Preparation of the Electronic Paper for the Proceedings of the

11th Int. WS on Simulation & EGSE facilities for Space Programmes
SESP 2010

28-30 September
at ESTEC, Noordwijk, the Netherlands

HARDWARE IN-THE-LOOP MULTI-SATELLITE SIMULATOR FOR PROXIMITY OPERATIONS

G. Gaias$^{(1)}$, S. D'Amico$^{(1)}$, J.-S. Ardaens$^{(1)}$, T. Boge$^{(1)}$

$^{(1)}$Deutsches Zentrum für Luft- und Raumfahrt (DLR), German Space Operations Center (GSOC)
Münchner Str. 20, 82234 Wessling, Germany

gabriella.gaias@dlr.d, simone.damico@dlr.de, jean-sebastien.ardaens@dlr.de, toralf.boge@dlr.de

INTRODUCTION

On-orbit servicing (OOS) activities constitute a novel class of spacecraft missions that recently has drawn the attention of agencies and industries. They are characterized by the interaction between two satellites, usually referred as *servicer* and *client* spacecraft, and their main objective is to navigate the servicer towards the client with the intention to manipulate it in a predefined manner. OOS mainly involve uncooperative satellites. Moreover it comprises several critical phases such as proximity motion, rendezvous and docking (RvD). Straightforward applications of the OOS concept deal with robotic inspection and assistance of flying satellites.
The German Space Operations Center (GSOC) is currently involved in two main projects, the DEOS (DEutsche Orbitale Servicing) mission and the international OLEV (Orbital Lifetime Extension Vehicle) mission [1-2]. DEOS is a technology demonstrator in low Earth orbit, in which two satellites shall perform various scenarios of RvD and, at the end of the mission, de-orbit in a coupled configuration. OLEV is meant to prolong the lifetime of telecommunication satellites in geostationary orbit. To this end the spacecraft has to dock to the client satellite and take over attitude and orbit control tasks for a maximum duration of 12 years. Furthermore OLEV shall be able to perform docking and undocking up to five times, in order to assist different client satellites.
Close-range autonomous relative navigation, docking and successive dynamics of a coupled system introduce various challenges in several areas of the mission design. Such issues concern the development and testing of specific hardware as, for example, docking mechanisms or visual based sensors to perform the close-range relative navigation. In the meantime there is the need to develop complete guidance navigation and control (GNC) architectures to accomplish peculiar tasks such as the approach to an uncooperative client taking into account collision avoidance and risk mitigation strategies, given the available sensors and actuators. In this frame, the capability to test the complete plant with a facility that operates in real-time and that runs hardware-based demonstrations constitutes a key factor. This definitely helps in analyzing the phenomenon and in improving the development process across instrument, subsystem and system levels. Moreover it would allow to validate simulations and to provide the technological maturity necessary to proceed with flight demonstrations and, eventually, a mission.

This paper addresses the first steps realized to set up a hardware in-the-loop (HWIL) multi-satellite simulator aimed at the support of closed-loop real-time real-scale OOS scenarios. The driving idea of this project is to join the advantages coming from a model-based design approach with the simulating capabilities provided by the new European Proximity Operations Simulator (EPOS 2.0). The result turns out into a flexible, application- and mission-independent facility capable to focus on the critical phase of separation ranging from 25m to 0m.
Testbeds can be divided into simulation and robotic types. Robotic testbeds physically move spacecraft models in response to algorithmic commands. Simulation testbeds do not have physical spacecraft-emulators, but can include hardware-in-the-loop (HWIL) for specific sub-systems such as GPS receivers and signal simulators [3]. Here the focus is on robotic testbeds since they are needed for the highest-fidelity pre-flight demonstrations. Robotic testbeds can be classified by the number of vehicles and degrees of freedom, but two distinct categories can be identified.

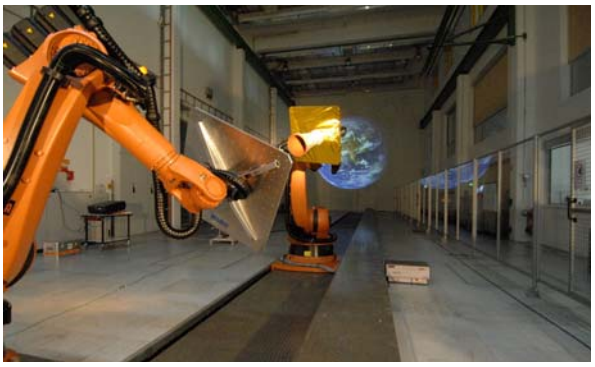


Fig. 1. The new EPOS 2.0 facility: robotics-based testbed (left) and operation station (right)

Kinetic testbeds are characterized by forces or torques which are applied as they would be in flight in at least one degree of freedom, for example by a thruster or reaction wheel. This type typically consists of air-levitated rigid bodies operating on a planar surface with two translational degrees of freedom and one rotational. *Kinematic* testbeds are actuated in response to simulated motion in all degrees of freedom, for example by a robotic arm. The most common kinematic testbed consists of spacecraft-emulators mounted on 6-DOF robotic arms. The arms can be fixed in location or mounted on planar gantry to allow larger motions in the gantry plane.
EPOS 2.0 is a 6-DOF kinematic testbed which is currently being upgraded to feature kinetic functionalities to fulfill specific OOS needs. In particular EPOS 2.0 has replaced the former EPOS facility which had been operating over several years mainly to support the development of the RvD systems for the unmanned supply and service-flights of the European Automated Transfer Vehicle (ATV) to the International Space Station (ISS) [4]. The new EPOS 2.0 has been specifically designed to provide test and verification capabilities for the complete RvD processes occurring in OOS missions. The facility comprises two industrial robots for physical real-time simulations of rendezvous and docking maneuvers. This testbed allows simulation of the last critical approaching phase, including the simulation of the contact dynamics during the docking process. With respect to the former set-up, the accuracy of the new testbed has been improved as measurement and positioning performances are increased by a factor of 10. Moreover dynamic capabilities allow for high commanding rates and provide force and torque measurements. Simulations of sunlight illumination conditions, as well as the compensation of Earth-gravity force, are both part of the assembly in order to generate an utmost realistic simulation of the real rendezvous and docking process. Finally, the utilization of standard industrial robotics hardware (HW) allows a very high flexibility related to different application scenarios.

A detailed overview of the HW included in the facility is given in the following. The paper continues with a description of the design approach which has been identified to support a flexible application set-up. The architecture of the whole simulation environment, interfaces and mechanism of the simulation are subsequently described. Finally results obtained from a first real-time HWIL application scenario are reported. Summary and way forward conclude this work.

THE NEW EUROPEAN PROXIMITY OPERATIONS SIMULATOR (EPOS 2.0) FACILITY

Future applications for satellite OOS missions require the EPOS 2.0 facility to be able to provide the following test and simulation capabilities:

- 6-DOF relative dynamic motion of two satellites in the final approaching phase from 25m to 0m,
- 6-DOF contact dynamic behavior during the entire docking process including the initial impact, soft docking, and hard (final rigidization),
- space-representative lighting and background conditions.

As sketched in Fig. 2, EPOS 2.0 consists of the following components:

- rail system mounted on the floor to move an industrial robot up to a distance of 25m,
- KUKA KR240 robot (Robot 1) mounted at the end of the rail system for simulating the client spacecraft,
- KUKA KR100HA robot (Robot 2) mounted on the rail system for simulating the servicer spacecraft,
- PC-based monitoring and control system to monitor and steer the RvD simulation on the facility. It can be divided into three levels:
    - Local Robot Control (LRC) system,
    - Facility Monitoring and Control (FMC) system,
    - Application Control System (ACS).

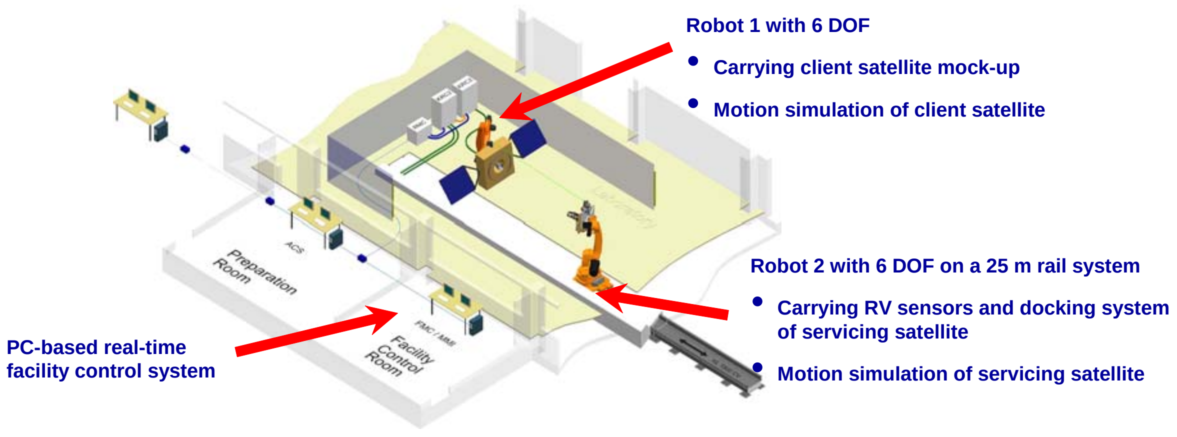
Each robot is independently controlled in real-time by its own LCR unit, provided by the robot manufacturer. At the FMC level the entire facility is controlled and monitored in real-time. Moreover, the FMC system allows the following tasks:

- operator's monitoring action over all the parameters and states of the facility,
- logging of all the parameters and states of the facility, including external synchronization signals,
- real-time control of the entire facility including synchronization of all motion devices and kinematical conversions of the external commands,
- choice among different interfaces (IF). The following options are available:
    - a synchronous IF (EtherCAT), for closed-loop applications,
    - an asynchronous IF, in order to run a predefined trajectory stored in a file,
    - a KUKA Robot Sensor Interface (RSI) IF to directly interface the FMC with the LRC units.

Finally, at an ACS level, the actual application of the dynamic system is run. In particular, the models of the satellites dynamics and the case-specific scenarios can be implemented in a MATLAB/Simulink environment. This means that the whole software related part of the simulation can exploit a model-based design approach. According to it, 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 the 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.

Since the EPOS 2.0 has been established, it was mainly used in its asynchronous working mode to support research and development tasks like the development of image processing algorithms for different rendezvous phases [6]. In asynchronous mode the robots replay a predefined trajectory, to support experiments involving docking devices and/or case-specific sensors. In this case the facility is said to work in *open-loop*. The topic of this work, however, relies in setting up an overall architecture, and in testing its functionalities, to make the facility running in a synchronous mode. Thus the robots execute a trajectory that is step-by-step computed by some internal logic embedded in the simulator. As a result the facility works based on *closed-loop* online computations, paving the way for future, more complex, fully closed-loop and HWIL demonstrations. It should be emphasized that our very final goal is the set-up of a testbed that allows the user to introduce real sensors in the loop. He should be also free to perform Software- and Processor- in the loop simulations, therefore embedding some on-board flight software. The overall architecture of the simulator is discussed in the second part of this report.

Finally, Tab. 1 specifies the EPOS 2.0 motion simulation performances and capabilities [10]. EPOS 2.0 will also be used for RvD sensor verification purpose, thus the facility has been extensively calibrated after its installation. With a laser tracker device an overall positioning accuracy of the facility of better than 2 mm (3D, 3σ) and an orientation accuracy of 0.2° (3D, 3σ) have been verified [5]. In addition it is planned to develop an online measurement system which measures the relative position between both robots and commands corrections to the robots. So the achieved position accuracy will be in sub millimeter range.



Robot 1 with 6 DOF
Carrying client satellite mock-up
Motion simulation of client satellite
Robot 2 with 6 DOF on a 25 m rail system
Carrying RV sensors and docking system of servicing satellite
Motion simulation of servicing satellite
PC-based real-time facility control system
Preparation Room
ACS
FMC / MMI
Facility Control Room

Fig. 2. Components of the new testbed – EPOS 2.0 [5]

Tab. 1 EPOS 2.0 motion capabilities [10]

Parameter	Robot 1	Robot 2
Position: (Global Lab)		
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
Attitude A_{ZYX}: (Global Lab)		
Roll [deg]	-300 - +300	-300 - +300
Pitch [deg]	-90 - +90	-90 - +90
Yaw [deg]	-90 - +90	-90 - +90
Max. tip velocity:		
Translational [m/s]	2	2
Rotational [deg/s]	180	180
Command IF		
Command rate [Hz]	250	250
First natural frequency [Hz]	8-10	8-10

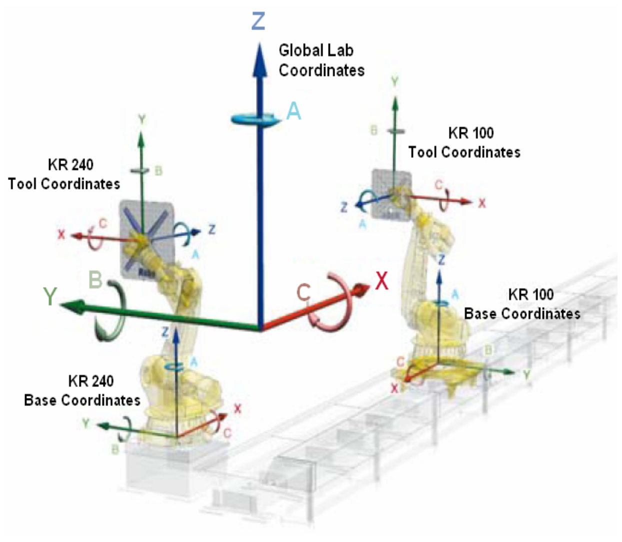


Fig. 3. Reference systems [10]

MODEL-BASED SIMULATION DESIGN

Model-based design (MBD) has become increasingly more popular in the last years mainly due to its efficient and cost-effective properties. Within the space field, it has been successfully applied in the on-board software design of the SMART-1 (Small Missions for Advanced Research in Technology) Moon probe and the PRISMA Formation Flying mission [7]. MBD consists in an iterative process in which models of the system, automatic code generation and tests and verifications campaigns are exploited. This approach starts with the very first design trade-offs and is carried out till the achievement of a prototype. Thus it goes rapidly through an increase of the level of detail of the implemented models and of the system specialization itself. Thanks to the capability to run straight faster than real-time and real-time simulations the designer can perform the usual design iterations in a definitely quicker way. In the meantime, the design process can take advantage of performing effective trade-offs and extensive investigations with reasonable costs in terms of time and resources.

Among the available products, the simulator addressed in this paper exploits the Mathworks related tools: Simulink and Real-Time Workshop (RTW). Such a choice is justified by the user-friendly interface and by the highly modular structure that Simulink can offer. Nevertheless, eventual complex functionalities can be introduced in the models through S-functions written in C/C++ language. RTW is then configured to generate automatic code as supported by VxWorks. The executable so produced is loaded on the target PC that communicates with the FMC system.

Regarding the ACS layer, non-RT pure Simulink simulations for software development purpose are run on a PC with Windows XP operating system. RT simulations are supported by a PC with VxWorks operating system.

Fig. 4 gives an overview of the whole architecture of the simulator. As mentioned before, both robots are locally controlled by their LRC units. They interface with the FMC system in real-time and require commands to be imparted every 4 ms. Such a constraint is reflected on the output of the ACS: all the kinematical information must be provided at that sample rate. Each robot requires as a command the position and the orientation of a frame. The definition of this local frame with respect to the robot flange can be specified in a mask at the FMC level. Thus, for example, the user could refer to the spacecraft center of mass or to another geometrical point of the mock up meaningful for his application. The local frame could be the orbital one centered in the satellite, the body-frame or whatever frame related to a specific device. This flexibility allows to set up different scenarios without affecting the ACS core. Moreover, it shall be emphasized that all additional kinematical conversions are carried out by the FMC system in real-time.

Finally, data logging, takes place both at FMC and ACS levels.

The right part of Fig. 4 focuses on the ACS/RT level; here the main functionalities needed for setting up an OOS scenario are reported. They consist of:

- propagator of the 6-DOF dynamics of both satellites, including the effects of the external environment,
- models of translational and rotational actuators acting on the dynamics of the spacecraft,
- models of the sensors,
- GNC core (observer, guidance and control profiles computation),

- IF with the operator to allow his manual guidance on the servicer (6-DOF mouse or direct commands from keyboard during the RT simulation).

With the increase of the maturity level of the facility, in order to build a fully closed-loop testbed, some of the functionalities listed above should be removed from the ACS and replaced by their real counterparts. Thus sensors' and GNC boxes could be substituted by respectively the real hardware and the on-board flight software. Thanks to its modular architecture the simulator is prearranged towards these improvements.

At the beginning of every simulation, a synchronization process between FMC/RT and ACS/RT takes place. According to it, first the aimed initial positions and attitudes are communicated to the FMC. Then, as soon as the robots achieve those conditions, a confirmation signal is sent back to the ACS in order to activate the start of the propagation. Due to the fact that the robots are real mechanical systems they can not realize arbitrary variations of their motion instantaneously. As a consequence specific constraints on maximum accelerations and jerks have to be reflected at an application layer. If for example a desired scenario requires a non-zero relative velocity at the initial time, a preparation phase is necessary to gradually lead the robots to the proper initial conditions – position and velocity.

THE APPLICATION CONTROL SYSTEM LAYER

This part of the report focuses on the ACS level and explains how the specific OOS problem is modeled within the Simulink environment. If we refer to Fig. 4 (right), in general terms, the problem can be divided into the propagation of the complete dynamics of each satellite and the whole GNC chain.

Modeling of the dynamics

The propagator block is composed by the following elements:

- solver engine that accomplishes the numerical integration,
- set of models covering all the relevant phenomena that influence the level of detail of the application,
- graphical user-friendly interface, i.e. mask, to let the user to set his preferences.

Integration methods vary according to solution scheme, order and size of the integration step. The choice of which method to use is strictly related to the nature of the problem studied. Moreover, in case of real-time applications, issues concerning synchronization of the operations and computational load play a fundamental role. Regarding the satellite orbit dynamics, to achieve a sufficient accuracy of the solution, a high order method is usually preferred. Reasonable integration step-sizes range around 1 s.

Both in Continuous-Time and Discrete-Time configurations Simulink provides pre-packed integrator blocks which offer families of fixed-step methods and which allow to set the desired integration macro-step. Despite this, the user is partially allowed to access the evaluations of the derivatives that occur at some internal, method-related, micro-steps. Moreover he is prevented to run any other block at a faster sample time than the macro-step of the integrator block. Other issues arise during the automatic code generation process. Here, in fact, the user has no easy control on how the synchronization between the temporization of the real-time target and the method-related micro-steps is managed. Whenever they do not perfectly coincide, the order of the integration method is simply degraded.

Therefore, the design of a dedicated solver engine has been necessary to take into account the lack of transparency in the automatic code generation, the 4 ms constraint set by the LCR units and the physics of the satellite dynamics. The newly developed integrator works in the Simulink Discrete-Time configuration and runs at the sample time required by the target machine, i.e. 4 ms for the FMC/RT. The solver exploits simple functionalities and blocks which are supported by the automatic code generation process.

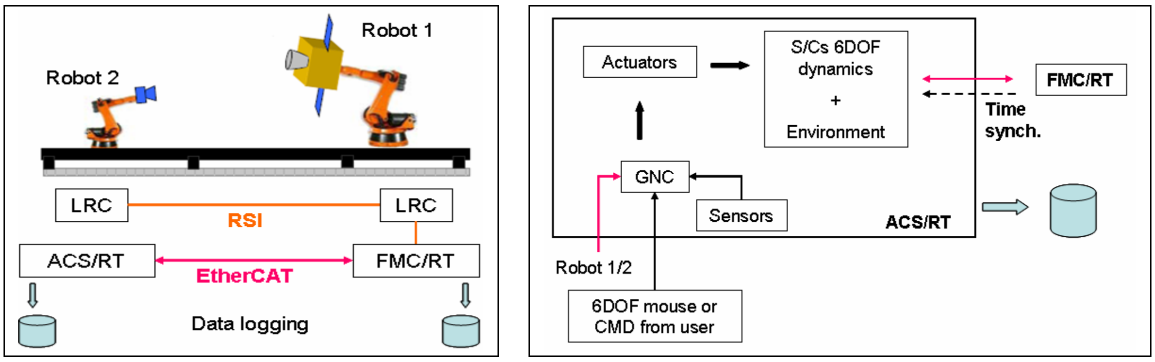


Fig. 4. The architecture of the simulator: system levels and IFs (left); view of ACS for OOS applications (right).

The user can choose among the following fixed-step solution methods: Heun, Bogacki-Shampine, Runge-Kutta 4th order and Dormand-Prince. To assure portability, the desired macro-step is eventually slightly corrected depending on both the method and the temporization required by the target machine. As a result both synchronization and a rigorous control on the flux of the information are possible. Nevertheless, considering that not necessarily every simulation-tic coincides with a micro-step of the integration method, an interpolator is needed to provide a continuous (C^0) output. In particular, here we make use of a Hermite interpolation to take advantage of the available knowledge of the derivates at each micro-step. The coupling of a fixed-step method and Hermite interpolation has been demonstrated to be very efficient for spacecraft on-board applications where a clear separation of time-consuming and rapid turn-around functions is necessary to cope with the limited processor resources. For reference the proposed approach was already used for the BIRD Small Satellite [8] and the PRISMA formation flying missions [9]. Here, that approach is generalized over more integration methods, still pursuing the idea of exploiting the periodicity of the time distribution of the data to be interpolated.
The following perturbation factors can be included in the orbit dynamics of each satellite:

- gravity potential of a non homogeneous mass distribution: maximum accuracy of the 30th order and degree,
- aerodynamic drag and solar radiation pressure,
- Moon Sun perturbations and tidal effects.

The following perturbation factors can be included in the attitude dynamics of each satellite:

- gravity gradient torque: Earth modeled as a sphere,
- torque due to Earth magnetic field according to the International Geomagnetic Reference Field (IGRF) 2010 model till order 13th. As an alternative a simpler dipole model can be exploited,
- solar radiation pressure and aerodynamic torques. To compute them a description of the geometrical and optical characteristics of each satellite is needed. To this extent spacecraft are modeled as a set of surfaces composed by rectangular mesh-elements. The finer the mesh the higher the accuracy of the computed torque. No shadowing effects are taken into account,
- internal torque due to misalignment of the total thrust vector.

All the preferences listed so far, both concerning the solver engine and the modeling of the external environment, can be specified by the user, separately for each spacecraft. A dedicated mask, at the ACS level, is devoted to this task.

GNC functionalities

The complete GNC chain should be constituted by sensors, observer, guidance profiles generator, controllers and actuators. Currently, just some of these functionalities are developed, as long as the aim of the work was to set up an unsophisticated show-case to test the functioning of the whole architecture. In particular, the behavior of sensors and actuators is not modeled. Therefore they act as they were ideal. Moreover it is assumed a perfect knowledge of all the information needed by the controllers. Thus no observer is included yet.
The guidance module, instead, is meant to define an aimed operative state for the servicer spacecraft and to generate a reference profile to let the servicer achieving the aforementioned operative condition. Regarding the first task, the user can specify a desired relative position and relative attitude that the servicer should assume with respect to the client satellite. Keeping in mind the objectives of OOS missions, in fact, it is more useful to reason in the body-frame of the client spacecraft. Practically, the user can introduce his requirements either off-line in the Simulink model of the application or with direct commands, through keyboard or using a 6-DOF mouse. Both these last two options occur during the real-time simulation. Hence they provide a great level of flexibility and the capability to set up realistic scenarios to test the reaction of the closed-loop system.
The guidance module combines desired, client-body-frame requirements with the motion of that satellite to compute the real state of the aimed operative point. This information is then used to accomplish its second task: to determine the reference profile that the controllers shall track. Currently, this functionality is simplified and the reference profile coincides with the evolution on time of the final condition. Thus just small deviations from the desired operative point can be recovered.
The implemented controllers are simple and not specifically suited on a given application. In particular they consist of:

- Linear Quadratic Regulator (LQR) for the absolute attitude control of the client satellite,
- LQR for the relative attitude control of the servicer to achieve the desired attitude of the operative condition,
- LQR for the relative orbital motion towards the operative point.

FIRST RESULTS FROM HARDWARE-IN-THE-LOOP DEMONSTRATION

A show-case scenario is here presented. It is performed by two satellites having the characteristics of the two spacecraft of the PRISMA formation flying mission [9].

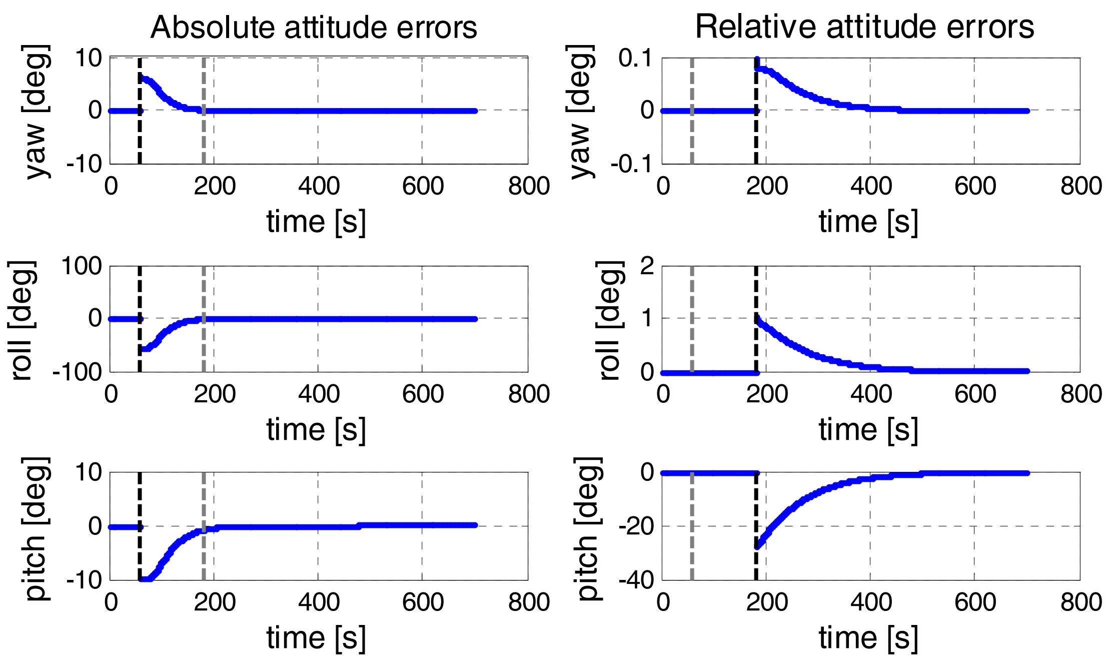


Fig. 5. Attitude error transients for client (left) and servicer (right).

In particular, the client is a nano-satellite with 3-axis coarse attitude control capability and without any orbital control, named *Target*. The servicer is a fully maneuverable micro-satellite, named *Main*. At the beginning of the simulation they are displaced by [0.0 5.1 0.0] m in the Radial-Tangential-Normal (RTN) client-centered co-moving orbital frame.
The servicer is asked to achieve a displacement of 5 m in the +y direction of the client body-frame. Moreover its body-frame shall be aligned with the one of the client, with +y direction of opposite verse.
As shown in Fig. 5, at initial time *Target* has a relevant misalignment with respect to its desired attitude, i.e. body-frame aligned with orbital frame RTN. *Main* is also not perfectly Earth pointing: it is a bit tilted and has some error in its angular velocity.
A predefined sequence of actions is planned in the scenario. Hence, in this case, the aimed operative point is not changed in real-time. The plan consists in correcting the absolute attitude of the client first, and then the servicer starts its approach. It corrects its relative position and afterwards its relative attitude. The absolute attitude control is activated after 60 s, whereas the relative attitude and relative position controls are respectively activated after 120 s and 180 s from the initial time.
As shown in Fig. 5 and Fig. 6 the servicer spacecraft is free drifting during the time in which all controllers are disabled (i.e., first 60 s) and when only the absolute attitude of the client is adjusted (i.e., from 60 s to 120 s). The applied initial conditions provide a small relative displacement and almost null relative velocity, thus the accumulated drift is rather small. When the relative control towards the operative point is enabled, the client attitude has almost converged, thus +y body-frame coincides with +T RTN. As a result the commanded position of robot-2 (i.e., 5 m offset in +y direction) consists in a maneuver mainly in the along-track direction. While approaching, the servicer starts performing the slew to achieve the aimed relative attitude. At convergence, the operative condition is kept stable by counteracting the external disturbances. The first 700s are reported in the figures.
Despite the apparent simplicity of the presented scenario, most of the intended goals for the EPOS 2.0 demonstration have been achieved. The simulation portability from a pure Matlab/Simulink environment to the real-time VxWorks target and finally to the hardware robots has been demonstrated. The real-time steering of the robots through remote motion commands sent every 4 ms has been demonstrated. Rigorous orbit and attitude dynamics are included in the simulation. Furthermore the autonomous control of a servicer with respect to the client spacecraft has been implemented for relative orbit and attitude motion.

SUMMARY AND WAY FORWARD

This paper has reported the first steps performed to set up a hardware in-the-loop multi-satellite simulator for proximity operations. Attention has been focused on establishing an overall architecture capable to perform mission-independent applications. Moreover the simulation environment is already structured to support future developments towards more representative, fully closed-loop hardware in the loop demonstrations.

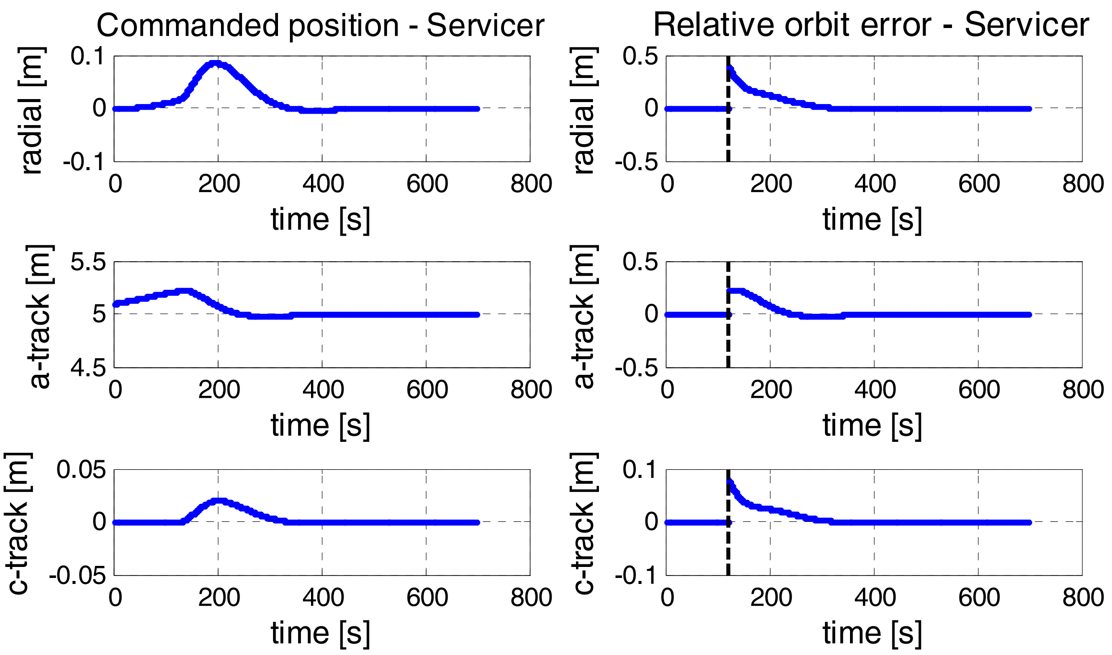


Fig. 6. Commanded position for the R2 robot in RTN-client (left). Relative orbit transient to achieve the aimed operative point (right). It is expressed in the RTN-servicer frame.

The main results gained so far give a greater insight into the synchronous working mode of the EPOS 2.0 facility and the interfaces among the different architecture blocks. Heritage of previous work in the frame of orbit and attitude modeling as well as numerical integration methods for space applications has been exploited to consolidate the real-time simulator and solve portability issues related to the model-based design approach.
Forthcoming upgrades in the simulation environment deal with the development of some missing functionalities which have been neglected so far. That concerns mainly the GNC chain which is mission dependent and requires the consideration of a specific application scenario. Future further developments include the introduction of real sensors and the implementation of more sophisticated scenarios as the ones involved in on-orbit servicing operations.

REFERENCES

[1] Th. Rupp, T. Boge, R. Kiehling, and F. Sellmaier, "Flight Dynamics Challenges of the German On-Orbit Servicing Mission DEOS," *21st International Symposium on Space Flight Dynamics*, Toulouse, France, 28 Sep. - 2 Oct., 2009.

[2] F. Sellmaier, T. Boge, J. Spurmann, S. Gully, T. Rupp, and F. Huber, "On-Orbit Servicing Missions: Challenges and Solutions for Spacecraft Operations," AIAA 2010-2159; *SpaceOps 2010 Conference*, Huntsville, Alabama, 25-30 April, 2010.

[3] J.-S. Ardaens and S. D'Amico, "Formation Flying Testbed," DLR-GSOC TN 09-01, Deutsches Zentrum für Luft- und Raumfahrt, Oberpfaffenhofen, 2009.

[4] R.L. Stapf and E. Schreutelkamp, "Preparing EPOS for the Future," *6th International Workshop on the Simulation of European Space Programmes*, SESP 2000, Noordwijk ,10-12 Oct., 2000.

[5] EPOS – Facility Manual, Robo-Technology GmbH, Puchheim, Germany, 2009.

[6] T. Tzschichholz and T. Boge, "GNC Systems Development in conjunction with a RVD Hardware-in-the-loop Simulator," *4th International Conference on Astrodynamics Tools and Techniques ICATT*, European Space Astronomy Centre ESAC, Madrid, Spain, 3-6 May, 2010.

[7] T. Olsson and A. Edfors, "Model-Based Onboard Software Design: The Prisma Case Study," *DAta Systems In Aerospace*, DASIA 2006, Berlin, Germany, 22-25 May 2006.

[8] E. Gill and O. Montenbruck, "The Onboard Navigation System for the BIRD Small Satellite," DLR FB 2002-06; Deutsches Zentrum für Luft- und Raumfahrt, Oberpfaffenhofen, 2002.

[9] S. D'Amico, J.-S. Ardaens, and O. Montenbruck, "Navigation of Formation Flying Spacecraft using GPS: the PRISMA Technology Demonstration," *ION-GNSS-2009*, Savannah, USA, 22 Sep. - 25 Sep., 2009.

[10] T. Boge, et al., "Hardware in the Loop Simulator von Rendezvous und Docking Manövern," *German Aerospace Congress of DGLR,* Aachen, Germany, 8-10 September, 2009.