.2018-5223.

[7] M.A. Shoemaker, M. Vavrina, D.E. Gaylor, R. Mcintosh, M. Volle, J. Jacobsohn, OSAM-1 decommissioning orbit design, in: AAS/AIAA Astrodynamics Specialist Conference, 2020. AAS 20-460 South Lake Tahoe, CA.

[8] B.B. Reed, R.C. Smith, B.J. Naasz, J.F. Pellegrino, C.E. Bacon, The restore-L servicing mission, AIAA 2016-5478, AIAA SPACE (2016). September 2016.

[9] L. Felicetti, P. Gasbarri, A. Pisculli, M. Sabatini, G.B. Palmerini, Design of robotic manipulators for orbit removal of spent launchers' stages, Acta Astronaut. 119 (2016) 118–130, https://doi.org/10.1016/j.actaastro.2015.11.012.

[10] R. Xu, J. Luo, M. Wang, Kinematic and dynamic manipulability analysis for free-floating space robots with closed chain constraints, Robot. Autonom. Syst. 130 (2020), https://doi.org/10.1016/j.robot.2020.103548.

[11] B.M. Moghaddam, R. Chhabra, On the guidance, navigation and control of in-orbit space robotic missions: a survey and prospective vision, Acta Astronaut. 184 (2021) 70–100, https://doi.org/10.1016/j.actaastro.2021.03.029.

[12] J. García, D. Gonzalez, A. Rodríguez, B. Santamaria, J. Estremera, M. Armendia, Application of impedance control in robotic manipulators for spacecraft on-orbit servicing, in: 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 2019, pp. 836–842, https://doi.org/10.1109/ETFA.2019.8869069.

[13] L.P. Cassinis, R. Fonod, Eberhard Gill, Review of the robustness and applicability of monocular pose estimation systems for relative navigation with an uncooperative spacecraft, Prog. Aero. Sci. 110 (2019), https://doi.org/10.1016/j.paerosci.2019.05.008.

[14] V. Pesce, R. Opromolla, S. Sarno, M. Lavagna, M. Grassi, Autonomous relative navigation around uncooperative spacecraft based on a single camera, Aero. Sci. Technol. 84 (2019) 1070–1080, https://doi.org/10.1016/j.ast.2018.11.042.

[15] J. Peng, W. Xu, B. Liang, A. Wu, Virtual stereovision pose measurement of noncooperative space targets for a dual-arm space robot, IEEE Trans. Instrum. Meas. 69 (1) (2020) 76–88, https://doi.org/10.1109/TIM.2019.2893010.

[16] A. Nocerino, R. Opromolla, G. Fasano, M. Grassi, LIDAR-based multi-step approach for relative state and inertia parameters determination of an uncooperative target, Acta Astronaut. 181 (2021) 662–678, https://doi.org/10.1016/j.actaastro.2021.02.019.

[17] J.W. Griffin, A.C. Lindberg, T.B. Ahn, P.L. Harton, Shuttle rendezvous radar performance evaluation and simulation, in: IEEE PLANS '88.,Position Location and Navigation Symposium, Record, Navigation into the 21st Century, 1988, pp. 236–245, https://doi.org/10.1109/PLANS.1988.195489.

[18] G.B. Palmerini, Combining thermal and visual imaging in spacecraft proximity operations, in: 13th International Conference on Control Automation Robotics & Vision (ICARCV), 2014, pp. 383–388, https://doi.org/10.1109/ICARCV.2014.7064336.

[19] G.B. Palmerini, Relative navigation in autonomous spacecraft formations, in: IEEE Aerospace Conference, 2016, pp. 1–10, https://doi.org/10.1109/AERO.2016.7500944.

[20] J. Peng, W. Xu, T. Liu, H. Yuan, B. Liang, End-effector pose and arm-shape synchronous planning methods of a hyper-redundant manipulator for spacecraft repairing, Mech. Mach. Theor. (2021) 155, https://doi.org/10.1016/j.mechmachtheory.2020.104062.
[21] H. Wang, D. Guo, H. Xu, W. Chen, T. Liu, K.K. Leang, Eye-in-Hand tracking control of a free-floating space manipulator, IEEE Trans. Aero. Electron. Syst. 53 (4) (2017) 1855–1865, https://doi.org/10.1109/TAES.2017.2674218.
[22] F. Chaumette, S. Hutchinson, Visual servo control. I. Basic approaches, IEEE Robot. Autom. Mag. 13 (4) (2006) 82–90.
[23] J.P. Alepuz, M.R. Emami, J. Pomares, Direct image-based visual servoing of free-floating space manipulators, Aero. Sci. Technol. 55 (2016) 1–9, https://doi.org/10.1016/j.ast.2016.05.012.
[24] A.H. Abdul Hafez, P. Mithun, V.V. Anurag, S.V. Shah, K. Madhava Krishna, Reactionless visual servoing of a multi-arm space robot combined with other manipulation tasks, Robot. Autonom. Syst. 91 (2017) 1–10, https://doi.org/10.1016/j.robot.2016.12.010.
[25] X. Zhao, Z. Xie, H. Yang, J. Liu, Minimum base disturbance control of free-floating space robot during visual servoing pre-capturing process, Robotica 38 (4) (2020) 652–668, https://doi.org/10.1017/S0263574719000924.
[26] G. Dong, Z.H. Zhu, Incremental inverse kinematics based vision servo for autonomous robotic capture of non-cooperative space debris, Adv. Space Res. 57 (7) (2016) 1508–1514, https://doi.org/10.1016/j.asr.2016.01.011.
[27] M. Jin, H. Yang, Z. Xie, K. Sun, H. Liu, The ground-based verification system of visual servoing control for a space robot, in: IEEE International Conference Mechatronics Automation, IEEE, 2013, pp. 1566–1570, https://doi.org/10.1109/ICMA.2013.6618147.
[28] H. Yang, M. Jin, Z. Xie, K. Sun, H. Liu, Ground verification of space robot capturing the free-floating target based on visual servoing control with time delay, Ind. Robot An Int. J. 41 (2014) 543–556, https://doi.org/10.1108/IR-05-2014-0339.
[29] G. Ma, Z. Jiang, H. Li, J. Gao, Z. Yu, X. Chen, Hand-eye servo and impedance control for manipulator arm to capture target satellite safely, Robotica 33 (2015) 848–864, https://doi.org/10.1017/S0263574714000587.
[30] J. Pomares, L. Felicetti, J. Pérez, M.R. Emami, Concurrent image-based visual servoing with adaptive zooming for non-cooperative rendezvous maneuvers, Adv. Space Res. 61 (3) (2018) 862–878, https://doi.org/10.1016/j.asr.2017.10.054.
[31] S. Hutchinson, G.D. Hager, P.I. Corke, A tutorial on visual servo control, IEEE Trans. Robot. Autom. 12 (5) (1996) 651–670, https://doi.org/10.1109/70.538972.
[32] A. Alabdo, J. Pérez, G.J. Garcia, J. Pomares, F. Torres, FPGA-based architecture for direct visual control robotic systems, Mechatronics 39 (2016) 204–216, https://doi.org/10.1016/j.mechatronics.2016.05.008.
[33] P. Gasbarri, M. Sabatini, G.B. Palmerini, Ground tests for vision based determination and control of formation flying spacecraft trajectories, Acta Astronaut. 102 (2014) 378–391, https://doi.org/10.1016/j.actaastro.2013.11.035.
[34] H. Benninghoff, F. Rems, T. Boge, Development and hardware-in-the-loop test of a guidance, navigation and control system for on-orbit servicing, Acta Astronaut. 102 (2014) 67–80, https://doi.org/10.1016/j.actaastro.2014.05.023.
[35] J. Garcia, A. Rodriguez, J. Estremera, B. Santamaria, D. Gonzalez, M. Armendia, Visual Servoing and Impedance Control in Robotic Manipulators for On-Orbit Servicing, 2020, pp. 734–741, https://doi.org/10.1109/ETFA46521.2020.9211989.
[36] Z. Mitros, G. Rekleitis, E. Papadopoulos, Impedance control design for on-orbit docking using an analytical and experimental approach, July 3-6, 2017, in: 25th Mediterranean Conference on Control and Automation (MED), Valletta, Malta, 2017.
[37] J. García, D. Gonzalez, A. Rodríguez, B. Santamaria, J. Estremera, M. Armendia, Application of Impedance Control in Robotic Manipulators for Spacecraft On-Orbit Servicing, in: 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 2019, pp. 836–842, https://doi.org/10.1109/ETFA.2019.8869069.
[38] I. Dalyaev, V. Titov, I. Shardyko, A concept of robotic system with force-controlled manipulators for on-orbit servicing spacecraft, in: Proceedings of the Scientific-Practical Conference "Research and Development, 2016, https://doi.org/10.1007/978-3-319-62870-7_26.
[39] S. Abiko, R. Lampariello, G. Hirzinger, Impedance control for a free-floating robot in the grasping of a tumbling target with parameter uncertainty, Int. Conf. Intell. Robot. Syst. (2006).
[40] J. Pomares, C.A. Jara, J. Pérez, F. Torres, Direct visual servoing framework based on optimal control for redundant joint structures, Int. J. Precis. Eng. Manuf. 16 (2015) 267–274, https://doi.org/10.1007/s12541-015-0035-z.
[41] F. Caccavale, C. Natale, B. Siciliano, V. Luigi, Resolved-acceleration control of robot manipulators: a critical review with experiments, Robotica 16 (1998) 565–573, https://doi.org/10.1017/S0263574798000290.
[42] P.C. Hughes, Spacecraft Attitude Dynamics, Dover, 2004.
[43] A. Pisculli, L. Felicetti, P. Gasbarri, G.B. Palmerini, M. Sabatini, A reaction-null/Jacobian transpose control strategy with gravity gradient compensation for on-orbit space manipulators, Aero. Sci. Technol. 38 (2014) 30–40, https://doi.org/10.1016/j.ast.2014.07.012.
[44] F. Chaumette, S. Hutchinson, Visual servo control. I. Basic approaches, IEEE Robot. Autom. Mag. 13 (4) (2006) 82–90, https://doi.org/10.1109/MRA.2006.250573.
[45] P. Corke, M. Goods, Controller design for high performance visual servoing, Proc. 12th World Congr. IFAC 93 (1993) 395–398. -July.
[46] J. Gangloff, M. de Mathelin, Visual servoing of a 6 dof manipulator for unknown 3-D profile following, IEEE Trans. Robot. Autom. 18 (4) (2002) 511–520, https://doi.org/10.1109/TRA.2002.802201.
[47] D. Tommasino, G. Cipriani, A. Doria, G. Rosati, Effect of end-effector compliance on collisions in robotic teleoperation, Appl. Sci. 10 (24) (2020) 9077, https://doi.org/10.3390/app10249077.
[48] P.D. Triantafyllou, G.A. Rovithakis, Z. Doulgeri, Guaranteeing field of view constraints in visual servoing tasks under uncertain dynamics, IFAC-PapersOnLine 50 (1) (2017) 2229–2234.
[49] T.-T. Wang, X. Han, J. Zhou, H. Chen, Path planning for visual servoing with search algorithm, Adv. Mech. Eng. (2018), https://doi.org/10.1177/1687814017750264.
[50] C. Liu, X. Huang, M. Wang, Target tracking for visual servoing systems based on an adaptive Kalman filter, Int. J. Adv. Rob. Syst. 9 (2012), https://doi.org/10.5772/52035.
[51] B.M. Nguyen, W. Ohnishi, Y. Wang, H. Fujimoto, Y. Hori, K. Ito, M. Odai, H. Ogawa, E. Takano, T. Inoue, et al., Dual rate Kalman filter considering delayed measurement and its application in visual servo, in: IEEE 13th International Workshop on Advanced Motion Control, 2014, pp. 494–499, https://doi.org/10.1109/AMC.2014.6823331.
[52] H. Wu, L. Lou, C.-C. Chen, S. Hirche, K. Kuhnlenz, Cloud-based networked visual servo control, IEEE Trans. Ind. Electron. 60 (2) (2013) 554–566, https://doi.org/10.1109/TIE.2012.2186775.