

# European Proximity Operations Simulator 2.0 (EPOS) - A Robotic-Based Rendezvous and Docking Simulator

Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)  
Space Operations and Astronaut Training<sup>\*</sup>

Instrument Scientists:

- H. Benninghoff, Space Operations and Astronaut Training, DLR, Oberpfaffenhofen, Germany, email: [heike.benninghoff@dlr.de](mailto:heike.benninghoff@dlr.de)
- F. Rems, Space Operations and Astronaut Training, DLR, Oberpfaffenhofen, Germany, email: [florian.rems@dlr.de](mailto:florian.rems@dlr.de)
- E.-A. Risse, Space Operations and Astronaut Training, DLR, Oberpfaffenhofen, Germany, email: [eicke-alexander.risse@dlr.de](mailto:eicke-alexander.risse@dlr.de)
- C. Mietner, Space Operations and Astronaut Training, DLR, Oberpfaffenhofen, Germany, email: [christian.mietner@dlr.de](mailto:christian.mietner@dlr.de)

**Abstract:** The European Proximity Operations Simulator (EPOS) 2.0 located at the German Space Operations Center (GSOC) in Oberpfaffenhofen, Germany, is a robotic based test facility of the German Aerospace Center (DLR) used for simulation of rendezvous and docking (RvD) processes. Hardware such as rendezvous sensors (cameras, laser scanners) or docking tools, as well as software (e.g. for navigation and control) can be tested and verified. The facility consists of two robotic manipulators with each six degrees of freedom, a linear slide of 25m length on which one robot can be moved in the laboratory, and a computer-based monitoring and control system. EPOS 2.0 allows for real-time simulations of the rendezvous and docking process during the most critical phase (separation from 25m to 0m) of proximity and docking/berthing operations.

## 1 Introduction

Test and verification play major roles in the preparation of space missions. Especially new classes of missions like on-orbit servicing (OOS) missions or other proximity operations between satellites require intensive tests of all involved systems.

<sup>\*</sup>Cite article as: DLR Space Operations and Astronaut Training. (2017). EPOS 2.0 RvD Simulator. *Journal of large-scale research facilities*, 3, A107. <http://dx.doi.org/jlsrf-3-155-6>

In an on-orbit servicing mission (Ellery et al., 2008; Nishida et al., 2009; Yasaka & Ashford, 1996), a service satellite approaches a client satellite in its orbit for life time extension or safe end-of-life de-orbiting. The client satellites can be partly damaged, can be out of fuel or can have a non-operative attitude and orbit control system to perform an end-of-life de-orbit on its own (Benninghoff, Boge, & Rems, 2014; Boge et al., 2009, 2010).

Concerning an OOS mission, there are new challenges for the rendezvous and docking process since the involved client satellites are non-cooperative in contrast to previous rendezvous and docking (RvD) missions in manned spaceflight. A typical target satellite in an OOS scenario has not been developed taking into account rendezvous and docking maneuvers. Therefore, there are no retro-reflectors, for example, which can be used for navigation, and no particular docking port. In addition, there is no communication between the service spacecraft and the client spacecraft. Furthermore, for on-orbit servicing in Low Earth Orbit (LEO), the RvD process should be highly autonomous due to long phases where there is no contact with the ground station. Concerning on-orbit servicing in the geostationary orbit (GEO), the operator has to cope with time delays during the robotic activities. A possible business-case in GEO is lifetime extension by station keeping of expensive and complex communication satellites which have run out of fuel.

Motivated by this new class of missions, the European Proximity Operations Simulator (EPOS) 2.0 has been developed. It is a robotic based test facility for real-time simulation of the final phase of an on-orbit servicing mission: the mid and close range rendezvous phase and the docking phase. The facility consists of two industrial, highly accurate robots with each six degrees of freedom which simulate the 6D dynamic motion of two satellites during rendezvous and docking. One robot is mounted on a linear slide (rail system) of 25 m length. Thus the final approach can be completely simulated. EPOS further contains a real-time facility monitoring and control system and a real-time application control system. It is possible to perform hybrid simulations with one part of the simulation represented by the robots with integrated hardware (such as sensors) and with the other part modeled and simulated in software (software based satellite simulator for instance).

This paper gives a technical description of the facility and examples of utilizations and previous test campaigns and experiments.

## 2 EPOS 2.0 - A Facility for RvD Simulation, Test and Verification

### 2.1 History and Motivation

The German Aerospace Center (DLR) has experience in the field of rendezvous and docking simulation since the 1980s (Boge & Schreutelkamp, 2002; Boge et al., 2010). The predecessor of the current RvD facility, EPOS 1.0, was a joint test facility of DLR and ESA for simulation of approach maneuvers during the most critical phase: the final meters of the rendezvous phase prior to docking. One of the last biggest test campaigns was the test and verification of the RvD sensors of the European ATV (Automated Transfer Vehicle). Also sensor tests of the Japanese HTV (H-2 Transfer Vehicle) have been conducted at the EPOS 1.0 facility. Figure 1 shows two views of the EPOS 1.0 facility; it consisted of a fixed part with the RvD sensors and of a mobile part where a target mockup was mounted.

After approximately two decades, EPOS 1.0 was replaced by a new rendezvous and docking test facility which provides the capabilities for complete RvD processes with special focus on on-orbit servicing and spacecraft de-orbit missions. The new EPOS 2.0 facility, which was developed and established from 2008 to 2009, was a joint work of two institutes of DLR: The first institute is GSOC, the German Space Operations Center, where the facility is located. The GSOC engineers and scientists were responsible for the overall design, construction and operation of the facility. GSOC further provided expertise in the field of operations, flight dynamics and navigation. The second institute is the Robotics and Mechatronics Institute of DLR which provided expertise in space robotics. Figure 2 shows the physical part of the facility consisting of two industrial robots and one linear slide to move one robot in the laboratory such that distances up to 25m can be realized. The image shows the robots of the facility in



Figure 1: The former EPOS 1.0 facility: a fixed part (left) and a movable part (right)

different simulation scenarios.



Figure 2: EPOS 2.0 facility - Physical part of the facility showing robots from different views and with different mockups and background. Left: two robots representing satellites in close proximity, center: large mockup with two antennas with a projection of the Earth in the background, right: large 3D mockup on a robot with black molton wrapping and with black background

As illustrated in Section 1, because of the high technological challenges of OOS missions, the rendezvous and docking system has to be tested and verified intensively on ground before an OOS mission can be launched. The tests comprise the rendezvous and docking hardware and software. The EPOS facility meets the following requirements for such tests (Boge et al., 2010):

- increased positioning accuracy (factor 10 compared to EPOS 1.0),
- capability to perform the 6D relative dynamic motion between two spacecrafts during the close range rendezvous phase ranging from 25m to 0m,
- dynamical capabilities such as high commanding rate necessary to simulate the 6D contact dynamic behavior during the docking process,
- nearly space-representative lightning conditions,
- capability to mount and move large client mockups and RvD sensors and equipment,
- capability to integrate on-board computers,
- capability to connect the facility with a control room (TM/TC exchange with RvD consoles) and
- capability to command the entire facility in real-time.

Details are given in the technical description and specification below.

## 2.2 Technical Description and Specification

### 2.2.1 Hardware Overview

The EPOS facility consists of the following hardware elements:

- a rail system KUKA KL1500 on the floor to move an industrial robot up to a distance of 25m,
- a KUKA KR100HA (HA = High Accuracy) robot (*robot 1*) mounted on the rail system for simulation of the 6 degree of freedom motion of one spacecraft,
- a KUKA KR240-2 robot (*robot 2*) mounted at one end of the rail system for simulation of the 6 degree of freedom motion of a second spacecraft,
- an ARRI Max 12/18 used as Sun simulator,
- a PC based monitoring and control system.

The technical data of the rail and the robots is summarized in Table 1 and Table 2.

Parameter	KUKA KL1500 (rail)
Type	Linear axis with rack-and-pinion drive
Payload mass maximum	3000 kg
Rail mass (without robot)	ca. 13500 kg
Repeatability	$\pm 0.02$ mm (ISO 9283)

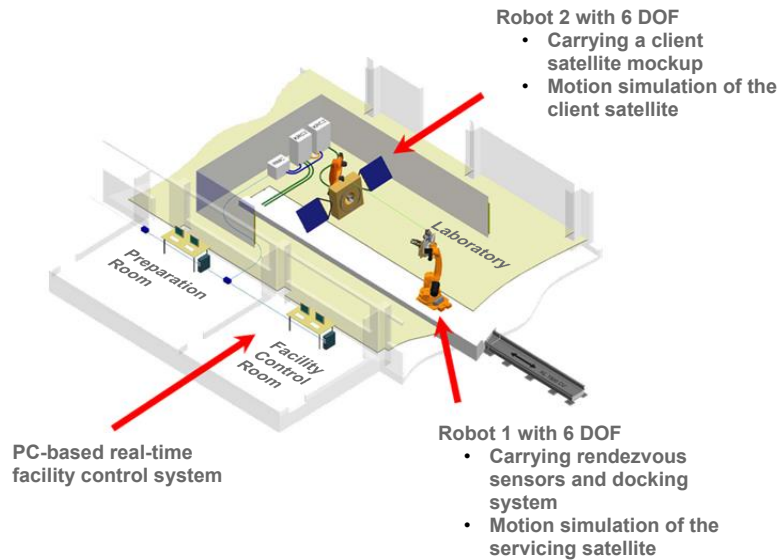
Table 1: Technical data of the rail.

Parameter	KUKA KR100HA (robot 1)	KUKA KR240-2 (robot 2)
Type	6-axis articulated robot	6-axis articulated robot
Payload mass maximum	100 kg	240 kg
Robot mass (without control part)	1200 kg	1267 kg
Reach (approx.)	2600 mm	2700 mm
Repeatability	$\pm 0.12$ mm (ISO 9283)	$\pm 0.12$ mm (ISO 9283)
Mounting	Rail	Floor

Table 2: Technical data of the robots.

Further, there is a large protective fence surrounding the rail system and the robots. During a simulation, the whole cabinet has to be locked for safety reasons. If the doors of the protective fence are not locked, only manual control via the KUKA control panels (KCPs) is possible.

A typical setup for RvD simulation is as follows: one robot carries a client satellite mockup and simulates the 6D motion of the client satellite. The other robot carries the rendezvous sensors (camera, LiDAR, etc.) and the docking system and simulates the 6D motion of the service satellite in an OOS mission. The simulation is controlled and monitored in real-time by a specific system of computers. Parts of a simulation which cannot be realized with real hardware can be simulated and calculated by software, for example orbit dynamics. Figure 3 gives an overview on an exemplary setup at the EPOS facility.



[b]

Figure 3: Possible setup of an RvD simulation with a client and a servicing satellite represented by the robots.

For visual navigation with optical sensors such as cameras or time-of-flight sensors, the client satellite mockup is often illuminated such that the test lightning conditions are close those in space. Further, one can simulate different lightning directions like front, side or back Sun illumination.

For Sun simulation, the powerful floodlight ARRI Max 12/18 is installed in the laboratory, see Figure 4 (left). It is equipped with a 12 kW HMI (hydrargyrum medium-arc iodide) lamp of Osram. This lamp is capable to generate a luminous flux of 1.15 million lumens. It can generate a spectrally realistic irradiation, i.e. the light of the Sun simulator is very close to real Sun light in the visible spectrum. This has been verified at 7 m distance to the lamp with a spectral irradiance meter.

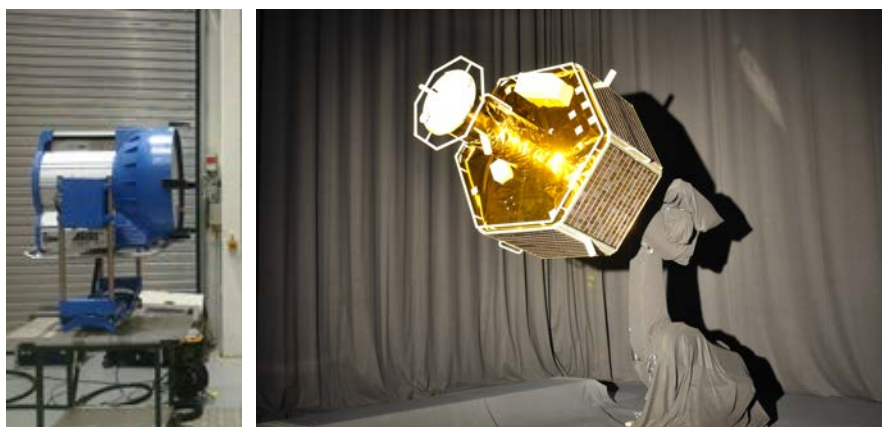


Figure 4: Sun Simulation. Left: spotlight ARRI Max 12/18 with a 12 kW HMI lamp. Right: robot with illuminated client mockup.

Using a beamer, an image of the Earth can be projected to a wall, using a black curtain and a black wrapping of one of the robots (material: molton) quite realistic images showing an illuminated client satellite with a black background can be captured. An example of such an image can be seen in Figure 5.



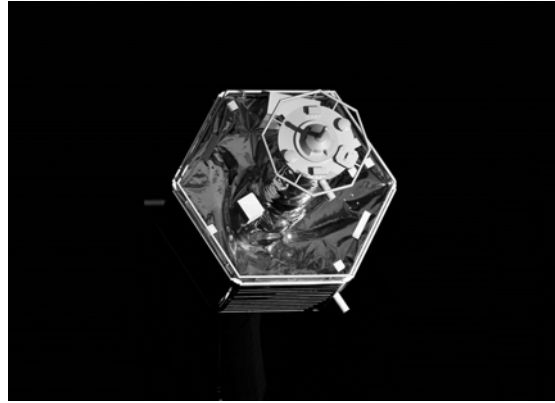


Figure 5: Gray-scaled image captured with a CCD camera. Due to the black curtain and black wrapping, very realistic images can be generated.

### 2.2.2 EPOS Control System

As mentioned before, the facility has a certain computer based control and monitoring system. An overview about the EPOS control system is given in Figure 6. Three different levels exist:

1. The **Local Robot Control (LRC)** controls the axes of the robots. Each robot is independently controlled in real-time by its own LRC unit, provided by the robot manufacturer. The KUKA Robot Control (KRC2) is a standard industrial robot control cabinet including the six robot servo amplifiers, a PC-based controller with a dual O/S programming environment and the required electronic safety systems for the standalone robot. Both KRC2 cabinets in the EPOS facility can be externally commanded with a command rate of 250 Hz and are synchronized to the EPOS external time base (ETB).
2. The **Facility Monitoring and Control system (FMC)** controls and monitors the entire facility in real-time. Moreover, the FMC system allows performing the following tasks:
  - monitoring of all parameters and states of the facility by an operator,
  - logging of all parameters and states of the facility, including external synchronization signals,
  - real-time control of the entire facility including synchronization of all motion devices and kinematic conversions of the external commands,
  - choice among different interfaces. The following options are available: a synchronous interface (EtherCAT), for closed-loop applications; an asynchronous interface, in order to run a predefined trajectory stored in a file, or a KUKA Robot Sensor Interface (RSI) to directly interface the FMC with the LRC units.
3. The **Application Control System (ACS)** runs the actual RvD-simulation application. In particular, models of the satellites dynamics and case-specific scenarios can be implemented in a MATLAB/Simulink environment. The MATLAB Real-Time Workshop can be used to accomplish the automatic code generation. Subsequently, the real-time executable is downloaded to a target platform running under a VxWorks operating system. Via EtherCAT this real-time PC can communicate with the FMC system. The desired motion commands must be sent every 4 ms to the facility, as requested by the LRC units.

Both, the FMC system and the ACS system, consist of a real-time computer with VxWorks as operating system, where the real-time executable runs, and a non real-time Windows computer for interaction with the user. The FMC computers are called FMC-MMI (Windows computer, MMI = Man Machine Interface) and FMC-RT (VxWorks computer, RT = real-time). The ACS computers are called ACS-MMI and ACS-RT, respectively.

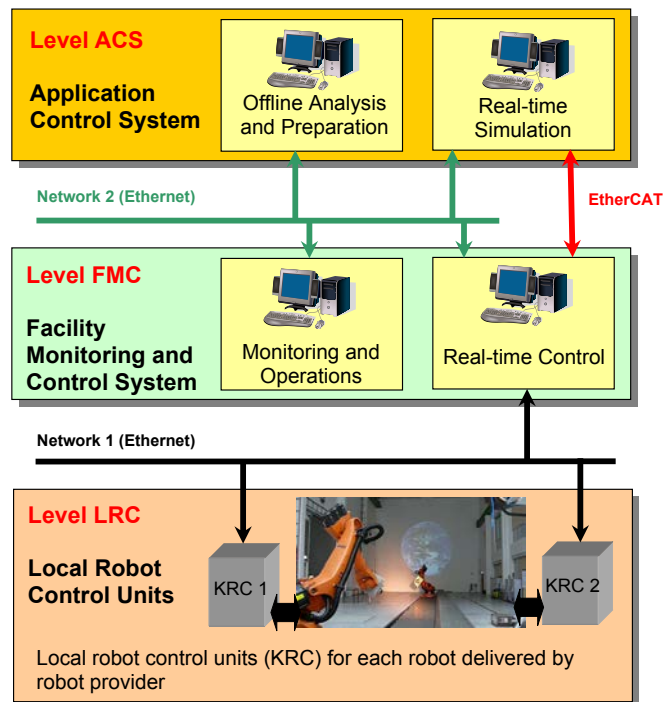


Figure 6: EPOS control system consisting of three different levels: LRC, FMC and ACS.

### 2.2.3 EPOS Coordinate Systems

The EPOS robots can be commanded using different reference coordinate systems which are shortly described in this section.

1. The **Ideal Robot Joint Coordinates / Ideal Joint Tool (IJT)**: Each robot has six independent servo controlled axes which allow to move the tool adapter relative to the robots base. The easiest way to command any motion is to directly define an angle for each of the robots axes. For completion, the position of robot 1 on the linear slide has to be commanded. Direct commanding of the axes is useful to move the robot manually using the KUKA KCP (KUKA Control Panel), i.e. to check for axis limits, but also other uses are possible.
2. The **Ideal Robot Device Coordinate System (IDC)**: Each robot can be commanded by prescribing the position and attitude of its adapter plate with respect to its base. In detail, the position and orientation of a so-called Tool Coordinate System with respect to its Base Coordinate System is commanded. Further the position of the base of robot 1 on the linear slide has to be commanded for a complete description. The Tool Coordinate System is a Cartesian coordinate system and has its origin in the middle of the robot's tool flange, the z-axis is oriented perpendicular outwards of the breadboard's mounting face and the x-axis is oriented towards the electrical interface block on the backside of the breadboard. Figure 7 (left) shows the Tool Coordinate System. The Robot Base Coordinate System is a Cartesian coordinate system and has its origin in the middle of the robots mounting face (the base), the z-axis is oriented towards the laboratory ceiling and the x-axis is oriented to the opposite of the cable plugs at the back of the robot. Figure 7 (center) shows the Base Coordinate System. IDC commanding is useful for various situations, in which the operator needs to command a single tool in Cartesian Coordinates.
3. The **Global / Lab Coordinate System (GLC)**: The Laboratory Coordinate System is a Cartesian coordinate system defined as follows: The z-axis is defined by the intersection of the xz-plane

of the KR100 Base Coordinate System on the slide and the yz-plane of the fixed KR240 Base Coordinate System, the z-axis is oriented towards the laboratory ceiling. The origin is 1500mm above the xy-plane of the KR240 Base Coordinate System, positive x-direction is opposite to the x-direction of the KR100 Base Coordinate System. The GLC System is useful whenever both robot tools need to be commanded together. Figure 7 (right) shows the Global / Lab Coordinate System.

4. The **Clohesy Wiltshire Coordinate System (CLW)**: The facility can be commanded in Clohesy Wiltshire (CLW) Coordinates. For the commanding of a Rendezvous and Docking simulation it might be useful i.e. to have the origin of a CLW Coordinate System in the center of gravity of the target spacecraft. To define a spacecraft related CLW System a so-called User Tool can be defined which is basically a fixed transformation relative to a robot's tool coordinate system. The orientation of the axes of the CLW System thus depends on the user's definition and may change from application to application. At EPOS three Cartesian coordinate frames can be commanded:
- Frame 1 - robot 1: Representing a defined reference frame fixed to spacecraft 1 (chaser or target depending on simulation set up at EPOS). The movement of spacecraft 1 results in a movement of robot 1 (KR100 HA).
  - Frame 2 - robot 2: Representing a defined reference frame fixed to spacecraft 2 (chaser or target depending on simulation set up at EPOS). The movement of spacecraft 2 results in a movement of robot 2 (KR240-2).
  - Frame 3 - point of view (POV): Representing a reference frame for the formation of the two spacecrafts. It describes the transformation from the formation scenario, used in the application, to the Global Laboratory Coordinate System.

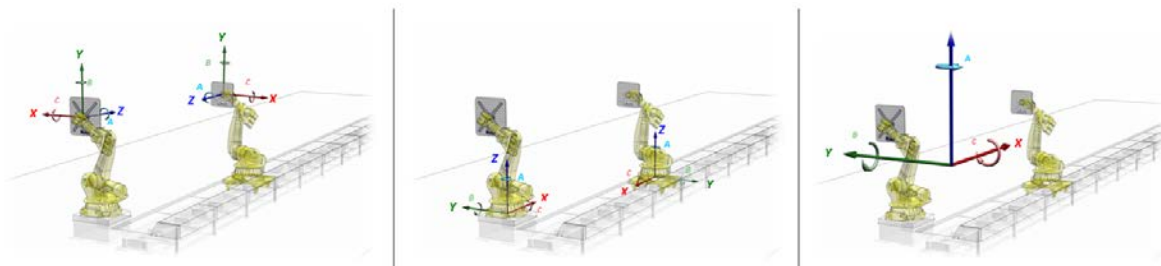


Figure 7: Coordinate Systems: Tool (left) and Base Coordinate System (center), and Global Lab Coordinate System (right)

## 2.2.4 Capabilities and Performances

The ranges and motion limits of the two EPOS robots are summarized in the following table.

The axes 4 and 6 of the KUKA KR240-2 Robot have been configured as endless axes. Simulation of tumbling satellites often results in several hundred complete rotations around the roll axis of the body. These rotations can be realized with the two endless axes 4 and 6.

The linear rail KUKA KL1500 has a motion range from 0 to 25 m with a maximum translational velocity of 1.45m/s.

Simulation with true-size mockups is possible up to a distance of approximately 25 m. For some applications (e.g. test of camera based rendezvous) also larger distances with downscaled mockups can be realized.

The positioning accuracy of the facility is required to be in the millimeter range. Measurements have been carried out with a Leica laser tracker by the company Robo Technology GmbH (<http://www.robo-technology.de/robo/de/>), allowing 3D precision measurements with micron-accuracy (absolute distance meter) and absolute angle detection with 0.5arcsec accuracy (ISO17123-3) with 0.07arcsec resolution. These measurements were used to determine the parameters of a static model of the facility which is



Parameter	Robot KUKA KR240-2 (from/to)	Robot KUKA KR100HA (from/to)
<b>Position</b>		
x [m]	-2.5 / +2.5	-2.5 / +24.5
y [m]	-1.0 / +4.0	-2.5 / +2.5
z [m]	-0.5 / +1.5	-0.5 / +1.2
Roll [deg]	endless	-300 / +300
Pitch [deg]	-90 / +90	-90 / +90
Yaw [deg]	-90 / +90	-90 / +90
<b>Velocity</b>		
Translational [m/s]	2	2
Rotational [deg/s]	180	180

Table 3: Motion ranges of the robots of the EPOS facility with respect to the Global / Lab Coordinate System.

running in real-time during commanding the facility. Based on this calibration the overall positioning accuracy is 1.56 mm (position,  $3\sigma$ ) and 0.2 deg (orientation,  $3\sigma$ ).

### 2.2.5 Command Interfaces

There are several options to command the facility: a so-called *Asynchronous Command Interface* reads a trajectory from a file stored on the FMC system (cf. Section 2.2.2). The *Synchronous Command Interface*, on the contrary, requires a command signal every 4 ms from the ACS system using EtherCAT. Further, a *Manual Control Unit* can be used to move the robots manually.

1. The **Asynchronous Command Interface**: For open loop tests, the asynchronous command interface is typically used since there is no need for an on-line command interface. The user can define a command trajectory, off-line in advance. The trajectory has to be provided as ASCII file and can be loaded by the FMC system and executed by the EPOS facility. The text file has to be structured as follows:
  - each line represents one motion command,
  - the time step between two consecutive lines is 4 ms,
  - each line consists of one int16 value followed by 22 double values separated by blanks, the detailed description is given in Table 4.
  - especially for quaternions a sufficient number of decimals are needed (15 should be sufficient)

The parameter definition (e.g. CMD Rob 1, CMD Rob 2 used in Table 4) depends on the reference coordinate system and is given in Appendix A.

2. The **Synchronous Command Interface**: This command interface uses the EtherCAT protocol to command the facility in real-time. It receives positioning commands and repeats with the real current robot positions as reported from the KUKA controller. The synchronous command interface sends and receives data using the same coordinate systems. The following conventions are used:
  - one command has to be sent every 4 ms
  - each command contains 22 double values and one int16 value. The int16 parameter is called mode and is a certain bit mask. The 22 double values are used to command the parameters CMD Rob 1, CMD Rob 2, CMD POV and data lin as used for the asynchronous command interface, recall Table 4.

The bit mask for the parameter mode is defined as follows:

- bits 0-3 define the reference coordinate system: 0001 = CLW, 0010 = GLC, 0011 = IJT, 0100 = IDC,
- bits 4-13 are not used

- bit 14: a flag used for checking the synchronous command status, 1: synchronous command mode is alive, the facility is commanded synchronously every 4 ms, 0: otherwise,
- bit 15: a flag used for checking the status of the starting phase, 1: FMC command "move to starting point" is running, the facility is moving to start point, 0: otherwise.

The parameter definition (e.g. CMD Rob 1, CMD Rob 2) depends on the reference coordinate system and is given in Appendix A. It is possible to generate a trajectory on-line in a MATLAB Simulink environment on the ACS level. Dynamical models for the orbit and attitude of the two spacecraft can be created in Simulink. When the Simulink model is executed with a rate of 250 Hz, the final motion for each spacecraft is sent every 4 ms using a special command interface block based on an S-function. Figure 8 shows the Simulink interface block.

3. The **Manual Control Unit**: This unit can be used to manually command the EPOS facility using a manual device (Keba). On the Keba manual device, there is an application called "EPOS MCU" which is used for both monitoring of variables of the data pool and for moving the robots manually. Figure 9 shows two different views of the MCU window. Variable lists can be loaded and displayed. Furthermore, a graphical view of the robots and the coordinate systems can be selected. Using the buttons on the manual device, the coordinate system and the robot to be commanded can be selected. The position and orientation or the individual axes values (in case of IJT) can be changed by using the  $+/-$  buttons on the right.

Nr	Parameter	Data Type	Description	Remark
1	ref coord	1 int16	EPOS coordinate system	1 = Clohessy Wiltshire Coordinate System (CLW) 2 = Global / Lab Coordinate System (GLC) 3 = Ideal Joint Tool System (IJT) 4 = Ideal Device Coordinate System (IDC)
2-8	CMD Rob 1	7 double	Position and orientation command for robot 1	Details see Appendix A
9-15	CMD Rob 2	7 double	Position and orientation command for robot 2	Details see Appendix A
16-22	CMD POV	7 double	Position and orientation command of a simulation reference frame (point of view)	Only used for CLW mode
23	data lin	1 double	CLW and GLC: offset in x-direction between robot base and tool frame IJT and IDC: position of robot 1 on the linear slide/rail	

Table 4: Description of the parameters of a command line of an asynchronous command ASCII file

### 2.2.6 Facility Monitoring, Operation and Control

On the FMC (Facility Monitoring and Control) level of the facility the entire simulation is controlled and monitored, recall Section 2.2.2. The FMC-MMI is a Windows computer and the interface between operator and facility. The main software application is the so-called *FMC Command Center*. A screenshot is shown in Figure 10.

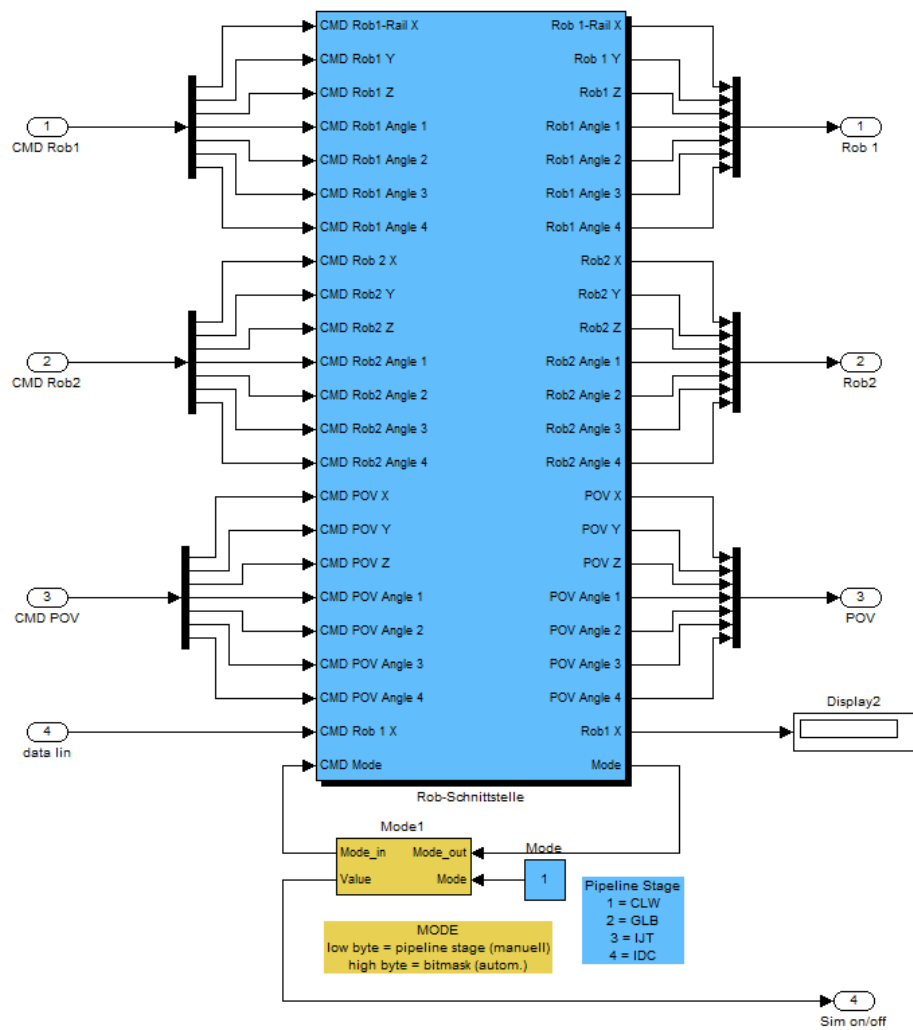


Figure 8: Synchronous Command Interface - Simulink Block



Figure 9: Manual Control Unit (MCU) - Numerical Display Window (left) and Graphical Visualization (right)

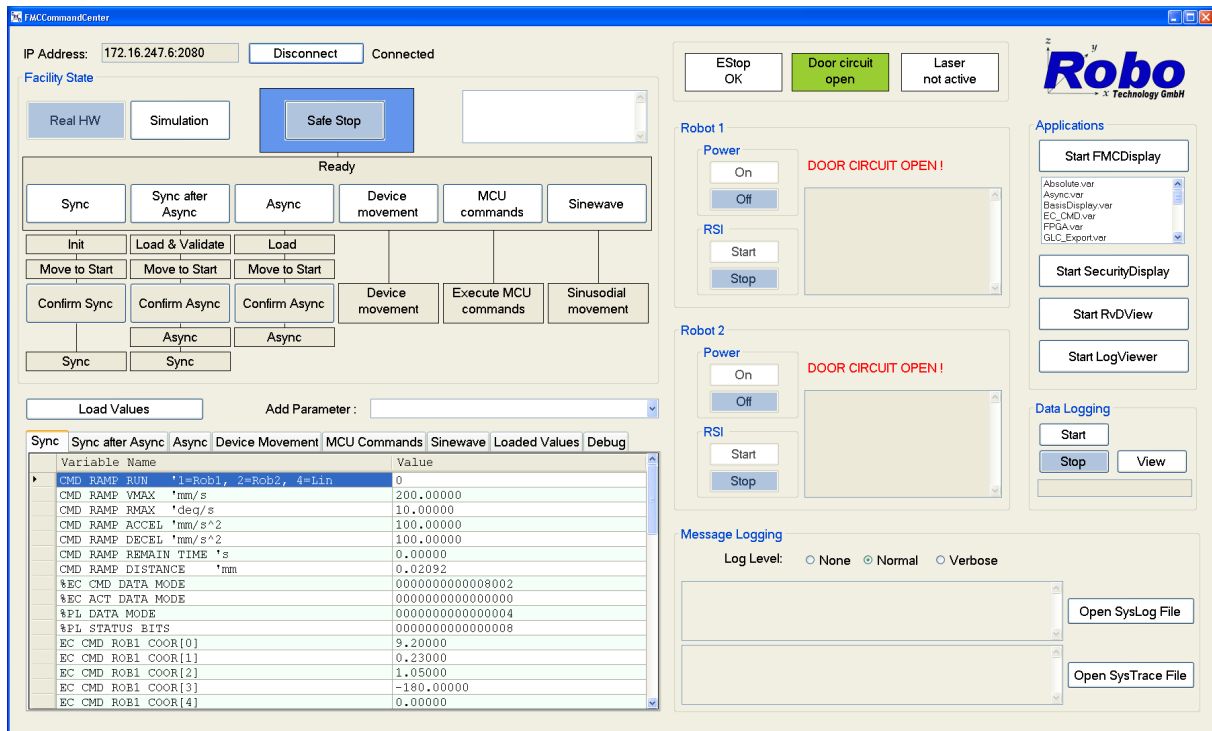


Figure 10: FMC Command Center - User interface software for monitoring and control of the simulation.

A simulation can be run either with real hardware or in simulation mode. When a new simulation is developed, it is first tested in simulation mode, i.e. without real robotic hardware. Via two buttons called *Real HW* and *Simulation*, one can switch between the two modes. By using additional buttons called *Sync*, *Asynch* and *MCU commands*, the mode of commanding can be selected, recall Section 2.2.5. In Figure 10, other buttons showing additional command modes can be found. However, command modes like *Sinewave* are dedicated to testing purposes. Therefore, this paper restricts to the three modes *Sync*, *Asynch* and *MCU commands*.

MCU commands can be executed directly without any starting phase. One starts always with the current pose of the robots. However, when commanding via the synchronous or asynchronous interface, the first command is usually not equal to the current pose of the robots. Therefore, there is a *Move to Start* phase, where the robots move to the first commanded position and attitude. To monitor this motion safely, the move to start phase is executed for robot 1, robot 2 and the linear rail successively. The order can be selected by the operator and is not prescribed. During the move to start phase the button *Move to Start* is marked with blue color. After this phase has been completed, the button *Confirm Sync* is active and the operator can manually start the synchronous or asynchronous simulation. During a simulation, the motion can be always stopped via the *Safe Stop* button.

The status of the simulation and of the facility can be observed with different tools: One is the so-called *RvD View* which is an optical visualization of the robots, see Figure 11. If CAD data of the mockups is available, it can also be loaded and the mockups can be visualized. The *RvD View* is an important tool for visual safety and collision avoidance checks. If a rendezvous or docking simulation is executed in simulation mode, the *RvD View* is used to observe if the calculated position and attitude of the robots is correct/as desired. Errors with e.g. coordinate transformations etc. can often be easily seen with that tool.

The second tool is the *FMC Display*, see Figure 12. With this tool, several numerical values of the facility and the robots can be displayed. The variables of interest are defined in so-called variable lists. For example, the variable list can contain the commanded and actual values for robot 1 and robot 2

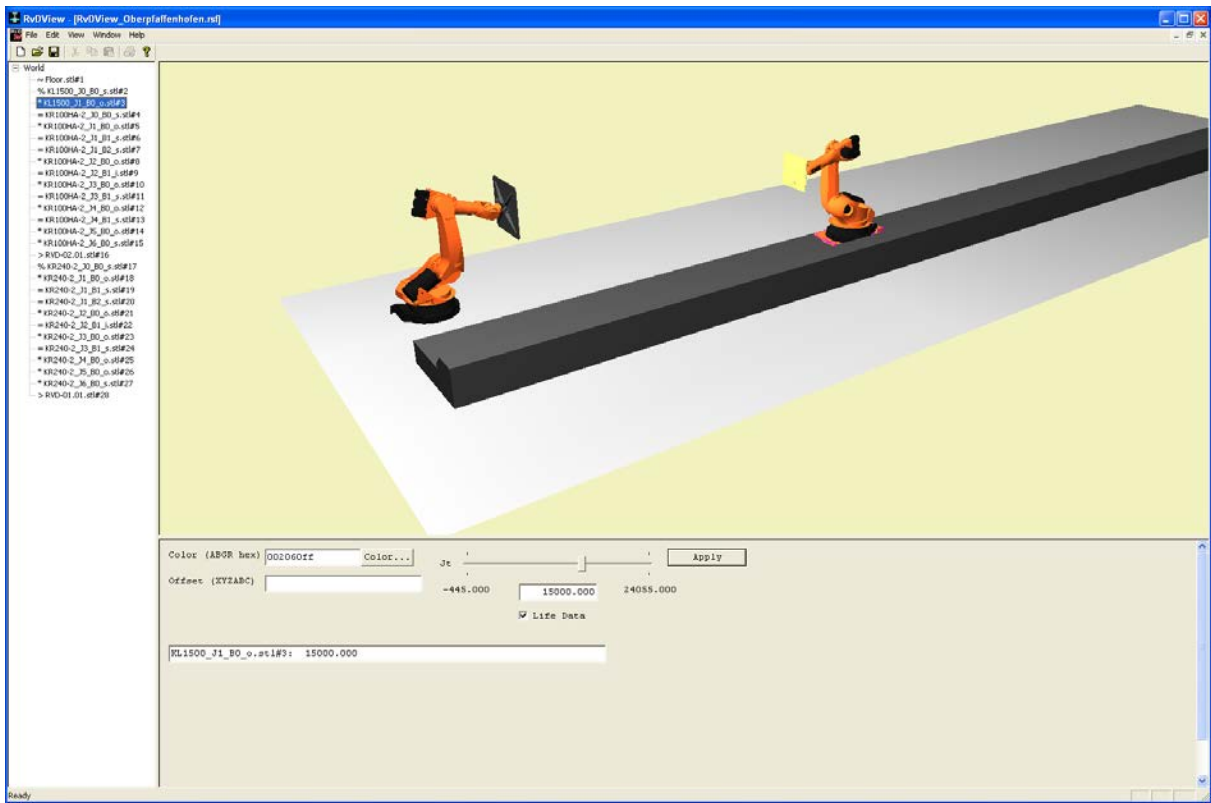


Figure 11: RvD View - Visualization of the motion of the robots.

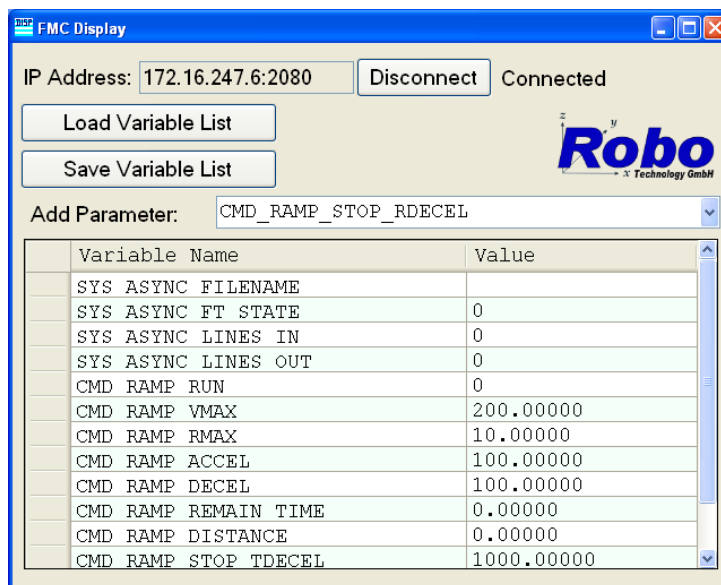


Figure 12: FMC Display - Numerical values of facility and robot states.



in several coordinate systems (CLW, GLC, IJT, IDC, recall Section 2.2.3). But also, other variables like temperature values, etc. exist.

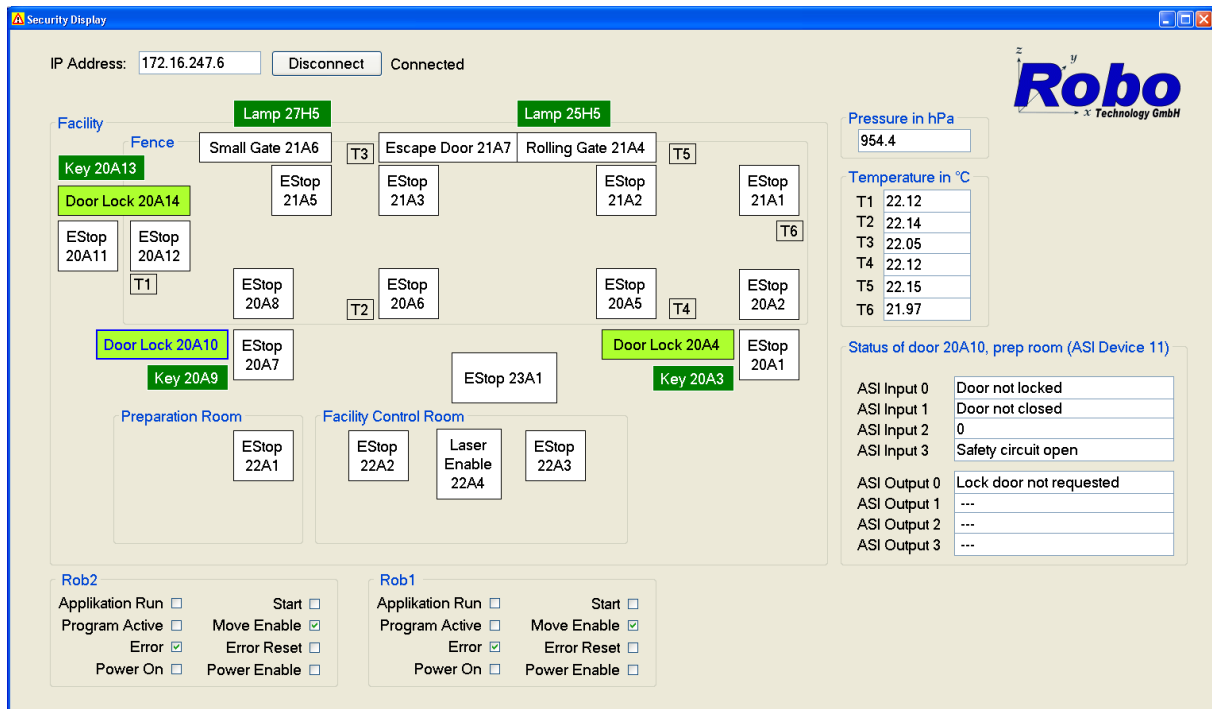


Figure 13: Security Display - Status of emergency stops, robots, doors, gates and lasers.

Apart of the simulation, the FMC operator has to continuously check the safety state of the facility during the simulation. For this purpose, a tool called *Security Display* exists, see Figure 13. The location and the status of all emergency stop buttons (abbreviated with *EStop*) are visualized. Those emergency stops are located at several places in the facility: for example at each door, around the the protective fence, at the local control units (LRCs) of the robots, in the preparation room and in the control room close to the FMC workstation.

Further, the status of all doors and rolling gates in the laboratory are shown in the Security Display. Before a simulation can be run, all doors and gates have to be closed. If, for example, one door is not properly closed, the display shows which door is still open.

For test campaigns and sensor tests at EPOS, it is possible to use sensors which actively emit light with harmful optical radiation since all doors and windows can be protected and a laser safety circuit can be activated. The status of such lasers is visualized also in the Security Display (button *Laser Enable*). Finally, errors occurring the robots and the status Power On/Off are also included in the security display. For monitoring apart of the FMC Command Center, four observation cameras exist. They can be used in addition to the RvD View (which is a pure software based visualization). With the observation cameras, those parts of the facility, which cannot be seen directly from the FMC operator sitting in the control room, can be observed. The camera based observation system is of great importance, if lasers are enabled or if a strong spotlight is switched on. To safe human eye and skin, blinds at all windows have to be shut down, including the windows between inner laboratory and control room.

## 2.2.7 Application Control

If the facility is commanded via the asynchronous command interface, one need to load a text file with a predefined trajectory, see Section 2.2.5. However if the synchronous command interface is used, there is a second workstation involved in addition to the FMC workstation: the Application Control System (ACS workstation), recall Section 2.2.2.

The ACS system consists of a Windows computer and a VxWorks real-time computer. On the Windows computer, called ACS-MMI, the application can be developed using MATLAB/Simulink. For the simulation the VxWorks computer, the ACS-RT, has to be configured as the target PC, the mode of the simulation has to be set to *External*.

The MATLAB Real Time Workshop is used to generate the real-time executable which is loaded to the ACS-RT afterwards. The Simulink model on the ACS-level must contain the EPOS command interface block, recall Figure 8.

A possible model is a numerical satellite simulator which contains dynamical models for the orbit and attitude of the two satellites. Further, environmental forces and torques acting on the satellite, as well as other disturbances can be simulated. All parts of the satellites, necessary for the application, which cannot be simulated with real hardware can be simulated by software. For example, actuators like thrusters or reaction wheels have to be simulated with software and can be included in the Simulink model.

The on-board computer can be part of the Simulink model, or can also be a separate computer. In the latter case, an interface between the computer simulating the on-board computer and the ACS computer has to be established. Ethernet can be used as interface. On the simulated on-board computer, interfaces to the rendezvous and docking sensors mounted on the robots are needed such that sensor data can be processed. Ethernet can be used as interface in this case, also.

### 2.2.8 Data Logging

All variables of the facility data pool can be logged every 4 ms to a binary log-file at the FMC level. A so-called EPOSLogViewer is a sub-program to view the logging data and to export logging data to other file formats, e.g. csv. Further, the log viewer can be used off-line for analysis; for example graphical illustrations and plots can be generated.

On the ACS level, data generated inside of Simulink models can be logged using for example the *To File* block of Simulink with user-defined sample rates.

### 2.2.9 Interfaces for Test Equipment

**Mechanical interfaces** Both EPOS robots are equipped with an adapter plate where the user can mount test equipment to the simulator. The specifications are given in Table 5. Before mounting any hardware to the tooling adapter plates, the overall mass and inertia properties have to be calculated and checked according to the KUKA manual, cf. KUKA Documentation (2009a,b). The parameters have to be entered in the KUKA controller via the KCP.

	<b>KR 100 HA Tooling Adapter</b>	<b>KR 240-2 Tooling Adapter</b>
<b>Size</b>	700 mm x 700 mm	1000 mm x 1000 mm
<b>Maximum Payload</b>	60 kg	200 kg
<b>Pattern</b>	Threaded hole pattern M6, 70mm pitch	Threaded hole pattern M6, 70mm pitch
<b>Mass</b>	47 kg	54 kg
<b>Moments of Inertia <math>I_{xx}</math></b>	19.7 kg m <sup>2</sup>	19.5 kg m <sup>2</sup>
<b>Moments of Inertia <math>I_{yy}</math></b>	1.0 kg m <sup>2</sup>	2.5 kg m <sup>2</sup>
<b>Moments of Inertia <math>I_{zz}</math></b>	20.3 kg m <sup>2</sup>	21.3 kg m <sup>2</sup>

Table 5: The KUKA KR100 HA and KR240-2 Tooling Adapter Plate

**Electrical and data interfaces** Each tooling adapter is equipped with an electrical interface block which allows the user to connect electrical test equipment (such as sensors like cameras, etc.) to a power

supply and to transmit the data to the control room. The interface block is located at the backside of the adapter plate.

The available interfaces are:

- DC Power, 5 V, 4 A max, 3 terminals,
- DC Power, 12 V, 2 A max, 2 terminals,
- DC Power, 24 V, 2.5 A max, 3 terminals,
- DC Power, 28 V, 2.5 A max, 3 terminals,
- AC Power, 230 V, 50 Hz, 10 A max, CEE 7/4 socket,
- 2x data link terminal: female FCC RJ-48 (i.e. for Ethernet),
- HF data link terminal: female BNC / RG59.

### 2.2.10 Synchronization of EPOS with External Components Using GPS

At EPOS, a GPS antenna signal is available which can be used to synchronize experimental data with the ground-truth of the EPOS robots (provided by the EPOS log-file): The FMC system logs the GPS time signal in the EPOS log-file together with other variables such as the robots' current states (in a user-defined coordinate system, recall Section 2.2.3).

If, for example, a sensor system is tested which is equipped with a GPS receiver, the sensor data and the EPOS log-file can be synchronized using the GPS time logged by both systems, the sensor system and the FMC system.

## 3 Using EPOS - Simulation and Utilization Concepts and Examples

### 3.1 Hybrid Simulation

The EPOS facility is a so-called *Hybrid Simulator* for simulation of relative motion processes. One part of the simulation is performed by numerical computations while the other part is executed by the robots.

Figure 14 visualizes the hybrid simulator concept. A typical setup is as follows: The calculation of the six degrees of freedom dynamical motion of the two satellites is part of the numerical simulator. On this level, equations of motion of the orbit and the attitude for both spacecrafts are solved. Further, equipment of the satellite which cannot be represented by real hardware in the laboratory such as actuators can be modeled on this level. The robots move such that their relative motion is equal to the relative motion of the spacecrafts computed by the numerical simulator. Part of the physical representation are sensors such as rendezvous sensors (camera, LiDAR, etc.) which are used for sensor verification experiments, or force/torque sensors which are used as measurement feedback to the numerical simulation (e.g. contact dynamics simulation).

### 3.2 Examples of RvD Simulations, Tests and Projects

#### 3.2.1 Camera Based Navigation

At EPOS, one major application is test and verification of camera based navigation systems. This comprises both camera hardware and software for image processing and pose estimation.

For a safe approach to the target during the approaching phase, one needs to continuously compute the relative position and attitude between servicer and client. Since the client can be completely passive and non-cooperative, optical sensors such as cameras are appropriate sensors for relative navigation.

If the target's geometry is known (e.g. provided by a CAD file) one can perform a model based pose estimation: One aims at finding the optimal relative pose such that the model best matches to the image. The underlying image processing often uses edge detection or image segmentation algorithms to find the target in the image. If the target's geometry is unknown, an inspection phase can be executed before

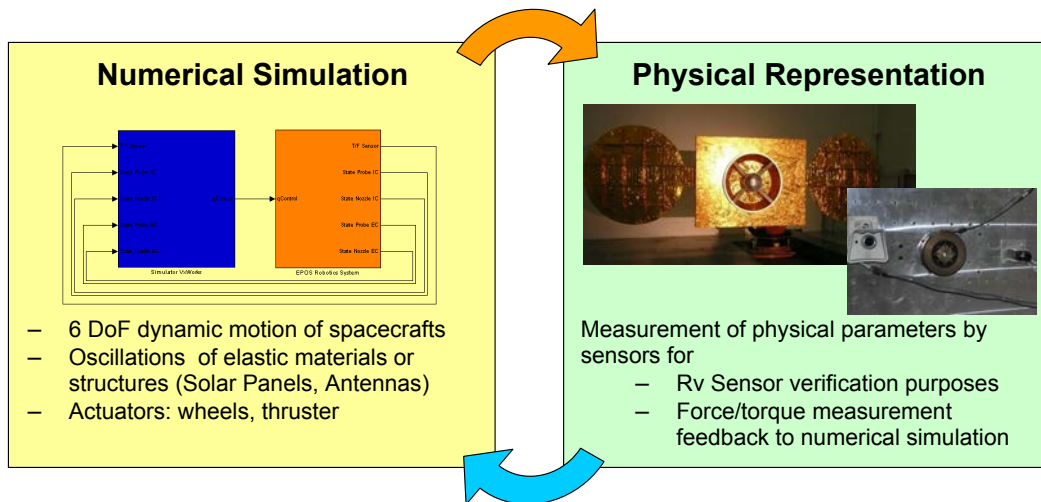


Figure 14: EPOS Hybrid Simulator - Concept

the final approach. During the inspection phase, all necessary information can be collected based on optical sensors, and a 3D modeling of the client is performed.

At EPOS, the user can mount its rendezvous sensor to one of the robots, and can perform sensor tests or advanced rendezvous simulations (cf. Section 2.2.9). Further, at EPOS, several sensors already exist and can be used for rendezvous simulation. Concerning cameras, two different kind of cameras are available and have been used for tests in different projects: cameras generating intensity images (visible spectrum) and cameras generating distance images. Intensity images can be captured with a CCD (charge coupled device) gray-scaled Prosilica GC-655M camera with a resolution of 640 x 480 pixels. Two such Prosilica cameras are available at EPOS. Distance images can be generated with PMD (photonic mixer device) cameras: One PMD camera at EPOS is the Camcube 3.0 of PMDtec (resolution 200 x 200 pixels) and the second PMD camera is the Argos3D-IRS1020 DLR Prototype of Bluetechnix (resolution 352 x 288 pixels). A PMD camera emits infrared light which is reflected by objects and received back by the camera. The camera computes distance images by calculating the phase shift between emitted light and received light. The Argon3D sensor contains also a CMOS sensor and captures therefore a distance image and an intensity image. Figure 15 shows the three different sensors.



Figure 15: Cameras used at EPOS: Prosilica GC-655M (left), PMDtec Camcube 3.0 (center) and Bluetechnix Argos3D-IRS1020 DLR Prototype (right).

Exemplary images with marked detected target are shown in Figure 16. A satellite mockup has been mounted on the tooling adapter of Robot 1, a Prosilica GC-655M has been installed at the tooling adapter of Robot 2. The camera has continuously captured images during an approach from 20 m to 5 m. With an edge detection algorithm based on a line-fit (Tzschichholz et al., 2011), the target has been tracked successfully in the sequence of gray-scaled images.

Another example is given in Figure 17 where a distance image is captured with a PMD camera. The



Figure 16: Examples of camera images captured with a Prosilica GC-655M showing a client satellite mockup in the laboratory at different distances. The detected edges are marked with orange color.

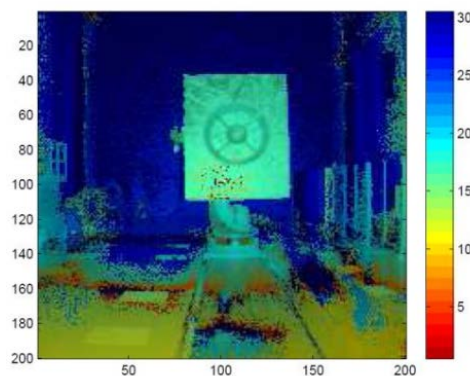


Figure 17: Exemplary distance image captured with the Camcube 3.0 of PMDtec showing a client satellite mockup in the laboratory. The distance information is visualized with different colors.

PMD image can be interpreted as 3D image; instead of a 2D line-fit a 3D plane-fit has been performed to estimate the pose from the given sensor data (Tzschichholz et al., 2015). For more details on camera based navigation experiments done on EPOS, we refer to Benninghoff, Boge, & Rems (2014); Klionovska & Benninghoff (2016); Tzschichholz (2014); Tzschichholz et al. (2011, 2015).

### 3.2.2 LiDAR Based Navigation

As an alternative to cameras, LiDAR (Light Detection And Ranging) sensors can be used as rendezvous sensors for relative navigation to a non-cooperative client satellite. LiDAR sensors provide a three-dimensional point cloud. At EPOS, we developed a 3D LiDAR sensor for real-time tracking and pose estimation (Rems et al., 2014). The sensor determines the distance to an object by emitting a laser beam and by analyzing the backscattered light.

By using a mirror system a certain 3D area can be scanned. Each point of the resulting 3D point cloud contains two direction angles (azimuth and elevation) and the measured distance - thus the full 3D information of a point. The relative pose of the client can be found by a variant of the Iterative Closest Point algorithm (Rusinkiewicz, 2001). A method to estimate the initial pose is proposed by Rems et al. (2015).

Figure 18 shows a photo of a robot carrying a client mockup and the corresponding scan of the LiDAR.

### 3.2.3 Hardware-in-the-Loop Rendezvous Tests

Additional to pure navigation tests, an entire Guidance, Navigation and Control (GNC) system can be embedded in a closed loop, hardware-in-the-loop simulation at EPOS. Even an On-Board Computer (OBC) or a representative computer playing the role of an OBC can be embedded in such a loop.



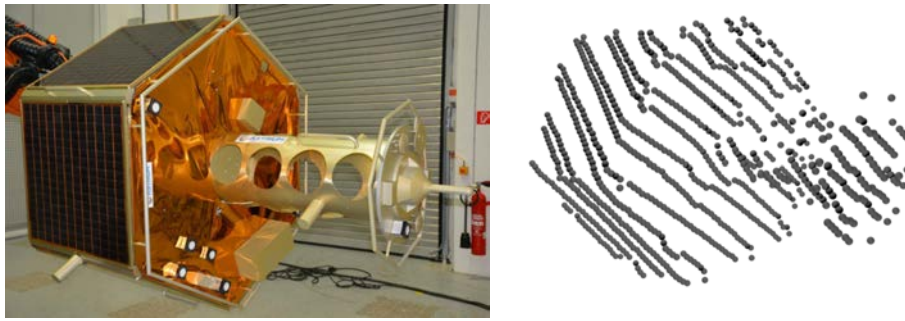


Figure 18: Exemplary LiDAR scan: reference photo showing a client satellite mockup at EPOS (left) and a scan set of the LiDAR (right).

Figure 19 visualizes a possible setup of such a hardware-in-the-loop test. One robot carries the mockup of a client satellite. The second robot carries a camera as rendezvous sensor. Using the available interfaces at EPOS (recall Section 2.2.9), the sensor data is transferred to a computer via Ethernet. This computer is called On-board Computer in the following since it simulates an On-board Computer of a real mission. On the On-board Computer the image data is processed by an image processing module followed by a navigation filter which estimates the relative position, velocity, attitude and attitude rate (12D estimated pose). A guidance and control module generates a reference guidance trajectory and attitude profile for a safe approach. A controller compares estimated state and guidance state and computes the necessary forces and torques, i.e. the actuator commands.

As discussed above, dynamic satellites models and actuator models have to be simulated numerically (see box Satellite Dynamic Simulator). This can be computed on the ACS-RT for example. The satellites' position and attitude values are sent to the FMC computer which sends final robot motion commands to the Local Robot Control Units (LRCs) of the robots.

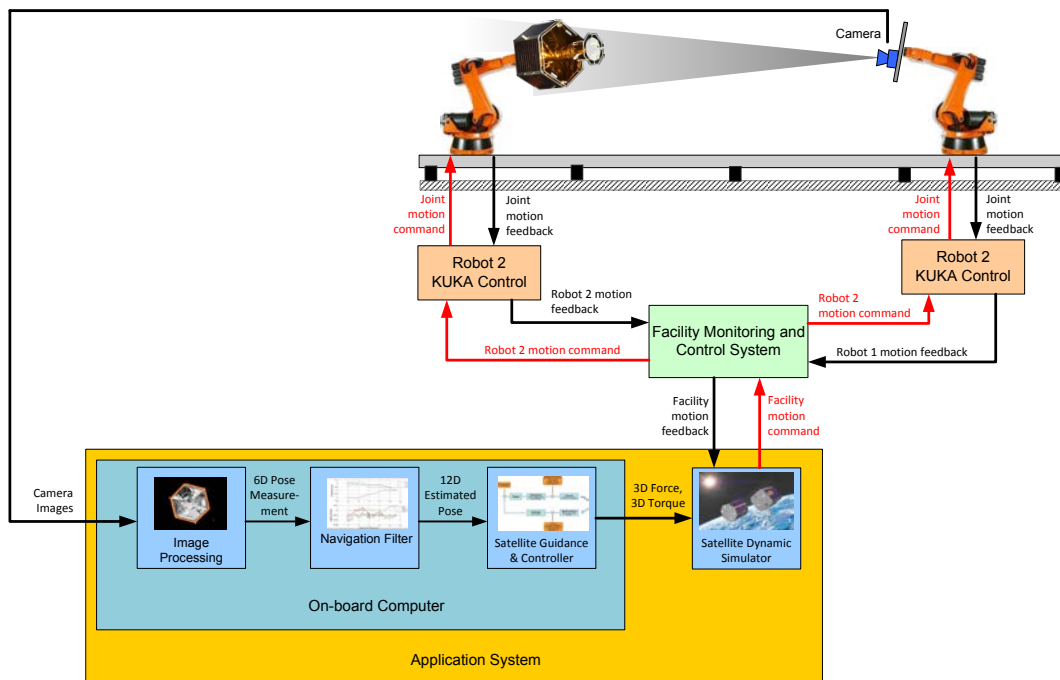


Figure 19: Hardware-in-the-Loop Simulation involving a camera as rendezvous sensor and an on-board computer.

Examples of hardware-in-the-loop simulations performed at EPOS can be found in Benninghoff et al. (2012); Benninghoff, Rems, & Boge (2014). Another hardware-in-the-loop simulation which uses advanced actuator models and which solves a mixed actuators optimization problem on-line in real-time is presented by Gomes dos Santos et al. (2015).

An operator in the loop test (Benninghoff, Rems, & Boge, 2014) has been conducted during the VIBANASS test campaign together with Kayser-Threde GmbH (now OHB-System AG) and DLR's institute of robotics. VIBANASS, VISION BASED NAVIGATION SENSOR SYSTEM, is a sensor system consisting of one mono mid range camera (CMOS chip) and a stereo close range camera system (2 cameras, CMOS chips). Further, VIBANASS contains a so-called Target Illumination System which sends out pulsed light and such improves the quality of images. Detailed technical descriptions of the VIBANASS system are given in Kaiser et al. (2011). Results of the test campaign at EPOS are presented by Mühlbauer et al. (2013) (mainly open loop sensor tests). Figure 20 shows the sensor system mounted on the tooling adapter plate of one of the EPOS robots and an operator on a local console who guided the image processing and the tracking of the target satellite using a joy stick. Details of an entire hardware-in-the-loop, closed-loop simulation performed during the VIBANASS test campaign, are given in Benninghoff, Rems, & Boge (2014).



Figure 20: VIBANASS sensor system (left) and operator-in-the-loop (right).

### 3.2.4 Contact Dynamic Simulations

In addition to rendezvous tests and simulations, the EPOS facility can also be used for validation and verification of satellite docking processes. The main concept is to create a closed loop docking simulation with force/torque feedback to the simulation. During contact, a force/torque sensor, see Figure 21, measures the actual forces and torques. This measurement is fed back to a numerical simulator. The concept is visualized in Figure 22.

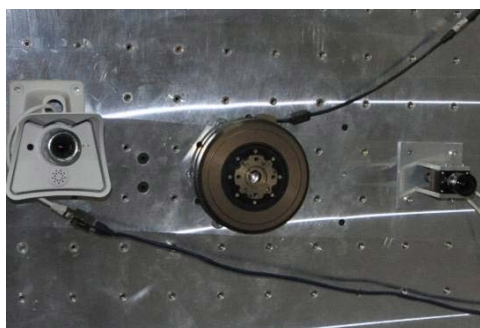


Figure 21: Force/Torque sensor mounted on a robot's Tooling Adapter Plate at EPOS.

The satellite simulator predicts the dynamic response of the client and servicing satellite based on a high-fidelity multi-body dynamics model of the satellite system. The robots of the EPOS facility

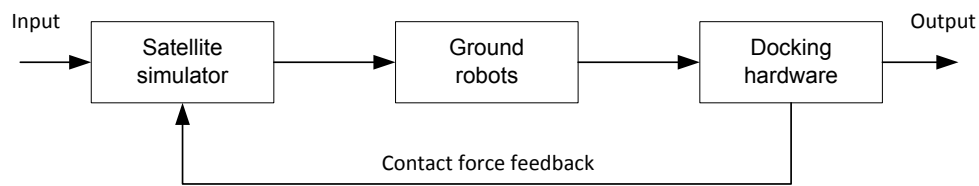


Figure 22: Force/Torque Feedback Loop of a Contact Dynamics Simulation

(ground robots) physically perform the 3D dynamic motion generated by the satellite simulator. A hardware mockup of the docking mechanism of the satellites is installed on the EPOS robots, which can make physical contact operations (docking hardware). Part of the docking hardware is the force/torque sensor. Its measurements are fed back to the simulation.

More details on contact dynamic and hybrid docking simulation can be found in Zebenay (2014); Zebe-nay et al. (2013, 2014).

### 3.3 On-Orbit Servicing End-to-End Simulation

Concerning on-orbit servicing missions, challenges arise not only in the space segment, but also in the ground segment. In addition to standard satellite consoles, a rendezvous console and a robotic console needs to be established and integrated in the ground segment. Robotic and GNC telemetry has to be received and corresponding telecommands from the consoles have to be sent via a certain communication system to the space segment. The idea of a DLR project called *On-Orbit Servicing End-to-End Simulation* is to not only simulate and verify sub-systems (like GNC systems, or robotic systems), but also to simulate the entire chain involving the control center, the communication system, a satellite simulator and two test beds for rendezvous and capturing: EPOS and a second facility called OOS-Sim located a DLR's institute of robotics. An overview is given by Figure 23.

For the end-to-end simulation of the rendezvous and berthing phase, the necessary components are developed and the entire chain will be tested and verified. Concerning the on-board system, this involves the robotic arm for grasping the client, the rendezvous sensors (CCD camera and LiDAR), and the on-board navigation and control software. A numerical satellite simulator contains the dynamic satellite models for orbit and attitude, and the on-board data handling system. The communication infrastructure is simulated, and the ground infrastructure is established including three consoles: a standard satellite bus console, a rendezvous console and a robotic console.

Finally, a series of end-to-end tests will be conducted; different approaches will be simulated. The first part, the approach to the final hold point will be performed using the EPOS facility. Then a switch to the OOS-Sim facility will be done.

In the project, the EPOS facility is connected to a real control room and telemetry and telecommands are exchanged between the on-board GNC system (at EPOS) and the consoles.

## 4 Conclusion

In this article, we presented a detailed technical description of the European Proximity Operations Simulator (EPOS) 2.0 located at DLR-German Space Operations Center in Oberpfaffenhofen, Germany. It is a robotic based test bed for test, verification and validation of rendezvous and docking systems (single hardware and/or software components, or even entire on-board computers or other complete systems).

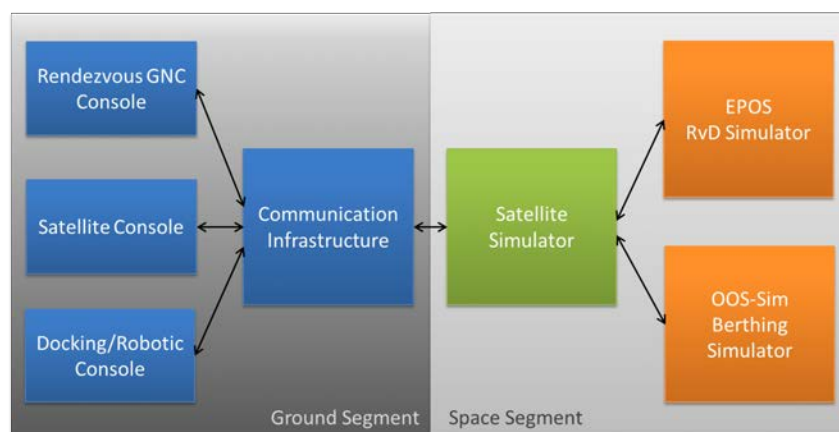


Figure 23: Overview on the elements of the On-Orbit Servicing End-to-End Simulation and their interaction.

Both open loop and closed loop tests can be performed at EPOS. The facility supports a variety of commanding modes, coordinate systems and interfaces. A Sun simulator allows to perform simulations with optical sensors under very realistic lightning conditions.

The last 25 m of the rendezvous phase (1:1 model) and the final docking phase can be simulated. Also longer rendezvous and inspection phases including a tumbling motion of the client can be realized, since two axes of robot 2 can be configured as endless axes. It is a highly accurate test bed with a high commanding rate of 250 Hz; therefore also contact dynamic simulations can be conducted.

To the authors' knowledge, the EPOS facility is unique in Europe with respect to its sizes, capabilities and performances.

## Acknowledgment

This paper is dedicated to Dr. Toralf Boge, who was mainly responsible for the establishment of the rendezvous and docking facility EPOS 2.0 and who built and formed a research group called "On-Orbit Servicing and Autonomy" based on his long experience in simulation, navigation and control. He passed away in November 2015, unexpected and far too early. His achievements are the basis of our current and future research work.

## A Appendix - Parameter Definition of the Command Interface

In this appendix the parameters used for the command interface (both synchronous and asynchronous interface) are described in detail.

As explained in Section 2.2.5, the commands of the position and attitude for robot 1 and robot 2 depend on the selected coordinate system. The four coordinate systems Ideal Joint Tool System (IJT), Ideal Device Coordinate System (IDC), Global / Lab Coordinate System (GLC) and Clohessy Wiltshire Coordinate System (CLW) exist for all interface modes described in Section 2.2.5. Recall Section 2.2.3 for the definition and description of the different coordinate systems.

The asynchronous and the synchronous command interface require 23 command values: one int16 value and 22 double values. The int16 value is a certain bit mask and mainly defines the coordinate system. The 22 double values of the command depend on the coordinate system and are given in Tables 6, 7, 8 and 9:

If the robots are commanded using the Ideal Joint Tool (IJT) coordinate system, see Table 6, an angle (in degree) for each of the six axes for both robots has to be provided. Further, since robot 1 is located on the linear rail system, an additional parameter (data lin) is needed which describes the position of

Nr	Command	Component	Description	Unit
1	CMD Rob 1	1	Robot 1, value of axis A1	deg
2		2	Robot 1, value of axis A2	deg
3		3	Robot 1, value of axis A3	deg
4		4	Robot 1, value of axis A4	deg
5		5	Robot 1, value of axis A5	deg
6		6	Robot 1, value of axis A6	deg
7		7	Not used	–
8	CMD Rob 2	1	Robot 2, value of axis A1	deg
9		2	Robot 2, value of axis A2	deg
10		3	Robot 2, value of axis A3	deg
11		4	Robot 2, value of axis A4	deg
12		5	Robot 2, value of axis A5	deg
13		6	Robot 2, value of axis A6	deg
14		7	Not used	–
15	CMD POV	1	Not used	–
16		2	Not used	–
17		3	Not used	–
18		4	Not used	–
19		5	Not used	–
20		6	Not used	–
21		7	Not used	–
22	data lin	1	Position of linear axis E1 (rail)	mm

Table 6: Description of the commanding parameters in case of Ideal Joint Tool (IJT) coordinate system

the robot (in mm) on the linear axis. The 7th index is not used for CMD Rob 1 and CMD Rob 2. Further, CMD POV is not used for this mode of commanding.

If the robots are commanded using the Ideal Device Coordinate system (IDC), see Table 7, the position and orientation of the tool system of each robot with respect to its base have to be provided. This is done by providing a 3D position (in mm) and a 3D orientation via Euler angles (in deg) for each of the two robots. For the Euler angle, the convention  $321 (= zyx)$  is used, i.e. the rotation is described by three consecutive rotations: a rotation with angle A around the z-axis, a rotation with angle B around the resulting y-axis (y-axis has changed due to the previous rotation) and finally by a rotation with angle C around the resulting x-axis (x-axis has changed due to the two previous rotations).

Further, since robot 1 is located on the linear rail system, an additional parameter (data lin) is needed which describes the position of the robot on the linear axis. The 7th index is not used for CMD Rob 1 and CMD Rob 2. Further, CMD POV is not used for this mode of commanding.



Nr	Command	Component	Description	Unit
1	CMD Rob 1	1	Robot 1, position tool w.r.t. base, x component	mm
2		2	Robot 1, position tool w.r.t. base, y component	mm
3		3	Robot 1, position tool w.r.t. base, z component	mm
4		4	Robot 1, orientation tool w.r.t. base, Euler-angle A rotation around z-axis	deg
5		5	Robot 1, orientation tool w.r.t. base, Euler-angle B rotation around resulting y-axis	deg
6		6	Robot 1, orientation tool w.r.t. base, Euler-angle C rotation around resulting x-axis	deg
7		7	Not used	–
8	CMD Rob 2	1	Robot 2, position tool w.r.t. base, x component	mm
9		2	Robot 2, position tool w.r.t. base, y component	mm
10		3	Robot 2, position tool w.r.t. base, z component	mm
11		4	Robot 2, orientation tool w.r.t. base, Euler-angle A rotation around z-axis	deg
12		5	Robot 2, orientation tool w.r.t. base, Euler-angle B rotation around resulting y-axis	deg
13		6	Robot 2, orientation tool w.r.t. base, Euler-angle C rotation around resulting x-axis	deg
14		7	Not used	–
15	CMD POV	1	Not used	–
16		2	Not used	–
17		3	Not used	–
18		4	Not used	–
19		5	Not used	–
20		6	Not used	–
21		7	Not used	–
22	data lin	1	Position of linear axis E1 (rail)	mm

Table 7: Description of the commanding parameters in case of Ideal Device Coordinate system (IDC)

Nr	Command	Component	Description	Unit
1	CMD Rob 1	1	Robot 1, position in the laboratory, x component (movement performed by rail system)	m
2		2	Robot 1, position in the laboratory, y component	m
3		3	Robot 1, position in the laboratory, z component	m
4		4	Robot 1, orientation w.r.t. laboratory, Euler-angle A rotation around z-axis	deg
5		5	Robot 1, orientation w.r.t. laboratory, Euler-angle B rotation around resulting y-axis	deg
6		6	Robot 1, orientation w.r.t. laboratory, Euler-angle C rotation around resulting x-axis	deg
7		7	Not used	–
8	CMD Rob 2	1	Robot 2, position in the laboratory, x component	m
9		2	Robot 2, position in the laboratory, y component	m
10		3	Robot 2, position in the laboratory, z component	m
11		4	Robot 2, orientation w.r.t. laboratory, Euler-angle A rotation around z-axis	deg
12		5	Robot 2, orientation w.r.t. laboratory, Euler-angle B rotation around resulting y-axis	deg
13		6	Robot 2, orientation w.r.t. laboratory, Euler-angle C rotation around resulting x-axis	deg
14		7	Not used	–
15	CMD POV	1	Not used	–
16		2	Not used	–
17		3	Not used	–
18		4	Not used	–
19		5	Not used	–
20		6	Not used	–
21		7	Not used	–
22	data lin	1	Linear displacement of Robot 1 Tool Coordinate System w.r.t. its Base Coordinate System	m

Table 8: Description of the commanding parameters in case of Global / Lab Coordinate System (GLC)

If the robots are commanded using the Global / Lab Coordinate system (GLC), see Table 8, the position and orientation of a frame fixed at the robot adapter plate w.r.t. the GLC system have to be commanded. The frame on the adapter plate is defined using a certain user tool which is defined in a text file and has to be loaded on the FMC computer before the commanding can be started. An exemplary user tool text file is shown in Figure 24. Units used for the user tool file are mm (positions) and deg (angles).

When commanding the robots in GLC coordinates, a 3D position (in m) and a 3D orientation via Euler angles (in deg) for each of the two robots have to be provided. For the Euler angles, the convention 321 (= zyx) is used. The x-coordinate motion for robot 1 (CMD Rob 1, x) is performed by the rail system. For a full description, one needs to define the x-displacement of the tool coordinate system of robot 1 with respect to its base coordinate system. This is done by providing the displacement (in m) via the parameter data lin.

The 7th index is not used for CMD Rob 1 and CMD Rob 2. Further, CMD POV is not used for this mode of commanding.

If we command using the Clohessy Wiltshire coordinate system (CLW), see Table 9, we need to command three reference frames: a frame fixed to spacecraft 1 represented by robot 1, a frame fixed to spacecraft 2 represented by robot 2 and the point of view, which is a reference frame describing the transformation from the scenario used in the application to the GLC system.

Nr	Command	Component	Description	Unit
1	CMD Rob 1	1	Robot 1, position w.r.t. reference frame, x component (movement performed by rail system)	m
2		2	Robot 1, position w.r.t. reference frame, y component	m
3		3	Robot 1, position w.r.t. reference frame, z component	m
4		4	Robot 1, orientation w.r.t. reference frame, Quaternion x component	-
5		5	Robot 1, orientation w.r.t. reference frame, Quaternion y component	-
6		6	Robot 1, orientation w.r.t. reference frame, Quaternion z component	-
7		7	Robot 1, orientation w.r.t. reference frame, Quaternion scalar component	-
8	CMD Rob 2	1	Robot 2, position w.r.t. reference frame, x component	m
9		2	Robot 2, position w.r.t. reference frame, y component	m
10		3	Robot 2, position w.r.t. reference frame, z component	m
11		4	Robot 2, orientation w.r.t. reference frame, Quaternion x component	-
12		5	Robot 2, orientation w.r.t. reference frame, Quaternion y component	-
13		6	Robot 2, orientation w.r.t. reference frame, Quaternion z component	-
14		7	Robot 2, orientation w.r.t. reference frame, Quaternion scalar component	-
15	CMD POV	1	Reference frame, position in the laboratory, x component	m
16		2	Reference frame, position in the laboratory, y component	m
17		3	Reference frame, position in the laboratory, z component	m
18		4	Reference frame, orientation w.r.t. laboratory, Quaternion x component	-
19		5	Reference frame, orientation w.r.t. laboratory, Quaternion y component	-
20		6	Reference frame, orientation w.r.t. laboratory, Quaternion z component	-
21		7	Reference frame, orientation w.r.t. laboratory, Quaternion scalar component	-
22	data lin	1	Linear displacement of Robot 1 Tool Coordinate System w.r.t. its Base Coordinate System	m

Table 9: Description of the commanding parameters in case of Clohessy Wiltshire Coordinate System (CLW)

```

1 USER_ROB1_TOOL_X=0.0
2 USER_ROB1_TOOL_Y=140.0
3 USER_ROB1_TOOL_Z=50.0
4 USER_ROB1_TOOL_A=0.0
5 USER_ROB1_TOOL_B=90.0
6 USER_ROB1_TOOL_C=-90.0
7 USER_ROB2_TOOL_X=0.0
8 USER_ROB2_TOOL_Y=0.0
9 USER_ROB2_TOOL_Z=480.0
10 USER_ROB2_TOOL_A=0.0
11 USER_ROB2_TOOL_B=-90.0
12 USER_ROB2_TOOL_C=-90.0
13
14

```

Figure 24: Example of a User Tool text file "user\_tool.val"

We need to provide a 3D position (in m) and an orientation via quaternions for each of the three reference frames. Therefore CMD Rob 1, CMD Rob 2 and CMD POV consist of 3 position and 4 attitude commands.

For a full description, one needs to define the x-displacement of the tool coordinate system of robot 1 with respect to its base coordinate system. This is done by providing the displace (in m) through the parameter data lin.

## References

- Benninghoff, H., Boge, T., & Rems, F. (2014). Autonomous Navigation for On-Orbit Servicing. *KI - Künstliche Intelligenz*, 28(2), 77–83. <http://dx.doi.org/10.1007/s13218-014-0297-0>
- Benninghoff, H., Boge, T., & Tzschichholz, T. (2012). Hardware-in-the-Loop Rendezvous Simulation Involving an Autonomous Guidance, Navigation and Control System. *Advances in the Astronautical Sciences*, 145.
- Benninghoff, H., Rems, F., & Boge, T. (2014). Development and hardware-in-the-loop test of a guidance, navigation and control system for on-orbit servicing. *Acta Astronautica*, 102, 67 - 80. <http://dx.doi.org/10.1016/j.actaastro.2014.05.023>
- Boge, T., Rupp, T., Landzettel, K., Wimmer, T., Mietner, C., Bosse, J., & Thaler, B. (2009). Hardware in the Loop Simulator for Rendezvous and Docking Maneuvers. In *Proc. German Aerospace Congress of DGLR*. Aachen, Germany.
- Boge, T., & Schreutelkamp, E. (2002). A New Commanding and Control Environment for Rendezvous and Docking Simulations at the EPOS-Facility. In *Proc. 7th Int. Workshop on Simulation for European Space Programmes (SESP)*. Noordwijk, The Netherlands.
- Boge, T., Wimmer, T., Ma, O., & Tzschichholz, T. (2010). EPOS - Using Robotics for RvD Simulation of On-Orbit Servicing Missions. In *Proc. AIAA Guidance, Navigation, and Control Conference*. Toronto, Canada.
- Ellery, A., Kreisel, J., & Sommer, B. (2008). The Case for Robotic On-Orbit Servicing of Spacecraft: Spacecraft Reliability is a Myth. *Acta Astronautica*, 63(5-6). <http://dx.doi.org/10.1016/j.actaastro.2008.01.042>

- Gomes dos Santos, W., Rocco, E. M., Boge, T., Rems, F., & Benninghoff, H. (2015). Discrete Multiobjective Optimization Methodology applied to the Mixed Actuators Problem and Tested in a Hardware-in-the-loop Rendezvous Simulator. In *Proc. AIAA Guidance, Navigation, and Control Conference*. Kissimmee, USA.
- Kaiser, C., Rank, P., Landzettel, K., Boge, T., & Turk, M. (2011). Vision Based Navigation for Future On-Orbit Servicing Missions. In *Proc. 62nd International Astronautical Congress (IAC)*.
- Klionovska, K., & Benninghoff, H. (2016). Visual Navigation for Rendezvous and Docking using PMD Camera. In *Proc. 2nd International Conference on Geographical Information Systems Theory, Applications and Management (GISTAM)*. Rome, Italy.
- KUKA Documentation. (2009a). *KUKA KR 100 HA Manual*.
- KUKA Documentation. (2009b). *KUKA KR 240-2 Manual*.
- Mühlbauer, Q., Rank, P., & Kaiser, C. (2013). On-Ground Verification of VIBANASS (VIsion Based NAvigation Sensor System): Capabilities and Results. In *Proc. 12th Symposium on Advanced Space Technologies in Robotics and Automation*. Noordwijk, The Netherlands.
- Nishida, S.-I., Kawamoto, S., Okawa, Y., Terui, F., & Kitamura, S. (2009). Space Debris Removal System Using a Small Satellite. *Acta Astronautica*, 65(1-2). <http://dx.doi.org/10.1016/j.actaastro.2009.01.041>
- Rems, F., Fritz, S., & Boge, T. (2014). Breadboard Model of a 3D LiDAR Sensor for Real-Time Pose Estimation of Spacecraft. In *Proc. 9th International ESA Conference on Guidance, Navigation & Control Systems*. Porto, Portugal.
- Rems, F., Moreno Gonzalez, J. A., Boge, T., Tuttas, S., & Stilla, U. (2015). Fast Initial Pose Estimation of Spacecraft from LiDAR Point Cloud Data. In *Proc. 13th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*. Noordwijk, The Netherlands.
- Rusinkiewicz, S. (2001). Efficient Variants of the ICP Algorithm. In *Proc. 3rd International Conference on 3D Digital Imaging and Modeling*. Quebec City, Canada.
- Tzschichholz, T. (2014). Relative Pose Estimation of Known Rigid Objects using a Novel Approach to High-Level PMD-/CCD- Sensor Data Fusion with regard to Applications in Space. *Dissertation, University Würzburg, DLR*.
- Tzschichholz, T., Boge, T., & Benninghoff, H. (2011). A Flexible Image Processing Framework for Vision-Based Navigation Using Monocular Imaging Sensors. In *Proc. 8th International ESA Conference on Guidance, Navigation & Control Systems*. Karlovy Vary, Czech Republic.
- Tzschichholz, T., Boge, T., & Schilling, K. (2015). Relative Pose Estimation of Satellites Using PMD-/CCD-Sensor Data Fusion. *Acta Astronautica*, 109(25-33). <http://dx.doi.org/10.1016/j.actaastro.2014.12.010>
- Yasaka, T., & Ashford, E. W. (1996). GSV: An Approach Toward Space System Servicing. *Earth Space Review*, 5(2).
- Zebenay, M. (2014). Development of a Robotics-based Satellites Docking Simulator. *Dissertation, TU Delft, DLR*.
- Zebenay, M., Boge, T., & Choukroun, D. (2013). Modelling, Stability Analysis, and Testing of a Hybrid Docking Simulator. In *Proc. AIAA Guidance Navigation and Control Conference*. Boston, USA.
- Zebenay, M., Boge, T., Krenn, R., & Choukroun, D. (2014). Analytical and Experimental Stability Investigation of a Hardware-in-the-Loop Satellite Docking Simulator. *Journal of Aerospace Engineering*. <http://dx.doi.org/10.1177/0954410014539290>