

EXPERIMENTAL SETUP AND TESTS' RESULTS FOR UNCOOPERATIVE OBJECTS CAPTURE AND MANOEUVRING WITH ROBOTIC ARM

R. Benvenuto⁽¹⁾, M. Lavagna⁽¹⁾, M. Schlotterer⁽²⁾, S. Theil⁽²⁾

⁽¹⁾ Politecnico di Milano, Department of Aerospace Science and Technologies, Via La Masa 34, 20156 Milano, Italy
Email: michelle.lavagna@polimi.it

⁽²⁾ DLR Institute of Space Systems, GNC Systems Department, Robert-Hooke-Str.7, 28359 Bremen, Germany
Email: stephan.theil@dlr.de

ABSTRACT

Active Debris Removal (ADR) and On-Orbit Servicing (OOS) are current research and development topics which are dealing with - either fully or partially - uncooperative orbiting objects to be approached and captured autonomously. For both scenarios, a viable solution is a robotic arm aboard the chaser which is in charge of the mechanical connection between the two objects. At the Department of Aerospace Science and Technologies of Politecnico di Milano, a validated software tool was developed to describe the multi-body dynamics involved in ADR/OOS scenarios and to enable fast analysis and guidance and control law design and testing. The tool was exploited to set up a robotic arm add-on on the Test Environment for Applications of Multiples Spacecraft (TEAMS) at the GNC Systems Department of the Institute of Space Systems of DLR in Bremen. Both numerical and experimental results are presented and discussed in the paper, to highlight drivers and requirements for arm-based mechanism design within the ADR/OOS scenarios, driven by GNC performance.

1 INTRODUCTION

Nowadays Active Debris Removal (ADR) and On Orbit Satellite Servicing (OOS) are hot topics in space engineering research: plenty of challenges, they deal with either fully or partially uncooperative orbiting objects to be approached and captured autonomously by another space vehicle. The high level of autonomy required by ADR and OOS tasks, defines new challenges for Guidance, Navigation and Control (GNC) systems: these missions cannot be tele-operated and ground-controlled due to communications delays, intermittence, and limited bandwidth between the ground and the chaser.

Even being different, ADR and OOS scenarios may present some commonalities as soon as the focus is on proximity GNC and on mechanisms to implement the chaser-target mechanical connection: indeed, to settle on board the chaser a robotic arm in charge of the whole mechanical connection between the two objects may be

seen as a winning solution for a multi-mission platform design; by tailoring the mission, changing the end effector and finely tuning the GNC, capture and connection with the target can be safely put in place all over the operational time window, either for disposal or servicing activities.

Several studies were carried out on the arm dynamics and its interaction with the non-cooperative vehicle, for example by Seweryn et al. [1] at ESA or by Rank et al. [2] in the framework of DEOS, which represent one of the most advanced studies on autonomous OOS mission today. Only a small handful, recently, considered the reference scenario of a massive satellite capture such as Envisat. Reiner [3], for example, developed in 2016 a simulation environment for GNC development using Modelica software and divided the control phases during capture and disposal. Wieser et al. [4] studied the Envisat capture in the framework of ESA e.Deorbit mission study [5], by focusing on the interaction between the gripper and Envisat and the choice of grasping interface with respect to expected loads.

At Politecnico di Milano – Department of Aerospace Science and Technologies (PoliMi-DAER), a validated software tool was developed to describe the multi-body dynamics involved in ADR/OOS scenarios and to enable fast analysis and guidance and control law design and testing [6]. The tool, being also well suited as a functional simulator for hardware-in-the-loop (HIL) test facilities implementation and tuning, was exploited to set up a robotic arm add-on on the TEAMS facility at the Institute of Space Systems of DLR in Bremen, Germany, to eventually run HIL simulation for ADR/OOS GNC testing. The Test Environment for Applications of Multiple Spacecraft is a laboratory for simulating the forces and torque free dynamics of satellites on ground, using air cushion gliding vehicles [7]. A robotic arm add-on was added to one vehicle to perform autonomous capture and manoeuvring of a second vehicle.

The experiment consisted of two vehicles, the chaser being actively controlled and the target being passive and slowly rotating about its z-axis. Coordinated control

strategies were used to capture the target. The control task for the chaser vehicle was to rendezvous with the target and synchronizing with its rotational motion. The gripper was controlled to grasp the capture interface with a constant pre-selected torque, which was also used as feedback for successful capture and to trigger the next guidance phase. Finally, the composite system was stabilized using both retrieval and rigidization of the arm as well as the chaser's on-board thrusters.

The paper details the work done to implement the breadboard on the DLR TEAMS facility, to run ADR/OOS simulations based on the robotic arm mechanism, to develop the arm control software and arm GNC, by exploiting the PoliMi-DAER software tool. In section 2 the multibody dynamics simulation environment and the implemented GNC laws are detailed followed by the analysis of numerical simulations output for the Envisat reference scenario. In section 3 the real-time hardware-in-the-loop test bed is presented and test results are discussed. Finally, in section 4, conclusions are drawn.

2 MULTIBODY DYNAMICS AND GNC SIMULATIONS

2.1 Multibody dynamics simulator

The multibody simulation tool was developed to reliably model the multibody dynamics involved in the above-mentioned ADR and OOS scenarios, effectively serving as a tool to support system design and enabling guidance and control laws implementation, testing and validation [6], [8]. The tool provides a fast and accurate simulation environment to describe multiple bodies six degrees of freedom dynamics, possibly linked by different flexible/rigid connections and including flexible appendages, fuel sloshing and a detailed environmental model. Fully integrated in Matlab/Simulink, the toolbox is partly based on SimMechanics multibody software: using this platform allowed to incorporate in one environment all the elements essential in numerical simulations of robotic manipulators and platform orbital/attitude dynamics. SimMechanics solver works autonomously and creates the mechanical model from the blocks of the system. The system of generalized equations of motion, represented in Equation 1 in the most general form to account for presence of flexible and rigid elements, is then constructed and solved in Simulink:

$$[\mathbf{M}(\mathbf{q})]\{\ddot{\mathbf{q}}\} + [\mathbf{D}(\mathbf{q}, \dot{\mathbf{q}})]\{\dot{\mathbf{q}}\} + [\mathbf{K}(\mathbf{q})]\{\mathbf{q}\} + \{\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\} \quad (1)$$

$$+\{\mathbf{G}(\mathbf{q})\} = \{\mathbf{Q}_{GNC}\} + \{\mathbf{Q}_{dist}\}$$

where \mathbf{q} is a state vector and includes satellite states and manipulators states, \mathbf{M} is a mass matrix, \mathbf{D} a damping matrix, \mathbf{K} stiffness matrix, \mathbf{C} is a Coriolis/centripetal

force vector, \mathbf{G} is a gravity force vector, \mathbf{Q}_{GNC} is a vector of generalized control forces acting on both satellite and manipulator, \mathbf{Q}_{dist} is a vector of generalized disturbing forces. The software is implemented as a combination of Matlab functions and Simulink/SimMechanics building blocks libraries. SimMechanics uses relative coordinates frames: bodies are initially given zero degrees of freedom, which are added by connecting joints to them. Therefore, few configuration variables and constraint equations are required. The drawback of this approach is a dense mass matrix, which contains the constraints implicitly. The structure of dynamics equations mostly depends on the choice of joints, the resulting differential algebraic equations (DAE) are transformed into ordinary differential equations (ODE) and solved with Runge-Kutta methods, after auto-coding it in C++ to improve time performances. SimMechanics also allows running analysis both in forward and inverse dynamics with the same block diagram, a useful feature to create inverse dynamics models and to simulate robotic systems.

The flexibility of manipulators and appendages, such as solar arrays or antennas, also needed to be taken into account for reliable dynamics and control simulations. These elements were modelled as chains of rigid bodies and viscoelastic joints, using lumped parameters method [9]. Each beam can be discretized with an arbitrary number of elements and axial, bending and torsional stiffness can be taken into account. Reactions forces and torques on the spacecraft body are the generalized forces/torques calculated at the interface and depend on the interface element (joint). In the case of robotic manipulators, joints can be externally controlled. In Figure 1, the model that was used for full-scale robotic capture simulations is depicted: arm and appendages discretization elements are shown.

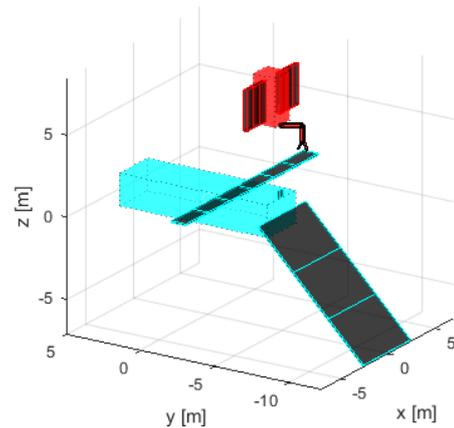


Figure 1: Robotic arm capture model, showing arm and appendages discretization elements

The modelled system dynamics are subject to the full

range of forces and torques expected in Earth orbit: the spatial motion of the system was studied in non-uniform Earth gravity field, under the action of chaser thrusters (when applicable), aerodynamic drag and solar pressure, which were taken into account as external perturbations on all the elements composing the system, both flexible and rigid. Propellant sloshing was also taken into account and modelled using the Abramson equivalent mechanical model [10] as a combination of a damped pendulum and a fixed mass.

A software verification and validation (V&V) activity was performed through test cases and bench-marking with analytical cases, before the execution of experiments. The V&V activity was carried out in an incremental bottom-up way, by first verifying single blocks implementation and validating dynamics models by using conservation laws, secondly by verifying software functionalities implementation and finally by analysing some relevant test cases [8].

2.2 GNC for autonomous capturing and manoeuvring

A closed loop control was implemented to simulate the whole target capturing and manoeuvring. The general scheme is given in Figure 2. In particular, during the operations, the GNC system has to face, at the same time:

- the chaser or stack position and attitude control and the trajectory planning;
- the robotic arm control;
- the uncertain relative target/chaser motion and the external environmental perturbations;
- the flexibility of different elements and connections and their vibrations damping;
- the correct thrust vector pointing performed through chaser or stack attitude control system (ACS).

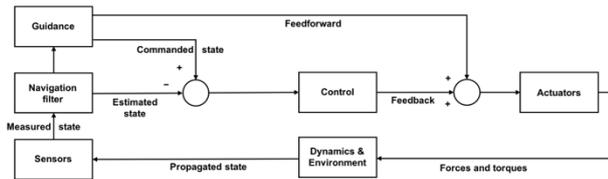


Figure 2: General GNC block diagram

As the actuators are concerned, thrusters were assumed to provide full three-dimensional translation capabilities and their dynamics were modelled as first order systems. A distinction was made between main engines for de-orbiting and reaction control system (RCS) for fine position control or attitude control. Reaction wheels for ACS were also modelled as first order elements and inserted in the mechanical system as rigid bodies connected to the chaser main body.

Detailed vision sensors simulations as well as on-line estimation algorithms for target inertial and dynamics characteristics were not part of the work. Only functional models were used for sensors, by approximating target real measurements using accuracies of real sensors, estimation algorithms delay and noise models. The same was done for chaser state measurements, by introducing similar models to estimate chaser position and velocity as well as attitude quaternion and angular velocity. Navigation filters (i.e. Kalman filters) were implemented to take as input these simulated measurements and estimate the actual state to be used by guidance and control.

The feedback controller for translation was implemented as a cascaded position-velocity controller: by implementing it in a cascaded outer-inner loop fashion it was possible to use the same controller and, depending on the phase, follow a reference trajectory or a reference velocity bypassing the outer loop. Only certain phases required, in fact, fine position control (for example the robotic capture phase) while for others only a velocity control was needed (for example during the de-orbiting phase). This concept was derived by the abovementioned Reiner's work [3]. The attitude controller was also implemented using the two degrees of freedom approach: the outer loop was set as a quaternion error feedback controller, while in the inner loop an angular rate controller was implemented. The quaternion feedback controller used for the simulations is a regulator centred around the quaternion error feedback of the vehicle attitude, inspired by the work of Wie et al. [11]. The control input \mathbf{u} has the form in Equation 2:

$$\mathbf{u} = \boldsymbol{\omega}_c \times \mathbf{I}_c \boldsymbol{\omega}_c - \mathbf{D} \boldsymbol{\omega}_c - \mathbf{K} \mathbf{q}_e \quad (2)$$

where \mathbf{q}_e is the error quaternion and $\boldsymbol{\omega}_c$ the chaser angular velocity. The first term is used to annul the gyroscopic effects of a dense inertia matrix. \mathbf{K} and \mathbf{D} are controller gains adapted on the configuration and selected, as a first guess, to be multiple of the spacecraft inertia and related to controller bandwidth ω_{BW} and damping factor ξ by the relations expressed in Equations 3 and 4:

$$\mathbf{K} = \omega_{BW}^2 \mathbf{I}_c \quad (3)$$

$$\mathbf{D} = 2\xi \omega_{BW} \mathbf{I}_c \quad (4)$$

In the guidance block, the reference trajectory was built and the navigation output was used to trigger different mission phases and switch control modes in an event driven fashion. In general, five guidance phases are distinguished:

- the chaser reaches a no relative motion condition with respect to the target, at a safe fixed distance;
- the arm is deployed (the end effector is moved to a certain point) and the chaser final approach is executed towards the grasping point: for this

manoeuvre, a straight-line trajectory (in relative frame) was adopted to ensure obstacle avoidance;

- the contact and grasping part in which the gripper is controlled using torque feedback to keep a constant grasping force (*rigidization*);
- de-tumbling and movement of the captured target towards the mechanical locking with the chaser.
- de-orbiting burn with orientation of the thrust in the anti-velocity direction and compensation of torques induced by misalignments between the thrust vector and the stack centre of mass.

The guidance block was also built, as visible in Figure 2, in order to provide feedforward translation control that was used either to improve the performance of the feedback controllers (devoted in this case just to the minimization of disturbances and handling of uncertainties). SimMechanics software automatically gives the possibility of using the built-up mechanical model in inverse dynamics mode: therefore, a feed-forward controller could be included to increase the accuracy and robustness of trajectory tracking. This was useful only when a fine position control was needed, for example during final approach, and the whole chaser-arm system was included in the inverse dynamics model.

During autonomous ADR/OOS operations, the chaser is controlled relative to the target in position and attitude and also needs to compensate for the arm motion. In the case of tumbling targets, the chaser GNC needs to be able to autonomously coordinating and synchronizing the chaser angular motion with the target one. The approaches to compensate the base motion due to arm movements, typically fall into two categories:

- use fixed-base arm control strategies but maintain the attitude of the vehicle using thrusters or reaction wheels;
- let the vehicle drift but modify the path of the arm to compensate for the base motion.

The first approach is called *coordinated control* and has the advantage of decoupling the manipulator from the satellite control but at the cost of increased fuel/power consumption [12]. The latter method, called *internal motion control*, uses less power but requires a more complex strategy for controlling the arm [13]. A coordinated control between the chaser and the arm was preferred in this work, dealing with fast tumbling targets. In the post-capture phases a coordinated control between the arm and the chaser is also needed to stabilize and de-tumble the stack, to be able to later carry out the needed de-orbiting operations. By using a coordinated control strategy, the arm control is decoupled from the chaser control: the chaser vehicle is responsible of rendezvousing with the target and synchronizing with its rotational motion, of compensating for arm reactions and, after capture, for stack reactions.

The arm trajectory can be designed off-line or constrained by the final state of the end-effector/gripper which should be the same as the grappling interface at the moment of contact. The chaser guidance feeds forward a signal to the arm PID controller and the manipulator can be controlled in joints coordinates or Cartesian coordinates (an inverse kinematics solver was included in the manipulator model), depending on the guidance phase, to increase the capture accuracy and reliability. In the second case, the reaction torques on the chaser vehicle are always bigger, to compensate for the arm reactions. Finally, the composite system is stabilized using both the arm retrieval and rigidization and the chaser on-board thrusters.

2.3 Simulations results and analysis

2.3.1 Reference Scenario: Envisat disposal

The reference mission for this study was based on the Envisat spacecraft capture and controlled re-entry and it was in line with the baseline of e.Deorbit mission [5], the first ESA's system level study on ADR. In the e.Deorbit study, Envisat was selected as main candidate target to be captured and de-orbited by a chaser satellite that will need to perform these operations safely, autonomously and propellant-efficiently.

Envisat is an ESA owned inactive large satellite, which was launched in 2002 and provided ten years of precious environmental data. Unfortunately, it suffered a major anomaly in 2012, close to the end of its operational life, resulting in loss of communication before the chance of performing any end of life manoeuvre. It is currently orbiting, uncontrolled, in a near-polar near-circular sun-synchronous orbit (SSO) at approximately 760 km of altitude. Envisat large size and mass (> 7 tons) and the crowded region it is orbiting in, made it a primary target for ADR, given the high risk and high chance of collision and the interest for future exploitation of its orbit. Radar measurements performed on Envisat in 2013 showed that its main attitude motion is a rotation of approximately 3.5 degrees per second around its orbital angular momentum vector, with on top, a slight tumbling around its other body axes [14]. ESA is currently analysing these measurements data to better predict the future evolution of its attitude motion, being the 2022 the presumed mission date. This sets a strong requirement for any capture/removal system to be able to deal with this tumbling and its uncertainty. Because of this uncertainty, Envisat angular motion was assumed, as in a previous work at DLR carried out by Vromen et al. [15], rotating at 3.5 degrees per second around its major axis with an axial precession at 0.2 degrees per second around the orbital momentum axis (z-axis) and an angle of 30 degrees between the two axes.

A reaction control system (RCS) was assumed to be

composed by 2 x 100 N thrusters on each long side of the chaser, giving a total torque of 600 Nm for attitude control during stack de-orbiting burn. The knowledge of grasping interface position and orientation were assumed to be known with a precision of 1 centimetre and 0.5 degrees, respectively.

The arm was designed with four degrees of freedom (joints) and two links plus gripper: two shoulder joint (J1 along arm axis and J2 perpendicular), one elbow joint (J2) between the two links and one gripper joint (J4) between the second link and the gripper.

In Table 1, main simulation parameters are reported:

Table 1: Simulation parameters

Parameter	Value			
Chaser mass at capture [Kg]	1500			
Target mass at capture [Kg]	7900			
Target angular velocity [deg s ⁻¹]	3.5			
Initial orbit altitude (circular) [km]	760			
Orbit inclination (SSO) [deg]	98.4			
Arm mass [kg]	15			
Arm links length [m]	1.5	1	0.5 (Gripper)	
Arm DOF [#]	4			
Arm forward reach [m]	3			
Joints mass [kg]	2			
Joints maximum velocity [deg s ⁻¹]	0.5	10	10	30 (Gripper)
Control set-point rate [Hz]	10			
Joints angular accuracy [deg]	0.3			

2.3.2 Target capture simulation

The following guidance phases were defined:

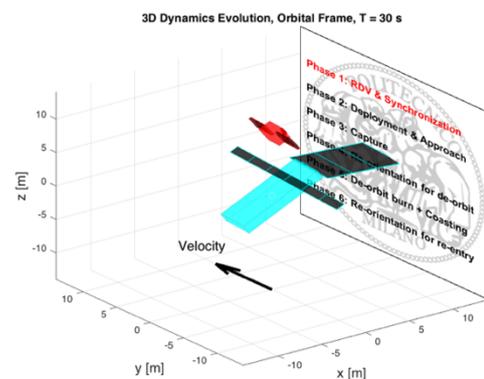
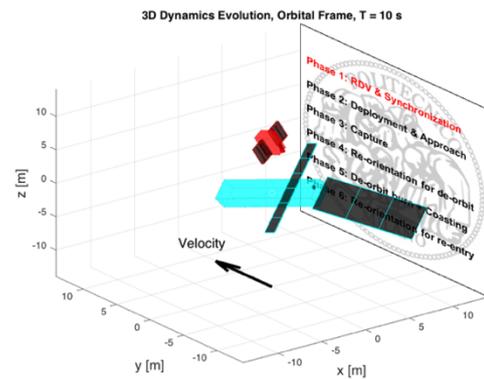
1. Rendezvous and motion synchronization: to synchronize chaser motion with Envisat tumbling at a fixed distance (10 metres).
2. Final approach and arm deployment: final approach with constant velocity along Envisat angular

momentum axis; arm deployment using constrained Cartesian scheme to minimize error between gripper and docking interface positions, while attitude of the gripper is constrained by the attitude of the docking port.

3. Gripper closure, i.e. capture, and rigidization.

The reference trajectory for the final approach of the chaser towards the target was constructed, as mentioned, in order for the docking axis to be aligned with the spin axis: it assumes a straight trajectory along the target spin axis with a constant velocity (0.1 metres per second) until a distance of around 4 metres from the target. Here the chaser no longer moves towards the target but keeps tracking the spin axis. The arm during was deployed the translational approach, by tuning both the chaser and joint velocity, in order to be able to track the docking position at the end of the approach by steering the arm through Cartesian control.

The results demonstrate the performances of the implemented coordinated control and investigate the cost in terms of ΔV and propellant for relative manoeuvring. Figure 3 reports the 3D visualization of the phases considered for the dynamics and control analysis.



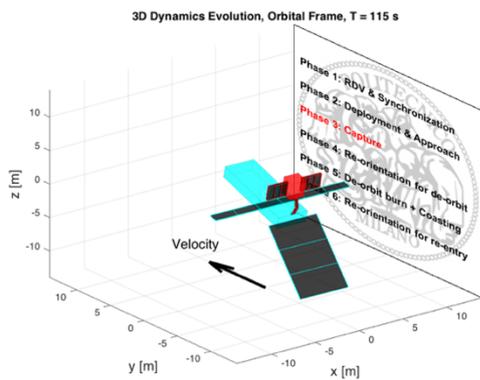
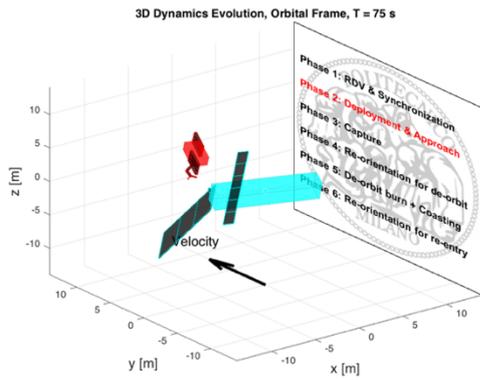


Figure 3: 3D visualization of phases 1 to 3, in orbital frame

Figure 4 and 5 show, respectively, the translational and rotational errors and highlight the control being quite effective in rapidly annulling the chaser state vector errors.

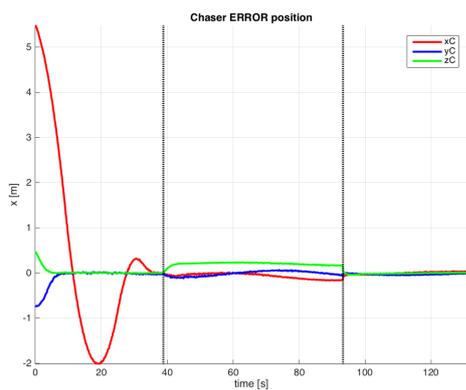


Figure 4: Chaser relative position errors, phases 1 to 3

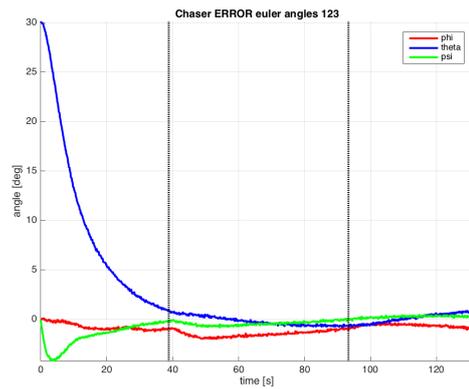


Figure 5: Chaser relative orientations errors, phases 1 to 3

Figure 6 and 7 highlight quantities related to the arm joints control. It can be appreciated how the arm configuration is feasible and the errors annul rapidly. The torques on each joint are affordable as well during these phases.

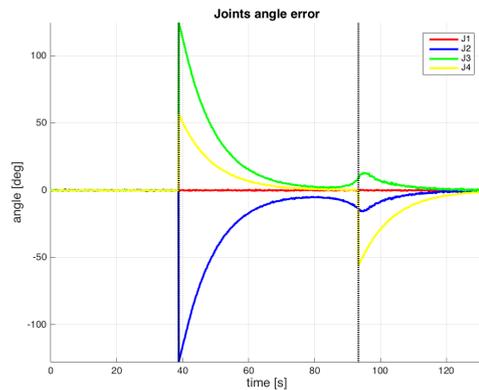


Figure 6: Arm joints error, phases 1 to 3

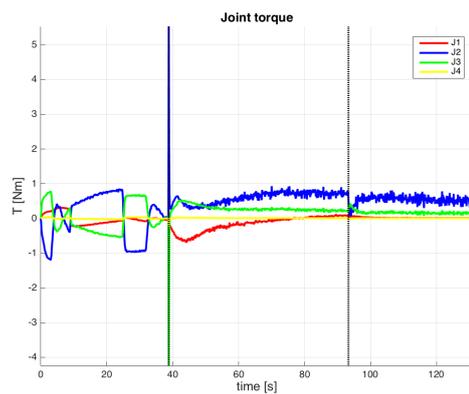


Figure 7: Arm joints torques, phases 1 to 3

Finally, Figure 8, wraps up the budgets in terms of ΔV and fuel mass demand (supposing a specific impulse of thrusters of 320 seconds) to perform the guidance

requested to answer the approach, synchronization and capture phases constraints.

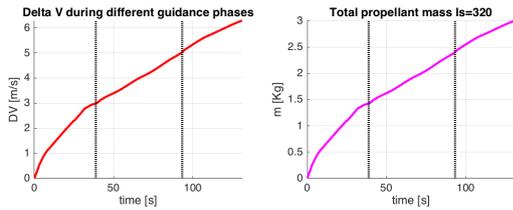


Figure 8: Control costs and budgets, phases 1 to 3

2.3.3 Stack disposal simulation

The stack disposal operations are analysed in this paragraph and the following guidance phases were defined:

4. Re-orientation for de-orbiting: the arm is retracted and stiffened and the stack is oriented in configuration where the main de-orbiting thrusters are correctly aligned in the orbital plane.
5. De-orbiting plus coasting: the burning and coasting time spans are reduced to 50 plus 50 seconds for clarity of results representation, still keeping the generality of results.
6. Re-orientation for re-entry and spin-up: before re-entry the stack is reoriented to increase the drag coefficient and spun-up to 2 degrees per second to reduce the re-entry path drift due to generation of lift forces.

Each re-orientation manoeuvre (phase 4 and 6) was split into two parts:

- the arm is reoriented to change the chaser relative attitude with respect to the target and the chaser absolute attitude is left unconstrained (attitude control off) to limit the control torques on the arm joints;
- afterwards, the arm joints are stiffened and the chaser is commanded to give a certain attitude to the stack.

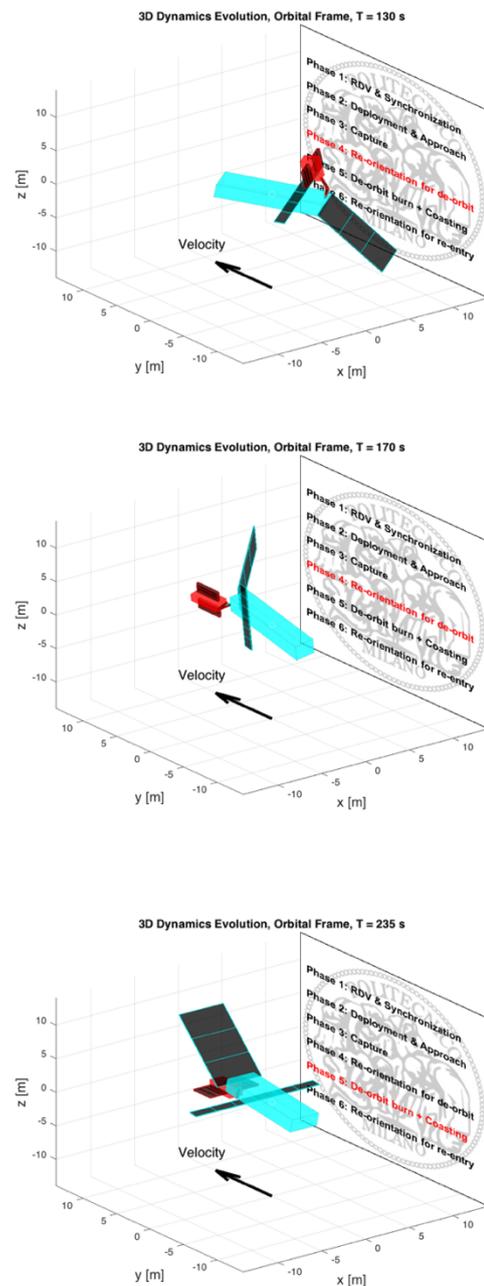
Now, a remark is necessary on the possible attitude control strategies that could be carried out during de-orbiting, both in terms of absolute stack orientation and relative orientation between chaser and target. The chaser main thruster need to be oriented opposite to V-bar for braking purposes and at the same time the communication and power generation must be guaranteed to meet the operations requirements. Two options might, then, be envisaged:

- either a relative configuration that minimize the control torque during de-orbiting burn
- or a fixed stack configuration and consequent compensation of torques due to misalignment with the composite COM (potentially helped by thrust vector control).

The trade-off, obviously, depends on the orbit

constraints, the requirements for power generation and communication and the platform configuration (thrust vector control and gimbals, clamping mechanisms, solar arrays, antennas and their respective DOFs, etc.). The second option was selected here, being Envisat orbit a dawn-dusk sun-synchronous orbit and being the most conservative and costly case. In this case, gimbaling the thrusters for vector control, would beneficially decrease the attitude control cost, but was avoided for the same rationale.

Figure 9 depicts the 3D visualization of the phases considered for the disposal dynamics and control analysis.



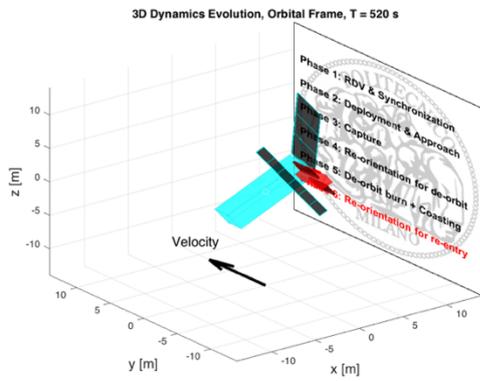
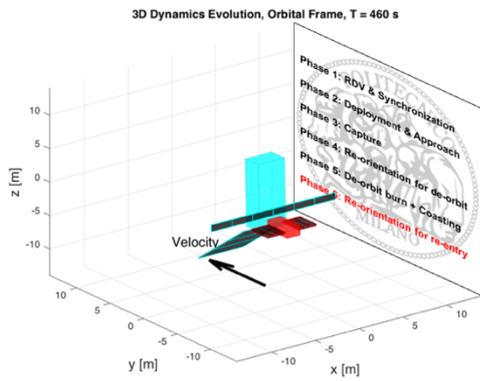
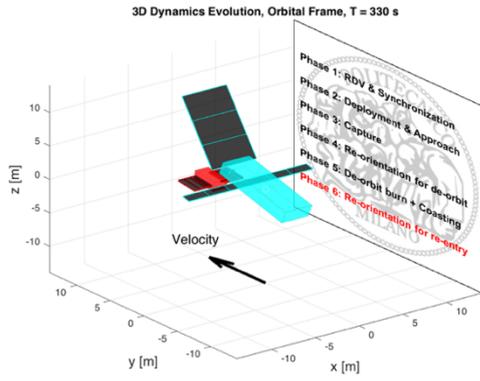


Figure 9: 3D visualization of phases 4 to 6, in orbital frame

Finally, as before, Figure 10 reports the main budgets for the phases discussed in this paragraph. As mentioned above, the fifth phase was here limited to 50 seconds burning plus 50 seconds coasting, for the sake of results representation.

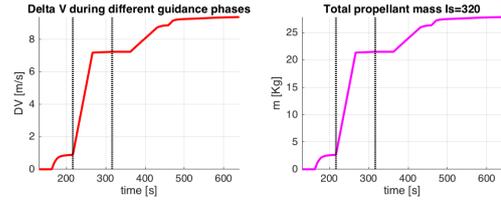


Figure 10: Control costs and budgets, phases 1 to 3

3 HIL TESTING ON TEAMS

3.1 Test set-up

The control loop performances were tested using real-time hardware-in-the-loop (HIL) simulations: the robotic arm multibody tool, being well suited as a functional simulator, was exploited to set up a robotic arm add-on on the TEAMS facility at the Institute of Space Systems of DLR in Bremen, Germany.

Low friction tables can reproduce the force and torque free dynamics of spacecraft on ground, using air cushion gliding vehicles. The DLR's TEAMS is one state of the art facility of its kind: the TEAMS3D vehicles glide frictionless through air pads, creating a thin air film, on two 4 x 2.5 meters smooth granite tables, therefore representing two force free translational and one torque free rotational degrees of freedom (DOF). The TEAMS5D vehicles extend the facility capabilities to five DOF by adding a three DOF attitude platform on top of the vehicles; nevertheless, this study only took advantage of the TEAMS3D vehicles. The vehicles are equipped with air tanks, feeding both the air cushions and the cold gas thrusters, and with a QNX real-time on board computer and Wireless LAN for communications, software uploads and data downloads. A *DTrack* infrared tracking system composed of five cameras is the main sensor for position and attitude measurements and is able to precisely track the pose at a frequency of 60Hz, using markers on top of vehicles. To control the position and attitude, the vehicles are equipped with 6 proportional cold gas thrusters (CGT), supported by 6-8 bar pressurized air, their maximum thrust being about 46 mN, each. The algorithm for thrusters control is treated in [7]. The characteristics of TEAMS3D vehicles positioning and control are reported in Table 2.

Table 2: TEAMS3D characteristics

Parameter	Value
Tracking frequency [Hz]	60
Tracking precision - position [m]	1E-3
Tracking precision - angle [deg]	0.1

CGT maximum thrust [N]	46E-3
CGT maximum torque [Nm]	5.3E-3
Vehicle mass [kg]	18.7

A four DOF robotic arm add-on was added to one vehicle to perform autonomous capture and manoeuvring of a second vehicle: the arm is a *Smart Robotic Arm AX Series*, produced by *CrustCrawler Robotics* and weighs around 0.5 kilograms. A battery was added on the vehicle to power the arm at 12 Volts. The robotic arm uses *Robotis' Dynamixel AX-12* servomotors, whose characteristics are reported in Table 3.

Table 3: Servos characteristics

Parameter	Value
Maximum torque [Nm]	1.47
Maximum speed (no load) [rpm]	59
Resolution [deg]	0.29
Maximum communication speed [Mbps]	1

The servos returned feedback on position, velocity, load and temperature and could be commanded through direct angle input, setting maximum velocity and torque. The motor response could be controlled using different settings, not reported here: by tuning these parameters arm jerks and vibrations were avoided. The type of internal controller in the Dynamixel was nowhere specified but based on their behaviour, the assumption of a PID controller was made. The internal controller bandwidth was anyways faster than the outer loop, so no more details are given here on the servos internal control and electronics (more details can be found in [16]). The servos were interfaced with the vehicle through a Simulink S-function using the servos Software Development Kit (SDK). The designed GNC algorithms were developed using Matlab/Simulink and Simulink coder for automatic generation of C code to be uploaded to the on board real-time operating system. The target vehicle was provided with a docking interface for the arm gripper to grasp: in Figure 11 both the arm and the docking interface are shown, integrated on the two vehicles, during a capture emulation.

The simulation environment and control scheme were adapted to the two-dimensional problem: the vehicle dynamics is in this case described by three uncoupled double integrators (two planar positions on the table and one for rotation) and therefore linear: the arm reactions are directly taken into account in the mechanical model and seen as perturbations in the control model. The arm

could again be controlled in the joint space or in the Cartesian space, depending on the guidance phase, to increase the capture accuracy and reliability: the planar inverse kinematics was simplified and constrained to have the gripper rightfully oriented with respect to the docking interface. The gripper was controlled to grasp the docking interface with a constant pre-selected torque, which was also used as feedback for successful capture and to trigger the next guidance phase: in case of unsuccessful capture a collision avoidance manoeuvre (CAM) was set-up. When the CAM was triggered, the arm was retracted and the chaser acquired a safe position with respect to the target before reattempting the capture. The CAM scheme proved to be effective, even if it was triggered only twice during more than fifty tests on the table, proving the robustness of the control scheme. Finally, the navigation filter was modified to estimate constant disturbances on the vehicles and the controller used direct feedback of the estimated disturbances: this approach, described in [7], proved to be efficient in increasing the targeting precision for approach and grasping.

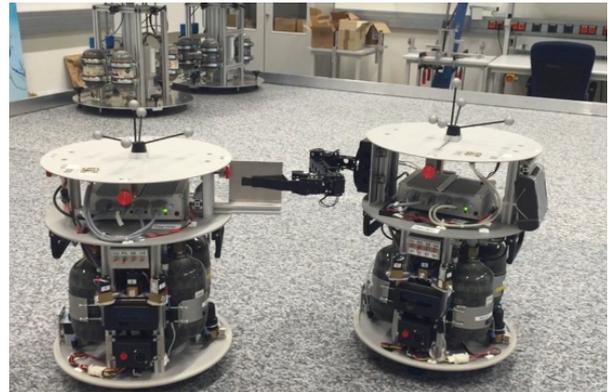


Figure 11: Arm and docking port integrated on the TEAMS3D vehicles, capture emulation

3.2 Test results

The test parameters are reported in Table 4. The control frequency was set to be compatible with the arm communication. In order to emulate the target as passive it was controlled to be in a fixed position and slowly rotating about its axis; at capture occurrence, the target control was switched off. The chaser on board computer received both its reconstructed state and the target one, in order to perform the relative navigation: the control task for the chaser vehicle was to rendezvous with the target and synchronizing with its rotational motion. Two different capture strategies were tested and are discussed below: in both cases the robotic arm was controlled in Cartesian coordinates during the final approach to steer the gripper over the docking interface with better precision. After capture, the gripper control torque was lowered with respect to the maximum

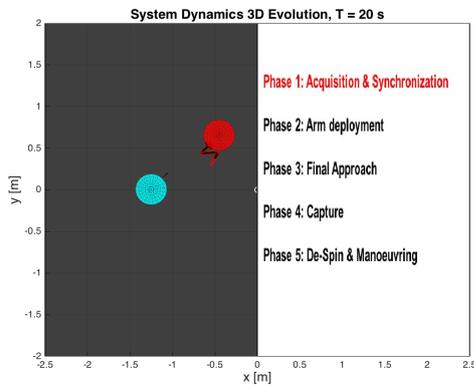
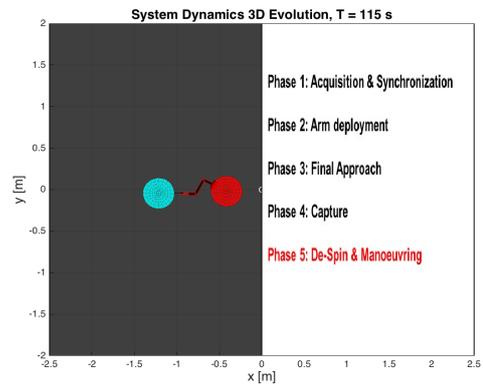
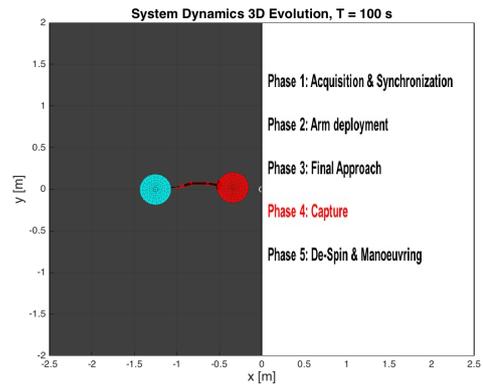
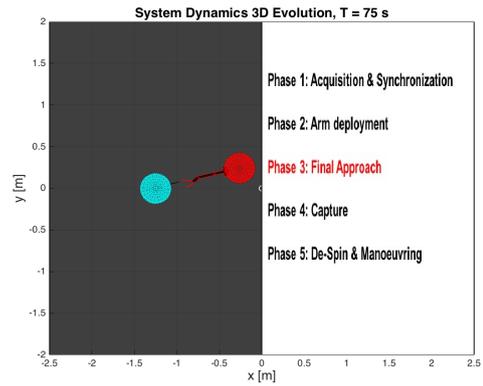
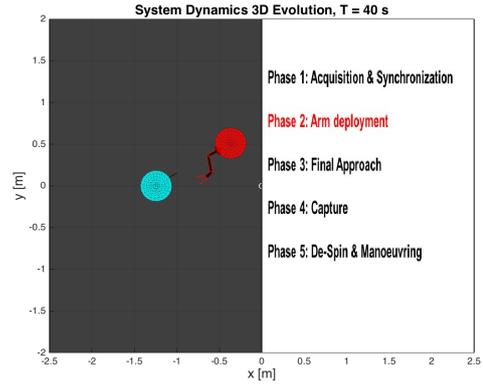
achievable, to avoid servos heating up excessively. Finally, the composite system was stabilized using both arm retrieval and rigidization as well as the chaser on-board thrusters.

Table 4: Test parameters

Parameter	Value			
Target angular velocity [deg s ⁻¹]	0.5			
Chaser approach velocity [m s ⁻¹]	5E-3			
Joints maximum velocity [deg s ⁻¹]	6.7	6.7	6.7	13.2 (Gripper)
Control set-point rate [Hz]	2			
Gripper torque [Nm]	0.8			

The first strategy was to move the gripper over the capture interface by slowly moving the chaser vehicle and steering the robotic manipulator to follow the capture interface. The trajectory was a spiralling trajectory in the table reference frame: the chaser approaches the target with a constant velocity along the rotating capture axis. In Figure 12, the test sequence is depicted. As visible, five main guidance phases were defined:

- acquisition and synchronization
- arm deployment
- final approach (constant approach velocity along docking axis, i.e. in plane “spiralling trajectory”)
- capture
- de-spin and manoeuvring (the test case presented here prescribed for the stabilization of the stack, a fixed position at capture with null angular velocity).



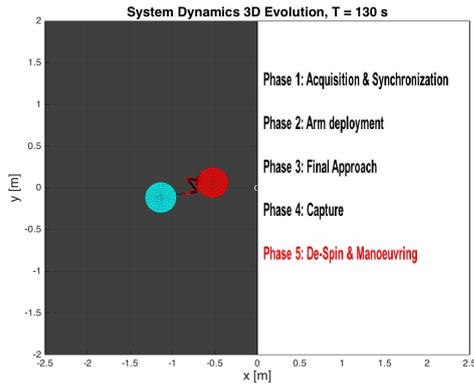


Figure 12: Frontal capture test sequence on TEAMS

In Figure 13 and 14, the joints angles and torques are reported, for the test under consideration: as visible the joints are controlled with very little jerk to steer the position of the gripper, depending on the target interface position. The effectiveness of the gripper torque control after capture is clearly visible in the second figure.

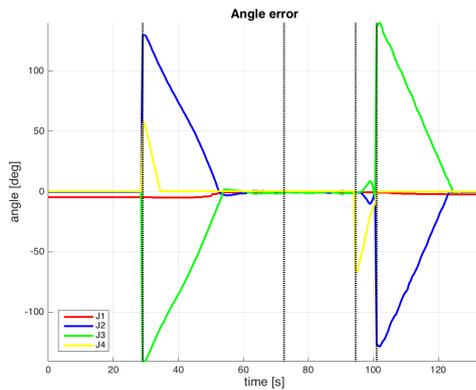


Figure 13: Arm angles error during different guidance phases (identified by black dotted lines)

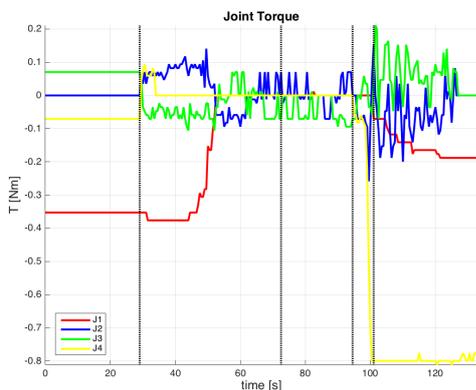


Figure 14: Arm joints torques during different guidance phases (identified by black dotted lines)

In Figure 15 and 16, the error between the desired

gripper position and grasping interface position and target angular velocity are presented. The error at capture is less than 2 centimetres, allowing for a precise interface targeting by the gripper. As visible in the second figure, the target angular velocity is strongly perturbed at capture, due to the positioning of the docking plate inside the gripper during its closure. This led to high jerks at capture, which were soon damped by the arm retrieval and the chaser control. A total impulse of less than 10 Ns was required for the thruster to perform all the phases.

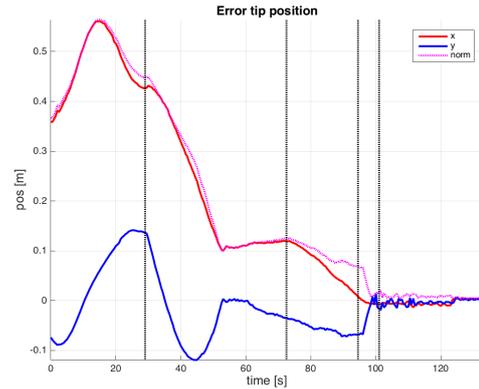


Figure 15: Error between desired gripper position and grasping interface position, during different guidance phases (identified by black dotted lines)

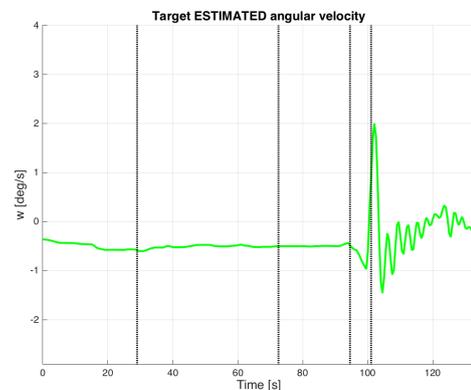


Figure 16: Target angular velocity during different guidance phases (identified by black dotted lines)

The second implemented strategy was, to move the gripper over the capture interface by steering the robotic arm and keeping a fixed relative position between target and chaser vehicles. Therefore, only four main guidance phases were defined merging the final approach with the arm deployment: in this case, in fact, a fixed positioning was kept during capture without the spiral trajectory foreseen in the previous case. This strategy would be better suited if the docking interface was not

aligned with the target angular momentum, as in the general 3D case. No particular differences in the performances were remarked, except for the obvious decrease in the capture time from acquisition and the consequent decrease in thrusters required impulse. For this reason, only the test sequence is reported below in Figure 17, where real recorded images are depicted, as published on the YouTube channel of the GNC Department of the DLR Institute of Space System, Bremen, Germany^{1,2}.

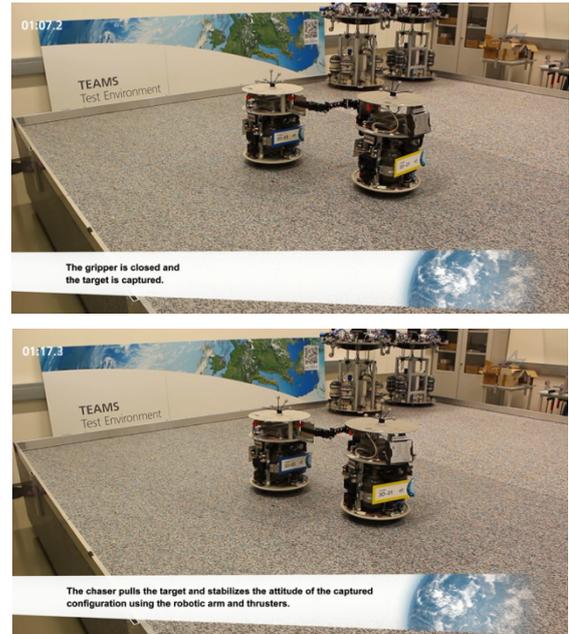


Figure 17: Lateral capture test sequence on TEAMS

4 CONCLUSIONS

The steadily increase of the space debris population around Earth is threatening the future of space utilization for both commercial and scientific purposes: both a disposal policy, to properly manage new space vehicles end-of-life, and active remediation are necessary to guarantee safe operational life time for current and future space systems in Earth orbit. The problem of remediation includes different aspects, the capture of the target and its manoeuvring being the most delicate phases. While Active Debris Removal (ADR) deals with inactive spacecraft disposal from operational orbits, On-Orbit Satellite Servicing (OOS) is aimed at extending the lifetime of operational satellites through refuelling and maintenance. The development of a chaser satellite able to perform autonomous servicing/removal mission is a complex, multidisciplinary and challenging task: such missions require the application of many sophisticated technologies and reliable lightweight manipulators capable of capturing non-cooperative or partially cooperative tumbling objects which are not equipped with dedicated docking ports and whose physical and dynamics characteristics are uncertain and not known a priori. Validated simulation tools are needed to develop these technologies, as well as to design future missions and systems.

The developed multibody simulator allowed to assess the capture techniques feasibility and to describe the overall dynamic behaviour, including the unforeseen dynamics arising from the interaction between two isolated bodies, becoming a single multibody system to manoeuvre. The tool was mainly used to drive the

¹<https://www.youtube.com/watch?v=KKfG85en8r4>

²<https://www.youtube.com/watch?v=pLj3vjOcu3c>

system design and to support the guidance, navigation and control (GNC) design and testing and ensured flexibility in different scenarios definition. The control performances were demonstrated through simulation of the reference orbital scenario, i.e. Envisat capture and disposal. The robotic arm simulation tool was also exploited to set up the implementation of a robotic arm add-on on the TEAMS facility at the Institute of Space Systems of DLR in Bremen, to eventually run HIL simulation for GNC testing. A robotic arm add-on was added to one gliding vehicle to perform autonomous capture and manoeuvring of a second uncooperative vehicle. HIL lab tests focused on the arm deployment impact on GNC laws design and on the simulator validation for the robotic arm scenario. Experimental results showed the robustness of the implemented control schemes to safely capture and stabilize the stack.

5 REFERENCES

1. Seweryn, K., Banaszkiwicz, M., Maediger, B., Rybus, T., Sommer, J. (2011, April), “*Dynamics of space robotic arm during interactions with non-cooperative objects*”, in proc. of 11th Symposium on Advanced Space Technologies in Robotic and Automation, ESA/ESTEC, Noordwijk, The Netherlands.
2. Rank, P., Mühlbauer, Q., Naumann, W. & Landzettel, K. (2011, April), “*The DEOS automation and robotics payload*”, in proc. of 11th Symposium on Advanced Space Technologies in Robotic and Automation, ESA/ESTEC, Noordwijk, The Netherlands.
3. Reiner, M. J., (2016, March), “*GNC simulation tool for active debris removal with a robot arm*”, in proc. of 6th International Conference on Astrodynamics Tools and Techniques (ICATT 2016), Darmstadt, Germany.
4. Wieser, M. et al., OHB-System AG, e.Deorbit Mission (2015, May), “*OHB debris removal concepts*”, in proc. of 13th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA 2015), ESA/ESTEC, Noordwijk, The Netherlands.
5. Biesbroek, R. et al. (2013, May), “*The e.Deorbit study CDF study: A design study for the safe removal of a large space debris*”, in proc. of 6th IAASS Conference, Montréal, Canada.
6. Benvenuto, R. & Lavagna, M. (2016, March), “*GNC techniques for proximity manoeuvring with uncooperative space objects*”, in proc. of 6th International Conference on Astrodynamics Tools and Techniques (ICATT 2016), Darmstadt, Germany.
7. Schlotterer, M. & Theil, S. (2010, August), “*Testbed for on-orbit servicing and formation flying dynamics emulation*”, in proc. of AIAA Guidance, Navigation, and Control Conference, Toronto, Canada.
8. Benvenuto, R., Lavagna, M., Cingoli, A., Yabar, C. & Casasco, M. (2014, June), “*MUST: Multi-body dynamics simulation tool to support the GNC design for active debris removal with flexible elements*”, in proc. of 9th International ESA Conference on Guidance, Navigation and Control Systems (ESA-GNC 2014), Porto, Portugal.
9. Benvenuto, R., Salvi, S., & Lavagna, M. (2015, May – June), “*Dynamics analysis and GNC design of flexible systems for space debris active removal*”, in Acta Astronautica, Vol. 110, pp. 247-265.
10. Dodge, F. (2000), “*The new dynamic behaviour of liquids in moving containers*”, at Southwest Research Institute.
11. Wie, B., Weiss, H. & Arapostathis, A. (1989, June – May), “*Quaternion feedback regulator for spacecraft eigenaxis rotations*”, in Guidance, Control and Dynamics, Vol. 12, No. 3, pp. 375-380.
12. Oda M., (1996, April), “*Coordinated control of spacecraft attitude and its manipulator*”, in proc. of IEEE Conf. Robotics and Automation, pp. 732-738.
13. Fernandes, C., Gurvits, L. & Li, Z. X. (1993), “*Attitude control of space platform/manipulator system using internal motion*”, in Space Robotics: Dynamics and Control, pp. 131-163.
14. Virgili, B.B., Lemmens, S. & Krag, H. (2014, May), “*Investigation on Envisat attitude motion*”, in proc. of e.Deorbit symposium.
15. Vromen, S., de Bruijny, F. J. & Mooij, E. (2015, June), “*Guidance for autonomous rendezvous and docking with Envisat using hardware-in-the-loop simulations*”, in proc. of 8th International Workshop on Satellite Constellations and formation flying, (IWSCFF-2015).
16. Mensink, A. (2008, November), “*Characterization and modelling of a Dynamixel servo*”, in EEMCS/Electrical Engineering, Report n. 035CE2008, University of Twente.