Tracking Control for the Grasping of a Tumbling Satellite with a Free-Floating Robot

Roberto Lampariello, Hrishik Mishra, Nassir Oumer, Phillip Schmidt, Marco De Stefano, Alin Albu-Schäffer

Abstract— In this paper a novel method is presented for grasping a partially cooperative tumbling satellite with a free-floating robot. A reference trajectory, provided by an offline motion planner and adapted online to account for modelling uncertainties, is tracked with a visual servo for the approach phase and with a joint-space controller for the rigidization phase which follows the grasp. An Extended Kalman filter ensures a Lipschitz continuity of the noisy pose estimates of a visual tracker, which makes the control method robust for executing the grasping task. The visual servo is in cascade with an impedance controller to achieve tracking while being compliant. The method is successfully validated on the OOS-SIM facility, which is used for simulating gravity-free post-grasping multibody dynamics with a novel momentum-based control method.

I. INTRODUCTION

Among the possible applications of robotic systems in an orbital scenario, this paper focusses on those which may operate in On-Orbit Servicing and in Active Debris Removal missions [1][2][3][4]. To tackle the related control challenges we present here an autonomy-based approach, in which the robot is commanded through a precomputed reference trajectory, corrected by a tracking controller, to account for intrinsic planning and execution errors. We focus on the task of grasping a defective, tumbling target satellite (from here on, the Target) by means of a free-floating robot, consisting of a non-actuated chaser satellite (from here on, the Chaser) carrying a kinematically redundant robot manipulator. These are shown in Fig. 1.

The task of grasping a target satellite with a space robot was already performed in [1] and [2], however only for the case of a cooperative Target. When the Target cannot be attitude controlled and does not present dedicated visual or structural features to aid its grasping, then it is generally termed non-cooperative. We consider here the case in which the Target presents a circular rail where it can be grasped, and assume that sufficient information is provided to allow generating a simplified geometric model of it for visual tracking purposes. Also in this case, the accomplishment of the robotic grasping task in autonomous mode presents some substantial challenges. The robot motion is dictated by that of the Target, which is generally unknown. The robot motion also has to be correctly synchronized with that of
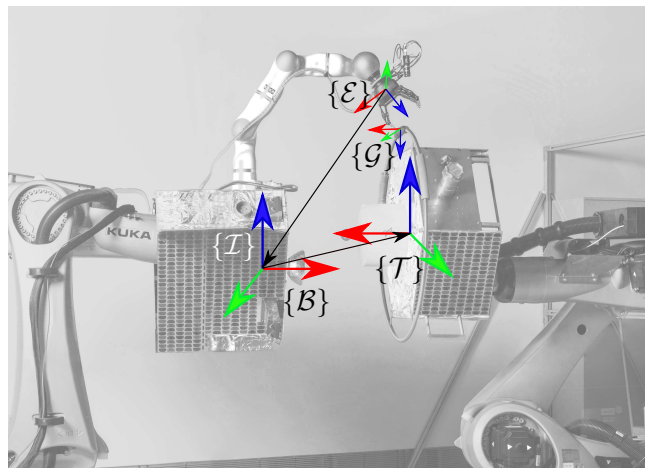
Fig. 1. OOS-SIM experimental facility used to reproduce the gravity-free dynamics of the Chaser (left), carrying a Light-Weight Robot, and the Target (right). A camera is mounted on the gripper of the kinematically redundant manipulator, to allow visual servoing. The reference frames of the Chaser $\mathcal{B}$, the robot end-effector ϵ, the Target $\mathcal{T}$ and the predefined grasping point $\mathcal{G}$ are also shown.

the Target, in order to meet and grasp some preselected grasping point on it, while satisfying motion constraints, such as workspace limits, kinematic and dynamic singularities, collision avoidance, as well as sensor-driven constraints, such as camera field-of-view boundaries and pixel velocity limits. Furthermore, due to the given free-floating dynamics in play, the effect of an impact during the contact phase can be very detrimental for accomplishing the task, since it can quickly bring the Target out of the range of the robot.

The grasping task of interest has been addressed extensively in the literature, first in the context of feedback control and then also in that of optimal control (see Section I-A). While in the former the aim is to solve a regulation control problem, often taking advantage of the free-floating dynamics and redundancy of the Chaser, in the latter an open-loop approach is preferred, based on the idea of computing a feasible and optimal trajectory in real-time. Due to the highly constrained task, explained above, the use of a feasible trajectory is recognized to be of great importance.

A gap in the methodology described above, which we want to close here, is that of having both the optimal control and the feedback control elements working together.

A feedback controller alone does not provide any guarantee of convergence, given the presence of the constraints, which give rise to local minima. Furthermore, the grasping task has a limited time window for its execution and local control methods generally do not ensure finite time convergence. At the same time, an open-loop approach is not robust to modelling errors (e.g., dynamic model) and to contingencies, such as an impact.

We therefore present here a tracking controller for executing the grasping task of interest which: firstly, takes as input a reference trajectory from a database, generated offline with a motion planner in simulation and feasible with respect to all relevant constraints; secondly, superimposes sensor-based corrections to the reference input, to account for the discrepancy between the simulated and the real world.

To this end, the overall tracking control task is composed of the following two sequential phases. The first phase of the task, the approach phase, is executed by a novel interconnected system which is a cascade of visual servo and an impedance controller. The visual servo appears as an outer-loop velocity command which brings the robot end-effector onto the grasping point on the tumbling Target. It is widely recognized that the robustness of a visual servo is significantly improved against modelling errors when performing tracking rather than regulation [5]. The use of impedance control is intended to minimize the detrimental effect of an impact between the robot manipulator and the Target. In the rigidization phase which follows the grasping task, the Target is stabilized with respect to the Chaser using the reference trajectory which in turn exploits the manipulator controller's dissipation to assure that the final configuration for the whole system is reached with a decaying velocity. Furthermore, in this phase, the intrinsic modelling uncertainty in the reference trajectory is handled with an on-line adaptation of the reference trajectory. An additional novelty in the controller is the use of robust estimates of pose and velocity which are obtained from an Extended Kalman Filter (EKF) for feedback and feed-forward respectively. The filter is fed with the noisy, outlier-affected, low-frequency pose estimates of a model-based visual tracking algorithm and produces filtered signals for the tracking controller. It is worth pointing out that the success of the aforesaid controller requires a judicious interplay between the different elements involved: the motion planner, the visual servo (including the EKF and the visual tracking) and the free-floating robot impedance controller.

In addition to defining the control laws for the aforementioned controller, experimental validations on DLR's OOS-SIM robotic facility, shown in Fig. 1, are also presented. This facility allows reproducing realistic orbital gravity-free dynamics in three-dimensions through hardware-in-the-loop simulation. From an infrastructure perspective, a novel momentum-based approach was also developed for the facility in order to simulate the post-grasping tumbling motion of the compound system which was necessary to validate the concepts presented in this paper.

The paper is then structured as follows. In the rest of this Section, we present a relevant bibliography and a more detailed problem statement. In Section II we provide details of the applied methods, while in Section IV we present the experimental results and their analysis. Finally, in Section V we provide a discussion and conclude with remarks outlining the scope of future work.

The mathematical notation is such that scalar constants are written in plain text, scalar variables in italics, vectors, vector arrays and matrices are always bold, coordinates of vector quantities are expressed with the left upper superscript. The right upper superscript $i\,j$ defined the direction of a vector from i to j. A homogeneous transformation matrix (pose) is denoted as $\mathbf{H} \equiv \mathbf{H}(R(q), r)$, where $q \in \mathbb{R}^4$ and $r \in \mathbb{R}^3$ are the quaternion orientation and the position respectively. A quaternion is written as $\theta = \begin{bmatrix} \theta_v^T & \theta_0 \end{bmatrix}$ where $\theta_v \in \mathbb{R}^3$ is the vector part. A pose $\mathbf{H}$ is driven by a twist $\dot{\mathbf{X}} = \begin{bmatrix} \dot{r}^T & \omega^{b^T} \end{bmatrix}^T$, where the superfix b denotes a body frame vector. All poses and velocities expressed relative to $\mathcal{I}$ are denoted with one subscript of the corresponding lowercase. For instance, the Target's inertial pose is defined as $\mathbf{H}_t = (R(\theta), r)$ where θ, r are indicated relative to the $\mathcal{I}$, and $\dot{\mathbf{x}}_t$ denotes the Target velocity twist. Between two non-inertial frames the pose/velocity notation is only different in using lowercase of both frames. An example is the measured pose using the vision sensor which is the pose of $\mathcal{G}$ relative to $\mathcal{E}$, and is denoted as $\mathbf{H}_{eg}(R(\mu), r_c)$. Following these definitions, the forward kinematics of the chaser end-effector is $\mathbf{H}_e(R(\eta_c), \rho_c) = \mathbf{H}_b \mathbf{H}_{be}$. The transformation of the grasping frame $\mathcal{G}$ relative to $\mathcal{T}$ is denoted as $\mathbf{H}_{tg}(R(\eta_t), r_t)$.

A. Related work

Many approaches for grasping a free-tumbling Target mostly focus on feedback control methods [6][7] and momentum control methods [8]. In [9], the motion planning problem is solved for the approach phase of the grasping task. The planning is supported by a prediction of the Target's motion, achieved through an extended Kalman filter and noisy pose measurements of a laser camera system. The motion prediction occurs before any motion of the robot and the control strategy does not include a continuous visual feedback. An optimization problem is solved partly analytically and partly numerically, to minimize a penalty cost, function of four weighted quantities: travel time, distance, line-of-sight angle of the laser camera and robot joint acceleration. Experiments are conducted on an experimental facility which reproduces the dynamics of the tumbling Target and of a robot manipulator end-effector with an attitude-controlled base. In [10] a path planning method is presented for the rigidization phase
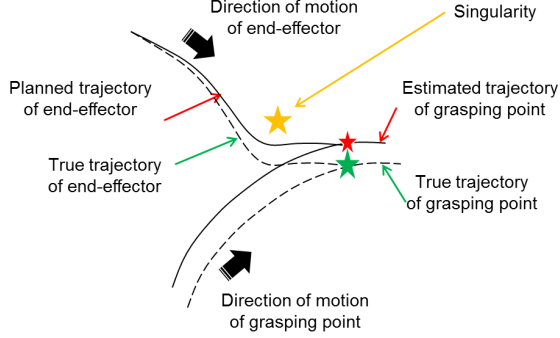
Fig. 2. The motion of the Target grasping point and of the robot end-effector are shwon for the model-based (solid line) and for the true (dotted line) cases.

which follows the grasping. It is assumed that the Chaser is attitude stabilized. As such, an optimal control problem is formulated for the detumbling of the Target to which an external moment is applied, solved analytically for the case of minimum time and zero final velocity.

In [11] an optimal control problem is solved numerically with nonlinear optimization in joint space, addressing the grasping task under the uncertainty of the initial and final positions of the robot end-effector resulting from tracking sensing data. The method is applied in simulation to a 2D problem, for a fully attitude controlled robotic system. In [12] a direct shooting method is used to treat the grasping problem in 3D, with inclusion of robot joint position and velocities contraints, as well as the Chaser free-floating dynamics. A look-up table approach is presented with which (close to) globally optimal soultions can be retreived in real-time for any possible tumbling state of the Target within a predefined range for the angular velocity.

In the context of state estimation, apart from the aforementioned contribution in [9], in [13] the authors implemented a filter cascade using range images for Target state and parameter estimation. In [14], a prediction method was adopted keeping autonomous grasping in focus. In [15], a closed-loop system was implemented for autonomous target tracking with visual servo and EKF as a part of the incremental inverse-kinematics controller. The idea of visual servoing and position-based impedance control were brought together in [16] for tasks in which contact with the environment is expected. As such, the aim here was to fuse the visual and joint position to improve the target's pose-estimation and hence improve tracking accuracy. Regarding the visual tracking, model-based methods which exploit robust edge features efficiently, were used in position-based control in [17] and in on-orbit servicing applications in [18].

B. Problem statement

In the grasping task of interest we assume that the Target has a known geometry, such that model-based computer

vision pose estimation can be used. Its angular rate is limited here to 2 deg/s (due to hardware limitations, the limit for the rigidization phase is set to 1 deg/s). The grasping point on the Target is predefined. The Chaser is not actuated (free-floating) and is initially in the same orbit as that of the Target with zero translational and angular velocities relative to the orbital frame.

The grasping task itself is commonly divided into three phases: an approach phase, in which the robot end-effector is brought in the vicinity of the moving grasping point of the Target; a tracking phase, in which the end-effector follows the grasping point with the same velocity, while homing in onto it and closing the grasp; a rigidization phase, in which the relative velocity between the Chaser and the Target is brought to rest by a suitable control of the robot manipulator.

A feasible trajectory to accomplish this task is provided by a motion planner, first described in [12] and extended here. The motion planner relies on a prediction of the Target motion, which can be accomplished as described in [14] [9]. However, in order to complete the task, the tracking controller will initially need to deviate from the reference trajectory, given that the latter is model-based and will differ from the real-world conditions. This fact is shown pictorially in Fig. 2. Note that the gross motion of the motion planner is still maintained, in order to avoid motion constraints such as, for example, a singularity.

Due to the deviation from the reference trajectory in the approach and tracking phases, the reference trajectory for the rigidization phase is adapted online, such that the deviation in joint space is recovered. This favours the fullfillment of joint position related constraints (collisions avoidance, singularity avoidance) throughout it.

II. METHODS

In this Section we present the methods used to solve the grasping task, to include elements of the motion planning, the visual-servo with manipulator dynamics, used in the approach and the tracking phases and the joint-position-based tracking control used in the rigidization phase. The control system architecture is shown in Fig. 3.

A. Motion planning and motion synthesizer

1) Approach and tracking phases: The free-floating dynamics of the robot manipulator are given by [5, §55]

$$\begin{bmatrix} \mathbf{M}_b & \mathbf{M}_{bm} \\ \mathbf{M}_{bm}^T & \mathbf{M}_m \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{x}}_b \\ \ddot{\mathbf{q}} \end{bmatrix} + \begin{bmatrix} \mathbf{C}_b & \mathbf{C}_{bm} \\ \mathbf{C}_{mb} & \mathbf{C}_m \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}}_b \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} 0 \\ \tau \end{bmatrix} \quad (1)$$

where $\mathbf{M}_b \in \mathbb{R}^{6\times6}$, $\mathbf{M}_{bm} \in \mathbb{R}^{6\times7}$, $\mathbf{M}_m \in \mathbb{R}^{7\times7}$ are the locked, coupled and 7-joint manipulator's inertias respectively. $\mathbf{C}_b \in \mathbb{R}^{6\times6}$, $\mathbf{C}_{bm} \in \mathbb{R}^{6\times7}$, $\mathbf{C}_{mb} \in \mathbb{R}^{7\times6}$, $\mathbf{C}_m \in \mathbb{R}^{6\times7}$ make up the whole Coriolis matrix, $\tau \in \mathbb{R}^7$ are the joint torques, $\dot{\mathbf{x}}_b \in \mathbb{R}^6$ is the twist of the base and $\mathbf{q} \in \mathbb{R}^7$ are the joint positions respectively. The motion planning problem
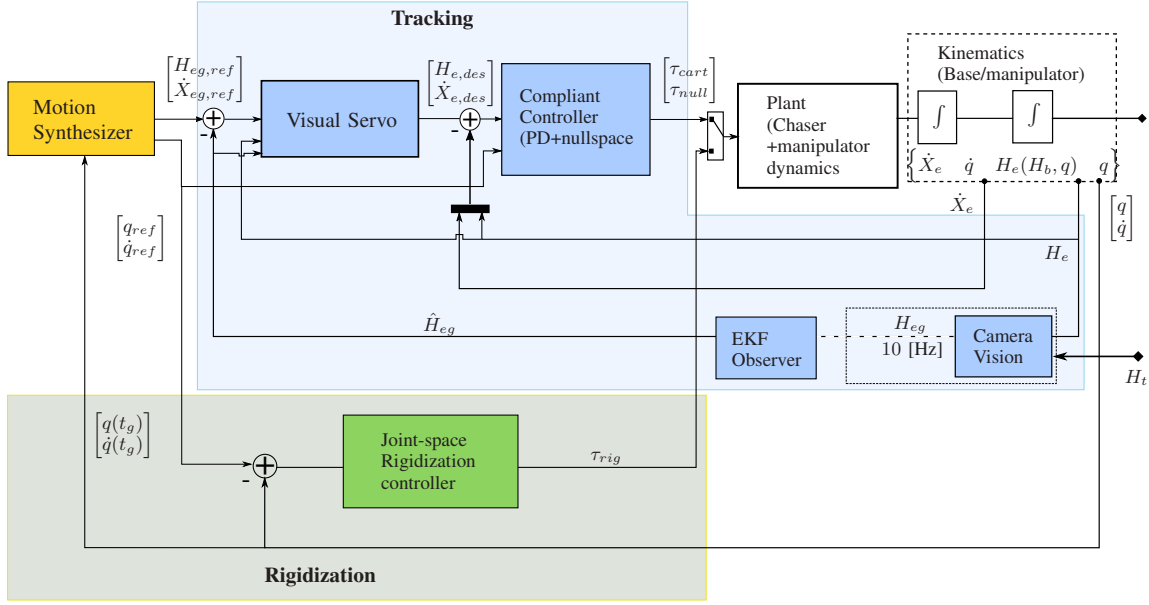
Fig. 3. Control system architecture for tracking or rigidization controllers.

relating to the approach and tracking phases is formulated as a point-to-point task (with non-zero final velocity) and an inverse differential kinematics task to follow it [12]. Mathematically, the first step can be formulated as the constrained optimal control problem

$$\min_{\mathbf{q}_1(t)} \Gamma_1(\mathbf{q}_1(t)) \qquad (2)$$

subject to

$$\mathbf{M}_b\, \ddot{\mathbf{x}}_\mathbf{b} + \mathbf{C}_b\, \dot{\mathbf{x}}_\mathbf{b} + \mathbf{C}_{bm}\, \dot{\mathbf{q}} = -\mathbf{M}_{bm}\, \ddot{\mathbf{q}}_\mathbf{1} \qquad (3)$$

$$\mathbf{h}_1(\mathbf{q}_1, \dot{\mathbf{q}}_\mathbf{1}) \leq 0 \qquad (4)$$

for $0 \leq t \leq t_{f1}$ and to the boundary conditions

$$\mathbf{f}_1(\mathbf{H}_e(t_{f1})) = 0,\ \mathbf{g}_1(\mathbf{x}^e(t_{f1})) = 0 \qquad (5)$$

$$\mathbf{q}_1(0) = \mathbf{q}_{1in}, \dot{\mathbf{q}}_\mathbf{1}(0) = 0. \qquad (6)$$

(3) then is an equality constraint stemming from the first line of these equations with a term on the right-hand side for rheonomically driven joints. t_{f1} is a predefined relative final time. Γ_1 is a predefined cost function. $\mathbf{h}_1$ include inequality box constraints of type $\mathbf{x}_{min} \leq \mathbf{x}(t) \leq \mathbf{x}_{max}$, for $\mathbf{x} = \{\mathbf{q}_1, \dot{\mathbf{q}}_1\}$.

Functions $\mathbf{f}_1$ and $\mathbf{g}_1$ are equality constraints on the final end-effector pose $\mathbf{H}_e(t_{f1})$ and twist $\dot{\mathbf{x}}_e(t_{f1})$ to be in a desired relative non-zero position and orientation with respect to the grasping point and travelling at its velocity. (6) expresses boundary conditions on position and on velocity.

The cost function is chosen to be the mechanical energy of the robot manipulator, in order to smoothen the solution,

i.e.

$$\mathbf{\Gamma} = \int_0^{t_{f1}} (\boldsymbol{\tau}_1{}^T(t)\, \dot{\mathbf{q}}_1(t))^2 dt. \qquad (7)$$

The tracking phase consists of an inverse differential kinematics solution, which is dictated by the motion of the Target. Its duration is defined by the relative time $t_{f2} - t_{f1}$. The same motion constraints apply for this phase as for the previous. During this phase, a constant homing-in velocity brings the end-effector onto the grasping point, closing the gap due to the non-zero relative position at the end of the previous phase. The solution is expressed by $\mathbf{q}_2(t)$.

The method in [12] is extended here to account for the requirements stemming specifically from the visual tracking. It is in fact required that throughout the approach phase sufficient features of the Target are in view of the camera (e.g., the Target's solar panel on its top suface) and that the pixel velocity in the image remains low. These result in motion constraints, the first of which is treated with an inequality constraint on the pitch angle ϕ_e of the end-effector frame $\mathcal{E}$ (rotation about its y-axis) CHECK relative to the target frame $\mathcal{T}$ (see Fig. 1), defined as

$$\phi_{e2} \leq \phi_{e\,mid2} + \phi_{e\,delta2}, \qquad (8)$$

where $\phi_{e\,mid}$ is a parameter to define a middle value for ϕ_e and $\phi_{e\,delta}$ is a paremeter which relates to the field-of-view of the camera.

Defining the pixel velocities in the x and y components of the end-effector frame $\mathcal{E}$ to be ${}^e v_{px} = {}^e v_y{}^{te}\,/{}^e r_z{}^{te}$ and ${}^e v_{py} = {}^e v_x{}^{te}\,/{}^e r_z{}^{te}$, inequality constraints are also

introduced as

$$-\Delta^e v_{px} \leq {}^e v_{px} \leq \Delta^e v_{px} \tag{9}$$

$$-\Delta^e v_{py} \leq {}^e v_{py} \leq \Delta^e v_{py}, \tag{10}$$

where $\Delta^e v_{px}$ and $\Delta^e v_{py}$ are parameters which relate to the pixel velocity limits in the camera image. Note that the velocity constraint in 5, not present in [12], contributes to minimizing the pixel velocity, by minimizing the relative velocity between the robot end-effector and the Target.

The output of the motion planner is judiciously defined to account for two important aspects. Firstly, a reference trajectory of the end-effector is provided relative to the grasping point frame $\mathcal{G}$ on the Target, i.e., as homogeneous trasformation $\mathbf{H}_{ge,ref}(t)$ and twist $\dot{\mathbf{x}}_{ge,ref}(t) = [{}^g v^{ge}\ {}^g \omega^{ge}]^T(t)$, for $0 \leq t \leq t_{f2}$. Note that the end-effector pose is effectively given by $\mathbf{H}_{ge} = \mathbf{H}_{gI}\mathbf{H}_{Ie}$. The tracking controller is designed to track this relative motion. This ensures that any error in $\mathbf{H}_{gI}$ (Target motion prediction errors) and in $\mathbf{H}_{Ie}$ (tracking control errors, robot forward kinematic errors) are intrinsically eliminated in following the desired relative motion $[\mathbf{H}_{ge,ref}](t)$, which is independent of these uncertainties. Note however, that the motion of the robot relative to absolute space, and therefore joint space, will deviate from the motion planning solution. This relates to the second aspect to be considered, namely that a reference trajectory is also provided in the joint space of the robot manipulator, $[\mathbf{q}_{ref}, \dot{\mathbf{q}}_{ref}](t)$. As will be shown in Section II-C, the tracking control law contains a main term to minimize the Cartesian tracking error and a second term, projected in the null-space of the robot manipulator, to minimize the tracking error in joint space (see (29) and (30)). This is intended to minimize the deviation of the robot motion in joint space from the feasible reference trajectory. The null-space term also cares for the kinematic redundancy of the robot manipulator.

2) Rigidization phase: The rigidization phase requires bringing the robot joints velocity to zero, while ensuring that the motion constraints are not violated. We solve

$$\min_{\mathbf{q}_3(t)} \Gamma_3(\mathbf{q}_3(t)) \tag{11}$$

subject to

$$\mathbf{M}_{b\,mod}\ddot{\mathbf{x}}_{\mathbf{b}} + \mathbf{C}_{b\,mod}\dot{\mathbf{x}}_{\mathbf{b}} + \mathbf{C}_{bm\,mod}\dot{\mathbf{q}} = -\mathbf{M}_{bm\,mod}\ddot{\mathbf{q}}_{\mathbf{1}} \tag{12}$$

$$\mathbf{q}(t_{03}) = \mathbf{q}(t_{f1} + t_{f2}), \dot{\mathbf{q}}(t_{03}) = \dot{\mathbf{q}}(t_{f1} + t_{f2}) \tag{13}$$

$$\dot{\mathbf{q}}(t_{03} + t_{f3}) = \mathbf{0} \tag{14}$$

$$\mathbf{h}_3(\mathbf{q}_3(t)) \leq \mathbf{0} \tag{15}$$

where the initial conditions in (13) express the dependency on the final conditions of the previous tracking phase. The matrices $\mathbf{M}_{b\,mod}$, $\mathbf{M}_{bm\,mod}$, $\mathbf{C}_{b\,mod}$ and $\mathbf{C}_{bm\,mod}$ of the free-floating robot dynamics are updated to include the inertial properties of the Target on the end-effector.

In this phase, the offline motion planner provides a reference trajectory in joint space. The tracking controller here is a joint space controller. The task is to rigidize the compound after the grasp, bringing the relative motion between the two satellites to zero. Given that only internal forces come into play, the angular momentum of the Target is shared with the Chaser and at the end of the maneuver the two satellites tumble with a new angular velocity, function of their total inertia. The cost function is chosen to be the mechanical energy of the robot manipulator, as in (7), in order to minimize its loads on the joints and on the end-effector. Due to the small Target tumbling velocities considered, it was found that these loads are very low and do not require relative motion constraints.

3) Method for the contrained optimization problem: The optimization problem described above is solved as a nonlinear programming problem (NLP). The inequality constraints $\mathbf{h}$ are satisifed at a finite number of k via points. The equality constraints in (3) and (12) are satisfied by numerical integration. As such, the NLP is solved with a single shooting approach, with parameterization of the joint positions in time, i.e., $\mathbf{q}_i = \mathbf{q}(t, \mathbf{p}_i)$, where $\mathbf{p}_i$ is a column matrix containing N_i optimization parameters and i is either 1 or 3. For the approach and for the rigidization problems described above, two parameter sets $\mathbf{p}_1$ and $\mathbf{p}_3$ result. The NPL is solved with the Sequential Quadratic Programming algorithm from the MOPS library [?]. The numerical integration is performed with an explicit fifth order Runge-Kutta method.

In the online setting however a modification of the reference trajectory for the ridigization is necessary, to account for the deviation from the reference trajectory in the previous approach phase. Each joint trajectory is parameterized with a uniform B-spline, the parameters of which are the optimization parameters to be found. The solution parameters generated offline are still used to synthesize online a new rigidization trajectory, by modifying them with the measured initial robot manipulator joint states. Let the measured initial position and velocity for a given joint be q_{0m} and $\dot{q}_{0m}$ respectively, and the B-spline parameters $q_i, 1 \leq i \leq N_B$ ($N_B < N$ and $N - N_B$ are the parameters which satisfy the boundary conditions). The modified parameters are then taken to be:

$$q_{i\,mod} = q_i + \left(q_0 - q_{0m} + \frac{\dot{q}_{0m}(t_{rf} - t_{ri})}{N_B}\right)\frac{(N_B - i)}{N_B}, \tag{16}$$

where the second term accounts for the initial position deviation $q_0 - q_{0m}$ and the joint velocity at the initial time times the duration of the first B-spline segment, $(t_{rf} - t_{ri})/N_B$, in which the spline cannot be modified. This effectively smoothly brings the joint configuration back to the one planned offline. As such, the fullfilment of the joint position related motion constraints is favoured. The chosen modification of the trajectory could however give rise to an increase

in robot joint torques, since it implies pushing the Target onto the predicted trajectory. Note however, that the choice of the cost function and of t_{rf} in Eqn. (7) minimize these, allowing for a larger margin from their limits.

B. Extended Kalman Filter (Model-based control)

The Extended Kalman Filter (EKF) is a ubiquitous state observer for nonlinear systems. It is a nonlinear observer in the sense that the model propagation is nonlinear whereas, the error injection is linear. In the context of this paper, the EKF provides robustness to the tracking controller against incoming measurement outliers from the computer vision algorithm by ensuring measurement continuity. Due to its predictor-update structure, it also provides a state prediction which allows us to compensate for image-processing time-delays and ensure continuous control errors during occlusion or measurement failures caused by degenerate observability conditions. From a tracking controller perspective, the EKF also estimates the target velocities for a feed-forward term for the Visual Servo, which compensates for the disturbance due to target motion that acts on the controller.

Assumption 1: Since the scope of this paper is limited to an operational time, $T < 20$ [sec], the contribution of the term which accounts for the relative translation due to orbital motion, and is given by the Clohessy-Wiltshire equations, is neglected.

The dynamic system for the EKF is modeled with the state, $x_a \in \mathbb{R}^{13} = \begin{bmatrix} \theta^T & r^T & \omega^{bT} & \dot{r}^T \end{bmatrix}^T$ as,

$$\frac{d}{dt} \begin{bmatrix} \dot{\theta} \\ r \\ \omega^b \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \omega^b \otimes \theta \\ \dot{r} \\ -\omega_\times^b I \omega^b \\ \Phi(n) \end{bmatrix} \tag{17}$$

where $\otimes$ denotes the quaternion multiplication tensor, $(.)_\times$ denotes the skew-symmetric form of $\mathbb{R}^3$. $\Phi(n)$ is the linear approximation of orbital dynamics given by the Clohessy-Wiltshire equations [19] for mean anomaly n. Based on ass. 1, $\Phi(n) = \mathbf{0}_{3,3}$.

For the quaternion group, Q, in the multiplicative variant of the EKF, the quaternion state is maintained as a 3-component vector as $\delta\theta(a_p) \in T_{\overline{\theta}}Q$, at a nominal value of the quaternion $\overline{\theta}$. $T_{\overline{\theta}}Q$ is the tangent space to the quaternion manifold and a_p is a 3-component parameterization vector. A detailed discussion on such vector expressions was given in [20]. In this text, the Modified Rodriguez Parameters (MRP), a_p was chosen so that singularity is not encountered before angular error 2π. The tangent vector $\delta\theta$, the parameterization vector a_p and the quaternion θ are related as follows,

$$a_p(\theta) = \frac{4}{1+\theta_0}\theta_v \quad , \delta\theta_v(a_p) = \frac{1}{16}\begin{bmatrix} 8a_p \\ 16 - a_p^T a_p \end{bmatrix} \tag{18}$$

The differential form for a_p is given as,

$$\dot{a}_p = \left(-\frac{1}{2}\omega_\times^b + \frac{1}{8}\right)a_p + \left(1 - \frac{1}{16}a_p^T a_p\right)\omega^b \tag{19}$$

Among the alternatives for scaled inertia parameterization for zero-torque motion propagation, [21, §4.6] the 3-parameter (p_x, p_y, p_z) form proposed in [19] is used. So, the angular dynamics can be written by reformulating the rotational dynamics Euler equation in eq. (17) as,

$$\dot{\omega}^b = \psi(\omega^b) = \begin{bmatrix} p_x \omega_y^b \omega_z^b \\ p_y \omega_x^b \omega_z^b \\ p_z \omega_x^b \omega_y^b \end{bmatrix} \tag{20}$$

which has a linear approximation about the current estimate, $\hat{\omega}^b$ as,

$$\delta\dot{\omega}^b = \mathbf{M}\delta\omega^b \tag{21}$$

where $M = \left(\frac{\partial \psi(\omega)}{\partial \omega^b}\right)_{\omega = \hat{\omega}^b}$ is a hollow matrix [19].

Finally, using eq. (19), (21) and eq. (**??**) from ass. 1, the linear approximation of the torque-free system (17) with state $x \in \mathbb{R}^{12} = \begin{bmatrix} a_p^T & r^T & \omega^{bT} & \dot{r}^T \end{bmatrix}$ is given in [21, eq. 4.79] as,

$$\dot{x} = \mathbf{A}x + \mathbf{B}w \tag{22}$$

From the Fig. 1, the measurement is derived by using current forward kinematics ($\mathbf{H}_e$) and known target geometry ($\mathbf{H}_{tg}$) is given as,

$$g = \begin{bmatrix} r \\ \theta \end{bmatrix} = \begin{bmatrix} R(\eta_c)^T r_c + \rho_c - R(\eta_t)\rho_t \\ \eta_t^* \otimes \mu \otimes \eta_c \end{bmatrix} \tag{23}$$

where $(.)^*$ refers to the quaternion inverse. In order to obtain the EKF measurement y from available measurement g, a_p is computed from eq. (23). The linear approximation for the EKF measurement was derived several times before in [19], [13], [21], and we only state the final formula as,

$$y = \mathbf{H}x, \quad \mathbf{H} = \begin{bmatrix} -R(\overline{q})\rho_{t\times} & 0_{3,3} & 1_{3,3} & 0_{3,3} \\ 1_{3,3} & & & 0_{3,9} \end{bmatrix} \tag{24}$$

where $\mathbf{H} = \frac{\partial y}{\partial x}\Big|_{x_a = \hat{x}_a}$ is obtained by linear approximation of the measurement function about the current state estimate $\hat{x}_a$. Since the Kalman Filter equations are standard, we do not explicitly write the prediction-update form. The interested reader is referred to [22] for seminal paper on equations. By using the aforementioned system of equations, the EKF was implemented with the Outlier-rejection (OR) scheme suggested in [23] to provide robustness against local outliers from the vision system. The EKF output $\hat{y} = \begin{bmatrix} \hat{r}_c^T & \hat{\mu}^T \end{bmatrix}^T \equiv \hat{\mathbf{H}}_{eg}$ is used for control in the next step.

C. Visual Servo for Approach and Tracking Phases

For the free-floating space manipulator, eq. (1) can be reformulated after eliminating terms explicitly depending on $\mathbf{x}_b$ and its derivatives as,

$$\hat{\mathbf{M}}_m(\mathbf{q}, \mathbf{x}_b)\ddot{\mathbf{q}} + \tilde{\mathbf{C}}_m(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{x}_b, \dot{\mathbf{x}}_b)\dot{\mathbf{q}} = \tau, \quad \text{(free-floating model)}$$
(25)

where $\hat{\mathbf{M}}_m = \mathbf{M}_m - \mathbf{M}_{bm}^T\mathbf{M}_b^{-1}\mathbf{M}_{bm}$, $\hat{\mathbf{C}}_b = \mathbf{C}_{mb} - \mathbf{M}_b^{-1}\mathbf{C}_b$, $\tilde{\mathbf{C}}_m = \hat{\mathbf{C}}_m(\mathbf{q}, \dot{\mathbf{x}}_b, \dot{\mathbf{q}}) - \hat{\mathbf{C}}_b(\mathbf{q}, \dot{\mathbf{x}}_b, \dot{\mathbf{q}})\mathbf{M}_b^{-1}\mathbf{M}_{bm}$, are the generalized terms (see [25]) and $\hat{\mathbf{C}}_m = \mathbf{C}_m - \mathbf{M}_b^{-1}\mathbf{C}_{bm}$. Using the formulation in [26, eq. 10], the eq. (25) can be written only for the Cartesian dynamics as,

$$\mathbf{\Lambda}_m(\mathbf{q}, \mathbf{x}_b)\ddot{\mathbf{x}}_e + \mu_m(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{x}_b, \dot{\mathbf{x}}_b)\dot{\mathbf{x}}_e = \mathbf{F}_e \quad (26)$$

where $\mathbf{\Lambda}_m = (\mathbf{J}_g(\mathbf{q}, \mathbf{x}_b)\hat{\mathbf{M}}_m^{-1}\mathbf{J}_g(\mathbf{q}, \mathbf{x}_b)^T)^{-1}$, $\mu = \mathbf{J}_g(\mathbf{q}, \mathbf{x}_b)^{-T}\tilde{\mathbf{C}}_m\mathbf{J}_g(\mathbf{q}, \mathbf{x}_b) + \mathbf{J}_g(\mathbf{q}, \mathbf{x}_b)^{-T}\hat{\mathbf{M}}_m\frac{d}{dt}\mathbf{J}_g(\mathbf{q}, \mathbf{x}_b)^{-1}$ where $\mathbf{F}_e$ is the Cartesian control torque.

In the context of On-Orbit Servicing, the target is partially cooperative in that its model is available in advance. As a result, in this paper, the kinematic-level control used is Position-based Visual Servo (PBVS), which means that the control law uses the pose error $\hat{\mathbf{H}}_{eg}$ that is reconstructed from image-space measurements. Based on the model-based visual tracking and EKF's pose results, this control law is used to provide a desired velocity ($\hat{\dot{X}}_e$) and pose for the inner loop impedance controller.

In accordance to fig. 3, a kinematic tracking velocity for the manipulator is generated using the visual servo law. Firstly, the reference trajectory is used to compute a desired end-effector pose $\mathbf{H}_{e,des} = (\mathbf{H}_e\hat{\mathbf{H}}_{eg})\mathbf{H}_{ge,ref}$. Secondly, this function is used as a pose error to construct the visual servo law $\hat{\dot{\mathbf{x}}}_e = \begin{bmatrix} \dot{r}_e^T & \omega_e^T \end{bmatrix}^T$. Using the slow-sampled $\hat{\dot{\mathbf{x}}}_e$, a virtual frame is generated by integrating $\dot{\mathbf{H}}_{e,v}(R_{e,v}, p_{e,v})$. Note here that this virtual frame is solved as an initial value problem between two EKF sampling times and acts like an intermediate trajectory for the manipulator between these sampled points. This is initialized as, $\mathbf{H}_{e,v}(n) = \mathbf{H}_e(n) \; \forall n \in [1..k]$ where n is the nth EKF sample time. The control law used is,

$$\hat{\dot{\mathbf{x}}}_e = K_1 e_{pose}(H_{e,v}, \; H_{e,des}) + \mathbf{v}_t$$
$$\mathbf{v}_t = \begin{bmatrix} \hat{\dot{r}} \\ R(\hat{\mu})\hat{\omega}^b \end{bmatrix}$$
(27)

where $k_{1,ii} > 0$, $\forall i \in [1...6]$, e_{pose} is the pose error function defined as, $e_{pose}(H_1, H_2) = \begin{bmatrix} (p_1 - p_2)^T & \psi(R_1^T R_2)^T \end{bmatrix}^T$, with the rotation error $\psi(R) = \frac{1}{2}[R - R^T]^\vee$ and $[a_\times]^\vee = a$, $a \in \mathbb{R}^6$ is the mapping from skew symmetric matrix to vector form. $\mathbf{v}_t$ is simply the estimated Target velocity applied as a feed-forward. A note on this error formulation and subsequent Lyapunov function analysis can be found in [27].

Finally, a compliant controller is placed in cascade after

the visual servo which acts upon the Cartesian error. The impedance error is a Cartesian error observed in the inertial frame and is denoted as $\mathbf{e} = e_{pose}(\mathbf{H}_{e,des}, \mathbf{H}_e)$. Using the corresponding velocity error $\dot{\mathbf{e}}$, a PD+ Cartesian control law is formulated as,

$$\mathcal{F}_m = \mathbf{\Lambda}_m(\mathbf{q}, \mathbf{x}_b)\hat{\ddot{\mathbf{x}}}_e + \mu_m(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{x}_b, \dot{\mathbf{x}}_b)\hat{\dot{\mathbf{x}}}_e - \mathbf{K}_P\mathbf{e} - \mathbf{K}_D\dot{\mathbf{e}}$$
(28)

where $k_{P,ii} > 0$ and $k_{D,ii} > 0$, $\forall i \in [1...6]$ are the choices for Cartesian stiffness and damping terms respectively. The corresponding joint torques with null-space control torque $\mathbf{\Gamma}$ are given as,

$$\tau = \mathbf{J}_g^T\mathcal{F}_m + (\mathbf{1}_{n,n} - \mathbf{J}_g^T\mathbf{J}_g^{*^T})\mathbf{\Gamma} \quad (29)$$

where $\mathbf{J}_g^*$ is the dynamically consistent Weighted Pseudo-inverse of $\mathbf{J}_g$ (generalized Jacobian) and Γ is given by

$$\Gamma = -\mathbf{K}_{PN}(\mathbf{q} - \mathbf{q}_{ref}) - \mathbf{K}_{DN}(\dot{\mathbf{q}} - \dot{\mathbf{q}}_{ref}) \quad (30)$$

where the two terms on the right hand side are a stiffness and damping terms respectively, with constant diagonal gain matrices $\mathbf{K}_{PN}$ and $\mathbf{K}_{DN}$.

In eq. 30, the two terms on the left hand side are a stiffness and damping terms for the null-space respectively, with constant diagonal gain matrices $\mathbf{K}_{PN}$, $k_{PN,ii} > 0$ and $\mathbf{K}_{DN}$, $k_{DN,ii} > 0$, $\forall i \in [1...7]$.

Note that the use of $\mathbf{J}_g$ ensures that the Cartesian and null-space coordinates are decoupled in the inertia matrix. However Coriolis/Centrifugal matrix remains coupled [26, eq. 10]. This coupling is argued in [26, pp. 2001] to be of minor importance in practice. The validity of the reference trajectory guarantees that the null-space projection in eq. 29 is not null. However, as shown in Fig. 2, uncertainties in the reference trajectory require departing from it in inertial space, but not relative to the target. As such, the null-space term will not converge but only minimize the error as $\min_{f(q) = \mathbf{H}_{e,des}} ||\mathbf{q} - \mathbf{q}_{ref}||_2$, where $f(q)$ is the forward kinematics function.

D. Computer Vision for Approach and Tracking Phases

The visual pose estimation is based on work of [17] which relies on a known object geometry. The algorithm consisits of sampling model contours and edge detection, followed by a local optimization of the 3D rigd-body transform. The re-projection error is then minmized to allign visible model contours and image edges using the Levenberg-Marquardt algorithm. The complex tumbling motion of the target poses difficulty in visual tracking. In order to address this problem, a motion prediction scheme based on an extended kalman filter is integrated into the local optimization. The prediction here relies on a simple kinematic motion model, assuming constant frame to frame motion. Under this assumption, the

visual tracker is able to cope with a relatively high image motion between two camera frames. Moreover, the short-term prediction scheme helps the visual tracker reduce the search space in order not to remain trapped in potential local optima during minimization of the cost function. If the tracker does remain trapped (see for example Fig.s 8 and 9), the algorithm needs to be reinitialized either with an external global detector or with the EKF presented in Section II.B CHECK. The latter in fact satisfies the realtime constraints and makes use of the dynamic model of the Target. This approach will be implemented in future work. The qualitative tracking results are illustrated in Fig. 4, with examplar tracked images from the approach and tracking phases. Those in red indicate the projections of the model contours onto the image edges at the estimated pose. The overlap of the model contours and image edges shows the successful tracking and pose estimation.

E. Joint Tracking Control for Stabilization Phase

The tracking controller for the rigidization phase is described in Fig. 3. The motion synthesizer provides an input expressed as $q_r, \dot{q}_r$. The PD+ control law for the tracking controller [28] adopted here is then as follows

$$
\begin{aligned}
\boldsymbol{\tau}_{rig} = &\hat{\mathbf{M}}_{m,t}(\mathbf{q}, \mathbf{x}_b)\ddot{\mathbf{q}}_r + \tilde{\mathbf{C}}_{m,t}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{x}_b, \dot{\mathbf{x}}_b)\dot{\mathbf{q}}_r \\
&+ \mathbf{K}_p(\mathbf{q}_r - \mathbf{q}) + \mathbf{K}_d(\dot{\mathbf{q}}_r - \dot{\mathbf{q}})
\end{aligned}
\tag{31}
$$

where $\mathbf{K}_p$, $k_{p,ii} > 0$ and $\mathbf{K}_d$, $k_{d,ii} > 0$, $\forall\ i = [1..7]$ are (7x7) diagonal matrices including the stiffness and damping gains respectively and, $\hat{\mathbf{M}}_{m,t}$ and $\tilde{\mathbf{C}}_{m,t}$ are inertia and Coriolis matrices computed by adding the transformed target inertia to $\hat{\mathbf{M}}_m(\mathbf{q}, \mathbf{x}_b)$ and $\tilde{\mathbf{C}}_m(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{x}_b, \dot{\mathbf{x}}_b)$ respectively from eq. (25). Note that, this is based on the assumption that the target CoM is the new end-effector frame. The third proportional term plays an important role, since it brings the system into a feasible final configuration, as dictated by the motion planner.

III. EXPERIMENTAL FACILITY

The OOS-SIM experimental facility is described in detail in [29]. However, in order to simulate the motion of the two satellites after the grasp, a new approach was used, based on the conservation of momentum. We note, in fact, that after the grasp the forces of interaction between the Target and the Light-Weight Robot (LWR CHECK) are internal and, as such, do not alter the total momentum of the Chaser-LWR-Target system. Based on the measured relative pose between the Target and the LWR, the dynamic model of the Chaser-LWR multibody system is updated at the time of grasping (end of closure of gripper), t_{ri}, to modify the inertial properties of the LWR's end-effector, to include those of the Target. From this point on, the laws of conservation of momentum are integrated in time, in function of the measured LWR joint positions and their first derivatives, as follows.

Given the relationship between the twist of the Chaser $\dot{\mathbf{x}}_e$ and the velocity of the LWR robot joints as [5]

$$
\mathbf{M}_b\dot{\mathbf{x}}_e + \mathbf{M}_{bm}\dot{\mathbf{q}} = \mathbf{L}
\tag{32}
$$

where $\mathbf{L}$ is the constant total momentum, we first note that the latter is assumen to be zero before the grasp. At $t = t_{stab}$, matrices $\mathbf{M}_m$ and $\mathbf{M}_{bm}$ are modified to include the modelling of the Target. The left-hand side is also updated to be equal to the measured momentum of the Target. Assuming that there is no impact at the grasp, the current state of the system is then valid as initial conditions for integrating Eq. CHECK (32) in $\dot{\mathbf{x}}_e$, i.e.,

$$
\int_{t_{ri}}^{t_{ri}+t} \dot{\mathbf{x}}_e \, \mathrm{d}t = \int_{t_{ri}}^{t_{ri}+t} \mathbf{M}_{b\ mod}^{-1} \left(\mathbf{L}_{mod} - \mathbf{M}_{bm\ mod}\,\dot{\mathbf{q}}\right) \, \mathrm{d}t
\tag{33}
$$

which provides the pose $\mathbf{H}_e$ to be commanded to the OOS-SIM industrial robot which simulates the Chaser. The Target was then commanded to followed the Light-Weight Robot through the internal actions which the latter imparted on it. These, as done in [29], were sensed by the FTS at the base of the Target and given as input to the online integration of its equations of motion.

With this approach we do not use the FTS of the Chaser to determine its motion, as done in [29], thus eliminating sources of simulation error. In fact, the FTS signal is affected by modelling errors (since the signal has to be corrected by the LWR gravity term) as well as by drift (due to sensor noise). We still use the FTS to determine the motion of the Target, since the inaccuracy of the forward kinematics of the facility does not favour a position control approach.

IV. RESULTS

Fig. 5 shows the motion planner solution for a Target spin velocity of 2 deg/s. The figure shows the three phases of the maneuver. The transition from the approach to the tracking phase is smooth, as the end-effector alligns itself with the grasping point on the Target. The values of the parameters $\phi_{e\ mid}$ and $\phi_{e\ delta}$ were set to 40 and 7.5 deg respectively. The tracking phase lasts 3.5 seconds to allow for the gripper to close onto it. The rigidization phase starts after the Target is fully grappled and lasts five seconds and the number of parameters per joint was $N_B = 6$. The solution resulted from a global search, which found different local minima, as expected, due to nonlinearity of the optimization problem.

Fig. 6 shows the measured robot manipulator joint position error, defined as $\mathbf{q} - \mathbf{q}_{ref}$, for the cases of Target spin velocities of 1 deg/s and 2 deg/s respectively. In the experiments on the OOS-SIM facility it was in fact found that the rigidization task for the 2 deg/s case was intractable with the current setup. CHECK need motivation As such the experimental results presented here for the rigidization phase only relate to the 1 deg/s case. In Fig. 7 the measurement of the Target motion is shown, in which we observed a
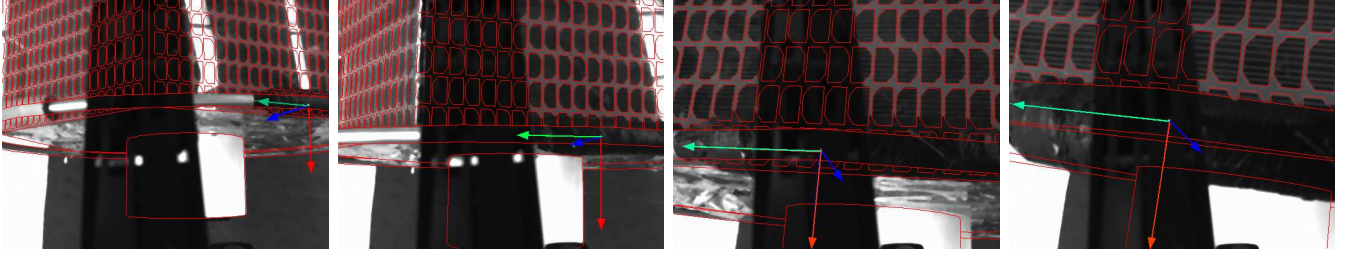
Fig. 4. Exemplar images during visual tracking for approach and tracking phase. The overlay of the model contours (in red) and image edges shows the successful tracking and pose estimation. The axes of the coordinate frame of the grasping point $\{\mathcal{G}\}$ are indicated in red, green and blue. The target features reduce significantly at very close range because of gripper finger occlusion (black in the image) and camera field of view.
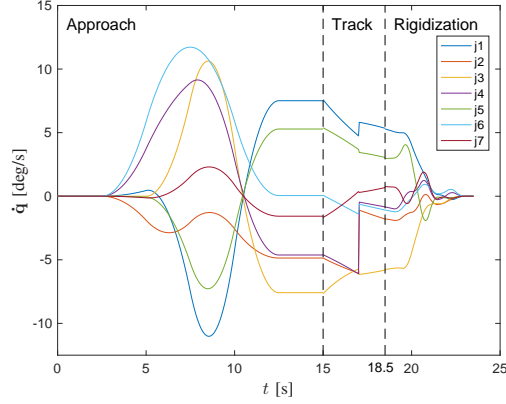

Fig. 5. Simulated robot manipulator joint velocities for a Target spin velocity of 2 deg/s. The transition between the phases, at t=15 s and at t=18.5 s, is smooth. The jump during the tracking phase is due to the homing-in motion, which is however directed towards the target and, as such, does not have a noticeable effect on the pixel velocity in the camera image.
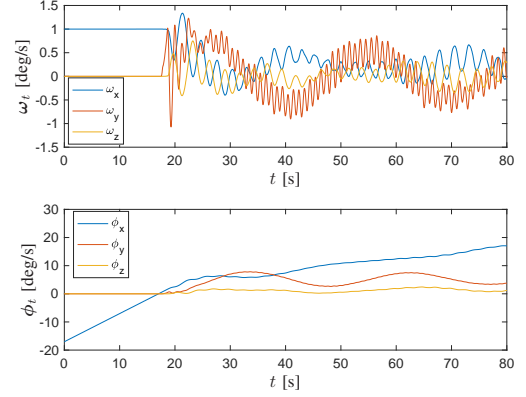

Fig. 7. Measured Target angular velocity (top) and orientation (bottom) for the 1 deg/s case. The post-grasping tumbling of the sattellite stack is visible in the (reduced) rotation about the x-axis.
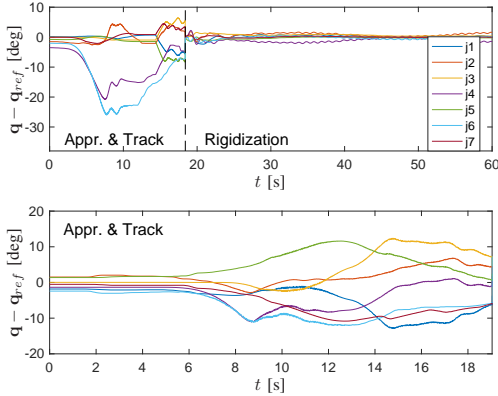

Fig. 6. Measured robot manipulator joint position error for Target spin velocity of 1 deg/s (top) and 2 deg/s (bottom). For the latter case, only the approach phase is shown, in which however the use of the EKF and of the extended parameterization of the joint motion give a net increase in joint-space tracking performance.

very slow damping of the angular velocity. The oscillations in the angular velocity appear to have two frequencies: the lower one is to be attributed to the imepdance control of the LWR; the higher one is to be attributed to the time delay and discretization effects in the control loop of the facility (independent of the LWR impedance parameters).

Returning to Fig. 6, the error in the joint position is shown to improve substantially for the second case, due to the fact that the EKF and the joint motion parameterization shown in Fig. 5 is used (the manuever was not possible without these new features). While the first provides a smoother estimate of $\mathbf{H}_{ge}$ (as shown below), the latter reduces the pixel velocity by a factor of two. At the point of transition between the tracking and the rigidization phase ($t = 18.5$), the joint position error becomes zero, due to the recomputation of the reference trajectory (see SectionII-A.2). The tracking error in the rigidization phase is then shown to remain small.

In Figures 8 the output error norm of the position r_c of EKF and vision is compared with respect to the ground truth. The EKF error norm is plotted as a heat-map of the statistical metrix, χ^2 of the incoming vision data. It is observed that the χ^2 measure degrades (yellow) for $t > 0$ and $||r_c|| \to 0$. In Fig. 8, the degradation of vision data cul-
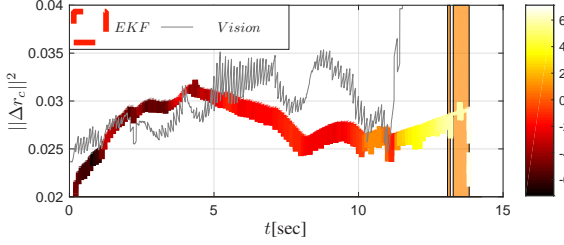
Fig. 8. Comparison of direct-Vision measurements with EKF, based on the norm $||\Delta r_c^2||$. As $||r_c|| \to 0$, the statistical measure of fitness, $\chi^2 \to \infty$ eventually leading to pose estimation failures due to degenerate observability.

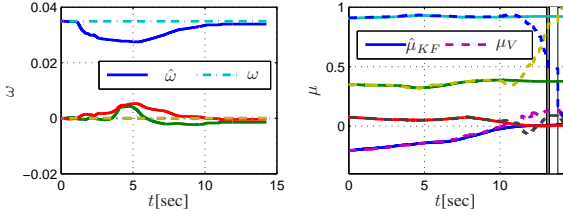Fig. 9. a) Target state estimation (ω) in EKF for tracking control. b) Divergence of visual tracking's orientation estimate towards end of maneuver.

minates in measurement failures (marked as rectangular area) towards the end. The EKF estimates though stay bounded. Fig. 9 shows the velocity convergence of the filter. It also demonstrates the orientation degradation as $||r_c|| \to 0$. This observation betrays a general problem with the eye-in-hand configuration of visual servoing, since, in close proximity to target, the observability degeneracy degrades camera-based measurement.

V. DISCUSSION AND CONCLUSION

The main achivement of this paper is to demonstrate the feasibility of a grasping strategy based on Target motion prediction, robot motion planning and trajectory tracking, specifically for the latter step. In fact, due to modelling uncertainties, the true trajectory deviates from the reference trajectory in joint space. The latter is therefore partially adapted online for the rigidizaiton phase. A method is presented and validated here, which does that while favouring joint position related motion constraints. The tracking controller is then shown to succesfully perform the task. In this way, the feasible (and optimal) motion planning solutions, which describe trajectories for the full degrees of freedom of the system throughout the approach and rigidization phases, and which are computed offline with much computational effort, are shown to be useful for control purposes.

A second major achievement of this paper is the implementation of an Extended Kalman Filter in between the visual tracker and the visual servo in cascade with an impedance controller. This ensures a Lipschitz continuity of measurements during the robot motion CHECK (make clear why

this is not a trivial engineering step), making the control method robust for executing the grasping task. It is in fact shown that the visual tracking algorithm adoped may become degenerate in the vicinity of the Target (CHECK but Aghili does this too). Furthermore, the new formulation to reduce the computational footprint has been shown to give a signifcant improvement (CHECK where?).

The momentum-based approach on the OOS-SIM facility has proven to be efficient in reducing drift due to noise and modelling errors. Future work will be dedicated to improving its performance in the coupled configuration. Furthermore, the stability of complete interconnected system will be addressed. Finally, the sensing of the Chaser, to include typical sensor noise and sampling times, will be considered.

REFERENCES

[1] T. Imaida, Y. Yokokohji, T. Doi, M. Oda, and T. Yoshikwa, "Ground-space bilateral teleoperation experiment using ets-vii robot arm with direct kinesthetic coupling," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation*, vol. 1, pp. 1031–1038 vol.1, 2001.

[2] M. W. James Shoemaker, "Orbital express space operations architecture program," 2003.

[3] F. Sellmaier, T. Boge, J. Spurmann, S. Gully, T. Rupp, and F. Huber, "On-orbit servicing missions: Challenges and solutions for spacecraft operations," in *SpaceOps 2010 Conference*, April 2010.

[4] J. Telaar, I. Ahrns, S. Estable, W. Rackl, M. De Stefano, R. Lampariello, N. Santos, P. Serra, M. Canetri, F. Ankersen, *et al.*, "Gnc architecture for the e. deorbit mission," in *7th European Conference for Aeronautics and Space Sciences*, 2017.

[5] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2016.

[6] S. A. A. Moosavian and E. Papadopoulos, "Free-flying robots in space: an overview of dynamics modeling, planning and control," *Robotica*, vol. 25, no. 5, pp. 537–547, 2007.

[7] M. D. Stefano, J. Artigas, A. Giordano, R. Lampariello, and A. Albu-Schaeffer, "On-ground experimental verification of a torque controlled free-floating robot," in *13th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA 2015)*, 2015.

[8] K. Yoshida, D. Dimitrov, and H. Nakanishi, "On the capture of tumbling satellite by a space robot," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 4127–4132, 2006.

[9] F. Aghili, "A prediction and motion-planning scheme for visually guided robotic capturing of free-floating tumbling objects with uncertain dynamics," *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 634–649, 2012.

[10] F. Aghili, "Optimal control of a space manipulator for detumbling of a target satellite," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 3019–3024, IEEE, 2009.

[11] A. Flores-Abad, Z. Wei, O. Ma, and K. D. Pham, "Optimal control of a space robot to approach a tumbling object for capture with uncertainties in the boundary conditions," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, p. 4522, 2013.

[12] R. Lampariello and G. Hirzinger, "Generating feasible trajectories for autonomous on-orbit grasping of spinning debris in a useful time," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 5652–5659, IEEE, 2013.

[13] M. D. Lichter and S. Dubowsky, "State, shape, and parameter estimation of space objects from range images," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 3, pp. 2974–2979 Vol.3, April 2004.

[14] U. Hillenbrand and R. Lampariello, "Motion and parameter estimation of a free-floating space object from range data for motion prediction," in *Proceedings of i-SAIRAS*, 2005.

[15] Z. Z. Hong and D. Gangqi, *Active Space Debris Removal by Visual Servo Controlled Autonomous Robotics*. AIAA SPACE Forum, American Institute of Aeronautics and Astronautics, Sept. 2016.

[16] V. Lippiello, B. Siciliano, and L. Villani, "A position-based visual impedance control for robot manipulators," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 2068–2073, April 2007.

[17] T. Drummond and R. Cipolla, *Real-time visual tracking of complex structures*. 2002.

[18] N. W. Oumer, G. Panin, Q. Mülbauer, and A. Tseneklidou, "Vision-based localization for on-orbit servicing of a partially cooperative satellite," *Acta Astronautica*, vol. 117, pp. 19–37, 2015.

[19] F. Aghili and K. Parsa, "Adaptive motion estimation of a tumbling satellite using laser-vision data with unknown noise characteristics," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 839–846, Oct 2007.

[20] M. Landis, "Attitude Error Representations for Kalman Filtering," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 2, pp. 311–317, 2003. doi: 10.2514/2.5048.

[21] B. E. Tweddle, *Computer Vision-Based Localization and Mapping of an Unknown, Uncooperative and Spinning Target for Spacecraft Proximity Operations*. PhD thesis, Massachusetts Institute of Technology, 2013.

[22] R. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[23] H. Mishra and P. Schmidt, "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)*, 2017.

[24] K. Yoshida, "Space robot dynamics and control: To orbit, from orbit, and future," in *Robotics Research* (J. M. Hollerbach and D. E. Koditschek, eds.), (London), pp. 449–456, Springer London, 2000.

[25] D. N. Nenchev and K. Yoshida, "Impact analysis and post-impact motion control issues of a free-floating space robot subject to a force impulse," *IEEE Transactions on Robotics and Automation*, vol. 15, pp. 548–557, Jun 1999.

[26] C. Ott, A. Kugi, and Y. Nakamura, "Resolving the problem of non-integrability of nullspace velocities for compliance control of redundant manipulators by using semi-definite lyapunov functions," in *2008 IEEE International Conference on Robotics and Automation*, pp. 1999–2004, May 2008.

[27] F. Bullo and R. M. Murray, "Tracking for fully actuated mechanical systems: a geometric framework," *Automatica*, vol. 35, no. 1, pp. 17–34, 1999.

[28] B. Paden and R. Panja, "Global asymptotically stable pd+ controller for robot manipulator," vol. 47, pp. 1697–1712, 06 1988.

[29] J. Artigas, M. D. Stefano, W. Rackl, R. Lampariello, B. Brunner, W. Bertleff, R. Burger, O. Porges, A. Giordano, C. Borst, and A. Albu-Schaeffer, "The oos-sim: An on-ground simulation facility for on-orbit servicing robotic operations," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2854–2860, May 2015.