INVERITAS: A Facility for Hardware-in-the-Loop Long Distance Movement Simulation for Rendezvous and Capture of Satellites and Other Autonomous Objects

J. Paul[a,*], A. Dettmann[b], B. Girault[a], J. Hilljegerdes[b], F. Kirchner[a,b], I. Ahrns[c], J. Sommer[c]

[a] *German Research Center for Artificial Intelligence, DFKI Bremen, Robotics Innovation Center, Robert-Hooke-Straße 1, D-28359 Bremen*
[b] *University of Bremen, Faculty 03: Mathematics/Computer Science, Robotics Group, Robert-Hooke-Straße 1, D-28359 Bremen, Germany*
[c] *Airbus Defence & Space*
Space Robotics Projects, Airbus - Allee 1, D-28199 Bremen, Germany

Abstract

This paper describes the hardware-in-the-loop (HIL) long distance movement simulation system that was designed and built at the DFKI RIC for the INVERITAS project. It can simulate rendezvous and capture maneuvers between a *Client* satellite and a *Servicer* satellite in Earth orbit for scenarios where the semi-automomous Servicer repairs, refuels, or re-orbits the Client which can otherwise become inoperable and eventually end up as space debris. The simulation system is a hardware-in-the-loop simulation system, meaning it incorporates real hardware like mock-ups of the Client and the Servicer, real sensors like stereo camera systems, a LIDAR, as well as sensor data processing hardware. Controlled by the simulation, the mock-ups are moved in reality so that the Servicer's sensors perceive the Client like in the real situation. One of the main tasks in the development of the simulator was the reduction of the twelve unconstrained degrees of freedom of two free-floating objects to ten constrained degrees of freedom of the INVERITAS movement hardware. A number of behaviors of the control system described in this paper enable it to use the given workspace efficiently to fit trajectories of the two satellites into it. The system reaches an accuracy of a few centimeters that is sufficient to test sensor data processing and navigation algorithms of the Servicer in closed-loop, meaning that the autonomous decisions of the Servicer can be based on the real sensor input. We also present methods and HIL test results concerning the sensors, sensory data processing and GNC (guidance, navigation and control) software of the functional Servicer mock-up that was developed in the INVERITAS project. Finally, the paper includes future plans of how the HIL simulator can be improved in accuracy and flexibility.

Keywords: Orbital servicing, Hardware-in-the-loop, Robotic manipulation, GNC, Pose estimation

1. Introduction

Space debris in Earth orbit is an increasing danger for manned and unmanned spaceflight. Removing space debris from orbit is a possible solution, but it is even better to repair or refuel satellites that need servicing or to only correct their current orbit, anything which prevents them from becoming space debris. This not only avoids new debris but enables expensive satellites to be used for a longer time. This has been the subject of the INVERITAS project. The overall goal was developing technologies for the rendezvous with an uncooperative target and a related hardware-in-the-loop (*HIL*) test facility for testing the Servicer *GNC* (guidance, navigation and control) system under conditions close to reality. In addition to this, technologies for a semi-autonomous on orbit servicing were envisaged to be verified using the HIL simulator.

This paper describes the HIL long distance movement simulation system (Figure 1) mentioned above that was designed and built at the DFKI RIC for the INVERITAS project [1] as well as the functional Servicer mock-up designed wihin the project to be tested in the HIL facility. The HIL system is capable of testing the sensor- and data-processing hardware and the software of the Servicer satellite that services other Client satellites. The simulation system is a HIL simulator involving a mock-up of the Servicer including real sensors like an observation camera, two stereo-camera systems, a LIDAR from Jena-Optronik, and processing units as well as a full-size mock-up of a Client satellite. The movement of both satellites in an Earth orbit is determined by a software simulation in real-time and the resulting absolute poses in orbit are used to map their relative movement to the movements of the Client and Servicer mock-ups in the testbed environment. The result is that the real hardware sensors of the Servicer can perceive the real mock-up of the Client very close to how they would do in space, enabling realistic tests of sensor

*Corresponding author
Email addresses: jan.paul@dfki.de (J. Paul), alexander.dettmann@uni-bremen.de (A. Dettmann), benjamin.girault@dfki.de (B. Girault), jens.hilljegerdes@uni-bremen.de (J. Hilljegerdes), frank.kirchner@uni-bremen.de (F. Kirchner), ingo.ahrns@astrium.eads.net (I. Ahrns), josef.sommer@astrium.eads.net (J. Sommer)

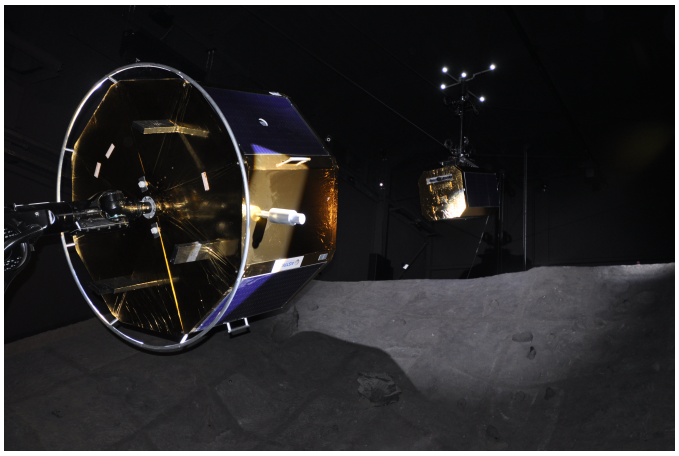


Figure 1: The INVERITAS System in the Exploration Hall of the DFKI RIC

hardware, its data processing algorithms, as well as of the autonomous navigation of the Servicer.

Common systems for HIL *RvC* (Rendezvous and capture) simulations consist of robotic arms, which approach each other with the help of linear rails or gantry cranes. EPOS 2.0 [2] (European Proximity Operation Simulator) at the German Space Operation Center in Oberpfaffenhofen consists of two industrial robotic arms. One arm is mounted on a linear rail, which allows an approach of up to 25 m. The whole system can utilize 13 degrees of freedom (*DOF*). The SOSC (Lockheed Martins Space Operations Simulation Center) [3] simulates on-orbit docking between a full-scaled ISS-docking system and the Orion spacecraft. The space simulator also includes two robotic arms and a rail for heavy weights. This common approach could not be realized in the INVERITAS project since the system had to be integrated into the Space Exploration Hall of the DFKI RIC, which contains an artificial moon crater and other testbeds to test mobility concepts of exploration systems for extraterrestrial celestial bodies [4]. So, the decision was made to use a robotic arm and a cable robot to realize the movement of two satellite mock-ups instead of two robotic arms where one of them is mounted on a linear rail. A cable robot has a large range of motion in all translational DOF without occupying the space when the system is not used. A long approach can be realized, especially when using the three dimensional space diagonal of the Space Exploration Hall which is 24 m long, 12 m wide and 10 m high. In addition, the combination of a cable robot able to move in all directions and an artificial Moon surface can be used to simulate landing maneuvers on extraterrestrial celestial bodies. More details on the INVERITAS system components are provided in section 2.1.

Systems which are designed to approach or dock to static obstacles have usually less than 12 DOF and thus are limited to their special application as well. Lama (Lande- und Mobiltaetstestanlage)[5] for example is a facility for lander touchdown and rover mobility testing, which consists of a very strong industrial robotic arm (max. 1000 kg load) on a rail to be able to test full-size and full-weight test objects. The system is able to simulate in real-time a controlled offloading force to compensate Earth gravitation to generate any desired gravitation. Tron (Testbed for Robotic Optical Navigation) [6] can simulate a very long descent towards Moon to test optical navigation algorithms. It consists of a robotic arm with 6 DOF on a rail. The rendezvous scenarios are always scaled, so the possible simulation distance and can be very large. Therefore, precision is even more important since all errors are scaled as well. Instead, the usage of a full-size Client mock-up in the INVERITAS system enables the acquisition of unscaled sensor data.

The INVERITAS HIL simulation system was designed to have a system available that is not restricted to one class of experiments like comparable systems, but can be used for a wider range of tests like rendezvous and docking-, capture- and landing-maneuvers. To simulate lighting conditions of different environments especially in space context it was also designed to have a dark environment with a configurable lighting setup to produce realistic inputs for real optical sensors.

The INVERITAS HIL simulation system has 10 controllable DOF, counting 6 DOF of the robotic arm moving the Client satellite (see table 1(c)) and 4 DOF of the cable robot moving the Servicer satellite (3 translational DOF plus one rotational DOF around its vertical axis, see table 1(b)). Nevertheless, one has to solve the problem that the 12 unconstrained DOF of two free-floating objects have to be mapped to these 10 constrained DOF of the INVERITAS system. Thus, other systems with less than 12 DOF often restrict their movements to a lower number of dimensions. For instance RACOON (Real-time Attitude Control and On-Orbit Navigation) [7] which has 5 DOF (two translational and one rotational DOF for the target satellite, two rotational DOF for the chaser) restricts the HIL simulation to in-plane approaches. To avoid loosing a movement dimension, we propose methods to realize the relative movement of two objects in 12 DOF without such a dimensional restriction, although there remains a restriction to the maximum distance between both objects. These methods efficiently use the limited space with 10 DOF as described in section 2.2 to reach the desired relative pose between the objects. This is possible because only the relative pose between the objects has to match the simulation so that not both mock-ups have to be movable in full 6 DOF each.

Section 2.3 describes methods developed to optimize the usage of the limited operation space inside the facility by adapting the movement paths accordingly.

Much work went into developing methods for measuring, calibrating, and controlling the components of the INVERITAS HIL simulation system so that it reaches a high positioning accuracy, which is important for verifying Servicer sensors and algorithms that measure distances and poses of the Client. Section 2.4 goes into detail about the

accuracy of the different system components and how we significantly increased the accuracy of the system especially by optimizing the precision of the cable robot.

Section 2.5 describes the sensors, sensor data processing algorithms, and the Guidance Navigation and Control (GNC) unit of the Servicer satellite.

Section 3 gives experimental results of the measured overall accuracy of the system and the performance of the sensor processing and GNC.

Section 4 discusses the results giving a conclusion and an outlook to future works.

2. Methods

2.1. System Overview

This section presents the different system components of the INVERITAS testbed, the way these systems are controlled, and how experiments are conducted. The key features of the robotic systems are described, as well as the motion tracking and lighting system. Then, the architecture of the system used to control the whole facility is presented. Figure 3 gives an overview of the Space Exploration Hall itself.

For simulating satellite rendezvous maneuvers we use a combination of software simulation and hardware. Large distances are covered in software simulation only, while the last meters (up to 16.5m) real hardware is included in the loop (HIL).

These two modes of operation are used because mission scenarios for satellite rendezvous usually cover distances between both involved satellites of more than 100 m. Because of limited space in testbeds, simulation or scaled mock-ups are used to simulate the whole approach. So the INVERITAS system uses only software simulation for the long-range as well, but then starts to use HIL as soon as the scenario is realizable in the available operation space. Thus, during the final approach, the Servicer satellite is able to collect realistic sensor data from the mock-up. For that, a full-size mock-up of the Client satellite is used, which is important because the GNC and the whole mission rely on these unscaled sensor informations. Consequently, a large operation space is desirable to maximize the amount of realistic data.

Figure 2 shows an overview over the system architecture. The GNC controls the simulated actuators of the Servicer, and the resulting movements are calculated by the orbital dynamics simulation. The result are new poses for the Client and the Servicer in space. These poses can be visualized in software, which is useful for observing the simulation and for producing simulated sensor data that can be sent to the GNC for a software simulation without hardware in the loop. The simulated poses are transferred to the DOF reduction module (see section 2.2) which translates them into new poses for the robotic arm (see section 2.1.2) and the mount of the cable robot (see section 2.1.1) of the movement system. These poses can be visualized before actually transferring them to the movement system, so that test runs of the movement system are possible without hardware in the loop. When the movement hardware is included into the simulation loop, it moves the Client and Servicer mock-ups so that their relative poses match the relative poses in space calculated by the orbital dynamics. Now, the real hardware sensors of the Servicer can receive realistic sensor data of the Client, as it has the correct pose relative to the Servicer. This sensor data can be returned to the GNC which can control the Servicer based on this data in closed loop.

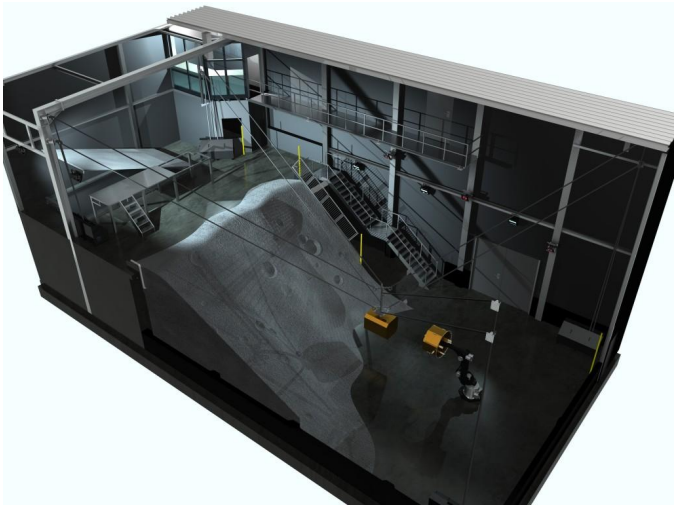


Figure 3: Overview of the Space Exploration Hall. One can identify the cable robot and the robotic arm which carry the satellite mock-ups. The control room on the upper left corner provides a good overview of the facility. A balcony has been added to be able to monitor the systems from above, with no risk of entering into the working areas of the robotic systems.

2.1.1. Cable Robot

A cable robot is able to use the three dimensional space diagonal of the given facility and therefore is best suited to realize a rendezvous scenario with an approach distance as long as possible. A greater distance does not significantly increase the effort of hardware installation compared to other solutions as only the cable lengths have to be increased, the rest of the hardware can stay unchanged as long as it can withstand the forces on the cables. The main applications for cable based equipment carrier systems are entertainment, e.g. spidercam®[1] or large scale manufacturing, e.g. Robocrane [8]). For these applications, an operator usually has the ability to manually control the position of a carrier or *mount* with a joystick. Stability and precision are not as important for such applications as the ability to move freely in a large operation space without disturbing anyone's view or action.

In the standard version of a spidercam® cable robot, the mount itself is lifted by four cables connected through

[1] www.spidercam.tv

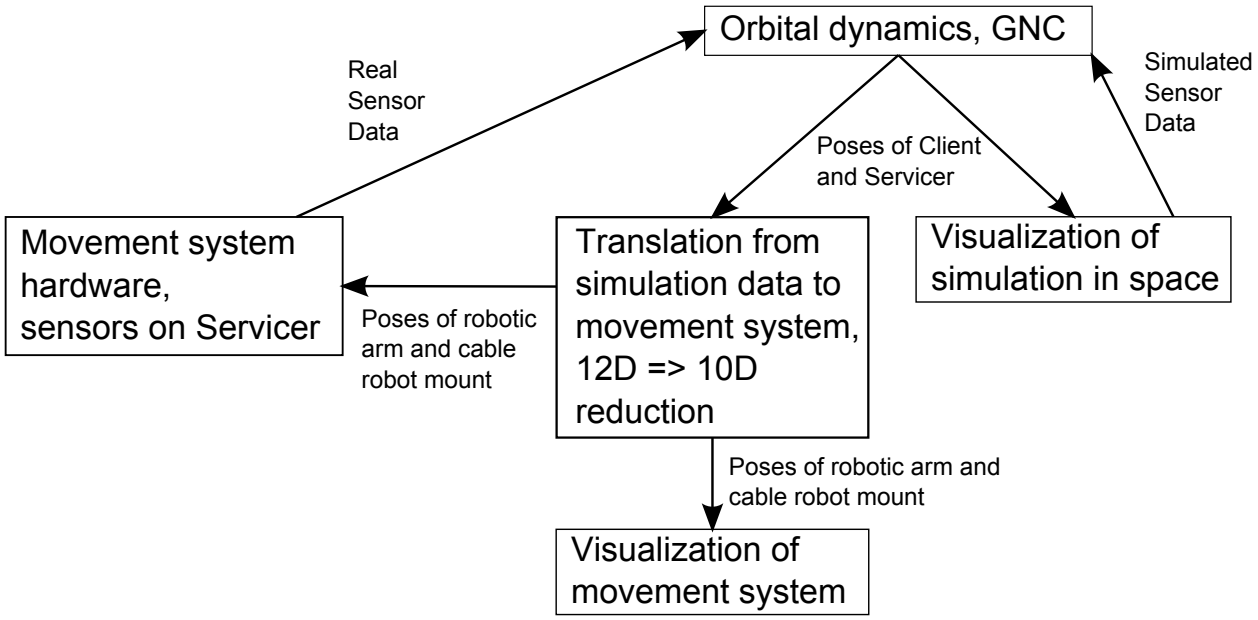
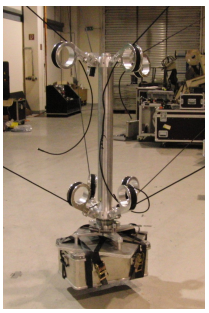


Figure 2: Overview of the system architecture



(a) First test of the mount with eight cables attached in two virtual points

(b) A winch for winding two cables simultaneously

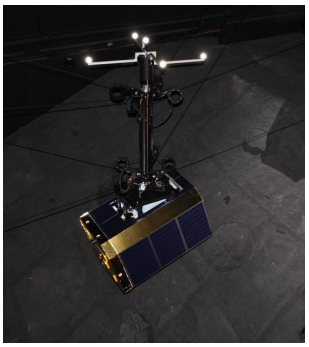
Figure 4: The cable robot hardware

pulleys to four independent winches. By winding and unwinding the cables, the force and thus the position of the mount is influenced. At the mount, all four cables are connected to ears close to the center pole on top of the the center of mass so that the mount stabilizes itself. The position and height of the pulleys define the possible maximum operating space of the mount. If the winches are ideally located on each corner of a square, the resulting operation space of the cable robot will be 30% smaller than the cuboid formed by the pulleys and the flat ground. The usage of four cables increases the complexity of the control algorithm because the solution to calculate the length of each winch is over-determined but also increases the safety and the stability.

For the INVERITAS project, spidercam® redesigned their common cable robot system to meet the required stability and precision for rendezvous maneuvers. The unique new concept is to use eight cables and keep the four winches. Therefore, two pairs of four ears are located on top of each other at a distance of ca. 1 m (Figure 4(a)). The two pairs of four pulleys are installed at the ceiling at the same distance to deflect the cables to one of the four winches (Figure. 4(b)). In this way, each winch can simultaneously control the length of two cables with the same control approach but the stability of the mount is increased. The cables also do not intersect the operation space like other approaches such as [9], decreasing the risks of collision. The cable robot could be installed in the Space Exploration Hall despite the artificial Moon crater. The space diagonal which limits the maximum approach distance is 16.5 m. In contrast to the standard version, accelerating or moving the center of mass away from the geometrical center of the mount will not influence the orientation of the mount.

Due to mechanical reasons, the cables are not fixed in two central points. This creates an additional torque around the vertical axis that on the one side stabilizes the mount rotation, but on the other side causes a relative rotation depending on the mount position in its operation space. To compensate this effect and increase the flexibility of the cable robot, an additional actuator to rotate the mount around the vertical axis was implemented.

In the INVERITAS system, the Servicer is attached to the cable robot (Figure 5(a)). Due to the data rates of the communication between the Servicer and the simulation system and the power requirements of the sensors and processing units in the Servicer, a WiFi connection and battery power supply were not suitable. To keep a simple experimental setup and to get in situ data, power and data are transferred via the cables. While power transmission is realized by copper wires, the data transmission relies on



(a) A cable robot holds the Servicer satellite

(b) A 6-DOF robotic arm is used to move the Client satellite

Figure 5: Photographs of the robotic systems used to move the mock-ups of the satellites

optical fiber. They are surrounded by Kevlar to support the weight of the Servicer and a black cable coating to protect the inner layers. The specifications of the cable robot (Table 1(a)) allow a wide range of applications.

The cable robot is controlled in Cartesian space via a CAN bus with a frequency of 250 Hz. The low-level control of the four winches is achieved by a proprietary software of the spidercam® company. The rotational actuator used for the vertical rotation of the Servicer with the mount and its electronic components were developed at the DFKI RIC. It is controlled via another CAN bus, including two CAN-Ethernet converters on both sides of the optical fiber that connects the mount to the control station via one of the wires. The control frequency is 250 Hz as well.

2.1.2. Robotic Arm

A robotic arm with six DOF is used to hold the mock-up of the Client satellite. In contrast to the cable robot, the robotic arm has a small range of translational motions but a large range of rotational capabilities. The arm chosen (Figure 5(b)) is an industrial Kuka[2] KR60-3 arm with a maximal payload of 60 kg at a lever of 200 mm. The specifications of the arm (Table 1(c)) allow very fast movements and a large range of rotation around the fourth and sixth joint axes, which are needed to simulate a tumbling satellite. The full-size Client mock-up is passive, i.e. no electronics are embedded. Nevertheless, the design of the arm allows the mockup to be wired for power supply or communication purposes. The arm itself is controlled via Ethernet with a frequency of 83,33 Hz. It can be controlled in Cartesian and in joint space.

2.1.3. Motion Tracking System

A motion tracking system (MTS) is used for offline calibration and online measurement of the cable robot pose.

Table 1: Robot Specifications

(a) Cable Robot

Max speed	$2\,\frac{m}{s}$
Max acceleration	$1.5\,\frac{m}{s^2}$
Max payload	$150\,kg$
Power supply	$4\,A$ @ $230\,V$
Communication	$10\,\frac{Gbit}{s}$
Range of rotation	$\pm 45°$

(b) Cable Robot DOFs

DOF	Actuator Type	Range	Max speed
1	winches & cables	$16m$	$2\,\frac{m}{s}$
2	winches & cables	$7m$	$2\,\frac{m}{s}$
3	winches & cables	$5.50m$	$2\,\frac{m}{s}$
4	rotational joint	$\pm 45°$	$3°/s$

(c) Robotic Arm DOFs

DOF	Actuator Type	Range	Max speed
1	rotational joint	$\pm 185°$	$128°/s$
2	rotational joint	$-135°$ to $35°$	$102°/s$
3	rotational joint	$-120°$ to $158°$	$128°/s$
4	rotational joint	$\pm 350°$	$260°/s$
5	rotational joint	$\pm 119°$	$245°/s$
6	rotational joint	$\pm 350°$	$322°/s$

This data is used to improve the precision at which the mock-ups reach the poses calculated by the orbital dynamics simulation of the HIL-simulator. It is not available to the Servicer itself, as it has to estimate the Client pose and its own relative position only using it's own sensors and data processing.

To improve the HIL-simulator precision with MTS data, the Space Exploration Hall is equipped with seven Vicon motion capture cameras[3] covering most of the operation space of the cable robot. Six cameras provide a 1 Mpx resolution and are equipped with lenses having a 6 mm focal length providing an opening angle of 68°. The seventh camera comes with a 16 Mpx resolution and a lens with 18 mm focal length providing an opening angle of 54°. The cameras are equipped with strobes emitting near-infrared light with a wavelength of 690 mm.

The pose of an object can be detected by the MTS by placing retro-reflective markers on the object which can be tracked by the cameras. The markers build a defined object with its own coordinate frame in the MTS coordinate system. The MTS coordinate system is also used as a world coordinate system and can be defined by placing a calibrated origin frame in the Space Exploration Hall. In order to guarantee a static world coordinate system origin with high accuracy, this frame was realized by fixed markers spanning a right angle with 3 m link length. These positions where precisely defined with a laser tracker [10]

[2] http://www.kuka-robotics.com

[3] http://vicon.com/System/TSeries

having a sub-millimeter accuracy.

The cable robot marker object was formed by nine markers in a non-symmetric fashion giving redundancy and preventing rotations from being ambiguous thus increasing the stability of the pose measurement. The pose of the cable robot is fed into the HIL-simulator's control system via ethernet with a frequency of 83,33 Hz, thus having the same period as the robotic arm. It is not possible to measure the Client pose directly with the MTS because the retro-reflective markers would also be seen by the Servicer sensors (e.g. LIDAR, cameras) and the respective processing algorithms that were tested during the experiments.

2.1.4. Lighting System

EPOS, TRON, SOSC, and RACOON are equipped with a lighting system that can simulate sunlight by creating almost parallel light beams that create hard shadows and large differences in light intensity between illuminated and dark regions, which are typical for objects in space illuminated by the sun without an atmosphere. In order to reproduce these lightning conditions as well, the Space Exploration Hall is equipped with six mobile spotlights[4]. Each spotlight is motorized, allowing pan and tilt rotations and the field of view can be varied between 12° and 30°. The spotlights can be moved up and down from 1 m to 6 m. The 575 W gas discharge lamps deliver a 6000 K light, with maximal intensity of 14500 Lux at 10 m distance and a 12° field of view. Combined with special light absorbing paint on the walls, ceiling, and on all visible components of the system, the camera sensors of the Servicer almost only see the Client. Under such lighting conditions, the cameras also see flares and other optical phenomena that would also appear in space. The resulting problems in the image processing algorithms can then be analyzed before launching a real Servicer satellite. Figure 6 shows the view on the Client mock-up from the Servicer when getting direct lighting from one spotlight.

2.1.5. System Control

For the control and the monitoring of the hardware devices, a control room was installed (Figure 7), which contains the needed equipment i.e., an uninterruptable power supply, a dSPACE[5] real-time core processing unit, a PC for the control software and the simulations, a computer with access to the eight surveillance cameras installed in the Space Exploration Hall, a computer for the user with access to the Servicer satellite, and the control units for the cable robot as well as for the MTS.

2.1.6. Core Processing

The communication with the different robotic systems, the transformation of the desired trajectory into a feasible

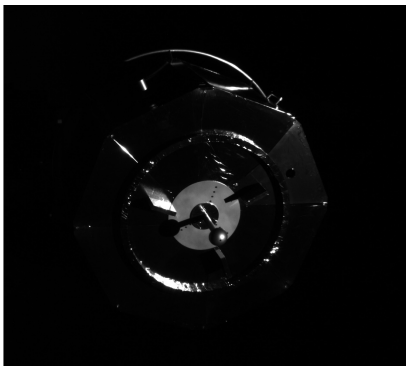
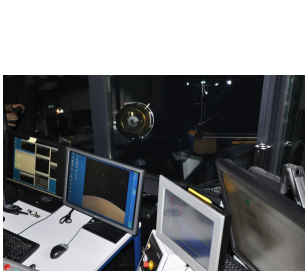
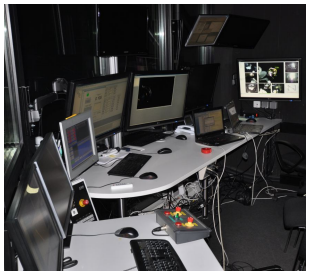


Figure 6: View on the Client from the Servicer satellite under realistic light conditions due to powerful gas discharge lamps and black light-absorbing paint



(a) View into Exploration Hall



(b) View into control room

Figure 7: Control Room

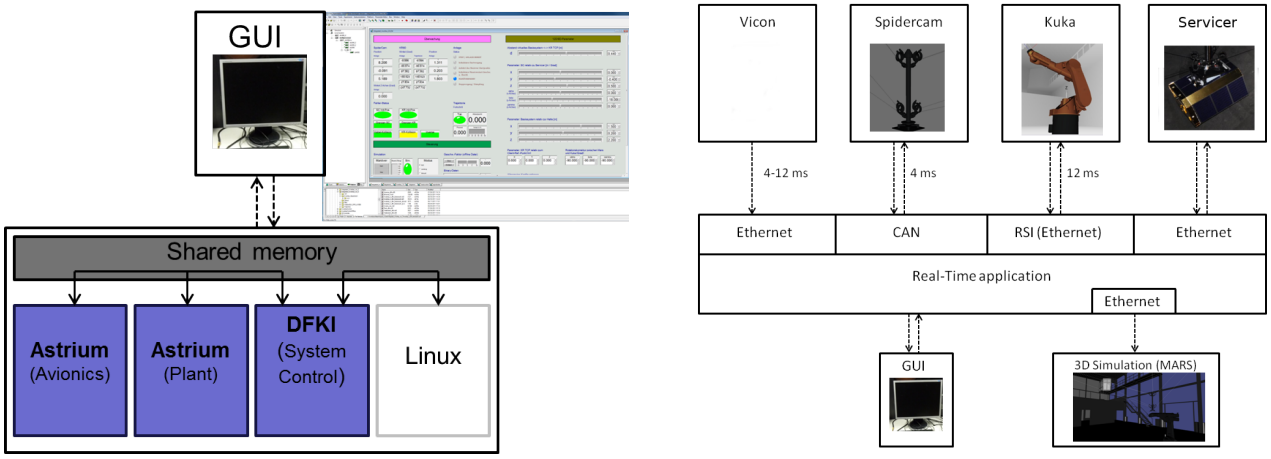
motion and the control of the robots to perform the motions is done on a real-time dSPACE 1006 platform. The interfaces and the algorithm implementation are done using Matlab[6]/Simulink[7] tools. The main processor board is connected by optical fiber to a dedicated computer allowing the user to control and monitor in real-time the whole mission. The quad-core architecture of the main processor board allows the control system to be split up into four different elements (Fig. 8(a)). One core is used for the control of the Space Exploration Hall and the CAN interfaces. One core runs on Linux and is dedicated to handle the Ethernet interfaces. The last two cores can be used to control and simulate the movements of both mock-ups according to the experiment. In the case of the project INVERITAS, one of these two cores was used to model the satellite dynamics and the other one to model the navigation modules. The communication interfaces between the cores is performed in a main Simulink model. Each core can then be compiled separately and only the binaries have to be exchanged, allowing external users to keep their algorithms confidential.

2.1.7. Communication

Figure 8(b) shows an overview of the communication interfaces between the different system components. The

[4]http://www.adblighting.com
[5]http://www.dspace.com

[6]http://www.mathworks.com/products/matlab/
[7]http://www.mathworks.com/products/simulink/



(a) Quad-Core processing architecture

(b) Communication interfaces

Figure 8: System Control

control period of the system control core was adapted to the fastest real-time component, i.e., the cable robot which has to be fed with position changes every 4 ms (250 Hz) via a CAN bus interface. The robotic arm, the MTS, and the software simulation communicate via TCP/IP Ethernet interfaces in a 12 ms period (83,33 Hz). The communication to the Servicer is also realized via Ethernet, but the protocol can be defined by the user. A 10 Gbit/s optical fiber communication guarantees sufficient bandwidth for diverse applications.

The dSPACE system allows different clock rates, so that the user is not restricted to a control frequency of 250 Hz, i.e., in the INVERITAS project the two cores for the the orbital dynamics and the navigation modules run with a clock of 10 Hz.

2.1.8. Simulation

Besides the HIL simulation in the Space Exploration Hall, the whole system is replicated in the real-time simulation software MARS (Machina Arte Robotum Simulans)[8]. MARS is an open-source simulation and visualization tool developed at DFKI for developing control algorithms and designing robots . It consists of a core framework containing all main simulation components, a GUI, an OpenGL based visualization, and a physics core that is currently based on ODE [11].

The simulation helps to test new developed software efficiently without the need of starting the hardware components. On the one side, one can test all missions beforehand to check if one of the objects would hit any limits of operation. On the other side, the whole set of transformations can be checked for plausibility. As you can see in Fig. 9, the simulation is used to compare the input data, meaning the absolute object poses given in a certain

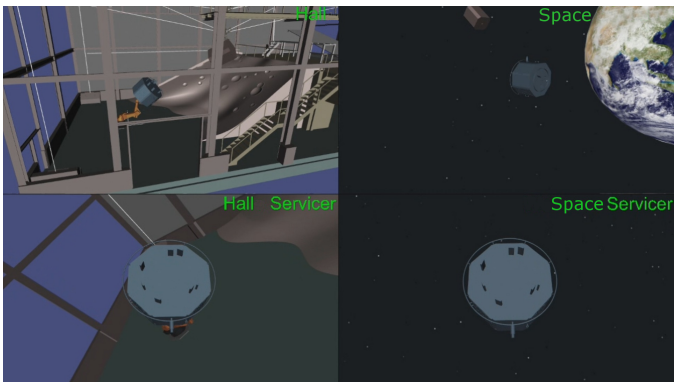


Figure 9: Different views on the simulated application: The top right picture shows the situation in the simulated in-orbit mission. The bottom right picture shows the same situation from the perspective of the Servicer satellite. The left side shows the same situation transformed into the Space Exploration Hall. Note that the absolute poses and coordinates are different but the relative pose stays the same.

coordinate system, with the output data of the HIL control system, meaning the resulting poses of the movement system components. Both should maintain the same relative poses. Qualitative differences can easily be seen which helps the programmer to track possible errors before testing the algorithms with the real hardware. In addition, the quantitative error is calculated as well to see if any inaccuracies are present at all (see section 2.4).

2.2. Reduction of the Degrees of Freedom

One of the key technologies of the INVERITAS system is the transformation of the movement of two free-floating objects, each having six unconstrained DOF, into the Space Exploration Hall with only ten constrained DOF. The objective is thus to transform the input trajectory given in Earth-Centered Inertial (ECI) coordinate frame into a feasible trajectory for the robotic systems. Since

[8] https://gitorious.org/rock-simulation/mars

the set of possible solutions of this transformation problem is most of the time infinite, but the working area of the systems is limited, the main issue is to find a solution which tries to optimize the range of trajectories that can be continuously simulated by simultaneously reducing the DOF from twelve to ten.

Two types of solutions have been implemented and tested. One solution, *Fixed Client*, assumes, that the relative rotation between Client and Servicer is exclusively realized by the robotic arm, whereas the cable robot solely executes the translational movement. The second solution, *3DOF Robotic Arm*, uses just three DOF of the robotic arm to realize the relative orientation of both objects. Thus, changes in the orientation result automatically in translational movements of the Client, which have to be compensated by the corresponding movements of the cable robot. This second solution increases the movement of the overall system, but has major advantages, described in section 2.3.

2.2.1. DOF Reduction - Fixed Client

In the description of the variables as well as in the following equations, the superscript describes the coordinate frame in which the transformation is considered. The subscript determines the considered object. T stands for a 4x4 transformation matrix, which includes a 3x3 rotation matrix R and a 3x1 position vector P.

The inputs for the DOF reduction are the absolute coordinates of the Client T^{ECI}_{Client} and Servicer $T^{ECI}_{Servicer}$ in the ECI coordinate system (Figure 10(a)). The required outputs of the DOF reduction are the poses of the robotic arm and the cable robot, which can then be used to command the devices (Figure 10(b)).

The first step consists of computing the relative pose of the Client satellite in the Servicer's coordinate frame $T^{Servicer}_{Client}$ (1).

$$T^{Servicer}_{Client} = (T^{ECI}_{Servicer})^{-1} \cdot T^{ECI}_{Client} \tag{1}$$

In a second step, the position of the Client P^{World}_{Client} is manually set. This position has to be chosen in a way that all of the required orientations can be realized by the robotic arm. This position can later automatically be adapted to avoid collisions or to stay inside the working range. The orientation of the Servicer relative to the world coordinate system $R^{World}_{Servicer}$ is also known (2).

$$R^{World}_{Servicer} = R_Z(\psi) \cdot R_Y(\theta) \cdot R_X(\phi) \tag{2}$$

with $\psi = 180° + \gamma$, $\theta = 18°$ and $\phi = 180°$. γ is the vertical rotation of the platform of the cable robot which is also user defined or automatically adapted during operation. The pitch angle θ is fixed and was set to an angle which allows to use the space diagonal of the operation area increasing the maximum approach distance.

The orientation of the Client satellite can then be derived (3) and the position of the Servicer satellite is obtained (4).

$$R^{World}_{Client} = R^{World}_{Servicer} \cdot R^{Servicer}_{Client} \tag{3}$$

$$P^{World}_{Servicer} = P^{World}_{Client} - R^{World}_{Servicer} \cdot P^{Servicer}_{Client} \tag{4}$$

After determining the positions and orientations of the two objects, the transformations from the satellites' mockups to the robotic systems have to be computed.

To calculate the joint angles for the robotic arm via the inverse kinematics, the pose of the tool center point (TCP) relative to the base of the robotic arm $T^{RoboticArm}_{TCP}$ has to be known. The pose of the TCP in world coordinates can be calculated by multiplying the known pose of the Client in world coordinates T^{World}_{Client} with the transformation from Client to the TCP T^{Client}_{TCP}, which was estimated using the CAD model of the mockup (5) (Figure 10(c)). Since world and robotic arm coordinates are not identical, a change in the coordinate system has to be applied as well (6). The required pose of the robotic arm in world coordinates $T^{World}_{RoboticArm}$ was experimentally determined by using a precise laser tracker [10].

$$T^{World}_{TCP} = T^{World}_{Client} \cdot T^{Client}_{TCP} \tag{5}$$

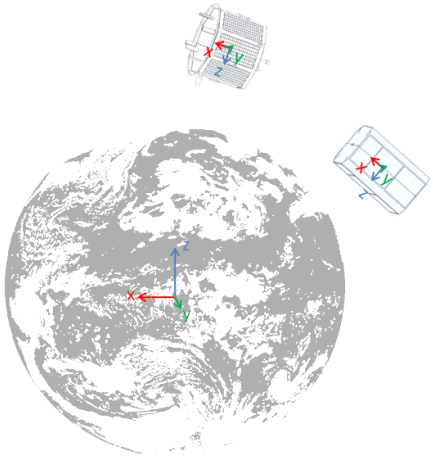
$$T^{RoboticArm}_{TCP} = (T^{World}_{RoboticArm})^{-1} \cdot T^{World}_{TCP} \tag{6}$$

The corresponding joint angles of the arm, presented in Figure (11), are calculated by using a geometrical solution for the inverse kinematics (7). The inverse kinematics gives up to 32 solutions. The user can choose the best solution manually at the beginning of the trajectory which he might prefer.

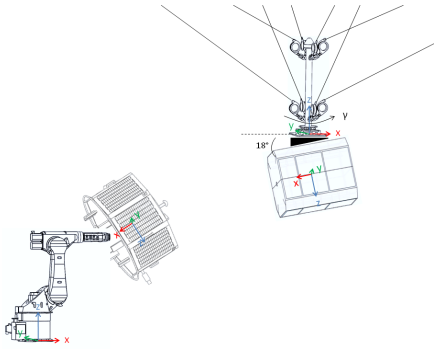
- *Overhead.* DOF_1 joint is rotated 180°
- *Elbow up/down.* DOF_2, DOF_3 and DOF_5 are modified accordingly
- *Inverse wrist.* Inverse value for DOF_5. DOF_4 and DOF_6 are rotated by 180°
- *DOF_4 and DOF_6 extra solution.* There exists in most cases up to three further solutions since the range of motion of these axes is between $-350°$ and $350°$

$$(DOF_1, DOF_2, DOF_3, DOF_4, DOF_5, DOF_6) = IK\left(T^{RoboticArm}_{TCP}\right) \tag{7}$$

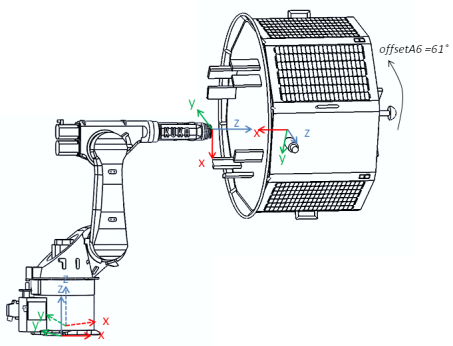
The position of the cable robot is derived knowing the translation between the origins of the Servicer and cable robot coordinate system. The horizontal and vertical offsets ($Offset_h$ and $Offset_v$) were estimated using the CAD models of the satellite and the cable robot (Fig.10(d)). The transformation also depends on the rotation γ of the cable robot platform.



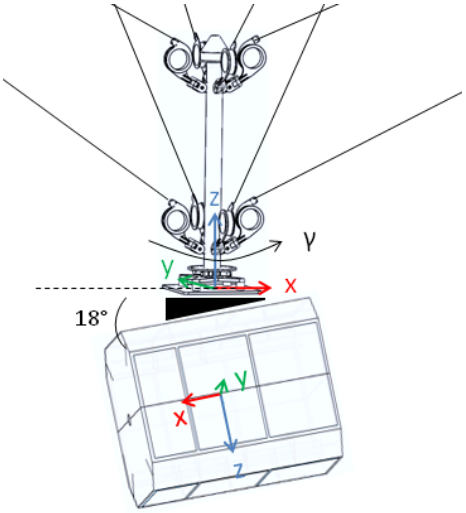
(a) The input data are the absolute coordinates of Client and Servicer satellite in the ECI coordinate system



(b) The outputs are the poses of cable robot and the Tool Center Point of the robotic arm



(c) Transformation between robotic arm and Client satellite



(d) Transformation between cable robot and Servicer satellite

Figure 10: Steps of the coordinate systems transformations

$$P^{World}_{Cable\,robot} = P^{World}_{Servicer} + \begin{bmatrix} Offset_h \cdot cos(\gamma) \\ Offset_h \cdot sin(\gamma) \\ Offset_v \end{bmatrix} \quad (8)$$

2.2.2. DOF Reduction - 3DOF Robotic Arm

This solution was developed to allow loads on the robotic arm slightly higher than the nominal loads. The load limit is mostly defined by the wrist joints of DOF_4 and DOF_5. The proposed solution minimizes the torques on these axes by setting DOF_5 to 0° and DOF_4 to $\pi/2 + k \cdot \pi$, $k \in \{-2, -1, 0, 1\}$. Since DOF_2 and DOF_3 have the same influence on the orientation of the TCP, DOF_2 was set to -90° to get maximum ground clearance but does not have to be static.

So, the robotic arm has only three DOF left (DOF_1, DOF_3 and DOF_6) to set the orientation and at the same time the position of the Client satellite.

In this approach the first step is to compute the relative pose between Client and Servicer $T^{Servicer}_{Client}$ based on their poses in the ECI coordinate system (1), too. The orientation of the Servicer and the Client satellite are also calculated in the same way (2) and (3).

The position of the Client P^{World}_{Client} has to be calculated, because it directly depends on its orientation. Therefore, the pose of the TCP has to be calculated with (5) and (6) and fed into the three-axes inverse kinematics of the robotic arm to get joint angles for the three variable rotational DOF (9).

$$(DOF_1,\, DOF_3,\, DOF_6) = IK\left(R^{RoboticArm}_{TCP},\, DOF_2,\, DOF_4\right) \quad (9)$$

Then, with all joint angles known, the forward kinematics can be used to calculate the pose of the TCP in the coordinate system of the robotic arm $T_{TCP}^{RoboticArm}$ (10).

$$T_{TCP}^{RoboticArm} = FK(DOF_1, ..., DOF_6) \tag{10}$$

The transformation between TCP and Client has to be multiplied and the result has to be transformed in the world coordinate system to obtain the final pose of the Client satellite T_{Client}^{World}(11).

$$T_{Client}^{World} = T_{RoboticArm}^{World} \cdot T_{TCP}^{RoboticArm} \cdot T_{Client}^{TCP} \tag{11}$$

The pose of the Servicer $T_{Servicer}^{World}$ can then be calculated (12).

$$T_{Servicer}^{World} = T_{Client}^{World} \cdot (T_{Client}^{Servicer})^{-1} \tag{12}$$

The position of the cable robot can then be derived according to the first method (8).

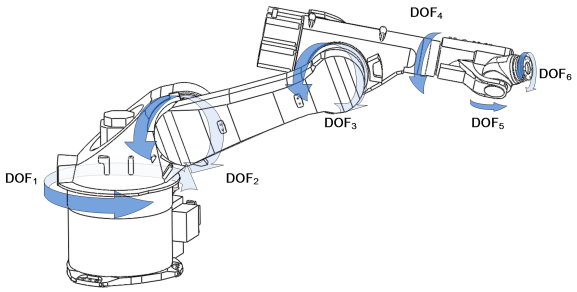


Figure 11: Degrees of Freedom (DOF) of the robotic arm

2.3. Optimized Workspace Utilization

Though the Client has the allowed mass for the robotic arm, its lever is too long so that DOF_4 and DOF_5 of the robotic arm are overloaded in some disadvantageous poses at low speeds. This situation forced us to use the solution using only three DOF of the robotic arm. So, the optimizations of the workspace utilization presented in this section are specific to this case. The improvements use the fact that some DOF are defined by the user at the beginning of the mission simulation but can later be automatically adapted in order to keep the robotic systems away from the limits of their working area.

2.3.1. Automatic Control of DOF_4 and DOF_6 of the Robotic Arm

DOF_4 and DOF_6 are redundant because DOF_5 is kept equal to zero. This means that DOF_4 can be used to extend the range of rotation of DOF_6. When $DOF_4 = \pi/2 + k \cdot \pi$, $k \in \{-2, -1, 0, 1\}$, the weight of the Client satellite is not acting on DOF_4 and DOF_5 since these joint axes are orthogonal to the gravity force vector. Consequently, angles for DOF_4 close to $k \cdot \pi$, $k \in \{-1, 0, 1\}$ have to be avoided, because they could cause an overload on DOF_5, especially at slow rotation speeds.

The proposed solution uses DOF_4 just before DOF_6 reaches its limit. When DOF_6 reaches $-340°$ or $340°$, both DOF_4 and DOF_6 are rotated $180°$ at a high speed in opposite directions to keep DOF_6 away from its limit (Figure 12). This maneuver takes only a few seconds in order to minimize the force on DOF_5 and can be performed up to three times, depending on the initial configuration. It also does not stop the simulation but takes the intended Client mock-up rotation into account, so that the Client's movement does not stop. Thanks to this automatic reconfiguration, the mockup of the Client can perform more than three full revolutions around its longitudinal axis, which increases the mission time without interrupts when simulating a tumbling satellite.

2.3.2. Automatic Control of DOF_2 and DOF_3 of the Robotic Arm

As the working space of the cable robot is limited especially in height, the configuration of the robotic arm can to be adapted to perform a part of the movement in the z-direction and help keeping the cable robot within its limits. For this purpose, the second and third joints of the arm are used. Usually, DOF_2 is fixed and DOF_3 is one of the three axes to rotate the Client. But if both axes are rotated in opposite directions the Client orientation will stay the same. Instead a translational movement will be created which has to be compensated by the cable robot to keep the same relative pose. So, both systems adapt their positions simultaneously. If the elbow is up and overhead solution is taken, an increase of DOF_2 from $-90°$ towards $0°$ will decrease the height of the mock-ups and move them away from the robotic arm. A decrease from $-90°$ towards $-135°$ will also result in less height but will move the mock-ups towards the robotic arm.

A controller has been implemented, that measures the distance of the cable robot to its upper and lower limit and a proportional controller modifies DOF_2 of the arm in a way that the cable robot is kept in the middle of its operation space (Figure 13). The changes in height are limited, but a change of 1 m can make the difference between being able to simulate a certain mission with HIL or not. Especially at long distances, the vertical space is limited due to the artificial Moon crater in the test facility.

2.3.3. Automatic Control of the Vertical Axis of the Cable Robot

The cable robot is equipped with a driven vertical axis, which is able to rotate the Servicer satellite around the world z-axis. On the one hand, this axis can be used to enlarge the operation space of the system by rotating the axis in a way that the Servicer approach can use the diagonal of the operation space. Therefore, the chosen angle γ is taken into consideration in (2) to calculate the pose of the Servicer satellite. On the other hand, the axis can continuously be used to compensate rotation errors of the

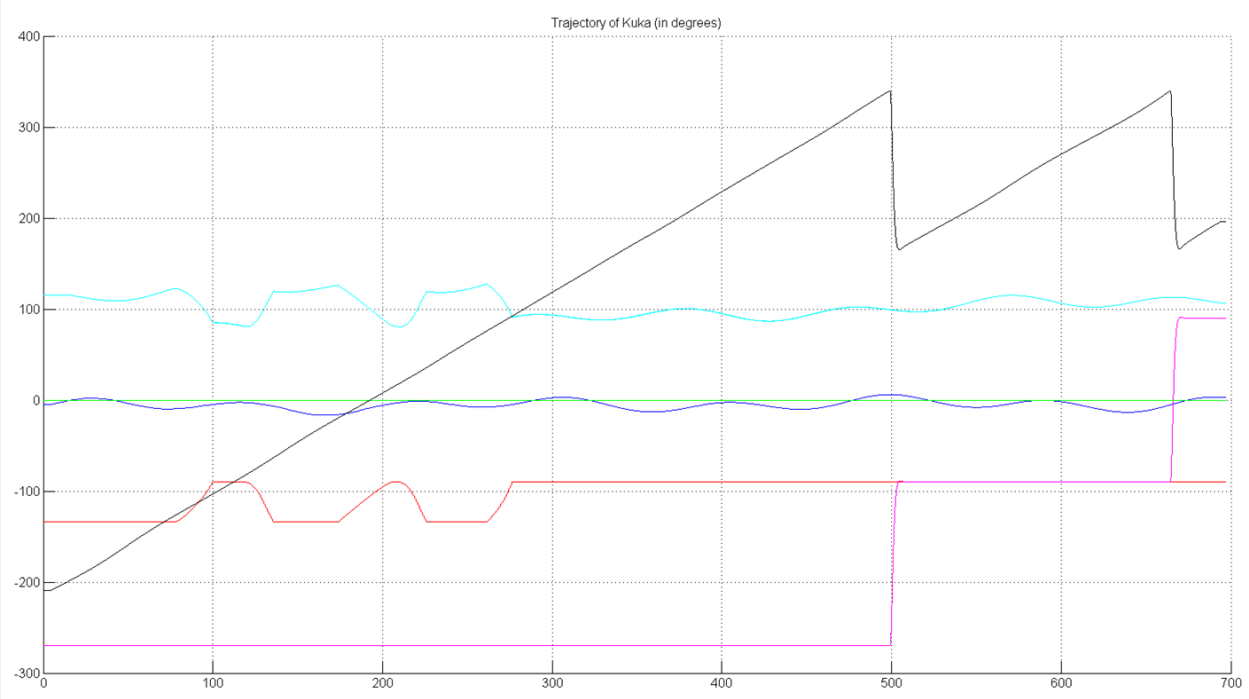


Figure 12: Effects of the automatic control of the fixed axes DOF 4 and DOF 6 of the robotic arm. Angles of the axes of the robotic arm (DOF_2 red, DOF_3 cyan, DOF_4 magenta, DOF_6 black). At time $t = 100\,s$, $t = 200\,s$ or $t = 280\,s$ DOF_2 and DOF_3 rotate in the opposite direction to prevent the cable robot from reaching its vertical limits. At time $t = 500\,s$ and $t = 660\,s$, DOF_4 and DOF_6 rotate in opposite directions to prevent DOF_6 from reaching its limit.

cable robot which depend on its position and are measured by the MTS.

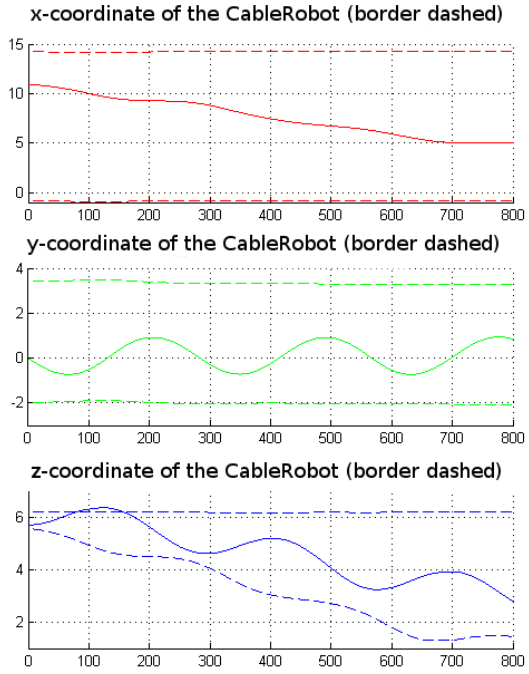
2.4. System Accuracy

The main objective of the testing facility is the ability to reproduce the relative pose of the satellites as accurate as possible. We can define three levels of accuracy depending on the requirements of the experiments.

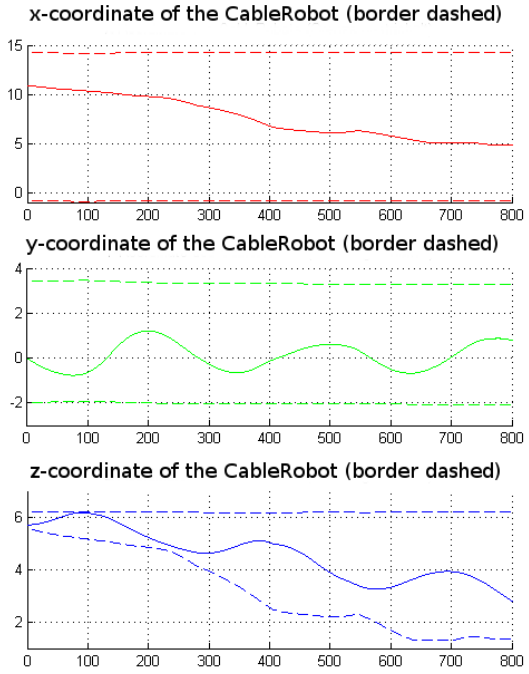
- The first level of accuracy is needed to perform realistic simulations of a mission. This allows to record stereo camera images or LIDAR point clouds in order to develop and test the algorithms for post-processing, feature detection, and pose estimation. For this purpose, accuracy is not very important as long as the performed motions are qualitatively similar to the desired ones.
- As soon as the algorithms prove to work, the second step consists of integrating the sensor results in the control loop. This means that the trajectory of the satellites is dynamically adapted taking the results of the camera or LIDAR algorithms into account. For these requirements, the accuracy should be around a few centimeters and degrees to avoid instability.
- The third level accuracy is required to measure the accuracy of the pose estimation algorithms. In that case, the overall accuracy of the testing facility should be a tenth of the accuracy that can be measured, i.e., the accuracy of the pose estimation of the LIDAR is around $1\,cm$, therefore, a $1\,mm$ accuracy for the INVERITAS system is needed.

To minimize the overall error which leads to inaccuracy, one has to take a look at the different sources where errors may occur (Figure 14) and rate their impact. In general, input data has to be provided by the end-user, i.e., trajectories in a certain coordinate system. Before this data can be processed by the INVERITAS control system, it has to be adapted, e.g. extraction of sub-data sets or interpolation or extrapolation to match different clock rates, which causes minor errors. A second source of error is the DOF reduction. Theoretically, this error is negligible since an analytical solution is used. But because some of the used transformations are based on experimental results or CAD data of the movement systems and mock-ups which can slightly differ from the manufactured versions, such deviations can lead to inaccuracies.

After calculation of the desired angles and positions, the hardware receives the corresponding commands. Between receiving a command and having the command executed with the hardware is a time delay, which causes a dynamic error that increases with the desired speeds. This is superimposed on the absolute inaccuracy of the systems themselves, due to internal sensor inaccuracies or control issues. As stated above, the mock-ups themselves can also



(a) Trajectory for the cable robot would normally cross the limits of the operation area (dotted lines)



(b) The DOF 2 and DOF 3 automatic control helps to keep the cable robot trajectory in the allowed operation area

Figure 13: Effects of the automatic control of the fixed axes DOF 2 and DOF 3 of the robotic arm.

cause errors. Not only can they differ from their CAD models, but they can also deform during operation. The deformation was however determined to be negligible.

Figure 14 shows how the different error sources sum up to the overall error. Besides the raw positioning error, the visual positioning error e_{vis} has a huge influence, which is caused by rotation errors of the sensor-carrying Servicer. These sensors will recognize the target at a different location, than it is supposed to be (Figure 15). This error is proportional to the distance d (13), e.g. a rotational error of the Servicer of $\epsilon = 1°$ at a distance of 8 m causes a visual positioning error of 140 mm. In the next sections we describe how the main error sources are minimized in order to reach an acceptable overall accuracy. We focus on the cable robot system since the robotic arm allows less room for improvements.

$$\begin{aligned} e_{Vis} &= \sqrt{e_1^2 + e_2^2} \\ &= \sqrt{(d \cdot sin(\epsilon))^2 + (d - d \cdot cos(\epsilon))^2} \\ &= d \cdot \sqrt{2 \cdot (1 - cos(\epsilon))} \end{aligned} \tag{13}$$

2.4.1. Cable Robot Accuracy

The inaccuracy of the cable robot in position and orientation was identified as the major source of error in

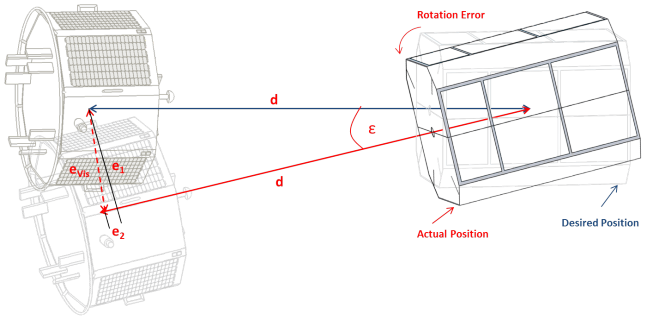


Figure 15: Visual positioning error

our testing facility. The cable robot appears to perform a slightly parabolic curve when commanded to perform a straight line. It also rotates around all three axes whereas it should always keep a vertical configuration. Since the cable robot is controlled via proprietary software, we have no access to it and cannot improve the internal models of the winches, the cables, and the mount.

Instead, the position accuracy of the cable robot has been measured with a laser tracker [10] according to the standard test methods defined in [12]. The cable robot had to perform a five-point-trajectory 30 times. These waypoints are placed on an inclined plane that covers the core working area of the cable robot. Four of them form

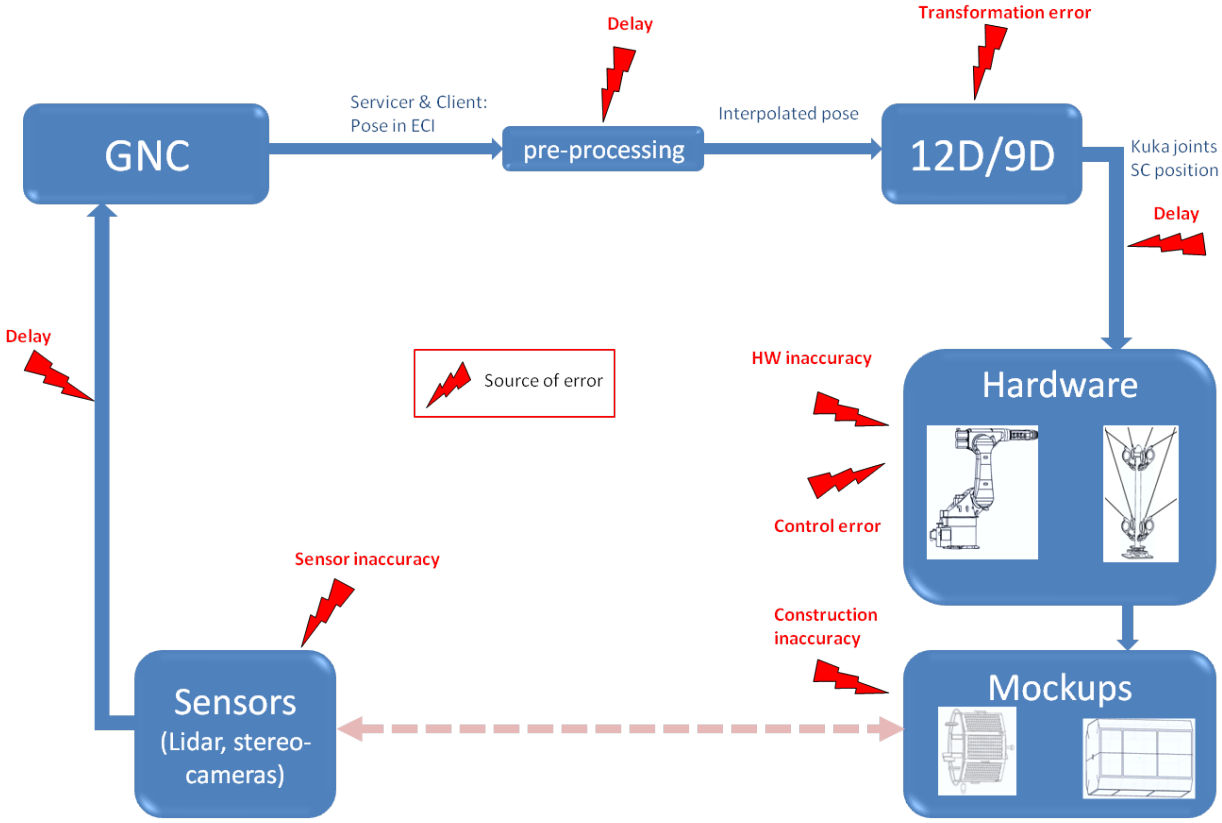


Figure 14: Potential error sources

a square, the fifth one is placed in the center. The results from this test series confirmed that the absolute accuracy was insufficient, almost 270 mm, but that the repeatability was very good with 1.1 mm.

Because of the good repeatability, it was possible to measure the error in position and rotation in the whole working area and approximate a 3D polynomial function which minimizes the overall error. This approximation can then be used inside the control loop to compensate the undesired errors.

Calibration Trajectory. A calibration trajectory covering most of the working area was defined as shown in Figure 16(a). This trajectory was generated with limited jerk [13] in order to minimize the dynamic effects and to prevent the cable robot from oscillating (Figure 16(b)). The cable robot position is recorded by the MTS and compared to the position given by the proprietary cable robot software. The undesired rotation of the mount is also measured.

Then, the error in absolute position (x, y, and z) and absolute orientation (using XYZ-Euler angles a, b, and c) are approximated using a third order 3D polynomial function. We have to solve for each data point of coordinate x, y, and z and for each error $\epsilon_j, j \in \{x, y, z, a, b, c\}$ the linear equation (14).

$$\epsilon_j = [x^3\ x^2y\ x^2z\ xy^2\ xyz\ xz^2\ y^3\ y^2z\ yz^2 \\ z^3\ x^2\ xy\ xz\ y^2\ yz\ z^2\ x\ y\ z\ 1] \begin{bmatrix} p^j_{19} \\ \dots \\ p^j_0 \end{bmatrix} \tag{14}$$

Combining the n equations of the n trajectory points, we obtain (15) for $j \in \{x, y, z, a, b, c\}$:

$$E_j = V \cdot P_j \tag{15}$$

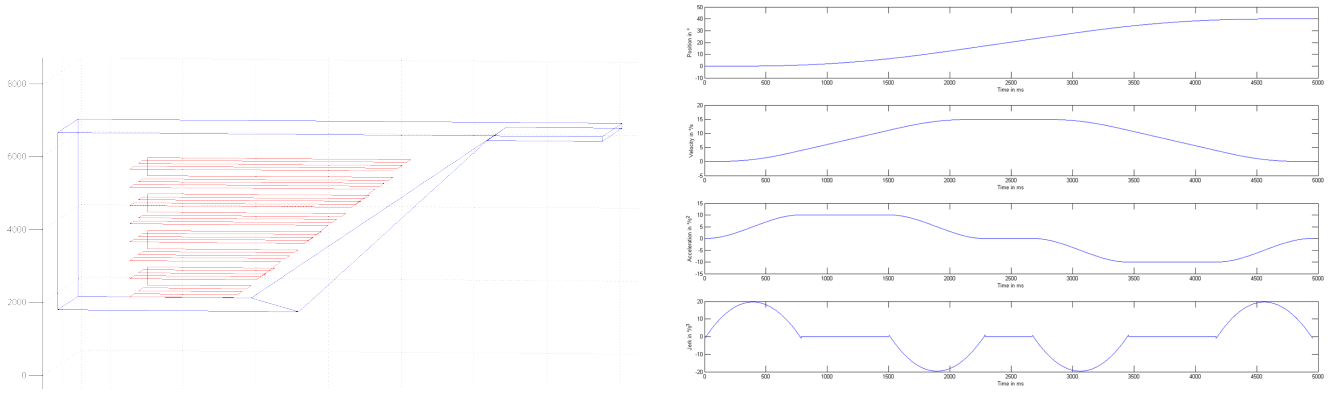
With

$$E_j = \begin{bmatrix} \epsilon^j_1 \\ \dots \\ \epsilon^j_n \end{bmatrix} \tag{16}$$

And

$$V = \begin{bmatrix} x_1^3\ x_1^2y_1\ x_1^2z_1\ \dots\ z_1^2\ x_1\ y_1\ z_1\ 1 \\ \dots \\ x_n^3\ x_n^2y_n\ x_n^2z_n\ \dots\ z_n^2\ x_n\ y_n\ z_n\ 1 \end{bmatrix} \tag{17}$$

The solution of this overdetermined linear equation is obtained using the QR decomposition [14] of the matrix V. We derive the coefficients of the polynomials that approximate, in a least square sense, the error in position or orientation of the cable robot.



(a) The trajectory covers most of the operation area

(b) A jerk bounded trajectory is used to minimize oscillations during acceleration and deceleration

Figure 16: Calibration Trajectory

Position Error Compensation. The position of the cable robot can be estimated in real time using the polynomial function to predict the error.(18).

$$\hat{X} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = \begin{bmatrix} x_{cable} + \hat{\epsilon_x} \\ y_{cable} + \hat{\epsilon_y} \\ z_{cable} + \hat{\epsilon_z} \end{bmatrix} = \begin{bmatrix} x_{cable} + \begin{bmatrix} x_{cable}^3 & \dots & y_{cable} & z_{cable} & 1 \end{bmatrix} \cdot P_x \\ y_{cable} + \begin{bmatrix} x_{cable}^3 & \dots & y_{cable} & z_{cable} & 1 \end{bmatrix} \cdot P_y \\ z_{cable} + \begin{bmatrix} x_{cable}^3 & \dots & y_{cable} & z_{cable} & 1 \end{bmatrix} \cdot P_z \end{bmatrix} \tag{18}$$

Figure 17 shows how the cable robot performs parabolic movements instead of staying on one horizontal plane. The red lines represent the experimental data measured by the MTS and the surface shows the estimation of the z-position using the polynomial function. We can see on Figure 17 that the errors are higher when approaching the side of the working area, where the forces applied to the cable are most asymmetrical.

The control loop of the cable robot was then modified in order to use the position estimated using the correction polynomial instead of using directly the values given by the cable robot software. Then the accuracy was tested again with the result that the cable robot's absolute accuracy was dramatically improved (Table 2). The same algorithms were simultaneously applied for the MTS data, which resulted in small improvements. This experiment also showed that the MTS is most accurate at the center of its operation space and becomes continuously less accurate towards its edges .

Rotation Error Compensation. Figure 18 shows the angle rotation of the cable robot when moving on a horizontal plane. We can notice that the rotation can reach 14° which would cause a visual positioning error of almost $2\,m$ at a distance of $8\,m$. Consequently, this undesired rotation of the cable robot needs to be compensated.

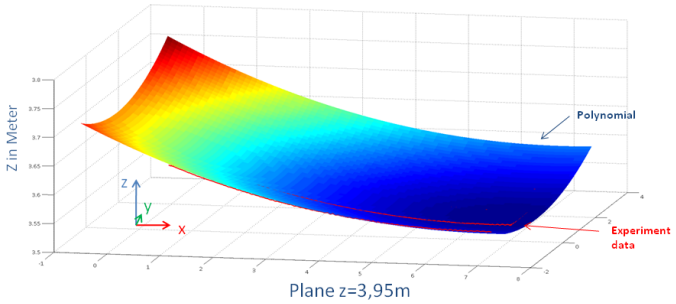


Figure 17: Position correction polynomial for one particular height of the cable robot

As the rotation of the cable robot is undesired and cannot be corrected in all directions even with the help of the movable vertical z-axis, the robotic arm has to compensate for the rotation of the cable robot. This means that the rotation of the cable robot influences the whole configuration of the robotic systems in the Space Exploration Hall. The computation of the rotation matrix of the Servicer described in (2) has to be modified accordingly (19). In this way, the DOF reduction automatically compensates the undesired cable robot rotation.

$$R_{Servicer}^{World} = R_{Cable\,robot}^{World} \cdot R_{Servicer}^{Cable\,robot} \tag{19}$$

Since the compensation of the rotation of the cable robot implies a change of its position, this could lead to instability of the system. But because the estimation of the rotation is obtained with a 3D polynomial function, the evolution of the rotation is continuous and continuously derivable. The polynomial function appears also to be bijective within the working area of the cable robot. This last assumption was nonetheless not further investigated or proved. The whole system converges thus after a few seconds and stays stable. This means that a short delay of a few seconds is necessary until the system has reached a stable initial configuration for both the robotic arm and

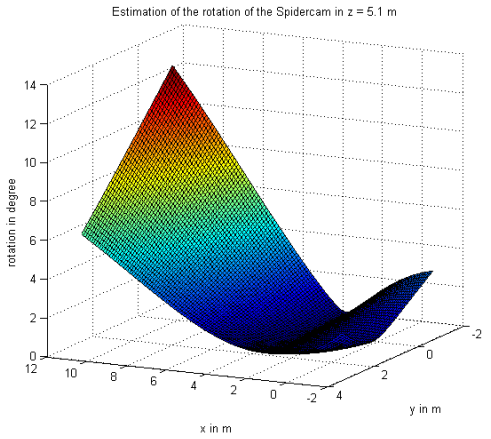


Figure 18: Rotation correction polynomial for one particular height of the cable robot

the cable robot before starting the trajectory. Afterwards, the movements are slow and continuous, which allows the system to stay stable and adapt the configuration in real-time.

Sensor Fusion. The position of the cable robot is measured internally using the cable lengths and additionally with the MTS, whereas the orientation is only measured by the MTS because changes in the rotation during a HIL-simulation experiment are unintended and assumed to be zero by the internal measurement. In the previous sections it was shown that the internal measurement and rotation assumption have to be corrected with polynomials to improve accuracy. The result right after calibration is an accurate and smooth position estimation. The drawback is the effort of calibration, which can take an hour at best. The MTS pose measurement on the other side is precise and always delivers up-to-date information. The drawbacks of the MTS are that it does not cover the whole operation area of the cable robot and its accuracy and noise depends on the position of the measured object, where the center of the operation space has better accuracy and lower noise than a position measurement close to the border of the measurement volume. Thus, relying on the MTS is desired in the center of the operation space, whereas the internal measurements of the cable robot combined with correction polynomials are necessary at the edges or were no MTS data is available at all. A Kalman filter is a good choice to fuse the sensor information in this manner and to provide smooth transitions between them.

First, a Kalman filter with constant covariance matrices was implemented. The model used for the prediction step is the following: the position is derived knowing the previous position and the relative displacement command sent to the cable robot software, and the rotation angles are modeled as constants. The observation step is done at each timestamp using on one side the polynomial function and on the other side the value measured by the MTS. The covariances, which characterize the confidence in the measurements, were chosen knowing the absolute position accuracy (Table 2) of both information sources.

Figure 19 shows the rotation of the cable robot during a trajectory, where the cable robot starts at the edge of the MTS working space and moves towards its center. At the beginning of the trajectory, the MTS data (blue) is probably on average the best, but has a lot of noise which would cause oscillating compensation movements of the cable robot when the rotation compensation is active which finally leads to instability. The provided data of the correction polynomial (red) are very smooth, but can be inaccurate due to unnoticed changes in the experiment setup and environment influences as well as due to an insufficient polynomial approximation. The result of the standard Kalman filter with constant trust values (cyan) will have more accuracy than the correction polynomial but filters just partially the noise. The more it trusts the MTS data, the more accurate it is while increasing the noise.

An improved version of the Kalman filter was implemented, using position-varying covariances for the MTS data. The covariance was defined proportionally to the standard deviation of the input signals. In this way, the MTS data is trusted less at the edge of the operation space and trusted more in its center (Figure 19 - magenta line). But this approach does not help to increase the accuracy at the edges. So, a further implementation of the Kalman filter was developed which filters the input data of the MTS with a Median filter before processing it in the Kalman filter itself (Figure 19 - black line). Consequently, the noise in the MTS data is lowered and the trust increased. The size of the Median filter determines the input noise but also generates a delay. It was possible to find an appropriate filter size which makes the last proposed Kalman filter best suited for the application.

2.4.2. Robotic Arm Calibration and Accuracy

The pose of the robotic arm with respect to the world coordinate system was experimentally measured using a laser tracker [10]. For this calibration, the robotic arm had to reach four different points of its working space, chosen in a way to have the maximal distance between them. The best fitting (in a least square way) transformation matrix between the world coordinate system and the robotic arm coordinate system was then derived from those four points.

The absolute accuracy of the robotic arm was also measured according to the standard test methods defined in [12], but without payload. The experiments showed an absolute accuracy of the arm of 1.6 mm and a repeatability of of 0.01 mm which corresponds to the technical data sheet of the arm.

2.5. Autonomous approach of a servicer satellite towards an uncooperative tumbling satellite

The INVERITAS test facility was used to verify sensor processing and GNC algorithms, which should enable a

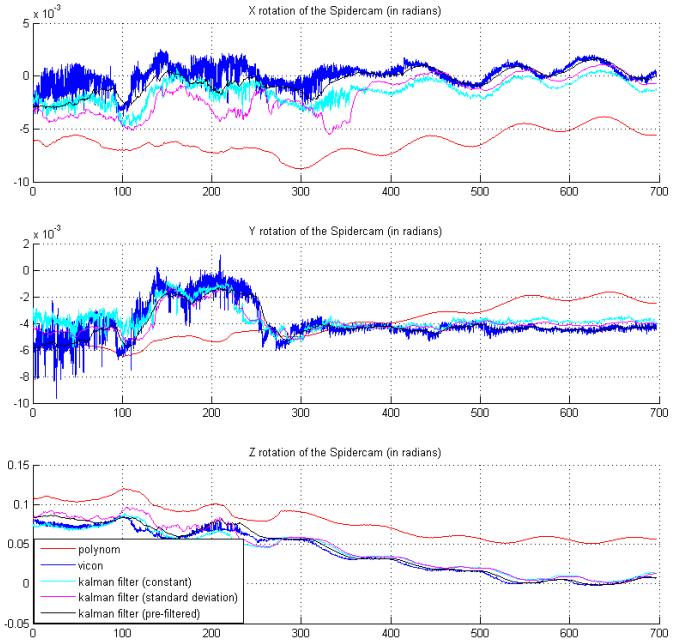


Figure 19: Sensor fusion with different variations of Kalman filtering

Servicer satellite to autonomously approach a Client satellite. Two operation modes were tested. In the open-loop mode, Servicer and Client satellite follow predefined trajectories, which represent a realistic mission scenario. This mode is used to collect sensor information to test sensor models and algorithms like line of sight or pose estimation with camera systems and a LIDAR. In the closed-loop mode, the GNC is fed by the sensor information of the Servicer which leads to possible pose corrections. A desired pose correction is internally mapped to thruster activations, which are transformed into new Earth-Centered Inertial (ECI) poses by the orbital dynamic simulation.

2.5.1. Servicer GNC

The following section gives a brief overview of the design of the Servicer GNC system with focus on the close range navigation in the vicinity of an uncooperative target and the rendezvous sensors, which were the first test subject of the INVERITAS robotic test facility.

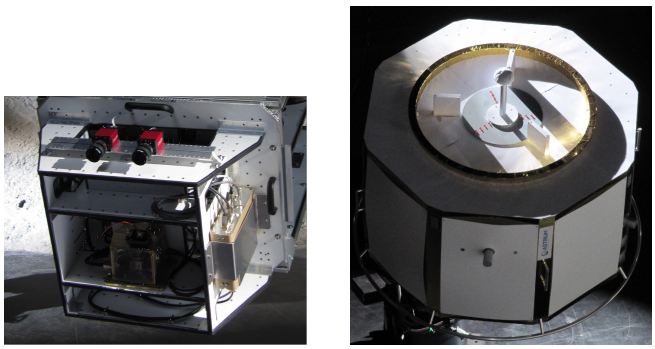
Outline of the GNC System. The GNC system comprises sensors, actuators and the software controlling the vehicle position and attitude in space and relative to an uncooperative, potentially tumbling target. The software is in charge of processing sensor measurements, computation of the flight trajectory, navigation, attitude and position control as well as flight control and safety monitoring of the servicer space-craft. Compared to existing rendezvous and docking systems as applied for ATV (Automated Transfer Vehicle), the new system is able to perform relative position and attitude control with respect to a non-cooperative passive and tumbling target on the basis of optical measurements. Hereafter the basic ideas and the preliminary performance of the major modules are presented.

Guidance. For the tests in the INVERITAS facility, only the last meters of the approach were investigated due to the size limitations of the facility. Therefore, the guidance starts at a distance of approximately 10 m. At this point, usually an inspection flight has already been performed in order to determine the target spacecraft's integrity. After the inspection a hold point on the *V-Bar* about 10 m behind the target is acquired. In this point the navigation sensor is switched to a mode where initial pose estimation is performed. Based on the inspection results the approach direction towards the target is defined. The V-Bar approach is advantageous in terms of complexity, flexibility (easy introduction of further hold point), and fuel consumption. Therefore it is selected whenever possible. Small compact targets without large appendices can be captured by a manipulator system. By appropriate impedance control the target angular momentum can be absorbed by the manipulator system and after rigidization the coupled system is stabilized and controlled by the chaser GNC (this is the reference case of the present GNC). For larger tumbling bodies with appendices the chaser motion needs to be synchronized with the target attitude motion. Since the necessary forces to control a synchronized motion increase with tumbling rate and distance to the target center of mass (centripetal and Coriolis forces) the distance should be kept small (typically $< 5\,m$). The last meter approach is then performed along a straight line w.r.t. the target grappling point

Control. The control relies to a large extent on proven methods from the ATV design. The attitude control is performed with a configurable PID controller. The same principle holds for the position control in close range. H_∞ control may be applied if stronger robust performance requirements arise from the manipulator system. Recent studies (e.g. [15]) propose a MPC (model predictive control) controller also for rendezvous, however for the cost of very high CPU load. The resulting force/torque command vector is processed by a thruster management function to find the best (fuel minimizing) combination of thruster firing for its realization.

Primary Navigation Sensor. The navigation for the close-range is mainly based on the measurements of a 3D-LIDAR sensor which is able to provide a 3D representation of the object in front of the chaser. For this experiment, a prototypic 3D-LIDAR developed by JenaOptronik GmbH, was used. Originally this sensor was not designed to be used for non-cooperative targets but was originally foreseen for the detection of retro-reflective targets such as corner-cubes. Therefore, the laser-power of the used 3D-LIDAR was too small to be used for real non-cooperative targets that use highly reflective materials like multi-layer-insulation (MLI) or GaAs solar cells. This is because the laser rays get reflected to a large part with specular reflection from such materials so that those rays never go back

to the LIDAR. Since the diffuse reflection of these materials is quite small, a lot more laser power is needed to receive more reflected rays with the LIDAR. However, in order to show the basic operation principles, the existing 3D-LIDAR (see Figure 20(a)) and a modified target covered by diffusive foils (see Figure 20(b)) have been used to compensate the lack of sufficient laser power of the 3D-LIDAR. Furthermore, the scanning speed of the 3D-LIDAR was not as fast as needed for real missions. Therefore, the tumbling-rate of the target object was reduced to fit the scan frequency of approximately 1 Hz. Typical scans of the used 3D-LIDAR have a number of 5000 scan points distributed over a field of view of $20° \times 20°$ along a sinusoidal scan pattern which is the result of the motion of a gimbal mounted mirror which performs oscillations in azimuth and elevation directions.



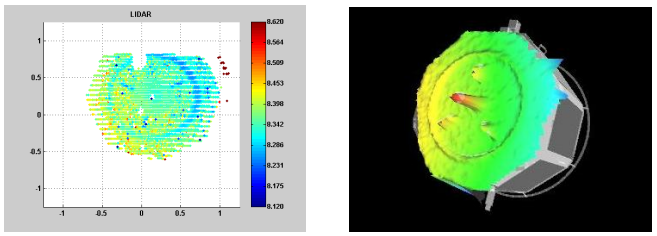
(a) Servicer mock-up equipped with stereo cameras and a LIDAR

(b) Mock-up of the target satellite covered with diffusive reflective material

Figure 20: Mock-ups of the servicer and the client satellites

Vision-Based Navigation. The vision-based navigation is one of the core-elements of the close-range rendezvous due to the fact that a high precision of the target position and attitude is required to perform rendezvous but also to enable following robotic manipulations like capturing. For this purpose, the position of the center of mass of the target is required and not only the centroid of all 3D-LIDAR measurements which might by several tens of centimeters or even meters away from the center of mass. The vision-based navigation which was one of the test subjects in the INVERITAS test facility is based on measurements provided by the 3D-LIDAR as described in the previous section. Since the target must be assumed to be non-cooperative (i.e. no visual markers and no attitude control) but not as unknown, it is possible to use the geometric appearance of the target as the essential feature that bears the position and attitude of the object. Therefore a model-based approach was chosen to estimate the pose of the target object. The 3D-LIDAR based pose-estimation is divided into two main stages: The pose-initialization and the pose-tracking.

Pose-Tracking Assumed that an initial pose of the target is known, the pose can be tracked over time very robustly by application of the iterative closest point algorithm (ICP) [16]. For this purpose, a model of the target object is used to be registered with the 3D point cloud acquired by the 3D-LIDAR. We use a specific version of the ICP algorithm that matches 3D points of the 3D-LIDAR with a model consisting of planar patches, lines and points [17]. Especially the usage of planar patches provides better accuracies compared to the point-to-point ICP due to the avoidance of a discretization of the model which imposes additional errors. A critical aspect is the limited field of view of the 3D-LIDAR which was used for the experiments. Since the geometric structure of the object is the only relevant pose-bearing feature, very close distances in connection with a small field of view can lead to single planes in the field of view which can no longer define all degrees of freedom of the relative pose. However, the exploitation of the knowledge of boundary points can solve most of the cases where only small portions of the boundary of the objects are visible. An increase of the field of view of the 3D-LIDAR can dramatically reduce this difficulty. In order to survive situations where only few geometric structures are available in the field of view, a Kalman filter is introduced modeling the system dynamics of a free-tumbling target object. The Euler-equations describe the rotational movement of the object. A sufficiently long observation of the object with larger distances between sensor and target enables the convergence of the Kalman filter. Once the filter converged, phases with limited view of the object and less geometric structures help surviving critical situations.



(a) Range data acquired by the 3D-LIDAR

(b) Match between 3D-LIDAR scan and the 3D model of the client

Figure 21: Raw data of the LIDAR and match between this data and the a priori model of the satellite

Pose-Initialization Before starting with the pose-tracking, a rough initial pose-estimation is essential for the convergence of the following tracking by means of the ICP algorithm. Therefore, a pose-initialization from scratch is necessary. Again, the 3D measurements of the 3D-LIDAR are used to solve this task. This choice has the advantage of being independent from illumination conditions. The proposed method applies a view-based approach. For this

purpose, a method similar to well-known correlation techniques from 2D template matching was developed for 3D templates. The proposed method exploits specific properties of the pose-initialization task:

- The object is known and can be represented as a model consisting of patches, lines and points.
- The object consists only of a single object in the field of view. At least in 3D-data the object-background separation is already solved.
- The 3D-sensor provides spatial measurements which especially contain depth information.

All three properties can be exploited for an algorithmic search of the target pose which we call *3D-Template-Matching*. While 2D-template matching methods [5] suffer from the lack of rotation and scale invariance and the lack of sufficient robustness against changing illumination conditions, 3D-Template-Matching does not show these drawbacks and thus can be used quite efficiently. The 3D-LIDAR provides 3D point clouds which are independent from illumination conditions. The missing scale invariance of 2D template matching can be compensated by using the knowledge about the range values measured by the 3D-LIDAR. In case of complete view of the target object, even the full 3D-position of the object can be defined by the position of the centroid of the 3D point cloud (at least for a specific attitude). Thus, only 3 degrees of freedom (i.e. the three angles of the attitude) are subject to a search in the search space. The 3D template matching consists of the following main steps:

- The 3D-point cloud is converted into a small matrix representing a depth image for a specific object attitude.
- For many different attitudes, similar views are represented in a model database.
- For every template of the database, the centroid of the 3D point cloud and its offset to the center of mass of the objects are stored.
- An orthographic projection of a view into a matrix of fixed size and discretization enables scale invariance.
- The storage of the offset between center of mass and the 3D-point cloud's centroid provides translation invariance.

Thus, the search of three of six degrees of freedom can be avoided. The search for the remaining 3 rotational degrees of freedom can be solved by storing many different reference views in a database of small reference templates. The templates do not have to be very large and detailed. For the experiments with the satellite mock-up of the INVERITAS facility, small reference views of 30x30 pixels were used. If the object shows further symmetries, the number of different attitudes can be further reduced. Two of the three attitude angles can be represented as points on a unit sphere whereas the third attitude angle is a rotation around the direction given by the spherical point. Figure 23 shows two different discretizations of the unit sphere representing different resolutions of the model database.

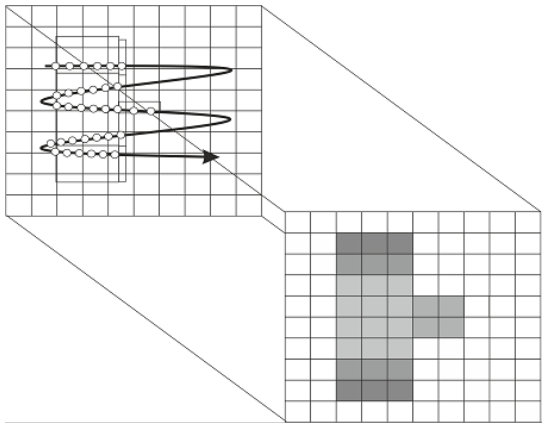


Figure 22: Generation of a small template from sinusoidal scan pattern provided by the 3D-LIDAR

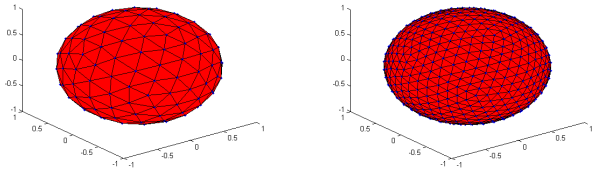


Figure 23: Corner points of a polyhedron span viewing directions for the computation of reference views. Two different resolutions are shown

The initial search of the target attitude and position can now be performed by comparing all reference views of the model database with the small template generated from the most recent 3D-LIDAR point cloud.

The first criterion for comparison between templates is the size of the bounding boxes of the 3D point clouds. This allows pruning of the search space with dramatic consequences for the computational loads. If the current view has a larger bounding box than the one stored in the database, the two cannot match. If the sizes correspond to each other, a sum of squared differences cost function can be computed. If it shall be possible to work on partial views of the object, the bounding box of the current view can also be smaller than the one stored in the database. This is the most time consuming step and does not need only the computation of the sum of squared differences of two templates but also requires the computation of the correct offset, i.e. 2D correlation is needed. The views of the model database always contain complete views. In this case, the best correlation value (or the minimum of

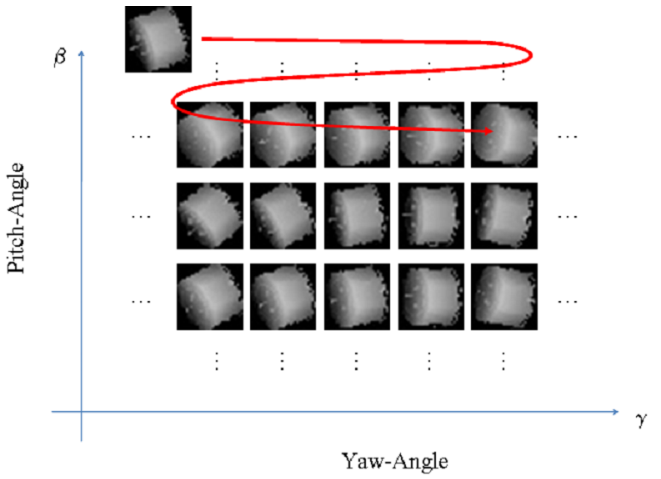


Figure 24: Principle of comparison between generated template from the LIDAR scan and the previously generated templates of the model database

	Absolute accuracy	Repeatability
Manipulator arm (no correction)	1.6 mm	0.01 mm
Cable robot without correction	267.4 mm	1.1 mm
Cable robot with correction	6.6 mm	1.1 mm
MTS without correction	8.9 mm	0.6 mm
MTS with correction	5.3 mm	0.6 mm

Table 2: Absolute accuracy and repeatability of the robotic arm, the cable robot and the MTS

the sum of squared differences) is regarded as the result of the comparison between the two templates. Finally, the best matching result gives a rough estimate of the attitude of the object and thanks to the known offsets between the center of mass and the centroids for every reference view also the initial position can be obtained. The resolution of the viewing angles represented in the template database depends on the required accuracy that assures convergence of the ICP algorithm. For the experiments shown in chapter 3.2, an angular step width of 10° was sufficient.

3. Results

3.1. Accuracy of the Test Facility

The accuracies of the different systems of the test facility were measured separately and the results are gathered in Table 2. However these values do not represent the overall accuracy of the system when reproducing a rendezvous scenario. As previously explained, the rotation of the cable robot has a strong impact on the visual positioning error, which is for the sensors the same as a physical positioning error of the mock-ups. The delay introduced by the control loop between the desired and the actual position or joint angles has also to be taken into account.

When performing a mission scenario, the original desired pose of the Client satellite, with respect to the Servicer has then to be compared with the current relative pose of the mock-ups in the Space Exploration Hall. The current position and orientation of the mock-ups is derived either from the joint angles given by the robotic arm or from the fused information given by the MTS. The accuracy experiments show that the cable robot can reach the desired position with a sub-centimeter accuracy. If we take the accuracy of the cable robot when we use the MTS, add the error of the robotic arm, and all other error sources, we obtain a position accuracy of around 22 mm. Since the undesired rotation can be compensated up to 0.2° on average, a visual position error of approx. 21 mm (for a maximal distance of 8 m) is left. The resulting overall accuracy is then above 4 cm (Figure 25).

These results show that the test facility reaches the second level of accuracy, as described above in chapter 2.4. This allows us to conduct open-loop experiments for visual navigation algorithms and closed-loop scenarios, where the results of the visual navigation algorithms are used in real-time to adapt the trajectory of the Servicer.

In order to further improve the accuracy of the test facility and be able to reach the third level of accuracy, several improvements are possible. An extension of the MTS or its replacement by a more accurate system can greatly improve the measurement of the cable robot pose and a direct tracking of the mock-ups can also eliminate some sources of errors. Improvements in the construction of the mock-ups are also necessary in order to avoid distortions.

3.2. Results of sensor processing and GNC tests

3.2.1. Test scenarios

Several mission scenarios were generated and tested on the INVERITAS system. These missions start at a distance of approx. 9 m and are 600 s to 2000 s long. They cover three main types of situations:

- *station keeping* at a distance of 9 m. Client stable or tumbling (rotation speed up to 0.3°/s).
- *straight approach.* Client stable or tumbling (rotation speed up to 0.3°/s).
- *fly around*

Figure 26 presents an image sequence from a typical straight approach from 9 m to 3 m, with a tumbling Client satellite. Each scenario starts with approximately three minutes of

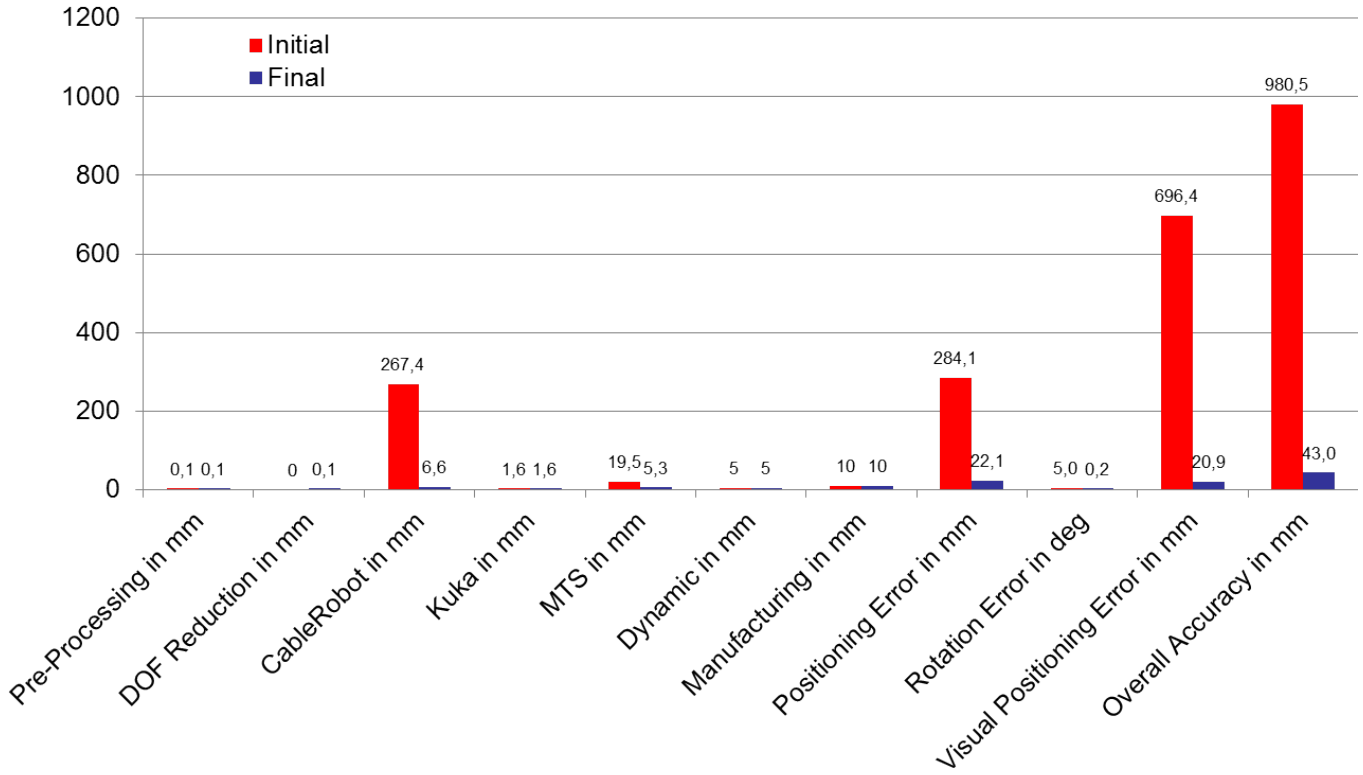


Figure 25: Accuracy of the INVERITAS system with and without improvements

station keeping to allow the VN algorithms to initialize and stabilize.

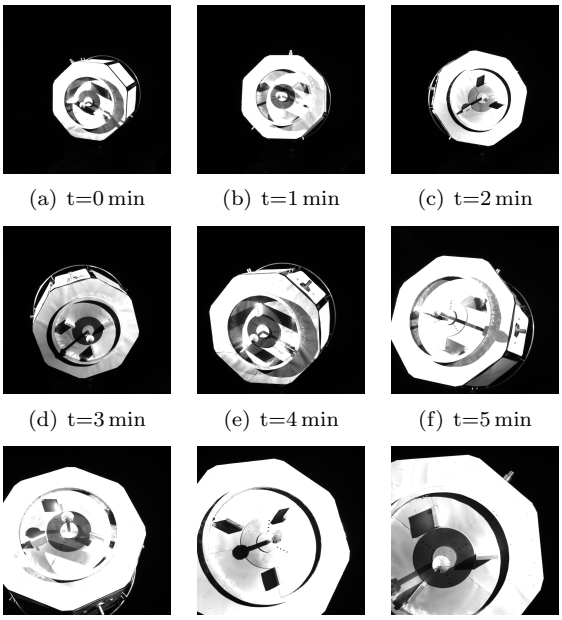
3.2.2. Setup parameters

Since the trajectories were pre-generated, it was possible to check in advance their feasibility and to adapt the setup parameters. Since the VN algorithms need a certain time to initialize, the objective was to maximize the length of the mission that can be simulated without interruption. In the case of a tumbling Client, the redundancy of DOF_4 and DOF_6 of the robotic arm was used to increase the length of the mission as described above in section 2.3.1. The starting angles for DOF_4 and DOF_6 were chosen in 180°-steps in order to extend the range of rotation of the Client. The variable parameters for DOF_2, DOF_3, and the vertical axis of the cable robot were tuned as well to keep the cable robot as far as possible from its limits. Tuning these parameters allowed to simulate between 85 and 100 percent of the missions without having to reconfigure. However, mission scenarios that demand Client rotations of more than 1400° need to be divided into several sections. The simulation stops at the end of a section and a reconfiguration is performed, for example by rotating DOF_4 and DOF_6 away from their joint limits, then the simulation process is resumed.

3.2.3. Experiment results

The first series of experiments were done in an open-loop mode: the visual navigation results were not used to adapt the trajectory of the Servicer in real-time. This phase allowed to test the visual navigation algorithms and to fine tune the different parameters. Once the visual navigation proved its reliability, the simulation was switched to closed-loop. In this case, the GNC system used the estimated pose of the Client satellite to modify the approach towards the satellite.

Figure 27 shows the results of the closed-loop experiments. The first three diagrams show the results of the position estimation while the last three diagrams show the errors of the positions. The blue line (marked as "ideal") depicts the positions as computed by the guidance function and thus represents the ideal position. The red lines represent the state of the space-craft (i.e. positions) as estimated by the GNC system based on ideal measurements of a behavior model of the navigation. The estimated states of the GNC system based on real measurements of the vision-based navigation (VN) are depicted by the green lines. The last three diagrams show the errors between ideal positions and the GNC states for simulated and real measurements of the VN system respectively. It could be demonstrated that the implemented system is able to successfully perform the close-range rendezvous maneuver based on the real sensor measurements of the 3D-LIDAR although the results were significantly worse



(a) t=0 min (b) t=1 min (c) t=2 min

(d) t=3 min (e) t=4 min (f) t=5 min

(g) t=6 min (h) t=7 min (i) t=8 min

Figure 26: View of the Servicer's right camera of its mid-range stereo camera system while approaching the tumbling Client

than predicted by an ideal sensor model. The errors of the VN system are in the order of several centimeters and the errors of the attitude angles (not shown here) are in the order of 1° for roll and pitch angles and $< 5°$ for the yaw angle (with the z-axis being the bore-sight direction). However, the measurement results also include some inaccuracies of the test facility itself. It has been observed that the accuracy of the VN system and the ground-truth measurements of the facility are in the same order of magnitude.

4. Discussion

4.1. Conclusions

In this paper we presented the abilities and architecture of the INVERITAS HIL long distance movement simulation system. The facility allows the relative pose simulation of two freely moving objects such as satellites with a maximal approach distance of $16.5\,m$.

We gave an overview of the hard- and software components and showed how a cable robot can be used to utilize the given space without interfering with the other experimental testbeds present in the facility. In addition, the software architecture of the system was described. We presented our solutions for reducing the 12 DOF of the two satellites to the constrained 10 DOF of the INVERITAS system while efficiently using the limited space in the Space Exploration Hall. This enables us to simulate diverse close-range approaches in our HIL simulator.

The improvements made on the systems, using offline calibration and sensor fusion, allowed us to reach a $6.6\,mm$ absolute position accuracy for the cable robot. The final overall accuracy, including the visual positioning errors, is $4\,cm$.

The testbed could be successfully used to test sensors such as LIDAR and stereo cameras during various mission scenarios. Visual servoing algorithms were tested in real-time, first open-loop and finally in closed loop.

The system's modular and flexible architecture enables us to keep using and improving the system in a wide range of other projects, which prevents the system from being a specialized solution for a single purpose.

4.2. General Usability for Rendezvous Tests

The INVERITAS HIL simulator is adaptable in use, making the simulation of a wide range of HIL rendezvous maneuvers possible. A user of the facility has to provide the mock-up to be picked up by sensors, the sensor hardware and binary Matlab/Simulink models for the behaviors of the objects. As only binaries have to be provided, a user does not have to provide the source codes of the the Matlab/Simulink models. The cable robot mount provides 230V AC at 1kW and 24V DC as well as a data connection of 10 Gbit/s via optical fiber, which can both be used for the sensors and connected data processing hardware.

4.3. Outlook

The INVERITAS simulator is part of planned as well as already running successor projects. In these projects we are continuing to improve as well as extending the system to increase the scope of possible uses for the system. In the follow-up project RTES-TA[9] we developed methods to further increase the versatility and accuracy of the system. New measurement systems like 6 DOF laser trackers or optical indoor GPS can be used to significantly increase the accuracy of the system, which remains an important factor for the usability of the system in possible future uses and projects. Until now, the Client was controlled open-loop by relying on the precision of the robotic arm. In future, the pose of the Client has to be tracked online with a measurement system which does not interfere with sensors of the Servicer.

Future plans also involve adding more moving objects in the HIL simulation, as some possible applications already show that for example simulating relative movements of three objects can be useful. Another possible use for moving more than two objects is the simulation of a realistic sunlight direction which adapts to the movement of the other objects by attaching the light source to one of the movement systems.

Additionally, increased allowed weights of Servicer and Client by using stronger manipulator systems are desirable, so that larger, heavier mock-ups with more realistic materials can be used.

[9]Funded by the Space Agency (DLR Agentur), acting on a mandate from the Federal Government, grant no. 50RA1224

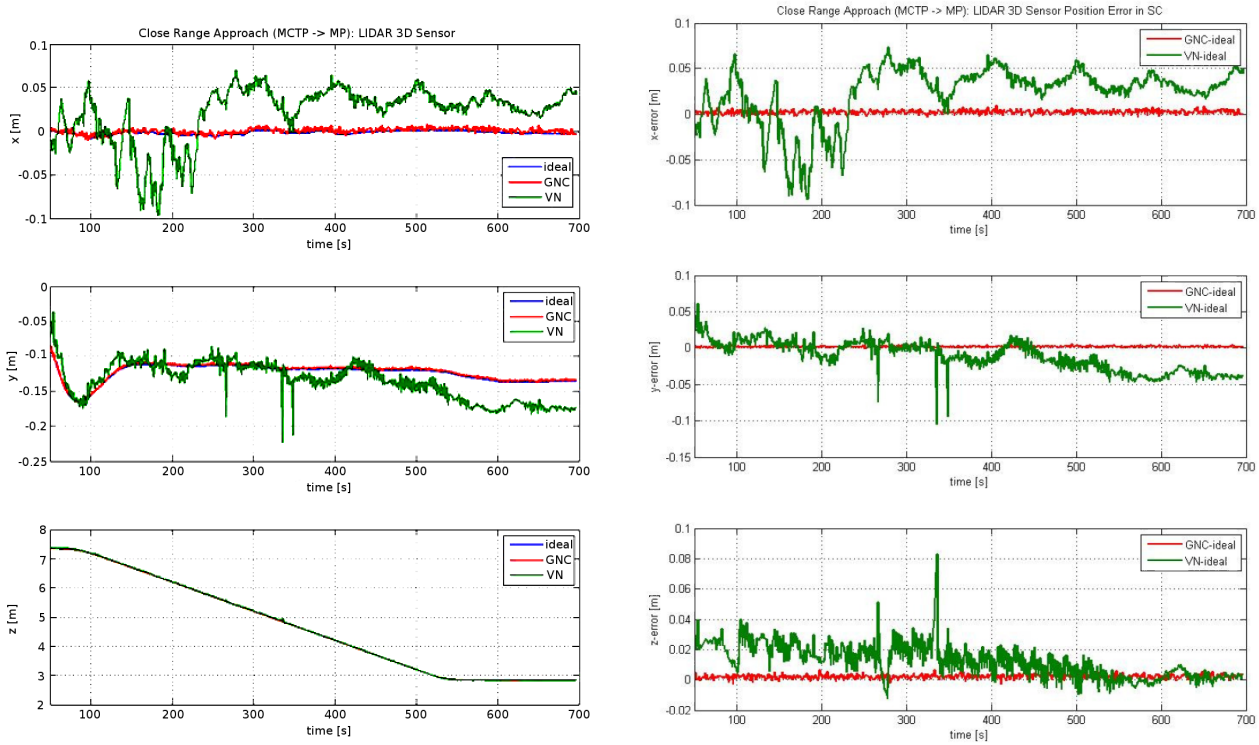


Figure 27: Results of the closed-loop experiments for close-range rendezvous in the test-facility

The system has the flexibility to be used in other future projects to simulate rendezvous and docking maneuvers between a wide range of autonomous objects like automatic refueling of aircrafts in flight, automatic docking of cars to form a road train, autonomous farm vehicles carrying out rendezvous maneuvers, or to simulate landing maneuvers. This keeps the system attractive not only for its original purpose but for a wide range of possible future projects that fit well in the scope of DFKI RIC research.

Acknowledgments

The project was funded by the Space Agency (DLR Agentur), acting on a mandate from the Federal Government, grants no. 50RA0908 and 50RA0910. It involved Airbus Defence & Space as project coordinator and Jena-Optronik as well as DFKI RIC[10] as project partners. The authors would like to thank the INVERITAS project partners Jena Optronik and the colleagues at the DFKI research group "Cyber-Physical Systems" for their support and cooperation.

References

[1] J. Paul, F. Kirchner, I. Ahrns, J. Sommer, Robotics Rendezvous & Capture Test Facility "Inveritas", in: Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS), 2014 International Symposium on, 2014.

[2] H. Benninghoff, F. Rems, T. Boge, Development and hardware-in-the-loop test of a guidance, navigation and control system for on-orbit servicing, Acta Astronautica 102 (2014) 67–80. doi:10.1016/j.actaastro.2014.05.023.

[3] Z. Milenkovich, Z. Wilson, D. Huish, J. Bendle, A. Kibler, The Space Operations Simulation Center (SOSC) and Closed-loop Hardware Testing for Orion Rendezvous System Design, AIAA Guidance, Navigation, and Control Conference (2012) 1–16doi:10.2514/6.2012-5034.
URL `http://arc.aiaa.org/doi/abs/10.2514/6.2012-5034`

[4] B. Girault, S. Bartsch, F. Kirchner, Multifunctional Robot Test Facility for on-orbit and extraterrestrial surface exploration, in: Proceedings of Ground-based Space facilities symposium. Ground-based Space facilities symposium (GBSF-2013), June 12-14, Paris, France, 2013.

[5] L. Richter, A. Brucks, L. Witte, A new facility for lander touchdown and rover mobility testing at dlr, in: 9th ILEWG International Conference on Exploration and Utilisation of the Moon (ICEUM9/ILC2007), 2007.

[6] H. Krueger, S. Theil, Tron-hardware-in-the-loop test facility for lunar descent and landing optical navigation, in: 18th IFAC Symposium on Automatic Control in Aerospace, 2010.

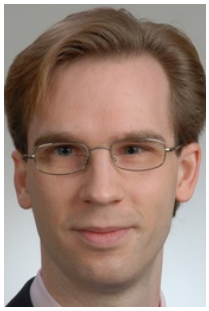
[7] A. Fleischner, M. Wilde, U. Walter, Racoon - a hardware-in-the-loop simulation environment for teleoperated proximity operations, in: I-SAIRAS 2012 TURIN, ITALY, Institute of Astronautics, TUM, 2012.

[8] J. Albus, R. Bostelman, N. Dagalakis, The NIST Robocrane, Journal of Robotic Systems 10 (1993) 709–724. doi:10.1002/rob.4620100509.

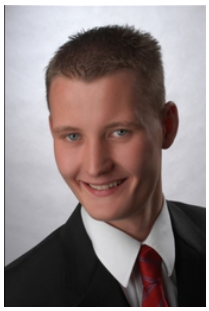
[9] A. Pott, C. Meyer, A. Verl, Large-scale assembly of solar power plants with parallel cable robots, in: Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK), 2010, pp. 1–6.
URL `http://ieeexplore.ieee.org/stamp/stamp.jsp?`

[10] German Research Center for Artificial Intelligence, DFKI Bremen, Robotics Innovation Center, Robert-Hooke-Straße 1, D-28359 Bremen

arnumber=5756909
[10] Automated Precision Inc., API product Specifications - Tracker3 Ultra Portable Laser Tracking System (2011).
[11] R. Smith, et al., Open dynamics engine (2005).
[12] International Organization for Standardization, Manipulating industrial robots - performance criteria and related test methods (iso 9283 : 1998) (1998).
[13] S. Macfarlane, E. A. Croft, Jerk-bounded manipulator trajectory planning: design for real-time applications, IEEE Transactions on Robotics and Automation 19 (2003) 42–52. doi:10.1109/TRA.2002.807548.
URL http://dx.doi.org/10.1109/TRA.2002.807548
[14] G. H. Golub, C. F. Van Loan, Matrix Computations, Vol. 10, 1996. doi:10.1063/1.3060478.
URL http://www.ncbi.nlm.nih.gov/pubmed/18273219
[15] H. Park, S. D. Cairano, I. Kolmanovsky, Model predictive control for spacecraft rendezvous and docking with a rotating/tumbling platform and for debris avoidance, in: Proceedings of the 2011 American Control Conference, 2011, pp. 1922–1927.
[16] F. Pomerleau, F. Colas, R. Siegwart, S. Magnenat, Comparing ICP variants on real-world data sets: Open-source library and experimental protocol, Autonomous Robots 34 (3) (2013) 133–148. doi:10.1007/s10514-013-9327-2.
URL http://dx.doi.org/10.1007/s10514-013-9327-2
[17] R. Brunelli, Template matching techniques in computer vision: theory and practice, 2009. doi:10.1002/9780470744055.
URL http://books.google.com/books?id=AowB9dRNTqYC



Jan Paul received his PhD in Computer Science from the Christian-Albrechts-Universität zu Kiel in 2008. Since he joined the DFKI-RIC in 2009, Dr. Paul was the leader of the simulation project "Virtual Crater" and of the hardware in the loop simulation projects "INVERITAS" and "RTES-TA". Dr. Paul's research interests are in the area of the simulation of physical and behavioral properties of robotic systems and their environments.



Alexander Dettmann received his M.Sc. degree in systems engineering from the University of Bremen, Germany in 2009. In 2009, he joined the DFKI working on the INVERITAS test facility. Since 2012, he became a research associate in the robotics group of the University of Bremen. His main interest is the control of kinematically complex robots in unstructured and unknown environments.

Benjamin Girault got his diploma in aerospace engineering with specializations in automatic control systems and computer science from the ISAE, Toulouse, France in 2009. He joined the DFKI in 2009 working on the development of the INVERITAS test facility. He focuses his work on modeling and control of complex kinematics such as robotic manipulators.

Jens Hilljegerdes received his diploma in mechanical engineering from the University of Applied Science in Wilhelmshaven in 1998. Since 2004 he works for the robotics workgroup of the University of Bremen in the field of design and construction. For this work the focus was to support the work on mechanical realisation, improvement and adaptation of the cable driven system.

Frank Kirchner is full professor at the Computer Science Faculty of the University of Bremen and CEO of the Robotics Innovation Center at DFKI. Prof. Kirchner is also Scientific Director of the Brazilian Institute of Robotics. He is a leading expert in research on biologically inspired locomotion and behaviour of highly redundant, multifunctional robotic systems. He is the principal supervisor for a number of PhD students and regularly serves as a reviewer for a series of international scientific journals and conferences. He also is author of more than 150 publications in the field of Robotics and AI, including space robotics.

Ingo Ahrns received his Ph.D. in computer science from the University of Ulm in 2000. Since 2000 he works for Airbus Defence and Space in the space robotics lab. He is an expert for computer vision and image processing and his areas of interest are vision-based navigation for rendezvous with uncooperative targets and planetary landing, as well as vison systems for robotic manipulation.

Josef Sommer is working since more than 30 years in the flight dynamics group of Airbus-DS in Bremen. After numerous GNC studies culminating in the ATV GNC definition he was responsible for the development of the flight control system of Phoenix, a RLV landing demonstrator, then led the IXV (reentry demonstrator). Actually he is responsible for the GNC system development of DEOS, a German demonstration mission for space debris removal and in orbit servicing.