Validation and Verification of Safety-Critical Aspects of Autonomy in Orbital Robotics

Roberto Lampariello
Institute of Robotics and Mechatronics
German Aerospace Center (DLR)
Wessling, Germany
roberto.lampariello@dlr.de

Caroline Specht
Institute of Robotics and Mechatronics
German Aerospace Center (DLR)
Wessling, Germany
caroline.specht@dlr.de

Margherita Piccinin
Institute of Robotics and Mechatronics
German Aerospace Center (DLR)
Wessling, Germany
margherita.piccinin@dlr.de

Hrishik Mishra
Institute of Robotics and Mechatronics
German Aerospace Center (DLR)
Wessling, Germany
hrishik.mishra@dlr.de

Marco De Stefano
Institute of Robotics and Mechatronics
German Aerospace Center (DLR)
Wessling, Germany
marco.destefano@dlr.de

Martin Stelzer
Institute of Robotics and Mechatronics
German Aerospace Center (DLR)
Wessling, Germany
martin.stelzer@dlr.de

***Abstract*—The validation and verification (V&V) of safety-critical functionalities of space robots engaged in autonomous task-execution is facing new challenges. The recognized advantage of optimal control and machine learning for providing planning and perception capabilities onboard a robotic spacecraft calls for the development of new V&V techniques to ensure robust, explainable, and resilient autonomous behavior. This paper presents V&V techniques and developments at DLR of safety-critical aspects specifically for the autonomous robotic capture of a target satellite. The paper first addresses optimal control for robustly planning the robot arm's interception with a predefined grasping point on the tumbling target satellite, as well as for providing the means of performing mission planning in view of operational and motion constraints. Deep-learning based perception algorithms are then addressed, underlining the advantage of these with respect to classical approaches and presenting new methods for their V&V. Finally, a novel model-based design approach is presented for rapid prototyping and simulation of orbital robotic spacecraft engaged in the close proximity operations of interest.**

***Index Terms*—Orbital Robotics, Validation & Verification, Deep Learning based Perception, Optimal Control, Model-based Design, Processor-in-Loop.**

I. INTRODUCTION

Developments in the orbital space sector over the last decades have led to a sharp increase in robotics-related activities. The current interest worldwide in ensuring the sustainability of the orbital environment and the security of space national assets, in the further commercialization of the space sector with satellite serving capabilities, as well as, more long-term, in on-orbit assembly, manufacturing and recycling, leverages robotics as a key enabling technology. Pioneering steps such as in Japan's ETS-VII and USA's Orbital Express demonstration missions [1, Chapter 55], will soon be followed by new missions, possibly USA-DARPA's RSGS [2], the European Commision's EROSS SC [3] and the European Space Agency's RISE [4] missions, to name some.

The robotic tasks involved in the above activities are best accomplished in the supervised autonomous operational mode, in which the operator on ground provides high-level commands and may intervene in case of contingencies. The on-board GNC of the robotic spacecraft performs safety-critical perception and control functionalities autonomously, to account for the remoteness, but also possibly for the dynamic character of the operational environment. One of the possibly most challenging tasks is the capture of a non-cooperative target object [5] [6], which is free-tumbling and which does not provide any information on its state, nor any support for its capture (dedicated mechanical interfaces or visual features). A comprehensive analysis of the autonomy challenges for next-generation space missions can be found in [7].

Optimal control has been recognized as a key technology in many space-related applications [8], of which spacecraft rendezvous and proximity operations [9] [10], attitude guidance and control [8], as well as space robot trajectory planning [11] [12] are some of the relevant examples. The need for validation and verification (V&V) of these methods was clearly outlined by the European Space Agency (ESA) in [13]. Most of the current applications are based on convexification of the optimal control problem. DLR has recently developed a method which serves as an alternative to embedded optimization to treat highly nonlinear robot trajectory planning tasks, recognized to be non-convexifiable in [11]. The method uses a sensitivity-based update of pre-computed feasible solutions, which is deterministic and not polynomial-time NP-hard in nature. The method also provides provable robustness for a predefined uncertainty in the given task. We expand on the discussions of requirement satisfaction and robustness of optimal control-based planning methods in [13] [14] to address non-convex methods in the presence of uncertainty and present a tool for verifying that mission requirements are satisfied on a given parameterization of task space.

Modern methods for robot perception and control are based nowadays on deep learning and optimal control. These methods are well-established for applications on ground but still need V&V tools. This is especially the case for their implementation in space, to possibly ensure robust, explainable, and resilient autonomous behavior for the safety-critical functionalities of interest. Deep learning (DL) based perception has been extensively developed for pose estimation, with visual cameras or LiDAR sensors, to outperform the more classical approaches [15], [16]. In this paper we summarize key insights from recent guidelines for V&V of space software based on DL components and we present the specific steps we performed in the V&V process of our safety-critical LiDAR-based pose estimation method.

DLR is also actively involved in the development of orbital robotic systems, contributing these in the EROSS SC and RISE missions. In these activities, a robotic arm is mounted on a satellite which can be controlled (combined control) or left free-to float (free-floating control). It soon became evident that adequate validation and verification tools for orbital robotic systems are missing. The model-based design approach, which strongly favors rapid prototyping in the first phases of a mission-oriented project, are currently well-developed for spacecraft but not for robotic spacecraft GNC systems. In this paper, a model-based design approach is outlined which allows the classical development steps of Model-in-the-loop (MIL), Software-in-the-loop (SIL), Processor-in-the-loop (PIL) for an orbital space robot. This allows the validation of control algorithms such as the combined or free-floating control, [9], [17].

The remainder of this paper is structured as follows. Section II provides a description of the software architecture for a free-flying robot in the capture of a target satellite. In Section III motion planning solution are discussed, the perception pipeline is presented in Section IV, and Section V presents the V&V verification process. Section VI concludes the paper.

II. Overview of the Operational Scenario and of the Related Software Components

On-orbit robotic rendezvous and proximity maneuvers leading to the capture of a target satellite, are completed in stages. This paper focuses on the phase consisting of the robotic approach and capture of the target satellite. This phase, depicted in Fig. 1, occurs after the completion of the close-range rendezvous, where the space robot has reached a predefined mating point relative to the target, from which the robot can perform the capture. The robot arm is maneuvered to bring its end-effector to a predefined grasping point (GP) on the target satellite (approach or interception task). The target satellite is non-cooperative and tumbling with an angular velocity $\boldsymbol{\omega}_t$. The target is subsequently grasped and stabilized by the arm (capture task).

In order to execute the approach and capture of the target satellite, several software modules are required. A Controller module determines the necessary control forces and torques with which the space robot should be actuated, based on

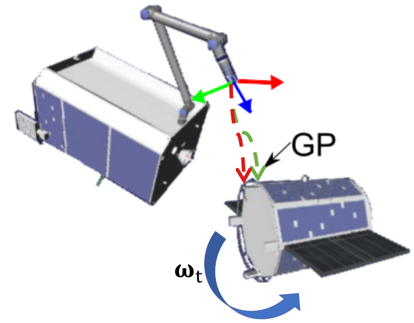

Fig. 1. A space robot approaching a grasping point (GP) located on a target satellite that is tumbling with uncertain angular velocity $\boldsymbol{\omega}_t$. The trajectory of the end-effector was planned on the red dashed line. As an online reaction to the deviation from the expected target satellite motion, the green dashed trajectory must be followed.

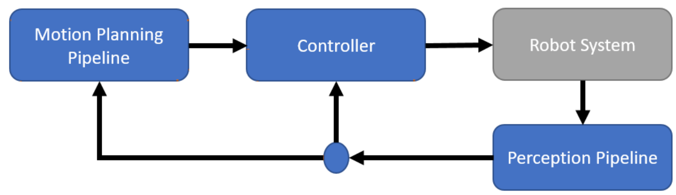

Fig. 2. A high-level block diagram of the software modules required for planning and executing the approach of the robot arm to the target satellite's grasping point.

sensory feedback and a reference trajectory. The Motion Planning Pipeline provides reference trajectories for the Controller module to track, so that the controlled motion of the robot remains feasible and the grasping point on the target satellite is reached, despite of operational and motion constraints. The Perception Pipeline provides sensor-based feedback to the Motion Planning Pipeline so that the motion parameters of the target satellite can be estimated and trajectories can be generated based on this information. The Controller also receives sensory input from the Perception Pipeline. Fig. 2 illustrates a high-level block diagram of these component modules and their interactions.

III. Optimal Control

A. Optimal Control in orbital robotics

Robots are involved in highly complex activities on-orbit. Extravehicular activities include the GNC approach to a target object [10] [18] [19], the approach of the robot to a grasping point on the target [12], and the capture of the target. Free-flying robots have also recently been involved in intravehicular activities on the ISS [20]–[22].

Various tactics are used in the motion planning and control of complex space robots. Optimal control has been demonstrated to be an effective tool – including convex [11] [23] and non-convex [10] [19] [24] [25] methods. In many situations, it is beneficial to track a pre-planned trajectory, see [9], [26] and the sources within.

As demonstrated in [10], [12], [19], [20], [27], and [28], the optimal control problems governing the motion of the space robot in the GNC approach of the target and the robotic approach to the grasping point can each be formulated as a parametric nonlinear program NLP($\boldsymbol{p}$). The NLP is parametric in $\boldsymbol{p} \in \mathbb{R}^{\mathrm{n_p}}$; on a given time interval $\mathrm{t_0} \leq t \leq \mathrm{t_f}$, discretized into n uniform steps, or via points, such that $\delta\mathrm{t} = (\mathrm{t_f} - \mathrm{t_0})/(\mathrm{n} - 1)$; and is of the general form

$$\begin{aligned} &\min_{\boldsymbol{z} \in \mathbb{R}^{\mathrm{n_z}}} \quad J(\boldsymbol{z}, \boldsymbol{p}) \\ &s.t.\ G_i(\boldsymbol{z}, \boldsymbol{p}) \leq 0, i = 0, ..., \mathrm{n_G} \\ &\qquad H_j(\boldsymbol{z}, \boldsymbol{p}) = 0, j = 0, ..., \mathrm{n_H}, \end{aligned} \tag{1}$$

where $\boldsymbol{z}$ is the solution in terms of a judiciously selected parameterization of the robot state; J is the discretized objective function; $\boldsymbol{G}_i := [g_{i,k}(\boldsymbol{z}, \boldsymbol{p}), k = 0 : n]^\top$ are the discretized inequality constraints; $\boldsymbol{H}$ are the discretized equality constraints; and $\mathrm{n_G}$ and $\mathrm{n_H}$ are respectively the number of inequality and equality constraints. The inequality constraints typically consist of, but are not limited to, boundary, collision, and camera field of view and/or pixel velocity constraints. Unlike the SCP applications in e.g. [11] and [23], the NLP formulation does not require the geometry of the target to be convex or convexified. The online solution of such a NLP($\boldsymbol{p}$) formulation has been demonstrated on the Astrobee testbed onboard the ISS in the MIT/DLR collaboration ROAM/TumbleDock [20].

Despite this success, common criticisms include: on-board embedded optimization is computational resource and time intensive, it is difficult to know that a solution to the optimization problem exits for a given set of parameters before beginning the costly optimization process, new V&V procedures are required, and it can be difficult for a human-in-the-loop operator to approve trajectories when little is known about the solution (e.g. proximity to constraints) [13] [24] [27].

B. Current methods for V&V of Optimal Control methods

V&V of a sub-system is an indispensable iterative process, which provides development and operation teams with confidence that the sub-system will operate as expected in the environment that it was developed for. For software-based pipelines, like that described in Sec. II, the environment refers to both the hardware that is to be employed (e.g. PC vs OBC) and the physical environment within which the hardware is to operate (e.g. ground vs on-orbit).

Traditional software V&V brings to mind the classic waterfall or V-shaped procedure. The waterfall description depicts the V&V process analogously to water flowing down stream – suggesting that the process flows from one step to the next, and that, as each is completed, it is left behind. This can discourage iteration and design improvement.

Alternatively, the V-shaped procedure, shown in Fig. 3, highlights the interconnected nature of the software design with its integration into a sub-system, as well as indicates the iterative nature that software design should take. The V-shaped description of V&V has therefore gained popularity, appearing in both ECSS [29] and NASA [30] literature. It is always possible to take a step (or many steps) back in the design process to rectify an issue determined in the design or

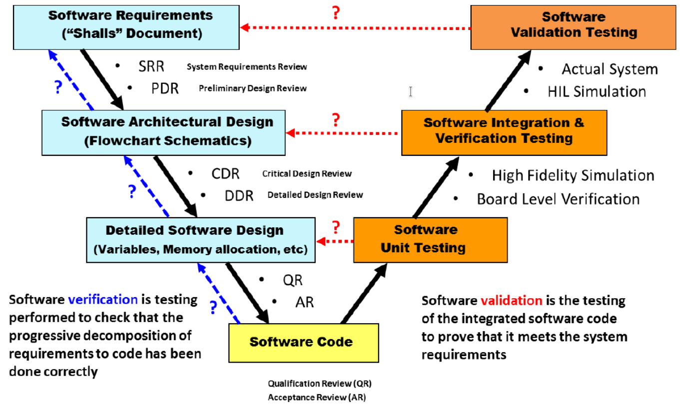

Fig. 3. The V-shaped software development life cycle used by NASA Marshall Spaceflight Center [30], which highlights the relationship of verification (blue arrows) and validation (red arrows) to the development process.

integration. However, any change to the software requirements or design triggers the repetition of all subsequent verification and validation steps to return to the developmental step previously reached in the design cycle. This can, of course, become very costly. It is clearly beneficial to fix feasible sub-system requirements as early in the design as possible.

Software V&V tools can be divided into three categories: simulation and testing, model-based design, and formal methods.

- Simulation and testing form the most classic tool set and remain the most widely used methods. In terms of optimal control for space robotics, this appears as unit testing of individual functions, graphical evaluation of a stochastic sample of solutions of the optimal control problem for requirement satisfaction, and integration testing of the interfacing sub-systems.
- Model-based design for optimal control in space robotics applications involves the construction of a modular optimal control software, commonly using MATLAB and/or Simulink to facilitate prototyping. After verification that task requirements are satisfied, the code can be exported to C++ by a certified Autocoder [31] [32] and compiled for a given target hardware.
- Formal methods attempt to prove the absence of error and detect for example dead logic, integer overflow, and division by zero. These methods come in two major varieties, static analysis and modal-checking. Static analysis does not execute code or find logic errors, but is used to detect lexical, syntactic, and semantic mistakes. Static analysis tools do not appear in literature for V&V of small satellites software. Modal-checking executes a discretized model of the software, performing an exhaustive search of every possible path through the software to verify that no erroneous results are produced. However, such methods cannot operate on floating point variables, and have been shown to be a poor choice for GNC software validation [14] [30] – in particular those dependent on optimal control-based motion planning and control [14] [33] [34].

At this time, no formal standard exists for the development of optimal control-based motion planners or controllers for on-orbit GNC or robotics, the solution of optimal control problems for such tasks offline, or the solution of optimal control problems embedded on-board beyond those for general software development. To move toward such a standard, ESA commissioned a study [13] of modern V&V methodologies and their application to GNC tasks. This call advocated specifically for optimal control-based methods to be used in the motion planning and control of satellites and space robots and for the abstraction of the GNC task into an architecture of distributed tasks from high level guidance and control to relevant algorithms to source code and hardware, indicating a number of relevant tools belonging to the three discussed categories to be investigated.

As part of the study [14], a benchmark optimal control problem was developed and an optimizer was implemented based on an existing algorithm in order to demonstrate a subset of the proposed topics and relevant tools. The presented optimal control problem was convexified and the implemented solver was based on the convex PIPG algorithm. The outcome of the study was a TRL 3/4 optimal control-based motion planner and MPC implementation, as well as a report of the sequence of steps which are parallel to the common approach used in software design for small satellites [30].

However, it should be noted that the study only considered convex, interior point method-based optimization. The study utilized an autocoder to generate C++ code from the model-designed MATLAB implementation of the motion planner and controller. While autocoders are gaining popularity, they do not replace verification of the model-designed code and must itself be verified after each generation. Finally, this study demonstrated that formal V&V methods are not yet mature for optimization algorithms. Formal methods can be used to evaluate discrete functions, but the non-deterministic nature of the search for minima and non-integer computation renders formal methods currently unsuitable for evaluating optimizer implementations.

C. Robust Optimal control methods

Until now, on-orbit robotics literature has largely considered the solution and V&V of the convexified optimal control problem. However, highly nonlinear problems are often not well suited to convexification or linearization [24] [35]. For example: (a) A chaser satellite approaching a non-cooperative tumbling satellite where the relative mating point is located within the convexified geometry of the target [10], [20], [28]; or (b) A space robot involved in the on-orbit capture of a non-cooperative tumbling satellite [12], [27]. V&V of non-convex optimal-control based methods have not yet been codified in literature, but a similar procedure [14], [30] can be followed:

- Set and verify task requirements early in the project life cycle.
- Use a modular code design to facilitate verification measures.
- Conduct thorough code review.
- Where appropriate, verification by comparison of algorithm implementations, e.g. integration, optimization, programming languages.
- Unit and integration testing of algorithms, functions, etc. Both black box and white box evaluations should be conducted, especially in the case where formal methods are not feasible.
- Where appropriate with respect to capabilities of the tools (only integer-based computation, evaluate on function by function basis), use formal static and modal methods.

The remainder of this Section addresses questions not yet discussed in the studies indicated in the preceding section.

A feature of optimal control-based motion planning and control is the incorporation of a model of the robot and its environment. Unfortunately, models are never perfect, and uncertainty in the parametric NLP($\boldsymbol{p}$) must be expected.

In the case of the GNC approach to a target or the approach of the robot to a grasping point on the target, one source of uncertainty is in the estimated target motion parameters. Fig. 1 illustrates the example of the robot approach to the grasping point. It is imperative to note that the trajectory generated by the optimal control-based motion planner (Fig. 1, red dashed line) is only valid for the set of task parameters for which the NLP is solved and highly depends on the quality of *in situ* target motion parameter estimates. If uncertainty exists in these target motion parameter estimates, the generated trajectory must be departed from to successfully and safely complete the rendezvous and robotic approach maneuvers (Fig. 1, green dashed line) [9] [10], necessitating replanning of the motion and increasing the complexity of the control task.

The V&V of modern GNC systems must be considered as early in system development as possible to reduce costly design and requirements iteration. An open question in this process is how uncertainty in related task parameters can be managed. In particular, the verification that a solution to the NLP($\boldsymbol{p}$) exists at $\boldsymbol{p}$ and the determination of how large the deviation from the planned trajectory may be without loss of feasibility are difficult tasks, but must be addressed as part of the first point in the V&V procedure outlined above.

A relevant tool is the parametric sensitivity analysis of the solution space of an NLP($\boldsymbol{p}$) with respect to a given parametric uncertainty, which can assist in understanding where the design requirements and robustness intersect: The Sensitivity Theorem [26] permits the approximation of solutions of a given parametric, constrained optimal control problem through a fast, online sensitivity-based update of a given nominal solution for perturbed task parameter values [36]. This approximation is only valid within an estimable neighborhood of the nominal parameter $\boldsymbol{p}$. Being able to estimate the neighborhood of validity of the sensitivity-based update [37], the task workspace which only includes admissible solution to the NLP($\boldsymbol{p}$) is derived [12]. It is also possible to display the neighborhoods of validity for collated nominal task parameter values, providing a neighborhood map, which indicates the robustness distribution on the task workspace [12], as demonstrated by the neighborhood map in Fig. 4.

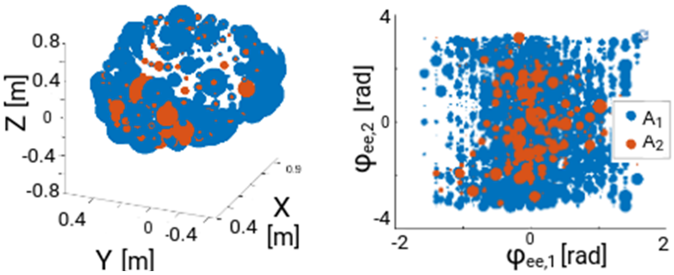

Fig. 4. Task workspace and neighborhood map for the approach of the robot arm to capture the target satellite, which provide an indication of where on the task workspace a robust solution to the NLP can be found. The two largest active constraint sets, $\mathbb{A}_1$ (no active inequality constraints) and $\mathbb{A}_2$ (active collision avoidance between robot link 7 and target satellite), are shown.

The significance of this result is that the regions of task parameter space wherein a feasible motion plan can efficiently, robustly, and provably be found is determinable in the early stages of mission planning, fulfilling the first point in the V&V procedure as early as possible. The safest regions (e.g. lowest number of active constraints, proximity to constraint boundaries) of the task workspace can be selected for autonomous system operations.

Finally, robust optimal control-based motion planning should be married to a robust controller, e.g. Tube-based Model Predictive control. This enforces the provable feasibility properties of the planned motion, while extending the overall mission robustness guarantees to the online disturbance rejection.

IV. Machine Learning based Perception

A. V&V process for DL-based software components

While deep learning (DL) methods have the potential to offer an excellent solution to many perception tasks in space applications, their development and testing process is still not well established in the space domain. The traditional process of software V&V ensures that software development meets the design requirements (verification) and that the right software products are produced during the building process (validation), however it requires adaptations for DL-based software components because of the substantially different development and testing processes being followed for data-driven algorithms. The rapidly evolving field of AI regulation involves various norms aimed at certifying AI products (e.g. ISO 42001, ISO 27001), although specific frameworks for V&V in space missions are lacking. In Europe, standards are developed through the European Cooperation for Space Standardization (ECSS), but there are currently no established regulations for AI software V&V. New guidelines for ML-based software have been recently introduced [38], even though not fully articulating the V&V process. A more comprehensive approach for safety-critical applications is provided by the European Union Aviation Safety Agency (EASA) standards [39], which describe the V&V as a W-shaped, iterative process encompassing all phases from subsystem requirements and design to verification, ensuring a thorough examination of the

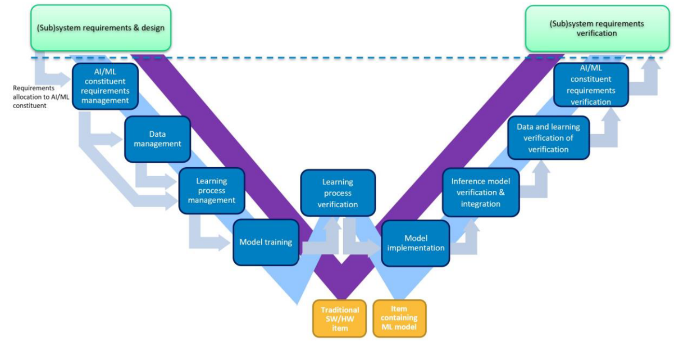

Fig. 5.

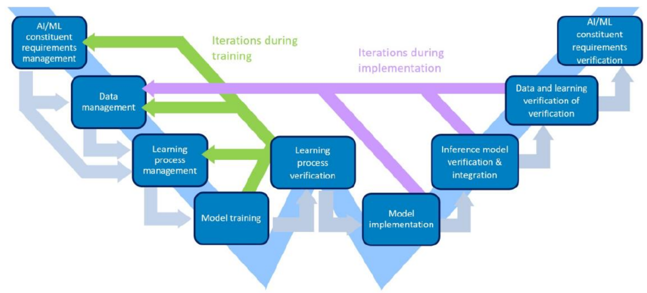

Fig. 6.

Fig. 7. The V&V cycle for DL software can be sketched as a W-shaped process. The figure displays the W-shaped learning assurance process used by EASA [39], which is concurrent with the traditional V-cycle for traditional software items (a) and requires several iterations involving both training and implementation (b).

system's safety and effectiveness (see Fig. 7). The main steps of the W-shaped process include management of data, model development, model testing, and system testing. In the following sub-sections we report the current guidelines for each of these major aspects recommended for space applications [38].

1) Guidelines on data quality: The guidelines on data quality emphasize the importance of both data management and the entire data life-cycle. Key aspects include ensuring high-quality labeling and metadata, maintaining a fully traceable data generation pipeline for reproducibility, and proper data storage. Test data management is also crucial, requiring appropriate dataset splitting, a probabilistic definition of the test dataset's ontology, and an alignment of operational parameters based on specific scenarios. The guidelines highlight distinct quality characteristics for various data types. Real data is valued for its inherent representativeness and necessitates high-quality metadata. In contrast, simulation data's representativeness is challenging to evaluate, while augmented data should focus on consistency with the original dataset. Lastly, laboratory data present their own challenges, as they are time-intensive to produce and often do not replicate the conditions found in the actual environment, which affects their representativeness and fitness for user needs.

2) Guidelines on model development and testing: The guidelines on model development emphasize that there is no favored framework for creating deep learning models, such as PyTorch, TensorFlow, or Scikit-learn. For safety categories mid to high criticality, it is recommended to assess the qualification needs of the inference engine or to consider compensatory measures. When designing a DL model, certain characteristics should be prioritized to promote the idea of “trustworthy AI,” including functionality, reliability, robustness, explainability, and resilience [40], [41]. Various methods can enhance these attributes, but the appropriate choice will depend on the specific application. Additionally, model testing is distinguished from evaluation on a test dataset, as it encompasses a wider range of techniques like operational design domain assessment, coverage testing, out-of-distribution testing, adversarial testing, and SEU testing [38], [41]. Effective testing may necessitate re-training the model with new data or revisiting its design to adapt to newly encountered conditions.

3) Guidelines on system testing: The guidelines on system testing for DL software components almost completely align with the traditional software V&V processes. The initial step involves assessing the safety criticality of the DL component, which helps classify its criticality level and allows for potential mitigation through operational, hardware, or software provisions. Following this, a Failure Mode, Effects, and Criticality Analysis (FMECA) must be performed on the DL methods in accordance with standard practices used in space systems. For DL method, to address the identified risks, mitigation strategies, termed ”safety cages”, can be implemented, incorporating various methods such as rule-based implementations, voting schemes, and performance monitoring [38], [39].

B. Status of V&V process for our perception methods

In [15] we have proposed a LiDAR-based DL global pose estimation method to provide a robust initial pose estimate of a known client satellite. Our lightweight DL method P2PReg (for Point cloud to Pose Regression) processes unordered point sets and regresses pose parameters which are adapted to the symmetries of the client object.

In this section we report the status of the V&V process for our pose estimation software, following the main steps of the guidelines reported above.

1) Data quality: We rely on two different datasets, a synthetic dataset and an experimental one, collected in our OOS-SIM facility (see Fig 8). The data quality characteristics are the following.

- Representativeness. Simulated data have been compared with experimental data to verify the correctness of the sensor model. The real data have been characterized regarding anisotropic noise on a flat surface (quantitative evaluation) and the presence of beam divergence (qualitative evaluation). The satellite mock-up is representative only in part, due to the use of materials which are not specifically for usage in space. The sensor (a dual-LiDAR system composed by two Velodyne VLP-16) comes from the automotive domain.
- Fitness. The synthetic dataset covers the whole pose space constrained to having $2\,\mathrm{m}$ up to $50\,\mathrm{cm}$ range distance from the client and constrained to have the client present in the FoV. No perfect client pointing is assumed. No dataset specifically for tracking has been produced up to now, however tracking trajectories are generated online during SIL and PIL tests. The pose space in the lab is very limited, however it would be mostly impossible to overcome such limitations with other set-ups and maintaining the same simulation capabilities. The limited pose space could be still fit for a mission, depending on the client satellite and on the operational trajectories.
- Metadata. Metadata allow to reproduce the data samples, however for synthetic data beam divergence is generated randomly and augmentations (like noise) are applied successively. Regarding the OOS-SIM, the current ground truth is obtained with kinematics, and is being currently improved by means of a VICON system.

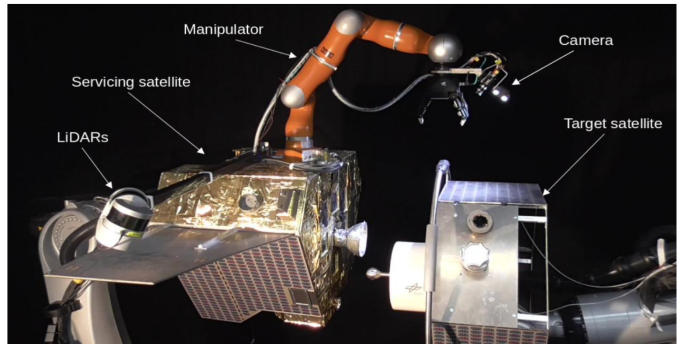

Fig. 8. The DLR OOS-SIM facility consists of a client satellite mock-up and a servicer satellite mock-up, equipped with a manipulator for robotic capture, a dual-LiDAR system and a stereo camera. We tested for sim2real transfer our LiDAR processing DL method P2PReg, which is trained solely on synthetic data, using data from the OOS-SIM [15].

2) Model development and testing: In [15] we show the method’s functionality by benchmarking against other DL and classical methods. The method’s robustness to data artifacts is also tackled, and a good sim2real transfer is achieved training solely on synthetic data. We focused our development on a lightweight method, to overcome possible shortcomings related to porting the method to space HW. We are confident that in the future more suitable space HW for DL methods will be available, as currently some alternative solutions relying on GPUs are being already tested for use in space [42].

3) System testing: Our DL method is at the moment not inserted in a safety-cage, however we evaluated possible fallback classical initialization methods in [15], [43]. After carrying out a preliminary requirements definition, the safety criticality assessment, and a FMECA analysis, we integrated our DL method in the SIL and PIL environments. For future work, we plan closed-loop tests of the integrated system, followed by HIL tests.

V. MODEL-BASED DESIGN

Once the mission requirements are roughly available, the initial phase progresses through a sequence of rapid prototyping using model-based design tools [44], [45]. However, contemporary tools have been aimed at GNC applications, and

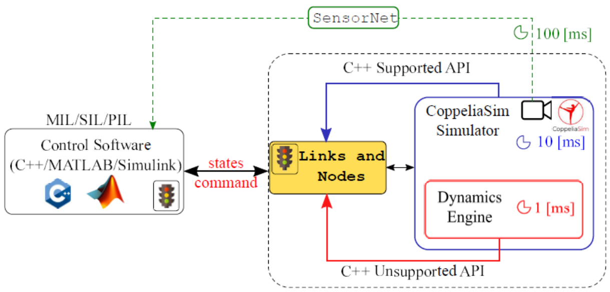

Fig. 9. DLR's co-simulation for control prototyping.

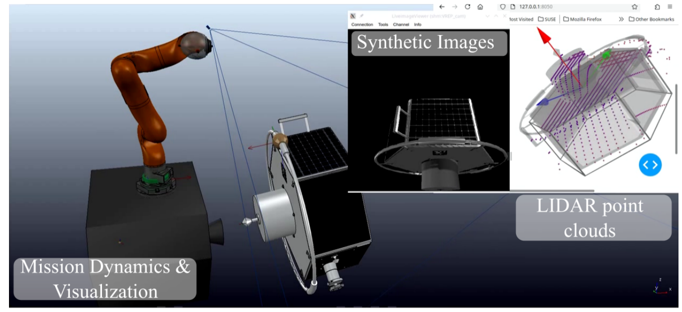

Fig. 10. Synthetic images (camera) and point-cloud (LiDAR) simulator for perception.

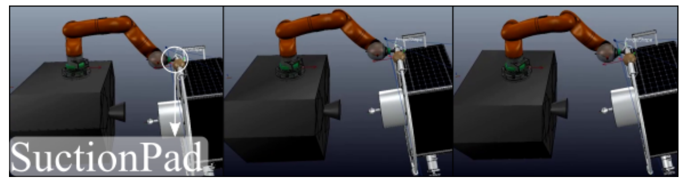

Fig. 11. Capture dynamics in DLR co-simulation.

the topics relevant to robotics prototyping, e.g. multibody dynamics, momentum conservation, capture dynamics, have been missing. At DLR, a co-simulation framework, see Fig. 9, has been developed that enables model/software/processor-in-the-loop, also known as MIL/SIL/PIL. In particular, the framework is composed of CoppeliaSim as the physics provider and MATLAB/Simulink as the control prototyping environment. The inter-process communication between the orbital robot and control software was developed using Links and Nodes, while high-bandwidth data communication, e.g. for synthetic camera images, was developed using SensorNet, both of which are developed at DLR. The main novelty over the state-of-the-art [46] is that, our approach allows the CoppeliaSim simulation to run with a lower time-step of 10 [ms], while the communication for robot joints runs at a faster and synchronized time-step of 1 [ms] with the control software for high-fidelity torque control. This ensures that the non-dynamics modules, e.g., graphics rendering, are computed slower than the dynamics engine, which is not possible with the supported Application Programming Interface (API) of CoppeliaSim. To achieve this, an unsupported API around the supported API was developed, which communicates directly with the underlying dynamics engine that runs with a finer time-step. This co-simulation framework is exploited in external projects in which DLR is part of, e.g., ESA MIRROR [47], EU EROSS+/IOD [3]. In the co-simulation, the motion of orbital robot is simulated using Open Dynamics Engine (ODE) because it ensures momentum conservation [48].

A snapshot of the co-simulation environment is shown in Fig. 10, in which the main scenario is shown. In particular, the spacecraft/satellite mock-ups are the same as on the DLR OOS-SIM facility (see [49]) whereas the robotic manipulator is the KUKA LBR 4+. In the inset figures, the synthetic camera images and the LiDAR point clouds are shown. While the synthetic images are published at 10 [Hz], the point clouds are generated at 1 [Hz]. With these components, the approach phase of the mission scenario can be validated. In Fig. 11, a capture scenario is shown. To demonstrate grasping for prototyping purposes, the suctionPad functionality is used, which creates a closure-constraint, i.e., the satellite is inertially linked to the end-effector of the robotic manipulator. This enables momentum transfer in the simulation, and gives a first prototypical values for post-grasping requirements.

The co-simulation framework developed at DLR provides a powerful prototyping tool to perform model/software/processor in the loop (MIL/SIL/PIL) tests. With the on-going external mission-oriented projects, this tool is useful to achieve DLR contributions.

VI. Conclusion

This paper first presents the status and some of the open problems in the validation and verification (V&V) of optimal control and machine-learning based algorithms, for motion planning and global pose estimation respectively. The outlined developments of a new optimal control methodology may pave the way for provable robustness to given uncertainty in the task parameters, as well as algorithmic simplicity for online applicability, for constrained nonlinear programs relevant to robotic control tasks. The developments in the LIDAR-based global pose estimation provide feasible and efficient V&V methods, based on state-of-the-art standards, of the relative algorithms. Results are being generated with the OOS-SIM experimental facility at the DLR, to validate the sim2real transfer. Finally, the tool developed at the DLR with the model-based design approach, brings robot systems on the same level as GNC systems, in terms of V&V capabilities. These capabilities are already being exploited in on-going mission oriented projects, such as ESA's RISE project.

The V&V task for orbital robots is an exciting field in full development. Space missions with robotic systems will become more and more of a reality in the close future and V&V capabilities will play a fundamental role in their successful realization. DLR aims to demonstrate the autonomous robot technologies discussed above in close-future missions.

References

[1] B. Siciliano and O. Khatib, "Robotics and the handbook," in *Springer Handbook of Robotics*. Springer, 2016, pp. 1–6.

[2] DARPA, "Rsgs: Robotic servicing of geosynchronous satellites," https://www.darpa.mil/research/programs/robotic-servicing-of-geosynchronous-satellites [Last viewed on 2025-06-06].

[3] M. A. Roa, A. Beyer, I. Rodríguez, M. Stelzer, M. de Stefano, J.-P. Lutze, H. Mishra, F. Elhardt, G. Grunwald, V. Dubanchet *et al.*, "Eross: In-orbit demonstration of european robotic orbital support services," in *2024 IEEE Aerospace Conference*. IEEE, 2024, pp. 1–9.

[4] E. S. Team, "Welcome to the space safety industry day 2024," pp. 62–66, 2024, https://indico.esa.int/event/544/attachments/6448/11369/Space Safey Industry Day 2024.pdf [Last viewed on 2025-06-06].

[5] D. Reintsema, J. Thaeter, A. Rathke, W. Naumann, P. Rank, and J. Sommer, "Deos – the german robotics approach to secure and de-orbit malfunctioned satellites from low earth orbits," in *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, Sapporo, Japan, 2010.

[6] "e.deorbit," European Space Agency, Tech. Rep., 2016. [Online]. Available: http://www.esa.int/ Our Activities/Space Engineering Technology/Clean Space/e.Deorbit

[7] J. A. Starek, B. Açıkmeşe, I. A. Nesnas, and M. Pavone, "Spacecraft autonomy challenges for next-generation space missions," in *Advances in Control System Technology for Aerospace Applications*. Springer, 2015, pp. 1–48.

[8] X. Liu, P. Lu, and B. Pan, "Survey of convex optimization for aerospace applications," *Astrodynamics*, vol. 1, pp. 23–40, 2017.

[9] R. Lampariello, H. Mishra, N. Oumer, P. Schmidt, M. De Stefano, and A. Albu-Schäffer, "Tracking control for the grasping of a tumbling satellite with a free-floating robot," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3638–3645, 2018.

[10] C. Specht, A. Bishnoi, and R. Lampariello, "Autonomous spacecraft rendezvous using tube-based model predictive control: Design and application," *Journal of Guidance, Control, and Dynamics*, vol. 46, no. 7, pp. 1243–1261, 2023.

[11] J. Virgili-Llop, C. Zagaris, I. Richard Zappulla, A. Bradstreet, and M. Romano, "A convex-programming-based guidance algorithm to capture a tumbling object on orbit using a spacecraft equipped with a robotic manipulator," *International Journal of Robotics Research*, vol. 38, no. 1, p. 40–72, 2019.

[12] C. Specht, R. Lampariello, and A. Albu-Schäffer, "Sensitivity analysis for a space robot in orbital interception tasks," *IEEE Robotics and Automatation Letters, submitted*, 2025.

[13] ESA, "Invitation to Tender for Verification and Validation of Real-Time Optimised Safety-Critical GNC SW Systems," *ITT AO/1-10895/21/NL/CRS*, 2021.

[14] P. Lourenço, H. Costa, J. Branco, P. Garoche, A. Sadeghzadeh, J. Frey, and et al., "Verification and validation of optimisation-based control systems: methods and outcomes of vv4rtos," in *ESA GNC and ICATT Conf.*, Sopot, Poland, 2023.

[15] M. Piccinin and U. Hillenbrand, "Deep learning-based pose regression for satellites: Handling orientation ambiguities in lidar data," *Journal of image and graphics-in print*, 2025.

[16] M. Ulmer, M. Durner, M. Sundermeyer, M. Stoiber, and R. Triebel, "6d object pose estimation from approximate 3d models for orbital robotics," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 10 749–10 756.

[17] M. De Stefano, H. Mishra, R. Balachandran, R. Lampariello, C. Ott, and C. Secchi, "Multi-rate tracking control for a space robot on a controlled satellite: A passivity-based strategy," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1319–1326, 2019.

[18] E. N. Hartley, P. A. Trodden, A. G. Richards, and J. M. Maciejowski, "Model predictive control system design and implementation for spacecraft rendezvous," vol. 20, no. 7, p. 695–713.

[19] S. Stoneman and R. Lampariello, "A nonlinear optimization method to provide real-time feasible reference trajectories to approach a tumbling target satellite," in *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, Beijing, China, 2016.

[20] K. Albee, C. Specht, H. Mishra, C. Oestreich, B. Brunner, R. Lampariello, and R. Linares, "Autonomous rendezvous with an uncertain, uncooperative tumbling target: The tumbledock flight experiments," in *ASTRA*, 2022.

[21] B. Doerr, K. Albee, M. Ekal, R. Ventura, and R. Linares, "The reswarm microgravity flight experiments: Planning, control, and model estimation for on-orbit close proximity operations," *Journal of Field Robotics*, vol. 41, no. 6, pp. 1645–1679, 2024. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.22308

[22] F. Turchetti, M. Ekal, N. Lii, and M. Roa, "Analysis of intra-vehicular robotic free-flyers and their manipulation capabilities," in *75th International Astronautical Congress (IAC)*, 2024.

[23] R. Bonalli, A. Cauligi, A. Bylard, and M. Pavone, "Gusto: Guaranteed sequential trajectory optimization via sequential convex programming," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6741–6747.

[24] R. Lampariello and G. Hirzinger, "Generating feasible trajectories for autonomous on-orbit grasping of spinning debris in a useful time," in *IEEE/RSJ International Conference on Intelligent Robots and Systems 2013 (IROS 13)*, Tokyo, Japan, November 2013.

[25] R. Lampariello, H. Mishra, N. W. Oumer, and J. Peters, "Robust motion prediction of a free-tumbling satellite with on-ground experimental validation," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 10, pp. 1777–1793, 2021.

[26] M. Gerdts, *Optimal Control of ODEs and DAEs*. de Gruyter, 2023.

[27] C. Specht, R. Lampariello, and A. Albu-Schäffer, "Robustness guarantees for nonlinear trajectory planning applied to orbital robots tasks," in *Inter-Agency GNC V&V Workshop (IAGNC 2025)*, 2025.

[28] K. Albee, C. Oestreich, C. Specht, A. Terán Espinoza, J. Todd, I. Hokaj, R. Lampariello, and R. Linares, "A Robust Observation, Planning, and Control Pipeline for Autonomous Rendezvous with Tumbling Targets," *Frontiers in Robotics and AI*, vol. 8, p. 234, 2021.

[29] ESA Requirements and Standards Division, Space project management – Project planning and implementation, *ECSS-M-ST-10C Rev.* [Online]. Available: https://ecss.nl/standard/ecss-m-st-10c-rev-1-project-planning-and-implementation/

[30] S. A. Jacklin, "Survey of verification and validation techniques for small satellite software development," in *2015 Space Tech Expo Conference*.

[31] T. Wang, R. Jobredeaux, M. Pantel, P.-L. Garoche, E. Feron, and D. Henrion, "Credible autocoding of convex optimization algorithms," vol. 17, no. 4, p. 781–812, December 2016.

[32] R. Cohen, E. Feron, and P.-L. Garoche, "Credible autocoding of the ellipsoid algorithm solving second-order cone programs," in *IEEE Conference on Decision and Control (CDC)*, December 2018, p. 3585–3591.

[33] F. Debrouwere, M. Vukov, R. Quirynen, M. Diehl, and J. Swevers, "Experimental validation of combined nonlinear optimal control and estimation of an overhead crane," in *IFAC World Congress*, 2014, p. 9617–9622.

[34] R. Cohen, E. Feron, and P.-L. Garoche, "Verification and validation of convex optimization algorithms for model predictive control," *Journal of Aerospace Information Systems*, vol. 17, no. 5, p. 257–270.

[35] R. Lampariello, D. Nguyen-Tuong, C. Castellini, G. Hirzinger, and J. Peters, "Trajectory planning for optimal robot catching in real-time," in *International Conference on Robotics and Automation 2011 (ICRA 2011)*, Shanghai, China, May 2011.

[36] C. Büskens, "Optimierungmethoden und Sensitivitätsanalyse für optimale Steuerprozesse mit Steuer- und Zustands-Beschränkungen," Ph.D. dissertation, Westfällischen Wilhelms-Universität Münster, 1998.

[37] C. Specht, M. Gerdts, and R. Lampariello, "Neighborhood estimation in sensitivity-based update rules for real-time optimal control," in *2020 European Control Conference (2020 ECC)*, 2020, pp. 1999–2006.

[38] ESA Requirements and Standards Division, Space project management – Project planning and implementation, "Space engineering - machine learning handbook," *ECSS-E-HB-40-02A*, 2024. [Online]. Available: https://ecss.nl/home/ecss-e-hb-40-02a-15-november-2024/

[39] EASA Artificial Intelligence Roadmap, "Easa concept paper: Guidance for level 1 & 2 machine learning applications," 2024.

[40] H. Delseny, C. Gabreau, A. Gauffriau, B. Beaudouin, L. Ponsolle, L. Alecu, H. Bonnin, B. Beltran, D. Duchel, J.-B. Ginestet *et al.*, "White paper machine learning in certified systems," *arXiv preprint arXiv:2103.10529*, 2021.

[41] F. Tambon, G. Laberge, L. An, A. Nikanjam, P. S. N. Mindom, Y. Pequignot, F. Khomh, G. Antoniol, E. Merlo, and F. Laviolette, "How to certify machine learning based safety-critical systems? a systematic literature review," *Automated Software Engineering*, vol. 29, no. 2, p. 38, 2022.

[42] W. S. Slater, N. P. Tiwari, T. M. Lovelly, and J. K. Mee, "Total ionizing dose radiation testing of nvidia jetson nano gpus," in *2020 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE, 2020, pp. 1–3.

[43] C. Tecchia, M. Piccinin, A. Nocerino, R. Opromolla, and U. Hillenbrand, "Pose acquisition of non-cooperative spacecraft from lidar: Comparison

of methods based on point correspondences," in *AIAA SciTech 2025 Forum*, 2025, p. 1416.

[44] V. Muralidharan, M. R. Makhdoomi, A. Žinys, B. Razgus, M. Klimavičius, M. Olivares-Mendez, and C. Martinez, "On-ground validation of orbital gnc: Visual navigation assessment in robotic testbed facility," *Astrodynamics*, pp. 1–25, 2024.

[45] P. Colmenarejo, F. Gandía, A. García Casas, A. Tomassini, and F. Ankersen, "Gncde: An integrated gnc development environment," in *6th International ESA Conference on Guidance, Navigation and Control Systems, held 17-20 October 2005 in Loutraki, Greece. Edited by D. Danesy. ESA SP-606. European Space Agency, 2006. Published on CDROM., id. 29*, vol. 606, 2006.

[46] R. Spica, G. Claudio, F. Spindler, and P. Robuffo Giordano, "Interfacing Matlab/Simulink with V-REP for an Easy Development of Sensor-Based Control Algorithms for Robotic Platforms," May 2014, 2014 IEEE Int. Conf. on Robotics and Automation Workshop on MATLAB/Simulink for Robotics Education and Research, Hong Kong, China. [Online]. Available: https://inria.hal.science/hal-01332960

[47] H. Mishra, M. De Stefano, and C. Ott, "Dynamics and control of a reconfigurable multi-arm robot for in-orbit assembly," *IFAC-PapersOnLine*, vol. 55, no. 20, pp. 235–240, 2022.

[48] T. Erez, Y. Tassa, and E. Todorov, "Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 4397–4404.

[49] M. De Stefano, M. A. Roa, R. Balachandran, A. M.Giordano, H. Mishra, N. Oumer, B. Brunner, M. Lingenauber, M. Stelzer, and R. Lampariello, *Design and Validation of Orbital Robotic Missions*. Cham: Springer Nature Switzerland, 2024, pp. 281–304. [Online]. Available: https://doi.org/10.1007/978-3-031-39214-6$_1$3