GNC SIMULATION TOOL FOR ACTIVE DEBRIS REMOVAL WITH A ROBOT ARM

Matthias J. Reiner

Institute of System Dynamics and Control
DLR - German Aerospace Center
Oberpfaffenhofen, Germany

ABSTRACT

The population of large debris has become a problem in Low Earth Orbit (LEO) and the danger of collisions is high. One proposed solution is Active Debris Removal (ADR, [1]), using a robotic arm mounted on a spacecraft that is able to connect to a debris part and deorbit it. To analyze this approach, the GNC simulation tool for Active Debris Removal with a robot arm was developed. A realistic benchmark scenario based on the capturing of the inactive Envisat satellite was chosen for a simulation study. The GNC simulation tool based on the object-oriented DLR Modelica SpaceSystems library [2] is used to design and simulate the control algorithms, satellite dynamics, kinematics and well as the robot arm control. Simulation results show that the ADR benchmark scenario can be successfully completed using the newly developed control algorithms based on Nonlinear Inversion and robust tuning using a multi-case and multi-objective optimization.

Index Terms— ADR, Modelica, Optimization, Nonlinear Inversion

1. INTRODUCTION

Studies have shown that the population of large objects in Low Earth Orbit (LEO) has become a problem. The danger of collisions is high and important objects are at risk of major damage. One proposed solution for this problem is Active Debris Removal (ADR) [1]. A possible way of realizing ADR is a satellite equipped with a robotic arm (chaser) which offers a very flexible way (compared to using nets or a harpoon) to handle various types of target objects. A gripper or another adequate tool installed on the robot arm flange can connect to almost every inactive satellite or other large debris parts (targets) that could endanger other satellites in the orbit. After a successful capturing, the chaser satellite can then be used to safely deorbit the target by transferring it to a disposal orbit (see Fig. 1).

To achieve this goal, a series of technical challenges has to be solved. An engineering simulation tool is useful to get better insight for the problem and to test out different scenarios. The GNC tool can be used to demonstrate and simulate important steps of the ADR scenario in order to verify and evaluate the satellite GNC and the robot arm control. It was developed at the DLR as part of a European Space Agency (ESA) project.

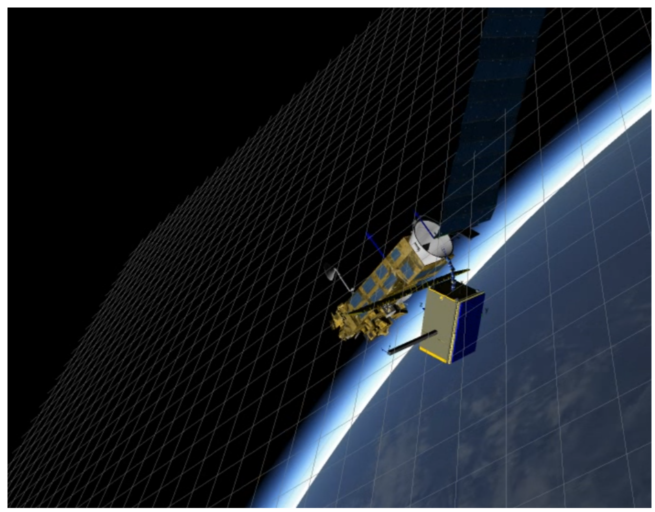

Fig. 1. Visualization of a successful capturing.

The chaser satellite with its robotic arm has to be controlled very precisely during the approaching phase close to the target. The coupled dynamics of the robot arm and the chaser satellite in a LEO environment have to be considered in the GNC approach. In addition, the dynamics of the combined system of chaser with robot arm and target changes drastically after a grip is established. Therefore, the GNC of the chaser satellite has to be able to handle the change in mass and inertia of the combined system. Since rotating targets have to be considered, it is necessary that the satellite has to be controlled simultaneously with the robot arm.

A prerequisite for capturing the target with the robot arm is the estimation of relevant properties and states of the target satellite. Only imperfect sensors are available for this task and uncertainties have to be taken into account. Mass and inertia of the target satellite as well as its rotating movement have to be estimated and are therefore only known within certain bounds. Based on this imperfect estimation, a trajectory has to be planned for the chaser satellite in order to move to an optimal starting position relative to the target for the grasping task. The robotic arm with its gripper tool is then used to establish a connection (grip) between the chaser and the target satellite. The connected system must then be controlled by the GNC of the chaser satellite in order to transfer it to a disposal orbit.

2. THE GNC SIMULATION TOOL SETUP

The developed GNC simulation tool for Active Debris Removal with a robot arm (short: GNC simulation tool) is based on the DLR Modelica SpaceSystems library [2]. The Modelica modeling language [3] offers many features that are helpful for an implementation of the engineering tool.

The language is object oriented and equation based. Modelica environments, such as Dymola [4], include a powerful symbolic engine that allows to handle large and complex nonlinear systems in the form of Differential Algebraic Equations (DAE). Advanced mathematical techniques are used to rearrange and simplify the equations, which are then solved numerically. The end result is C-code that can be compiled as an executable, exported to MATLAB/Simulink as an S-function or to other simulation tools as a Functional Mock-up Unit (FMU) for model exchange and co-Simulation [5].

Modelica models are stored in tree structured code libraries and can be used as subsystems for different models. These models can contain changeable parameters what allows to adapt components for a specific problem. The parameters can also still be changed for the compiled complete model. This allows modifying the resulting models, for example for robustness analysis or controller parameter tuning.

For the GNC simulation tool, various libraries were used developed at the DLR Institute of System Dynamics and Control. The libraries are set up in such a way that they can be combined together using standardized connectors.

Apart from the extensive Modelica Standard library, the most important libraries used for the GNC simulation tool implementation are briefly described in the following paragraphs.

The SpaceSystems library

The Modelica SpaceSystems library (SSL) was built to model and develop advanced control systems for satellites and other spacecraft with flexible structures. In particular, one goal of this library is the generation of nonlinear inverse models for control. The library contains Low Earth Orbit (LEO) space environment models and components. The SSL enables object-oriented, acausal, and equation-based modeling of space system dynamics and its corresponding orbital environments. This in turn allows controller design and verification as well as development of path planning and other algorithms. Details can be found in [2]. A screenshot overview of the SSL components is shown in Fig. 2.

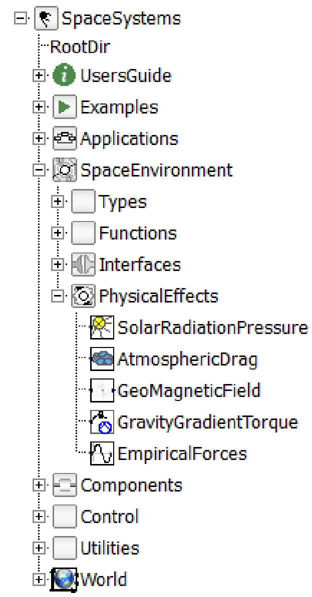

Fig. 2. Screenshot of the Modelica SpaceSystems library.

Robots and RobotDynamics library

The Robots and RobotDynamics library were designed to model serial kinematic robots. The DLR internal RobotDynamics library consists of components for the mechanical design of robots, including flexible elements and powertrains of the robots as well as models for different robot control structures. The RobotDynamics library was developed and refined over many years and used in various projects (e.g. [6, 7]). The Robots library focuses on the efficient and exchangeable implementation of robot kinematics. It also provides algorithms to solve forward and inverse kinematics problems. In addition, the library provides models for the visualization of robots.

DLR Visualization library

The DLR Visualization library provides an advanced, model-integrated visualization tool for Modelica models. It is especially useful in the mechanical, fluid and electrical area. Many components are available for offline, online and real-time animation. The library contains visualizers for basic shapes, CAD files, flexible bodies and surfaces, textures, light, energy and mass flow visualizers, analogue instruments and weather effects. A virtual camera system can be used to define the point of view manually or controlled by simulation. For space applications with a large difference in distances, a logarithmic z-buffer has been implemented. Details are given in [8].

MOPS - Multi-Objective Parameter Synthesis

The optimization software environment MOPS [9] supports a general controller design process, especially in the following three tasks: robust control law tuning, control law robustness assessment and parameter estimation of nonlinear dynamical systems. These tasks can be solved by optimization. MOPS supports features like multi-model/multi-case design problems, parallel computation and Monte-Carlo simulation. The DLR Modelica Optimization library [10] implements most of the MATLAB based MOPS features directly in Modelica.

LEO environmental models

The environmental models for the GNC simulation tool are part of the DLR Modelica SpaceSystems library. They are used for both the chaser and target satellite. The components used as part of the GNC simulation tool are described in the following section. Formulae and additional details are given in [2] and [11].

Global world model

The default "world" model of the Modelica MultiBody library is replaced by the SpaceSystems library default world model. It offers additional options and methods for space environment simulations. The world model handles the global simulation time that is used to calculate planet positions and various transformations. The initial time can be given in calendar- or Julian date format.

The basis coordinate system is chosen to be the Earth Centered Inertial (ECI) coordinate system, which is suitable for near Earth satellite simulations. The world model offers a connector and transformation for the Earth Centered Earth Fixed (ECEF) coordinate system, which is useful for the simulation of objects on Earth, like emitter stations. The world model implements the gravity acceleration for all multi-body objects. Multiple gravity models of different complexity are implemented in the SSL. The EGM96 gravity model [12] is used for the GNC simulation tool. A computational efficient approximation of this model was implemented, which uses terms up to the second degree of the zonal harmonic coefficients of the gravitational potential. In addition, the Moon and sun gravities have been included, considering them as important perturbation factors for the gravity. These are modeled as point gravity. The location of the sun and the Moon is calculated based on the current simulation date using approximation formulae or pre-computed table data.

Gravity gradient torque

The gravity gradient torque is modeled as a torque that acts on the center of mass (CoM) of the chaser and target. The torque is caused by the mass distribution of the body in consideration, and depends on the inertia tensor and the current gravity acceleration vector. The gravity and Gravity gradient torque are functions of the position and the Julian date because of the considered perturbation terms from the planet positions.

Solar radiation pressure

The effect of the solar radiation pressure is modeled as a force element that acts on the CoM of the chaser and target. Shadows caused by the position of the Earth and Moon relative to the sun are considered, using a cylindrical shadow model.

Atmospheric drag

The atmospheric drag is caused by friction with the atmosphere depending on the height of the satellite above the Earth. Like the radiation pressure, it is modeled as a force and torque element acting on the attached body, which is located at the center of pressure. The density of the atmosphere is computed using the NRL-MSISE-00 atmospheric density model [13]. It depends on the current longitude, latitude and height above Earth that can be computed from the current position of the chaser and target.

3. ADR BENCHMARK SCENARIO

The ADR benchmark scenario for the GNC simulation tool is the controlled deorbiting of an inactive satellite (target) using a satellite with a mounted robot arm (chaser). The chaser is equipped with cameras (VIS and IR) as well as a LIDAR system. The focus of the benchmark scenario is to show the capabilities of the GNC simulation tool and the preliminary design of the control algorithms for the chaser satellite including the robot arm.

The chaser and target satellite are modeled using the Modelica libraries described in Section 2. Both are implemented as rigid bodies, flexible and elastic effects are neglected. Additional investigation regarding the influence of flexible deformations and vibrations are planned for a future project.

Target satellite model

The target satellite is modeled after the inactive Envisat. For the simulation, the satellite is assumed as a passive rigid body with a defined rotation movement and orbit. Envisat is a large satellite with a mass of about 7.5 t and a length of about 10 m in addition to a large attached solar array. Figure 3 shows a SimVis visualization of Envisat using CAD data provided by ESA. The visualization is part of the GNC simulation tool.

The initial orbit and rotation state of the target is implemented for the GNC tool using data provided by ESA. As a simplification, the tumbling movement of the main spin axis is neglected before a connection is established between the

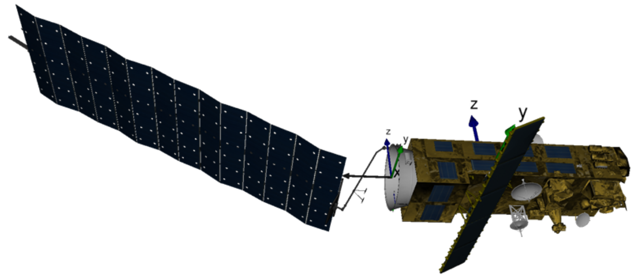

Fig. 3. ADR benchmark scenario visualization of Envisat.

chaser and target satellite by the robot arm. After the capturing, the simulation takes into account the full rigid body dynamics of the target (including tumbling movement).

One important aspects of the benchmark ADR scenario is the large mass and inertia difference of the chaser ($\sim$1500 kg) and the target ($\sim$7500 kg) and a (assumed) rotation speed of the target of 5°/s around the body z-axis.

For this reason it is important to select a suitable grip point on the target satellite. According to ESA information a possible grip location is on the satellite's launch adapter ring. This ring is used to mount the satellite in the launcher vehicle and can withstand large forces. A grip point on the ring was chosen such that the robot arm can reach it without risk of collision of the satellites and the large solar panel of Envisat.

For the simulation model, the grip point is defined as a frame connector to which the robot TCP (Tool Center Point) can be connected. In addition, the relative kinematics for the optimal docking position for the chaser satellite is defined in the target satellite model.

The target satellite model also provides an interface for additional force and toque inputs that act directly on the center of mass. These are used to simulate disturbances: the gravity gradient torque, solar radiation pressure and atmospheric drag or other user-defined forces.

Chaser satellite model

The chaser satellite bus is modeled as a rigid body together with three rigid body models for the reaction wheels. The dynamics of the wheel actuators are modeled as first order elements. Models of the thrusters for the translation of the satellite are considered as dynamic first order approximations. The thrusters are assumed to provide full 3D-translation capabilities.

For the chaser satellite, the dynamic coupling of the mounted robot structure and the motor inertias of the robot are considered. Modelica allows for efficient implementation of the gyroscopic coupling effects of (motor-) inertia on connected bodies. More details are given in [14].

The chaser satellite model contains sensor models to compute the angular rate and quaternion as well as the position and velocity of the satellite. These values are not used directly by the GNC, but provide ideal references for the inertial measurement unit (IMU) simulation.

Robot arm model

The robot arm kinematics is roughly based on a scaled DLR Light Weight Robot (LWR, [15, 16]) with only three axes. The powertrains and structure of the robot are implemented using the DLR Modelica Robots and RobotDynamics libraries. Using geometric scaling laws the geometry, mass and inertia tensors of the arm have been changed to achieve an arm length of 2 m.

The consequence of only having three axes is that the robot TCP can only be commanded a desired position; the orientation of the TCP is neglected. For the simulation it is assumed that a suitable gripper at the TCP is able to handle the orientation or is able to establish a grip regardless of the orientation of the TCP.

The three powertrains for each axis use the same model. They consist of a first order motor dynamic approximation to simulate the response time and delay from motor current to motor torque. The motor inertias are modeled as one-dimensional inertias for which the 3D-gyroscopic coupling effects on other robot components and on the chaser satellite are considered. This is implemented in Modelica according to [14] and allows for a computational efficient implementation.

The motor and gear friction is modeled using an invertible approximation according to [7] and consists of sticking and viscous friction. The flexibility of the robot gear is neglected.

For every axis, a joint position sensor is included in the powertrain. Sensor noise is added to simulate a real measurement.

No detailed grasp planning for the gripper at the robot arm is considered for the GNC simulation tool. Only the location of the robot arm TCP is relevant for the simulation. A grasp is considered as successful if the TCP is located at a pre-defined connection point on the target. It is assumed that a suitable tool or gripper at the TCP of the robot arm is able to establish a firm grip or other form of connection to dock the chaser satellite at the target satellite. The connection is considered as ideal for the simulation and forces and torques can be transferred as if the connection would be an ideal rigid link.

Due to the ideal connection, the robot TCP and the connected satellite would act as only one body with identical states instead of two separate bodies with each different states as before. Thus, the transition to a grip connection would imply a state reduction of the simulation model. Since a change of the number of states is not allowed during a Modelica simulation, the grip connection is realized with a force/torque element.

The force/torque element is an extension of a connector developed for the connection of launcher stages [17]. If a connection is established, there can always be a small error regarding the position and orientation of the two frames to be connected. The difference (mismatch) can be the control error for the robot TCP arm as well as numeric deviation (integrator tolerance). Therefore, the force/torque element is designed in such a way that a small mismatch between the two frames is allowed and does not result in extremely large forces and torques when the element is enabled (the grip closed). The force/torque element is based on the Baumgarte-version from [17]. A constraint force and torque is computed based on the relative motion between the two connected frames using a second order differential equation with a damping parameter.

Sensor models

Since highly detailed simulations of all involved sensors were not implemented for the GNC simulation tool yet, only functional models for required sensors are used for the simulation.

As such the detailed optical sensor simulation (e.g. Vis, IR and LIDAR) and estimation algorithms for the properties (e.g. tumbling, mass, inertia) of the target satellite are not part of the engineering tool. An approximation of real measurements is implemented using accuracies of real sensors and a noise model. This allows to easily simulating different sensor accuracies.

Sensor measurement values are simulated by calculating the ideal value directly from the simulation and adding a predefined noise. The noise is computed using random numbers according to the three sigma knowledge errors of the sensors.

For the orientation estimation of the chaser satellite, an Extended Kalman Filter (EKF, [18]) was implemented using simulated star- and sun-tracker data and angular rate measurement from an IMU. The EKF can handle multiple vector measurements with known location in a known inertial frame of reference (e.g. sun and star position vectors). It can combine the vector measurements with the simulated IMU rate measurements for an estimation of the current orientation of the satellite as quaternion.

4. GNC FOR THE CHASER SATELLITE WITH ROBOT ARM

The object-oriented GNC simulation tool allows implementing complex nonlinear control systems, which can easily be exchanged. For the ADR benchmark scenario, different control modules were implemented:

- Global trajectory planning module
- Chaser translation control
- Chaser attitude control
- Robot arm control

The benchmark ADR task itself was divided in different phases which are described in the following:

1. Rendezvous and approach Phase: The chaser performs a rendezvous with the target object, evaluates its attitude dynamics and CoM position.
2. Capture Phase: The chaser performs a final approach to the distance required to initiate the grasping using the robot arm.
3. Target Stabilization Phase: Once the grip is established, the de-tumbling of the target is started.
4. Disposal Phase: The target satellite is brought to a lower disposal orbit using a controlled descent.

The feedback-controller for the chaser position (translation) is implemented as a cascaded position - velocity controller. This allows using the same controllers for trajectory following control as well as following only a reference velocity without a given explicit desired position. This is helpful for different required steps of the path planning module.

The feedback-controller for the orientation of the chaser satellite is a cascaded controller as well. The outer loop is a feedback of the quaternion error between the desired and measured quaternion of the satellite (similar to [19]). In the inner loop, an angular rate controller is implemented.

Both controllers do not contain an explicit model of the system. To improve the performance of the feedback controllers, a two degree of freedom control approach is used. The feedback controllers main task is to ensure the minimization of disturbances and handling of parameter uncertainties, while the feed-forward controller improves the trajectory tracking performance. The model information is included in the feed-forward controller, which is implemented as a nonlinear inverse model of the system to provide the Inverse Dynamics of the system. Figure 4 shows an overview of the concept.

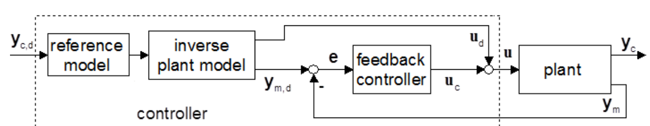

Fig. 4. Two degree of freedom controller using inverse plant model as feed-forward part (from [20]).

The feed-forward controller is a nonlinear inverse plant model of the satellite with its robotic arm. Such models can be designed using the powerful capabilities of the Modelica language. The use of nonlinear inverse models has proven to be useful to improve the control performance of various mechatronic systems (e.g. [7, 20]). The nonlinear inverse model includes all relevant dynamical effects and allows very precise calculation of the required actuator forces and torques.

The design of the inverse model for the chaser satellite was developed using the methods described in [2]. The inverse model considers the fully coupled chaser rigid body dynamics, including the reaction wheels and the mounted robot.

The friction model used for the robot is an invertible approximation that is also suitable for an inversion (see [7] for details). The environmental effects, such as atmospheric drag, are not considered for the inverse model and a perfect zero gravity environment is assumed. The feedback controllers have to correct this approximation in addition to other uncertainties and sensor noise.

Because the computation of nonlinear inverse models can be difficult for weak on-board CPUs of todays satellites, the idea is to store the resulting feed-forward trajectory, computed using the inverse model, as memory efficient B-spline data. The data is stored as relative translation and orientation to the target satellite. It can be used to precisely follow the approach trajectory.

The robot arm is controlled using direct joint control. The controller works directly on the difference between desired and measured motor angle only. In addition, to achieve an accurate grip position, the three individual joint controllers are extended with an outer loop Cartesian controller which allows for a correction of the TCP location. The Cartesian error between actual and desired position is converted to axis angle correction terms by the use of an analytical Jacobian matrix of the kinematic and a Damped Least Square (DLS) algorithm [21].

The outer Cartesian loop is not always active and only used for the fine correction of the TCP position. It is assumed that the chaser is equipped with suitable cameras, either directly at the TCP or another suitable location together with a suitable illumination system, that allow to measure the relative distance between the TCP location and the desired grip position.

Phase 1: Approach and rendezvous with target

Starting from an Inspection Point (IP) in a safe distance to the rotating target satellite, the chaser initiates a fly around the target satellite (see Fig. 5), keeping the optical sensors pointed on the target. It is assumed that the onboard optical

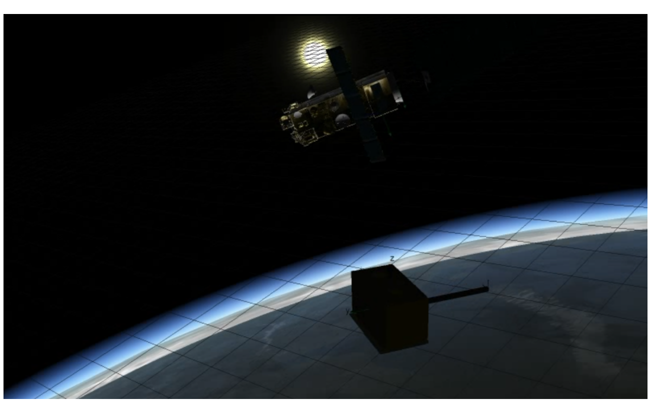

Fig. 5. Observing the target satellite.

sensors of the chaser are used for pose and state estimation of the target satellite during this phase. The optical sensors on board would estimate the CoM, attitude and exact relative position of the target. The complete fly around should give enough information to estimate the exact tumbling motion of the chaser using suitable algorithms. For the GNC-simulation tool the detailed simulation of these estimation algorithms is not considered, and it is assumed that the algorithms result in a good estimation within the knowledge error range provided by ESA.

The path for the approach (see Fig. 6) can be pre-computed starting for a given IP relative to the chaser satellite in LVLH coordinate system. For the simulation, a known axis of rotation and rotation speed of the target satellite is assumed. This allows the planning of the approach trajectory along the axis of rotation without danger of collision between the chaser satellite and the target satellite or its large solar panel.

For a real mission the pre-computation could be done in more steps. Starting from an estimation at the IP or after the fly around, using the actual measured motion of the target, a trajectory could be computed on Earth and the end result could be sent to the chaser satellite as B-spline data via a suitable uplink. For the GNC simulation tool, the approach trajectory is pre-computed based on the nonlinear inverse model of the chaser satellite with the robot arm. The inverse model is used as part of an optimization. For the optimization, the geometric path of the trajectory is given, but the velocity along the path is parametrized to allow the optimizer to find a time optimal trajectory that is within the actuator limits. The method is based on a similar approach that was used in [2]. The trajectory starts at the Inspection Point (IP) and after a fly around of the target ends at the Mating Point (MP). The resulting trajectory can directly be used as reference and feed-forward input for the chaser controllers and is stored as B-spline data.

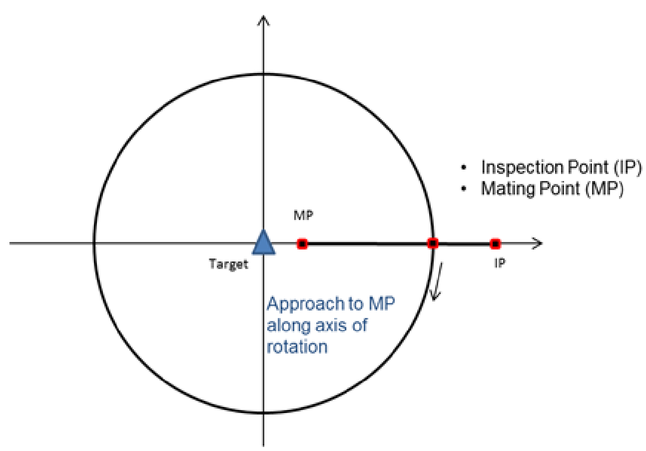

Fig. 6. Approach trajectory to target satellite.

There are multiple advantages of using the feed-forward data from the trajectory optimization for a two degree of freedom controller structure:

- Full consideration of coupling effects for the different controllers (translation and rotation).
- More precise path following is possible which allows a safer approach to the target satellite.
- Faster approach within actuator limits possible (also other optimization criteria would be possible, for example ideal energy consumption).
- Pre-computed data leads to less computational requirement for the on-board CPU. B-spline data can be evaluated easily and can be stored memory efficient.
- The usage of the inverse model as feed-forward control allows designing the feedback controllers with damping as priority.

The disadvantage of using the pre-computed data is the reduced flexibility. The path has to be known in advance and a good accuracy for the pose estimation is required. If the pre-computation is done on Earth, there also has to be a suitable communication interface on the satellite. Should this not be possible, a different solution (e.g. pure feedback control) could be chosen or a better CPU has to be installed on the chaser.

Phase 2: Capturing of target satellite

Phase 2 is the most critical phase of the mission, since a collision with the target satellite has to be avoided. To enable a safe approach to the robot arm range, starting from the MP, the chaser has to match the target rotation (see Fig. 7). A smooth transfer from the MP to the optimal grip position relative to the chaser satellite is necessary to avoid actuator saturation. The optimal relative grip position for the chaser rotates with the target satellite. The distance is such, that the robot arm on the chaser can easily reach the grip point. It is assumed that cameras on board of the chaser are able to measure this optimal chaser grip point. For the simulation, the ideal position is computed directly from the target satellite model and measurement noise is added to simulate the cameras. The optimal grip position for the chaser satellite is chosen such that a safe distance to the target is maintained but the center of mass of the chaser and target are as close as possible.

The capturing motion consists of four different steps:

1. The chaser mimics the target rotation, still positioned on the axis of rotation of the chaser.
2. The chaser satellite moves to the optimal grip point. Because of the rotation of the target, this leads to a spiral shaped movement.

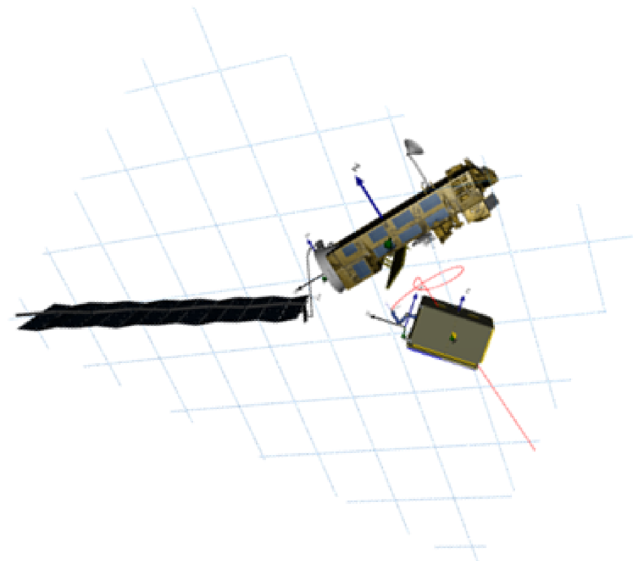

Fig. 7. Matching the target rotation for the capturing.

3. Using a smooth Point to Point (PTP) movement, the TCP of the robot is moved close to the estimated grip location.
4. The Cartesian controller makes a fine correction of the TCP location to the exact grip location and the grip is closed.

The robot PTP-movement, after the forced translation to the target, is filtered to avoid disturbance of the chaser movement and to minimize coupling effects between the robot arm controllers and the chaser controllers, which are all active simultaneously. After the PTP-movement to the pre-defined joint angles is completed, the Cartesian robot controller is activated to move the TCP to the exact grip location using simulated measurements. The reason for the separation is to have a more defined motion for the robot arm, since the Cartesian controller uses a Jacobian matrix for the transformation from the Cartesian space to the robot joint space. The fine correction using the Cartesian controller is necessary to compensate for the small lag error between the chaser and target movement.

For this phase, no feed-forward control is used because the exact rotation and grip position is assumed as uncertain and depends on measurement information. It can therefore not be calculated in advance. Analytical smooth transitions using sine shaped transition functions are used for all controller demand values and filtered using low-pass filters.

Phase 3: Target stabilization phase

The closing of the grip initiates the target stabilization phase. The connection of the two satellites with large difference in inertia and mass leads to a force and torque impulse at the robot TCP. The robot joint controllers dampen this impulse such that no collision between the two satellites occurs.

The detumbling of the target is controlled by the inner loop velocity component of the thrust control as well as the angular velocity controller of the reaction wheels. Using both controllers simultaneously allows for a fast detumbling of the target. It would also be possible to just use one of the controllers and actuators, but this would require much more time, since the actuators are relatively weak compared to the large inertia and mass of the coupled system of chaser and target.

For the detumbling phase, the joint controlled robot arm acts as a spring damper system between the chaser and target satellite to avoid collisions between the two objects.

Since for the detumbling, the exact orientation and position of the satellites is uncritical, only the velocity feedback loops of thruster and reaction wheel controllers are used.

Phase 4: Deorbiting of the target satellite

The deorbiting of the target satellite (see Fig. 8) is implemented in the simulation as a velocity controlled descent along the local LVLH z-axis towards the Earth. The demand vector for the thruster velocity controller is a smooth transition to the defined maximum disposal velocity vector to avoid unnecessary excitation of the coupled system. The robot joint and satellite rate controllers dampen the movement. The joint controlled robot acts again as spring damper system between the two satellites to avoid collisions and tries to maintain the grip axes configuration. The reaction wheel angular rate controller dampens the rotating movement of the coupled system to avoid a fast spinning motion. Since a full three axis thrust control is assumed for the simulation, the exact orientation of the coupled system is not relevant and the demand vector for the velocity thrust controller changes with the current orientation of the chaser satellite.

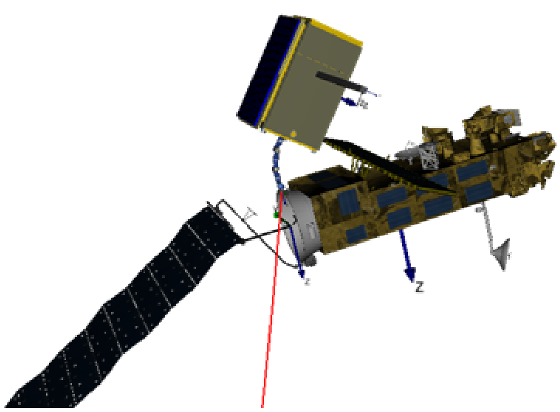

Fig. 8. Deorbiting of the target satellite

5. OPTIMIZATION BASED CONTROLLER SYNTHESIS

The controller synthesis is a difficult task for the capturing scenario, since multiple controllers have to work together to achieve the desired result. In addition, the possible range for uncertain mechanical parameters has to be considered as well as the accuracy and knowledge error for the required sensors. A tuning of all involved parameters directly by hand is nearly impossible. For this reason, an optimization based tuning of the controller parameters was chosen.

The tuning of the parameters has to ensure the stability of the strongly coupled system with all involved controllers and to provide a good performance, especially considering the trajectory following behavior.

In multiple phases, more than one controller is active at the same time. Therefore, it is difficult to optimize the controllers separately, since the coupled system could still become unstable or lead to an insufficient accuracy. In certain cases, the controllers can work against each other which can lead to actuator saturation. A suitable parametrized anti-windup is therefore required for the integral action of the controllers.

To overcome this problem, the DLR multi-case and multi-criteria optimization tool MOPS [9] was used to find suitable parameters for the controllers. Starting point for the optimization of the complete scenario were individually tuned controllers for the robot joint control, robot Cartesian control extension and thruster and reaction wheel control. The total criterion for the tuning is a weighted sum of sub-criteria:

- The weighted summed L2-control errors (integral of the squared error) of all involved controllers: to give a measure for performance and accuracy.
- Required L2-magnitude of the control action for all controllers: to avoid excessive energy consumption.
- Derivative of the magnitude of the control action: to avoid jittering and extreme aggressive controller behavior.

The same criteria were used for the local optimization of the individual controllers as well as for the complete scenario but the weighting factors for the individual sub-criteria were changed to achieve a good compromise between robustness and performance. To ensure the robustness and stability of the controllers for all the different phases of the scenario, a multi-case optimization was set up in MOPS for the complete scenario of the docking.

Each case consists of a Modelica model of the complete ADR scenario but uses different parameters. The controller parameters and the timing of the different phases are optimization tuners and are equal for all cases. This ensures robustness and stability for all considered variants. The four cases used for the optimization are:

1. Nominal case: Ideal scenario without noise and mechanical parameters at their nominal value.

2. Noisy uncertain measurements: Measurement noise and knowledge error for all involved sensors are considered for the simulation.

3. Maximum load: For this case, all mechanical parameters for weight and inertia tensors were set to their maximum allowed values (uncertainty). In addition, the centers of masses of the chaser and target satellite are not located at their nominal positions but are moved according to the maximum allowed deviation from their nominal position. Noisy measurements are considered as well.

4. Minimum load: Similar to the last case but all masses and inertia tensors are assumed at the lowest allowed values.

Figure 9 shows an overview of the multi-case and multi-criteria optimization setup. The optimization of the con-

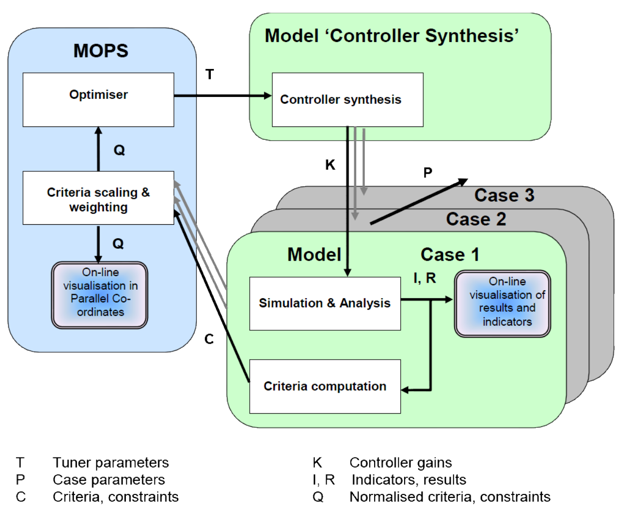

Fig. 9. Multi-case and multi-criteria optimization setup using the DLR tool MOPS.

trollers was done using a "Pattern-Search" optimization algorithm implemented in MOPS [9]. This algorithm is a robust local optimization algorithm. For the optimization, the described four cases where considered and the simulation included all phases of the scenario.

The four considered cases already lead to a good robustness for the controllers because the most extreme cases were considered. The results could be improved and verified by a Monte Carlo analysis or anti-optimization, but this was out of the scope of this project.

6. CONCLUSION AND OUTLOOK

The GNC simulation tool for Active Debris Removal with a robot arm was implemented in the Modelica language based on various Modelica libraries that were developed at the DLR Institute of System Dynamics and Control. The tool allows simulating the GNC for an ADR scenario and includes environmental models to simulate the LEO-space environment.

For the simulation, a chaser satellite model with a robot arm and the target satellite, based on Envisat, were implemented.

A trajectory planning module coordinates the different controllers of the chaser satellite and the robot arm. It provides reference trajectories for a successful capturing of the target satellite. The approach trajectory was designed using an inverse satellite model for the chaser including the robot arm. Functional sensor models simulate noisy measurements for the involved sensors. The synthesis and tuning of the feedback controllers for the thrusters, reaction wheels and robot control was done using a multi-case and multi-objective optimization. This results in a robust control setup that is able to handle mechanical uncertainties and sensor noise.

The simulation tool can be used to visualize the ADR scenario using CAD models of the satellites and the robot arm. The object-oriented design allows changing different components of the simulation. Mechanical parameters and the controllers can be easily modified to test different scenarios.

The simulation results showed that the ADR benchmark scenario (capturing of Envisat) can be successfully completed using the developed control algorithms.

In a future project, more complex models will be implemented to improve the simulation accuracy. Interesting aspects are the consideration of the flexibility of the robot and satellites as well as more detailed sensor and actuator models.

7. ACKNOWLEDGMENT

The work of this paper was funded by the ESA. The opinions expressed in the paper can in no way be taken to reflect the official opinion of the ESA.

8. REFERENCES

[1] J. Liou, N. Johnson, and N. Hill, "Controlling the growth of future LEO debris populations with active debris removal," *Acta Astronautica*, vol. 66, no. 56, pp. 648 – 653, 2010.

[2] M. Reiner and J. Bals, "Nonlinear inverse models for the control of satellites with flexible structures," *Proceedings of the 10th International Modelica Conference*, pp. 577–587, 2014.

[3] Modelica Association, Ed., *Modelica - A Unified Object-Oriented Language for Physical Systems Modeling Language Specification Version 3.2*, 2010.

[4] Hilding Elmquist, Dag Brück, Sven Erik Mattsson, and Hans Olsson und Martin Otter, *Dymola, Dynamic Modeling Laboratory Users Manual*, Dynasim AB, Sweden, 4.2b edition, 2002.

[5] FMI development group, “Functional mock-up interface for model exchange and co-simulation,” Tech. Rep., MODELISAR consortium, 2014.

[6] M. Thümmel, *Modellbasierte Regelung mit nichtlinearen inversen Systemen und Beobachtern von Robotern mit elastischen Gelenken*, Ph.D. thesis, Technische Universität München,Fakultät für Elektrotechnik und Informationstechnik, 2006.

[7] M. Reiner, *Modellierung und Steuerung von strukturelastischen Robotern*, Ph.D. thesis, Technische Universität München, Fakultät für Maschinenwesen, 2010.

[8] T. Bellmann, “Interactive simulations and advanced visualization with modelica,” *Proceedings of the 7th Modelica Conference*, pp. 541–550, 2009.

[9] H. Joos, J. Bals, G. Looye, K. Schnepper, and A. Varga, “A multi-objective optimisation based software environment for control systems design,” *Proc. of 2002 IEEE International Conference on Control Applications and International Symposium on Computer Aided Control Systems Design, CCA/CACSD*, 2002.

[10] A. Pfeiffer, “Optimization library for interactive multicriteria optimization tasks,” *Proceedings of the 9th International Modelica Conference*, 2012.

[11] O. Montenbruck and E. Gill, *Satellite Orbits - Models, Methods, and Applications*, Springer Verlag, Heidelberg, 2000.

[12] F. Lemoine, S. Kenyon, J. Factor, and R. Trimmer et al., “The development of the joint nasa gsfc and nima geopotential model egm96,” Tech. Rep., NASA Goddard Space Flight Center, 1998.

[13] J. Picone, A. Hedin, and D. Dro, “NRL-MSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues,” *Journal of Geophysical Research*, 2001.

[14] C. Schweiger and M. Otter, “Modeling 3D mechanical effects of 1D powertrains,” *Proceedings of the third International Modelica Conference*, 2003.

[15] Alin Albu-Schäffer, *Regelung von Robotern mit elastischen Gelenken am Beispiel der DLR-Leichtbauarme*, Ph.D. thesis, Technische Universität München, Fakultät für Elektrotechnik und Informationstechnik, 2001.

[16] Klaus Landzettel, Carsten Preusche, Alin Albu-Schaffer, Detlef Reintsema, Bernhard Rebele, and Gerhard Hirzinger, “Robotic on-orbit servicing - dlr’s experience and perspective,” in *Proceedings of teh IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006, pp. 4587–4594.

[17] P. Acquatella and M. Reiner, “Modelica stage separation dynamics modeling for end-to-end launch vehicle trajectory simulations,” *Proceedings of the 10th International Modelica Conference*, pp. 589–598, 2014.

[18] J. Crassidis and J. Junkins, Eds., *Optimal Estimation of Dynamic Systems*, CRC Press, 2012.

[19] Bong Wie and Peter M. Barba, “Quaternion Feedback for Spacecraft Large Angle Maneuvers,” *Journal of Guidance, Control and Dynamics*, vol. 8, no. 3, pp. 360–365, 1985.

[20] M. Thümmel, G. Looye, M. Kurze, M. Otter, and J. Bals, “Nonlinear inverse models for control,” *Proceedings of the 4th International Modelica Conference,Hamburg, March 7-8*, 2005.

[21] C. Wampler, “Manipulator inverse kinematic solutions based on vector formulations and damped least squares methods,” *IEEE Transactions on Systems, Man, and Cybernetics, 16*, 1986.