Guidance, navigation and control for autonomous close-range-rendezvous

E. Risse, K. Schwenk, H. Benninghoff, F. Rems

German Aerospace Center (DLR), German Space Operations Center (GSOC), Münchner Str. 20, 82234 Wessling, Germany

Abstract

This article presents the Guidance, Navigation and Control system (Gnc-system) developed by GSOC's On Orbit Servicing (OOS)-group. It is used for research on autonomous rendezvous, optical sensors and operational concepts with the European Proximity Operations Simulator 2.0 (EPOS) facility and inside an End-to-End simulation. It's modular design intends to make it capable of deployment in space on a novel on-board computer with less effort. Major design decisions as well as plans for future development are introduced.

1 INTRODUCTION

Since the early 70's, space debris has been identified for the first time as a major potential issue for future spaceflight missions. It's impact has increased ever since. In recent years, with new mega constellations such as Oneweb [1] and Starlink [2] to be launched, the necessity for a more sustainable use of space resources is gaining new momentum. In this context, OOS can provide a valuable and indispensable way to keep important orbits usable. The recent successful deployment of a Mission Extension Vehicle [3] gives an example of cooperative partly automated OOS, while far more challenging tasks like ClearSpace [4] lie still ahead.

With it's large-scale Hardware in the Loop close-range rendezvous test facility EPOS [5], German Aerospace Center (DLR) contributes to explore possible solutions. It allows experimental development and research on more challenging issues of OOS in a space-like environment. The main use-cases of EPOS focus on close-range rendezvous experiments and sensor verification for external customers. An overview of recent product development and missions to which EPOS contributed can be found in [6].

A Gnc-system uses the EPOS facility for testing and developing experimental sensors, on-board hardware and novel processing algorithms. The capabilities of the Gnc-system are continuously extended, and offer End-to-End (E2E) simulation capability [7], that are aimed toward covering different aspects of ground-systems and mission-operation of rendezvous scenarios.

This article presents the current status of the Gnc-system. It includes a short overview of the features, the concept and the system architecture.

2 CLOSE RANGE RENDEZVOUS WITH EPOS

This chapter introduces EPOS, outlines it's technical design, typical use cases and limitations.

EPOS is a robotic test bed with two robots as outlined in figure 1. Each robot has six degrees of freedom. One robot is mounted on a linear slide of 25 m length. Thus, the final 25 meters of the close range rendezvous phase can be simulated with 1:1 models[1]. The two robots simulate the motion of the chaser and the target spacecraft. Figure 1 shows a typical setup for rendezvous:

[1] and even larger distances with scaled models

The robot representing the chaser carries an adapter board with several sensors mounted - as described later in chapter 3. The other robot carries a mock up of the target satellite. A wheel-mounted 12kW gas-discharge lamp serves as sun-simulator and achieves lighting conditions that are realistic in terms of solar spectrum and intensity. [5] provides a more detailed desription of EPOS.

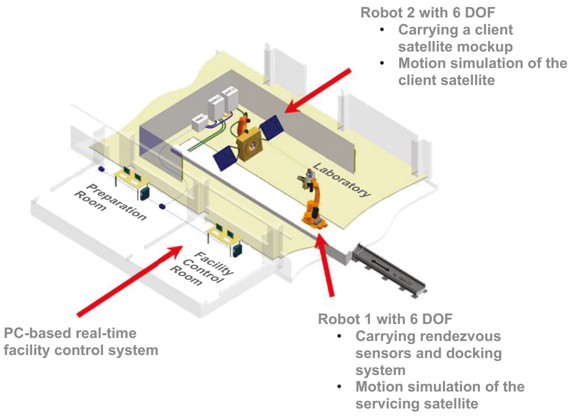

Figure 1: Layout of the EPOS Hardware in the Loop (HIL) testing facility

Robot dynamics are commanded in real-time (RT) by a cascaded control loop as shown in figure 2, which enables close-loop simulation of satellites in orbital relative motion. This is exploited to generate realistic data for optical sensors that can be used for experiments with uncooperative target satellites. EPOS can also be used with predefined trajectories for pure sensor testing. See [6] for examples of test-campaigns executed for external customers.
On the highest abstraction layer of the EPOS controller, a software satellite dynamics simulator is integrated as a Simulink [8] model on top of the intelligent External EPOS Interface (ExtEPOS). ExtEPOS optimizes robot joint attitudes to avoid laboratory limitations like singularities and walls as far as possible. It also provides additional safety margins and can automatically uncouple robot movements from the simulation in case of an unstable control loop. See [9] for more information.

The typical use case is a close-range approach towards an uncooperative target satellite. Feasible chaser to target center-of-mass distances range from 3m to about 20m depending on model dimensions. With EPOS, the simulation space, defined as the region where the robots don't block the sensor's view on the mock up, covers almost a half-sphere.

Starting from a target distance of around 15m, maneuvers of the chaser, like straight-line-approaches or fly-around for a lower earth orbit (LEO) orbit, can be simulated. A virtual reference sensor is available, providing the exact relative pose between the service satellite and the target. Allowing to bridge gaps in sensor coverage. The reference sensor also enables pure "image-acquisition flights". Ultimately, the Gnc-system will allow complete approach maneuvers with physical sensors alone.

3 GNC SYSTEM

This chapter provides a rough overview over the physical part of the Gnc-system developed at the EPOS facility. The software system of the Gnc-system is treated in chapter 5.

The Gnc-system has all necessary components required for simulating a real space Rendezvous and Docking (RvD) mission: Physical sensors mounted on the chaser robot as described in section 2, an onboard computer and the necessary system-internal and to-ground-interfaces. It also applies a Telemetry and Telecommand (TMTC) receiver naturally belonging to the ground segment for surveillance of software-internal functions, so this is also considered a part of the Gnc-system.

The Guidance, Navigation and Control (GNC)-software is executed on a standard x86-64 linux workstation in the EPOS laboratory environment. It additionally supports execution on an arm-based on-board computer system, developed at the DLR (see section 4). For more information on the software see section 5. The computer is connected via Gigabit-Ethernet to the sensors mounted on a carbon-fiber-board on the chaser-robot.

Part of the Gnc-system is also an Satellite Dynamic Simulator, providing navigational data to

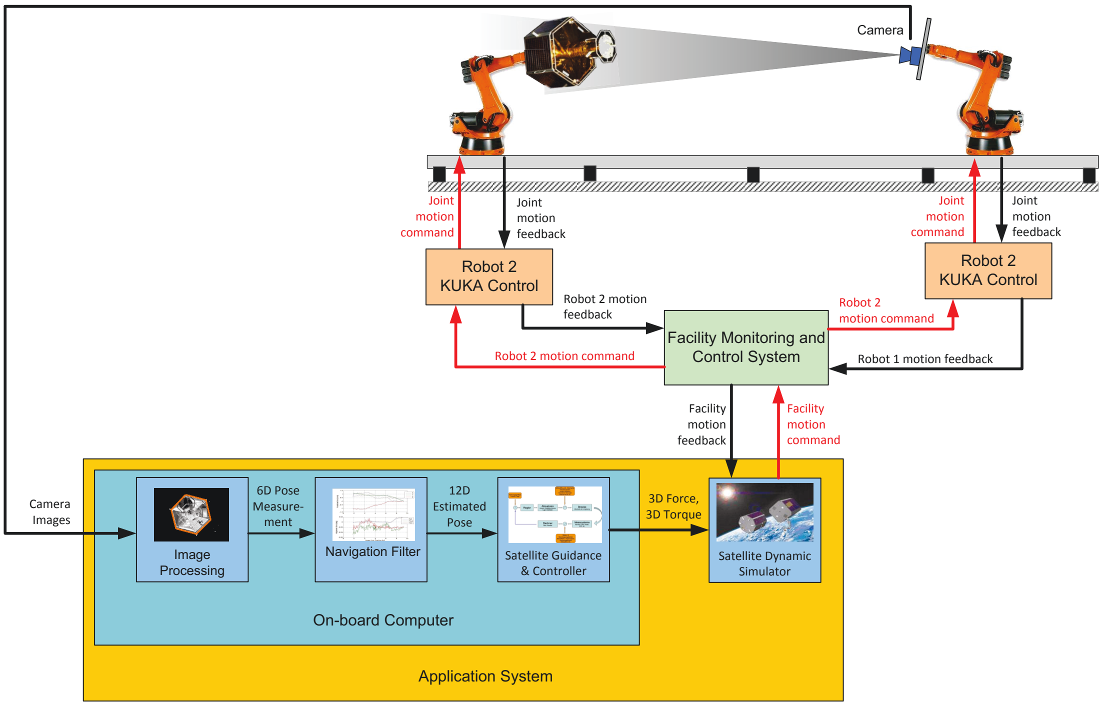

Figure 2: Schema of a closed-loop simulation with EPOS

the GNC-software system, and to a virtual sensor that is used in experiments where an accurate reference-position of the target satellite is needed. The satellite-simulator in turn receives force-commands from the GNC-software system, destined for it's simulated actuators. It can be used for a pure dynamics simulation and also optionally to forward TMTC to the ground-segment that is part of the E2E-Simulation.

Since routing of TMTC to the ground-station is quite complex and not necessary for all experiments, a separate TMTC monitoring and commanding-tool can be used on the laboratory workstation or any other computer.

A range of 2D and 3D Rendezvous-Sensors is mounted on the sensor-board, since one research purpose of EPOS is to find suitable combinations of sensors for a possible OOS mission:

- 2 640x480 pixel 8-bit monochromatic Prosilica GC655 Charge-Coupled Device (CCD)-cameras
- 2048x2048 pixel 12-bit monochromatic Prosilica GT2050 Complementary Metal-Oxide-Semiconductor (CMOS)-camera with 8° field of view (f.o.v.)
- 2048x2048 pixel 12-bit monochromatic Prosilica GT2050 CMOS-camera with 35° f.o.v.
- Bluetechnix Photonic Mixer Device (PMD)-camera providing seperate depth and amplitude images
- Livox Light Detection and Ranging (Lidar)-sensor

4 ON BOARD COMPUTING PLATFORM

An x86-64 based powerful workstation is useful for quick testing of new SW and easy integration of new hardware and drivers. Nevertheless the ultimate goal remains developing a space-capable Gnc-system on space-capable hardware. The Scalable On-board Computing for Space Avionics (ScOSA) system [10] is a compromise that eases development -compared to space-qualified computers- but by it's special design still provides many features making it compatible towards space-hardware. This section introduces the ScOSA on-board computing platform, which opens up a feasible way towards a space mission and sheds some light into how the Gnc-system is adapted to work on this platform.

The ScOSA system is a next generation computing on-board platform developed as an DLR wide endeavour. Its main focus relies on reliability, performance, usability and cost-efficiency. It is a distributed computing system, consisting of several independent computing nodes, connected by a space-wire network [10, 11]. There are different kinds of computing nodes available, each serving a different task:

- High Performance Nodes for executing computing tasks
- Reliable nodes meant for supervision of the system and controlling Fault detection, isolation, and recovery (FDIR)
- Interface Nodes meant for attaching to external system components as the satellite bus and sensors

A scheme of the ScOSA distributed platform is shown in figure 3 as well as an image of the development board we are currently using for prototyping in figure 4.

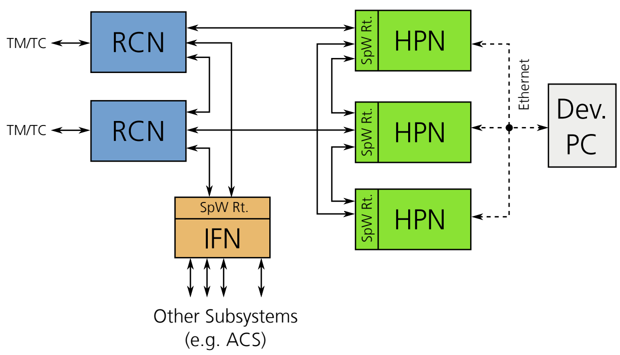

Figure 3: Block diagram of the ScOSA system (source [10])

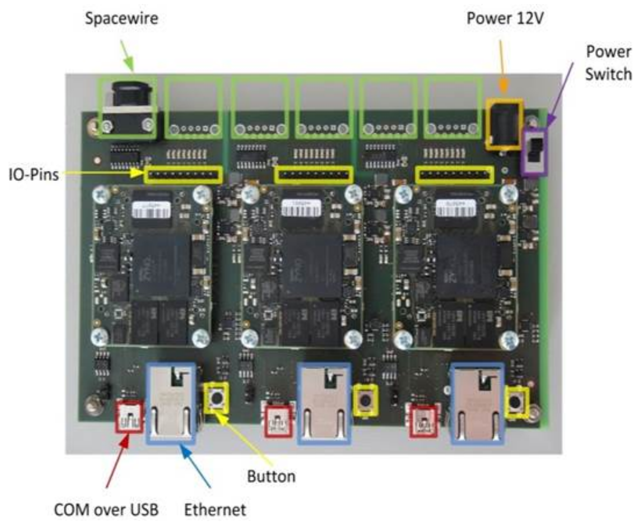

Figure 4: ScOSA development board containing three High Performance Node (HPN)

The HPNs are currently based on the programmable System on a chip (SOC) Xilinx-Zynq Z7020 device, which has a dual core ARMCortex-A9@1 GHz combined with a Field Programmable Gate Array (FPGA) [12]. For more detailed specification see [10].

Each node is an unique computing device, running either an embedded Linux or an Real-Time Executive for Multiprocessor Systems (RTEMS) Operating System (OS), depending on the use case. The ScOSA computing platform at a whole is composed of the ScOSA system software running on top of the OS. From the Operating System's perspective, the scosa software is a system process, containing management services and the different user applications (e.g. On board data analysis and real-time information system (ODARIS)).

The management services are responsible for initialization and (re-)configuration the complete system, handling (inter-)node communication, providing FDIR layer and task management.

User applications have to be integrated via a provided data-flow programming API. A scheme for illustrating this programming pattern is shown in figure 5. The data-flow programming scheme was chosen, as it naturally reflects the distributed hardware architecture of the system. An application can be divided into sub tasks, that can be executed on different HPNs. In case of a node failure, they can be migrated to other functional nodes. More details about application development for the ScOSA system can be found in [13].

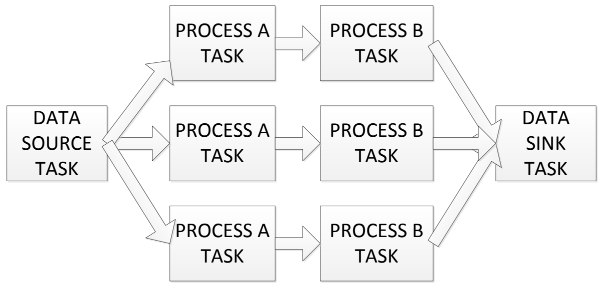

Figure 5: Illustration of data-flow driven application, using parallel blocks of execution

The major hurdles when running the GNC-System on ScOSA are:

- conceptual differences between data-flow oriented design and original centralized-data architecture
- change of platforms: x86 to armv7
- decrease of performance by roughly factor 10 and subsequent extreme latencies

Some measures to address these problems are also introduced in section 5.
This section is written in co-authorship with Kurt Schwenk and is in most parts the same as the correspondent section in [14] which is yet to be published by DLRK.

5 GNC SOFTWARE

This section introduces the software-part of the Gnc-system. It outlines the overall architecture, the important subsystems and highlights the recent development. Special challenges resulting from the multi-platform-approach (see also section 4) and sensor integration as well as their existing and possible future solutions are covered as well.

5.1 Modular Structure

One major problem is the discrepancy of architectures: The central-memory thread-based model (see 6) that is convenient to use for prototyping must be transferred into the data-flow-oriented ScOSA system with physically separated nodes (see figure 7 and subsection 5.3 for more details on the implementation). This challenge is solved by a modularized approach (the modules are explained in subsection 5.2): Each module consists of a callable class that is encapsulated by a so called Executor, which is architecture-dependent. Each module itself is independent of the run time environment and the same in the lab-environment as on the ScOSA on-board computer (OBC). The Executor is responsible for receiving and sending data either via channels or from/to the central data source. This approach decouples the architectural framework from it's content but also demands the modules to be compliant to both x86 and armv7 computing architectures.

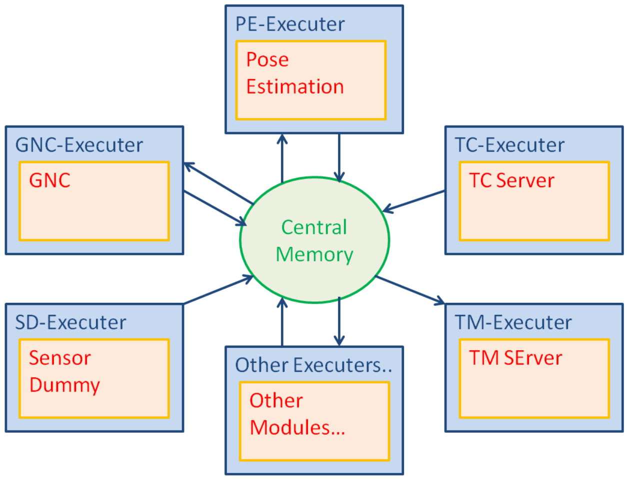

Figure 6: Simplified schema of Gnc-system in central-memory thread-based architecture

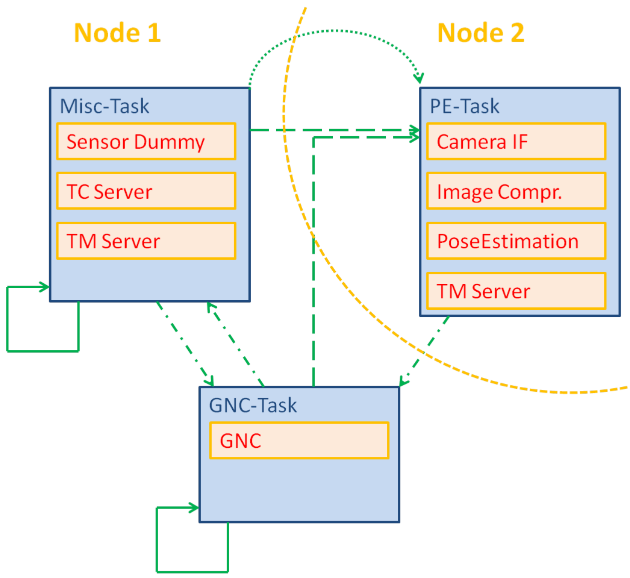

Figure 7: Simplified schema of Gnc-system in ScOSA data-flow-oriented architecture with physically separated nodes

5.2 Modules

Figures 6 and 7 show the modules constituting the GNC SW.
The module **"GNC"** covers the classical aspects of GNC: It contains a guidance sub component responsible for calculating trajectories in an coordinate frame relative to the target satellite (local-vertical local-horizontal (LVLH) frame) as commanded by the operator via telecommand (TC). The navigation is realized as Extended Kalman Filter for smoothing sensor-input, sensor fusion and updating the estimated position in an earth-centered inertial (ECI) frame. The ScOSA OBC and the separated-node-architecture cause some special demands on the navigation filter due to the increased latency by sensor-processing and routing through the nodes, these are explained separately in subsection 5.3. A Linear-quadratic-optimized orbit controller generates forces in the satellite-body-frame to be forwarded to the actuators. It contains a prefilter for compensating pseudo-forces from the LVLH non-inertial frame. This feature plays a major role in stabilizing the closed-loop system with the sensor latencies around 4 seconds, as for the ScOSA OBC.

There exist different **"Pose Estimation"** (PE) modules, one for each sensor (see section 3 for a list of sensors). Those compute poses of the target w.r.t. their own sensor frame. Currently the mono-camera images are processed with a line-search tracking algorithm that has been proven to perform well for a frontal approach on the target which is rotating around one fixed axis (depicted in figure 9 during a typical approach). The PMD sensor works with a feature-based tracker described in [15], an initialization routine is also implemented. Currently, a Lidar based tracker is under development, as well as a robust feature extraction algorithm [16] for mono-cameras. These latter two algorithms shall overcome the very high processing times of the line-search and pmd trackers which could cause problems in very dynamic situations, like a fast rotating target. The robust feature extraction has the potential to be much easier adjustable to new targets, be robust with respect to a tumbling target seen from different directions and is partially suited for parallel data processing. All trackers can be run "loosely" coupled - initializing themselves with the pose of the previous iteration or "tightly" coupled with initialization by the navigation filter. The latter has advantages in scenarios with high latency's and fast movements, the former provides better recovery-behavior under certain circumstances.

"Camera Interface", responsible fro image acquisition is loosely coupled with Pose Estimation, enabling also offline image-capturing.

A **"Sensor Dummy"** (SD) module represents a virtual sensor that feeds directly from the satellite simulator. It provides an estimated pose the same way as other "Pose Estimation" modules. It can provide Gaussian noise and biases for testing purposes. Also in demonstration and testing it provides a "bridge" to reliably initialize pure trackers or recover the simulation in case of failure. For image collection campaigns it provides an easy way to navigate around a target with high precision also at places where pose-estimation fails. It is further useful as ground truth provider for performance experiments.

Each module provides a TMTC interface via Packet Utilization Standard [17] (PUS) packets. Outgoing housekeeping telemetry (TM) must be collected at a central server, which is placed in the **"Telemetry Server"** module. From there it is forwarded either as part of the E2E Simulation to the satellite simulator and routed via a partly simulated communication track to the ground, or simply sent to a local receiver console "clic". In the other direction, TC is routed via the **"Telecommand Server"** module to each receiver.

5.3 Challenges of data-flow design

The philosophy of the thread based architecture of the GNC-software system with a centralized memory and a RT core is that non real time processes like TMTC or sensor-data-processes can communicate with the real-time-core (navigation and control) anytime without influencing the system stability. This requires some special adjustments in the RT-core that are described in [9]. However this predicate does not apply to the data-flow-driven architecture required for

ScOSA. When designing the flow-graph, all connections of RT-modules to non-RT modules have to be modeled explicitly; The following questions arise:

- how shall a task/module be triggered?
- how large shall the input/output buffers be?
- how can unnecessary latency's be avoided or at least minimized?
- what input-data shall be processed? all elements of a buffer? only last or first?
- are there unwanted side-effects to be avoided?

The flow graph in figure 7 shows the design answering these questions. One example of a special adaption is the double connection of the GNC-Task to the CamPe-Task: Triggering a new iteration of pose-estimation makes sense only as soon as a pose update from the filter has arrived as well as a camera image. Both these data come from separate sources and therefore on different channels, but can only trigger once both signals are on. Though the according TC channel shares source as well as destination with the other stream, it must be separated as it shall not trigger. Another undesired effect of waiting for two signals is the increases in latency. This is in principle never avoidable in a data-flow architecture, but some measures are applied to decrease latency's to a minimum:

- apply trigger rates with a common denominator, so the slower signal triggers instantly
- apply trigger rates which are considerably faster than processing time
- have the sensors, especially the cameras with large images constantly stream data into a buffer, so there is always data available when required

These measures lead to an acceptable signal-transmission-related latency's that are small compared to delay by signal processing.
Processing data on multiple nodes requires a precise synchronization of node clocks and sensor-clocks, the image timestamps are stricly required to be consistent with the clock of the navigation node. For this purpose, the CCD and CMOS cameras as well as OBC-nodes are equipped with a Precision-Time-Protocol (version 2) (PTP2)-daemon which picks the best available clock automatically (see [18]). With these synchronized clocks the cameras add very exact time stamps to images at the point of acquisition start. The remaining discrepancy through exposure time is in the order of magnitude of max. 50 microseconds which has shown to be acceptable. It translates to a rotational miscalculation of less than tenth a degree at 1 deg/s rotation.

Latencies from processing time and data-flow play subtly different roles for the navigation filter: Figure 8 illustrates how out-of-sync measurements have to be explicitly accounted for to not severely affect filter-performance. It is not sufficient to consider the sensor processing time with a time-stamp of the image-acquisition, but also the time of arrival at the filter is important. The following is a simplified explanation of how delayed measurements are considered in the filter: A ring buffer stores filter-states (estimations of the target state) of the past for up to 20 seconds. While the filter as part of the GNC-RT-process is executed in a 100 ms frame, sensor measurements arrive asynchronously with a non-deterministic delay (typically for ScOSA between two and five seconds). So called updates of the filter state happen only once a new measurement arrives, if no measurement is available, the filter state gets propagated only from it's previous estimation. The naturally imprecise model used for propagation leads to a saw tooth shaped error as displayed schematically in figure 8, the filter residual. The update-step leads to a feedback of the filter-state with measurements and therefore reduces the residual. The filter-state at time now can only be propagated from filter-states of the past, which are most accurate (assuming sufficiently accurate measurements) immediately after an update step. This is why backwards-propagation is important. If a measurement arrives matching a time right before an update step, where the filter residual is maximal, the high residual will be partially carried on to the current estimate. The resulting decrease in accuracy would often be dramatic enough to destabilize the control loop. Using backwards-propagation from a filter-state not yet known at the time of the measurement leads to drastically increased robustness against delayed and out-of-sync measurements.

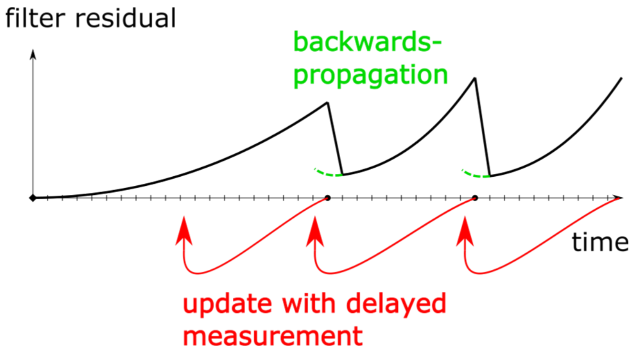

Figure 8: The delay of out-of-sync measurements is explicitly considered. If the measurement's time-stamp is before the last filter update, the filter-state used for the update is backwards-propagated.

5.4 Ground segment and TMTC concept

One research topic is how an autonomous rendezvous can be integrated into a classical mission infrastructure. This challenge is adressed with the E2E Simulation: It covers every aspect of a space mission from ground-segment to space-segment, see [7] for a detailed description. The GNC system is located at the space-end of E2E, but of course interacts closely with it's ground counterpart: The Rendezvous Console (RECO). The RECO basically consists of a set of virtual machines that are typically accessed from a terminal server in a German Space Operations Center (GSOC) multi-mission control room, along with other subsystem consoles. There are three machines acting as user interface:

- a Satmon (see [19]) console used for online-supervision of housekeeping-data
- a GECCOS (see [20]) console that is mainly used for commanding
- a Camera-console used to display online-data from pose-estimation and navigation in a user-friendly way.

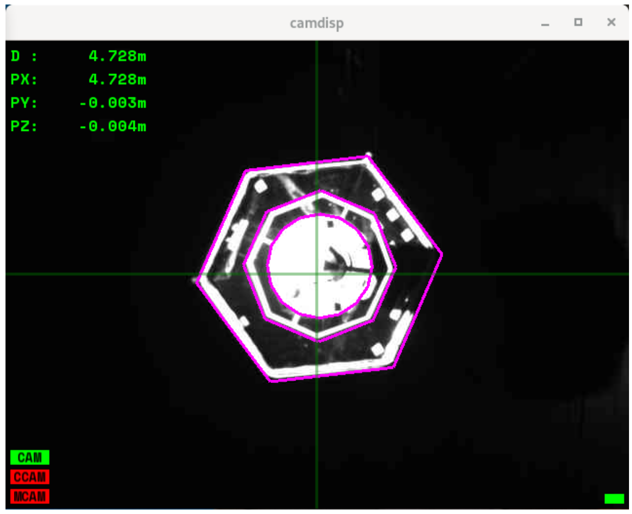

Figure 9: The visual TM-monitor "camdisp" during a frontal close-range approach

While the flexible tools Satmon and Scos merely have to be tailored for use in RECO (as well as for other consoles) via configuration. (E.g. a Mission Information Base (MIB) and operation procedures have to be created and maintained, etc.), The camera-console is closely coupled to the GNC system: It receives telemetry from sensor-interfaces, pose-estimation (multiple sensors possible) and navigation-filter in an asynchronous way and merges this information on a graphical display shown in figure 9. This is extremely helpful and necessary for an operator: The visualized target pose in magenta helps to detect critical situations i.e. loss of tracking "in one glimpse". A numerical display of the pose relative to servicer body frame in the upper left corner enables a quick assessment of the approach status. There is further a green connection indicator signal in the lower right and a sensor switch in the lower left, in case more than one sensor is active at a time.

It is planned to further enhance user-friendlyness by introducing a "Navigation display" that renders target and chaser in 3D and visualizes intuitively the current pose, current commanded pose and desired end pose.

6 CONCLUSION AND OUTLOOK

The Gnc-system developed by GSOC's OOS-group has been developed in different directions: Sensor-fusion capability of the navigation-filter,

additional sensors and a new more robust and scalable mono-camera image-processing algorithm aim to make the automated rendezvous safer and more reliable. On the operational side, a new TMTC monitoring and -commanding tool "camdisp" has been developed and is still being improved. It interacts with a guidance-system that is conveniently usable for manoeuvers in a relative frame that quickly overstrain the human mind. Those help to handle the increasing complexity resulting from more sensors and simultaneously make operations more intuitive.
In parallel, the Gnc-system has been adopted and modularized to be employed in the ScOSA-framework on a different hardware and in a data-flow oriented architecture. This parallel development creates new challenges for software development, but it is intended to make the classical x86-based Gnc-system and the ScOSA version grow closer together. Simplified module interfaces, low-intensity algorithms and the application of easily portable software-libraries will help in this context.
Our vision for the future is a flexible and scalable Gnc-system that supports experimental development on EPOS as well as space-missions on OBCs of the future without much friction loss.

References

[1] Hill, J., "Oneweb explains fcc application for 48,000 constellation satellites - https://www.satellitetoday.com/broadband/2020/05/27/oneweb-explains-fcc-application-for-48000-constellation-satellites/," (2020).

[2] Henry, C., "Spacex submits paperwork for 30,000 more starlink satellites - https://spacenews.com/spacex-submits-paperwork-for-30000-more-starlink-satellites/," (2019).

[3] Clark, S., "Two commercial satellites link up in space for first time - https://spaceflightnow.com/2020/02/26/two-commercial-satellites-link-up-in-space-for-first-time," (2020).

[4] ESA, "Clearspace-1 - https://www.esa.int/ESA_Multimedia/Images/2019/12/ClearSpace-1," (2020).

[5] Benninghoff, H., Rems, F., Risse, E.-A., and Mietner, C., "European Proximity Operations Simulator 2.0 (EPOS) - a robotic-based rendezvous and docking simulator," *Journal of Large-Scale Research Facilities* (2017).

[6] Rems, F., Benninghoff, H., Mietner, C., Risse, E., and Burri, M., "10-year anniversary of the european proximity operations simulator 2.0 - looking back at test campaig ns, rendezvous research and facility improvements," *Deutscher Luft- und Raumfahrtkongress (DLRK) 2020* (2020).

[7] Benninghoff, H., Rems, F., Brunner, B., Stelzer, M., Schmidt, P., Krenn, R., Stangl, C., and Gnat, M., "End-to-end simulation and verification of gnc and robotic systems considering both space and ground segment," in [*10th International ESA Conference on Guidance, Navigation & Control Systems*], (29 May - 2 June 2017).

[8] The Math Works Inc., "Simulink product page - https://www.mathworks.com/products/simulink," (2020).

[9] Rems, F., "Robotic verification of spacecraft rendezvous in-loop with real-time satellite dynamics simulation," in [*German Aerospace Conference*], (September 5–7 2017).

[10] Treudler, C. J., Benninghoff, H., Borchers, K., Brunner, B., Cremer, J., Dumke, M., Gärtner, T., Höflinger, K. J., Lüdtke, D., Peng, T., Risse, E.-A., Schwenk, K., Stelzer, M., Ulmer, M., Vellas, S., and Westerdorff, K., "Scosa - scalable on-board computing for space avionics," in [*International Astronautical Congress IAC 2018*], *Proceedings of the International Astronautical Congress* (2018).

[11] Lüdtke, D., Westerdorff, K., Stohlmann, K., Börner, A., Maibaum, O., Peng, T., Weps, B., Fey, G., and Gerndt, A., "Obc-ng: Towards a reconfigurable on-board computing

architecture for spacecraft," in [*2014 IEEE Aerospace Conference*], (2014).

[12] Xilinx Inc., "Zynq-7000 soc product description - https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html," (2020).

[13] Schwenk, K., Ulmer, M., and Peng, T., "Scosa: application development for a high-performance space qualified onboard computing platform," in [*Proc. SPIE 10792, High-Performance Computing in Geoscience and Remote Sensing VIII*], *Proceedings of SPIE* **VIII**, SPIE Remote Sensing (2018).

[14] K. Schwenk, D. H., "On-board data analysis and real-time information system," in [*Deutscher Luft- und Raumfahrtkongress (DLRK) 2020*], Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V., Bonn (2020).

[15] Klionovska, K., Ventura, J., Benninghoff, H., and Huber, F., "Close range tracking of an uncooperative target in a sequence of photonic mixer device (PMD) images," *Robotics* **7**, 5 (jan 2018).

[16] Capuano, V., Alimo, S. R., Ho, A. Q., and Chung, S.-J., "Robust features extraction for on-board monocular-based spacecraft pose acquisition," in [*AIAA Scitech 2019 Forum*], American Institute of Aeronautics and Astronautics (jan 2019).

[17] for Space Standardization, E. C., "Ecss-e-st-70-41c - telemetry and telecommand packet utilization," (2016).

[18] "Ieee 1588-2019 - ieee standard for a precision clock synchronization protocol for networked measurement and control systems."

[19] Noguero, J., Reid, S., Gil, J. C., Honold, A. P., and Ceballos, A., [*SATMON - A Generic User Interface for Satellite Control*].

[20] Stangl, C., Lotko, B., Geyer, M., Oswald, M., and Braun, A., "Geccos – the new monitoring and control system at dlr-gsoc for space operations, based on scos-2000," (2014).