Executive Summary Report

Artificial intelligence techniques for GNC design, implementation, and verification

AI4GNC

Code:	AI4GNC-DME-TEC-FIR02-10-E
Issue:	1.0
Approval Date:	28/11/2022
Confidentiality Level:	ESA UNCLASSIFIED – Releasable to the Public

Approval Signature:

Prepared by: AI4GNC Consortium (DME, INESC-ID, TASC, LUND)

Reviewed by: AI4GNC Consortium (DME, INESC-ID, TASC, LUND)

Approved by: Paulo Rosa / AI4GNC Project Manager

Paulo Rosa
2022.11.28
22:32:55 Z

Document Status Log

Issue	Section	Change Description	Date
1.0	all	First version released to the Agency	28/11/2022

Table of Contents

1. RELATED DOCUMENTS

1.1. Applicable Documents

The following table specifies the applicable documents complied with during project development.

Table 1-1: Applicable documents

Reference	Code	Title	Issue
[AD 1]	AO/1-10403/20/NL/CRS	ESA Invitation to Tender AO/1-10403/20/NL/CRS: "Artificial intelligence techniques for GNC design, implementation and verification"	10/07/2020
[AD 2]	ESA-TECSAG-SOW-019234	Appendix 1 to ESA ITT AO/1-10403/20/NL/CRS: "Artificial intelligence techniques for GNC design, implementation and verification". Statement of Work	1.0 25/06/2020

1.2. Reference Documents

The following table specifies the reference documents taken into account during project development and that are relevant to this report.

Table 1-2: Reference documents

Reference	Code	Title	Issue
[ADD2020]	N/A	Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., ... & Chatila, R. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion, 58, 82-115	2020
[ADL2019]	N/A	J. Anderson, J.C. Doyle, S.H. Low and N. Matni (2019). System level synthesis. Annual Reviews in Control 47: 364-393.	2019
[BDG2005]	N/A	Brat, G., Denney, E., Giannakopoulou, D., Frank, J., & Jonsson, A..Verification of Autonomous Systems for Space Applications. IEEEAC, 1488, 2005	2005
[BKM2021]	N/A	J. Berberich, J. Köhler, M.A. Müller and F. Allgöwer (2021). Data-driven model predictive control with stability and robustness guarantees. https://arxiv.org/abs/1906.04679	2021
[CM2014]	N/A	Chien, S., & Morris, R..Article: Space Applications of Artificial Intelligence. AAAI, 2014	2014
[CTS2020]	N/A	Chai, R.; A. Tsourdos, Al Savaris, S. Chai, Y. Chia and C. Philipe Chen (2020). Six-DOF spacecraft optimal trajectpry planning and real-time attitude control: a deep neural network-based approach. IEEE Trans. Neural Networks and Learning Systems, 31(11): 5005-5013.	2020
[D1]	AI4GNC-DME-TEC-TNO01	Review Document	1.1
[DLB2018]	N/A	Daudt, R. C., Le Saux, B., Boulch, A., & Gousseau, Y. (2018, July). Urban change detection for multispectral earth observation using convolutional neural networks. InIGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium(pp. 2115-2118). IEEE.	2018
[EHH2021]	N/A	Everett M, Habibi G, How JP. Robustness Analysis of Neural Networks via Efficient Partitioning with Applications in Control Systems. Proc Am Control Conf. 2021;2021-May (6):888-893. doi:10.23919/ACC50511.2021.9483033	N/A
[EHH2021B]	N/A	Everett M, Habibi G, How JP. Efficient Reachability Analysis of Closed-Loop Systems with Neural Network Controllers. 2021:4384-4390. doi:10.1109/icra48506.2021.9561348 https://github.com/mit-acl/nn_robustness_analysis	N/A

Reference	Code	Title	Issue
[GSH2017]	N/A	Grosu, R., Smolka, S.A., Havelund, K, Zadok, E., Ratasich, D., Bartocci, E., Selyunin, K., Nguyen, T., Stoller, S., Huang, X., Seyster, J., & Callanan, S. Project: ARRIVE -Adaptive Runtime Verification and Recovery for Mission-Critical Software. ResearchGate, 2017.	2017
[H2021]	N/A	Hua, C. (2021). Reinforcement learning aided performance optimization of feedback control systems. Springer Vieweg.	2021
[LV2009]	N/A	F. L. Lewis and D. Vrabie (2009). Reinforcement Learning and Adaptive Dynamic Programming for Feedback Control. IEEE Circuits and Systems Magazine, 9(3): 32-50.	2009
[LHP2016]	N/A	T. P. Lillicrap, J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver and D. Wierstra (2016). Continuous control with deep reinforcement learning. ICLR 2016.	2016
[MGK2019]	N/A	Retro Propulsion Assisted Landing Technologies (RETALT): Current Status and Outlook of the EU Funded Project on Reusable Launch Vehicles	2019
[MKD2015]	N/A	Makantasis, K., Karantzalos, K., Doulamis, A., & Doulamis, N. (2015, July). Deep supervised learning for hyperspectral data classification through convolutional neural networks. In2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)(pp. 4959-4962). IEEE.	2015
[NBM2017]	N/A	Neyshabur, B., Bhojanapalli, S., McAllester, D., & Srebro, N. (2017). Exploring generalization in deep learning. InAdvances in Neural Information Processing Systems(pp. 5947-5956).	2017
[RD1]	N/A	https://en.wikipedia.org/wiki/MNIST_database	2019 (accessed)
[S2003]	N/A	Stein, G. (2003). Respect the unstable. IEEE Control Systems Magazine, 23(4), 12-25	2003
[STB2018]	N/A	Speretta, S., Topputo, F., Biggs, J., Di Lizia, P., Massari, M., Mani, K., ... & Sundaramoorthy, P. (2018). LUMIO: achieving autonomous operations for Lunar exploration with a CubeSat. In2018 SpaceOps Conference(p. 2599).	2018
[SUM]	AI4GNC-DME-TEC-TNO09	Software User Manual	1.0
[SZS2013]	N/A	Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. and Fergus, R., Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199., 2013	2013
[TFS2015]	N/A	Tian, B.; W. Fan, R. Su and Q. Zong (2015). Real-Time Trajectory and Attitude Coordination Control for Reusable Launch Vehicle in Reentry Phase, IEEE Transactions On Industrial Electronics, Vol. 62, Pag. 1639-1650, 2015.	2015
[VKK2017]	N/A	Vassilyev, S. N., Kelina, A. Y., Kudinov, Y. I., & Pashchenko, F. F. (2017). Intelligent control systems. Procedia Computer Science, 103, 623-628.	2017

2. CONTEXT

Future spacecraft (S/C) missions will require the **ability to adapt to (at least partially) unknown conditions** and have the ability to perform control reconfiguration. Some examples are In-Orbit Servicing (IOS) and Entry, Descent and precise Landing (EDL) missions.

Among all the technical challenges that characterize **EDL missions**, it is highlighted that they may experience significant changes and uncertainty in aerodynamic coefficients, actuator characteristics and initial conditions. Therefore, the ability to **autonomously** adapt to variations in the dynamics and to re-plan the trajectory under the presence of changes to the descent conditions, can play a major role in ensuring mission success.

Similarly, in **in-orbit servicing and useful life extension missions**, there is the need to control the attitude of the composite vehicle with the same performance requirements that apply to the target S/C during its nominal operations.

One of the current control design philosophies consists in employing robust control techniques, such as structured H_∞, to synthetize a limited set of controllers that can robustly stabilize the whole range of potential S/C. This solution has the advantage of minimizing the synthesis effort but also has **one major drawback**, which is the well-known paradigm of automatic control: the **price to be paid to ensure robustness is that of performance, and vice-versa**.

Directly related to the need for adaption is the **ability to recognize performance degradation** during the mission time, which can be done by monitoring relevant performance indexes. Once the performance degradation reaches a pre-defined threshold, the parameters of the system may need to be re-evaluated, and the controller re-configured.

For model-based method unable to incorporate past experience from data, the behaviour of the dynamics under different situations has to be explicitly modelled, which rapidly increases the complexity of the model. Modelling all possible situations is often an unfeasible task, and thus the **use of AI/ML[1] techniques that provide the means to introduce knowledge from data have gained increased popularity and offer key benefits**. Such techniques promise to provide some advantages over traditional methods regarding the ability to explicitly take into account **nonlinear environments**, the **robustness to unmodeled events** [NBM2017], the support to **fast development**, and the **superior performance levels**.

Moreover, the **increasingly more powerful processing units (HW)**, together with **enhanced and less computationally intensive AI algorithms**, make it possible to explore new AI applications. Given that results in this direction indicated the plausibility of implementing onboard spacecraft these algorithms, the use of AI within the avionics system is within reach and this is a remarkable target to be pursued[2]. The use of AI in the space domain has been applied to both upstream applications ([STB2018], [S2003]), and for downstream ones ([SZS2013], [MKD2015], [DLB2018]), with ESA also promoting AI for space in activities such as AI4EO.

Nevertheless, a significant number of **challenges** still exist in order to reach the goal of fully exploiting AI for space applications. While challenges such as the computational burden of AI algorithms are fairly well addressed, both from the HW and the SW sides, solutions for the <u>validation and explainability of AI implementations</u> that can be conveyed to **critical** (including) space applications has only recently started to appear [GSH2017], [BDG2005], [CM2014], [ADD2020].

These challenges are also motivated by **the gap between the performance of AI and the theoretical understanding** and modelling of the behaviour of those algorithms. Moreover, due to the intrinsic nonlinear structure of the AI algorithms, <u>the generalization of the learned features to different scenarios is not straightforward</u>. As a consequence, Monte-Carlo (MC) simulations per se are not guaranteed to <u>cover</u> the whole space of potential configurations of parameters, as minor changes in the inputs of the AI can lead to completely different results in the output [SZS2013], while probability distribution functions are typically unknown for these problems.

[1] Throughout this report, the terms AI and ML are used interchangeably.

[2] Notably, DME is the prime contractor, and participates in, the ESA **FSSCAT** mission, technically lead by the UPC, that launched in the summer of 2020, and performed the 1st demonstration of on-board AI using a VPU. DME also led the application of the same AI HW to implement AI for GNC in the ESA **AIVIONIC** activity.

When it comes to the Space sector, this challenge is further amplified by the lack of existing large data sets that could be used for training and validation purposes. In applications such as EDL, this is further exacerbated by the need to fly-right in the first flight, where no incremental validation is possible. In addition, there are also **challenges coming from the software side**, such as predictability, clear/understandable behaviour, memory and CPU usages, compatibility with real-time constraints, etc. Hence, **validation limitations need to be considered at all levels**, especially because GNC has an impact on the S/C itself and possibly on others.

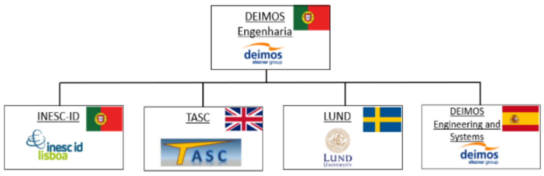

Figure 2-1 - AI4GNC consortium structure

The structure of the AI4GNC consortium and the associated work breakdown structure are represented in Figure 2-1 and Figure 2-2, respectively.

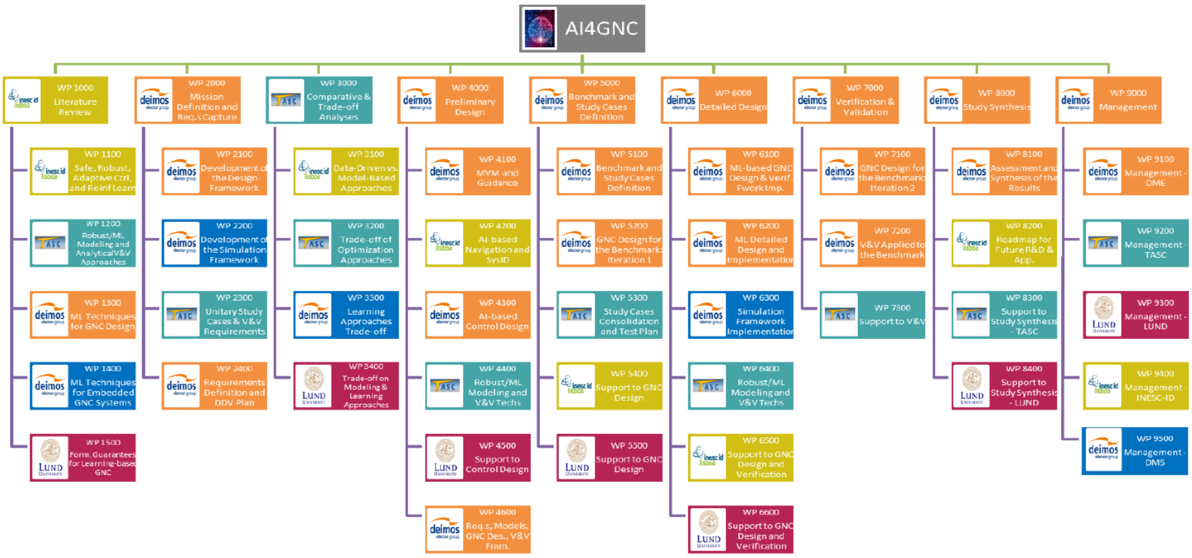

Figure 2-2 - AI4GNC Work Breakdown Structure (WBS)

3. OBJECTIVES

The overall objectives of AI4GNC are as follows:

Objective 1: Implement ESA-i4GNC, an AI-based GNC E2E design & analysis framework for layered architectures

- Cover the GNC system modeling, design and V&V process
- Supported by efficient optimization algorithms and formal mathematical techniques
- Ensuring robustness, performance, convergence, and explainable results

Objective 2: Exploit recent advances in control and AI

- Revisit the theory and techniques developed in the last two decades, including, but not limited to, fields such as IQCs, robust control, adaptive control, safe and robust reinforcement learning, and system identification
- Increase autonomy through onboard intelligence

Objective 3: Perform Trade-off analyses

- Different concepts to be considered, including full dedicated design architectures and augmentation strategies for already-existing control architectures
- Trade-off the offline design effort with the online real-time implementation requirements

Objective 4: Evaluate the proposed AI-based GNC design and V&V tool in a representative benchmark

- Define the criteria to sleect the benchmark
- Derive study cases and apply the tool to those
- Apply the tool to the benchmark

Objective 1: Implement ESA-i4GNC, an AI-based GNC E2E design & analysis (D&A) framework for layered architectures

The first goal of AI4GNC is to develop the **Enhanced Safe Artificial-intelligence Guidance, Navigation, and Control (ESA-i4GNC) tool**, which is an AI-based GNC E2E D&A framework for layered architectures. This tool is able to cover modelling, GNC design, V&V, training of ML algorithms, requirements specification, etc., being a key building block for, and enabler of, Model-Based Systems Engineering (MBSE). The tool has been built on well-established design and validation paradigms and frameworks, exploiting optimization approaches with solid theoretical backgrounds, in order to ensure the robustness of complex, multi-layered GNC systems.

Objective 2: Exploit recent advances in control and AI

There are clear advantages and interest in the community in integrating classical methods of automatic control with methods of AI for complex objects and processes [VKK2017]. The use of AI can be introduced to work in conjunction with the control systems, to expand the capacity of control systems when there exists a failure or a significant mismatch of the actual dynamics to the model used for design, when there is no availability of input data, or where quantitative models show worse performance compared to AI-based solutions.

Objective 3: Perform trade-off analyses

A detailed literature review and **thorough comparisons between the different methods**, theoretical backgrounds, optimization approaches, and AI/ML techniques, has been performed within AI4GNC. Performance metrics such trade-offs were defined, being drivers for these metrics the computational effort, the availability of dedicated HW architectures, the convergence rate, the usefulness (sub-optimal) of the results at intermediate iterations, and the ability to explain the observed results.

Objective 4: Evaluate the proposed AI-based GNC design and V&V tool in a representative benchmark

One of the key goals of AI4GNC has been to **critically analyze the benefits and drawbacks of using ML techniques** in the design, analysis, and incorporation (being one additional element) of GNC systems. To do so, one benchmark – a reusable launch vehicle - has been proposed in AI4GNC. E2E GNC design and analyses have been performed for this benchmark, by using the ESA-i4GNC tool. This allowed demonstrating the applicability of the tool and support the definition of a roadmap for future progress in the field. In order to complement this analysis, and to **support a more agile development process**, case studies, derived from the benchmark, were proposed. **These case studies aimed to allow the assessment of specific elements of the ML-assisted GNC solutions to be implemented**, and can be interpreted as preparatory steps or early iterations for the benchmark GNC design.

4. RESULTS

An overview of the main tasks within the AI4GNC project, including the trade-offs between several AI- and non-AI-based techniques for GNC, learning approaches, Machine Learning (ML) libraries and tools, and modelling techniques, is presented next. The selected benchmark to showcase the techniques, libraries and tools is presented, together with the ESA-i4GNC framework. Finally, the main results obtained are presented and discussed, and conclusions are drawn.

4.1. Trade-offs between techniques

This topic was subdivided into several tasks related to data-driven, model-based, modelling and learning methods, and libraries and tools that can be used for this purpose. It is remarked that the trade-offs presented in this section are a result of an extensive state-of-the-art literature review presented in [D1].

4.1.1. Data-driven vs. Model-based approaches

A qualitative trade-off between the techniques presented in the literature review on safe, robust, adaptive control and RL methods has been performed in AI4GNC. The control algorithms considered for comparison are as follows: Adaptive ADP/RL control [LV2009],Youla-Kucera RL control [H2021], Deep RL-based control [LHP2016], Switched NNs [TFS2015], [CTS2020], Data-driven MPC [BKM2021], and System Level Synthesis [ADL2019].

Overall, it was concluded that model-based techniques are, in general, easier to be verified and validated, but techniques like deep-RL can lead to remarkable levels of performance and flexibility (i.e., ability to address different types of GNC problems and different models), although posing challenges on the design and verification stages. Those techniques were, therefore, further explored and developed in this project.

4.1.2. Learning approaches

A quantitative trade-off between the most relevant ML methods, focusing especially on methods with GNC applications, was performed. The trade-off was based on the following GNC relevant criteria: explainability – 20%, onboard implementability – 10%, training datasets needs – 20%, performance – 30% and generalizability – 20%. In a scale from 1 to 5, the methods with the highest score in terms of the criteria described above are based upon NNs, due to their capability to generalize, while entailing high levels of performance in most nominal scenarios, within the supervised learning methods. Regarding RL methods, both DQN and DDPG scored 3.6, although DDPG is preferable for the majority of space-related problems, due to its ability to work with continuous-time action and observation spaces.

4.1.3. ML Toolboxes

For supervised and unsupervised learning, the comparison criteria considered are: algorithms available, documentation, support for embedded devices, and integration with MATLAB. The comparison between the most used tools was made, and it can be concluded that the tool with the highest score is the TensorFlow library. This library was developed and is maintained by Google and provides support for Deep Neural Networks, Convolutional Neural Networks and Recurrent Neural Networks.

For reinforcement learning, the criteria considered were: algorithms available, documentation, code readability, and supported environments. From the analysis performed, it is possible to conclude that the tools with the highest score are the MATLAB RL Toolbox and RL Coach. The main disadvantage of the MATLAB RL Toolbox is that it is not open-source, when compared to RL Coach, this being the reason why RL Coach was selected to implement the RL methods for one of the case studies described below. Since the RL coach framework is implemented in Python, a from/to connection had to be established between the MATLAB and Python environments. This has been done by implementing a TCP connection between the ESA-i4GNC tool and Python which allows (in case study #8, for example) to exchange data needed for NN training in a reasonably fast way.

4.1.4. Modeling & Learning approaches

The goal of this section is to briefly present two different approaches at opposite ends of the guarantee spectrum of learning-based control: IQC and learning-based MPC.

4.1.4.1. Integral Quadratic Constraints (IQCs)

The Integral Quadratic Constraints (IQC) framework provides methods for obtaining global guarantees of closed-loop performance for systems that include nonlinear elements. In particular, this is relevant for systems with NN as one of its nonlinear parts, that can act as, for instance, the controller or to model the nonlinear dynamics of a subsystem. The IQC framework requires the characterization of any nonlinearity or uncertainty using a quadratic form, and also the characterization of a target guarantee, e.g., closed-loop stability, as a quadratic form. From those quadratic forms, an SDP problem can be obtained and, if the SDP problem has a solution, then the solution is a verification that a guarantee exists for the system.

4.1.4.2. Learning-Based MPC

Learning-Based Model Predictive Control (LBMPC) provides a methodical formulation of the incorporation of statistical and machine learning techniques into the control design process. The underlying methodology behind this method is: (1) use a constant LTI model to predict the system's response alongside the robustified constraints which guarantees robust constraint satisfaction; (2) use a separate model which adaptively approximates the nonlinear dynamics as new data becomes available.

The comparison between the two approaches is based on the following criteria: Flexibility (10%), Easy design (10%) Online computational requirements (15%), Performance (25%), Robustness and verifiability (25%), Explainability (10%) and Mission criteria maturity level (5%). The IQC-based method of guaranteeing closed-loop stability is currently intended for off-line use, and hence received the highest score with respect to online computational requirements. It is also superior in terms of robustness and verifiability because of its hard robust performance and stability guarantees. Learning-based MPC, on the other hand, can currently handle a much wider range of problems and hence receives the higher performance and flexibility scores. Hence, as final result, the overall score is slightly higher for LBMPC. However, both methods contain a mix of data-driven and model-based approaches, something that will probably be appropriate for future space applications.

4.2. ESA-i4GNC overview

This tool was designed such that a general GNC architecture could be implemented in a systematic and modular manner, while allowing the implementation of diverse algorithms and the satisfaction of predefined requirements, supporting and managing models with different levels of fidelity/complexity. To fulfil these goals, the an Object-Oriented Programming (OOP) approach was considered, that allows different levels of abstraction and also the definition of their properties and methods.

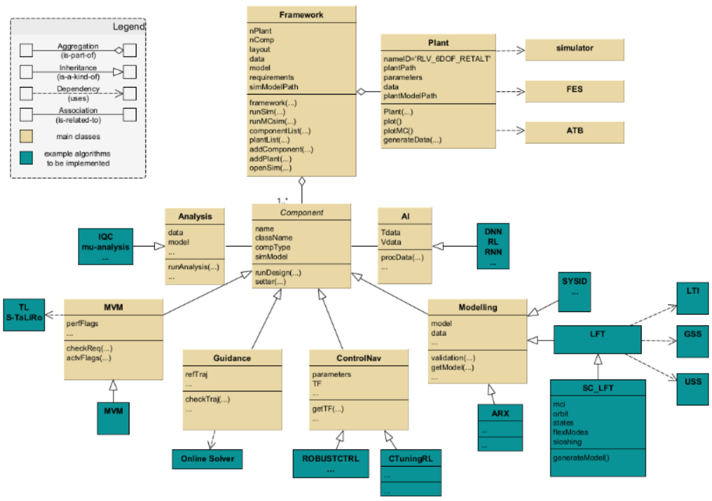

Figure 4-1 - Proposed GNC Framework architecture

In Figure 4-1, the yellow boxes represent the main OOP classes that, depending on the algorithms to be implemented (blue boxes), are instantiated to create objects. In addition to the implementation of the main classes, there are other libraries and functionalities of the tool that further generalize the tool, such as: CVX, MPT 3.0, S-TaLiRo Runtime Verification and Falsification, M2HTML, esai4gnc_install, esai4gnc_clean, autoDoc_ESAi4GNC, and profiler.

4.3. Benchmark selection

The benchmark selected within the AI4GNC project is the RETALT-1 Reusable Launch Vehicle (RLV) from the H2020 RETALT project. In terms of actuators, the launcher is equipped with: Thrust Vector Control (TVC), Reaction Control System (RCS), and Aerodynamics fins. For the purpose of this project, the flight phase of interest is the landing, that involve only the first stage of the launcher after the separation from the second stage.

The state model of the burn landing problem is defined by the following equation in the state space form $\dot{x} = f\,(x(t), u(t))$, where $x(t) = [\,m(t), p_{UEN}(t), v_{UEN}(t), q(t), \omega_B(t)]$ is the state vector composed of the mass of the rocket, the rocket position and velocity in UEN frame (Up-East-North), the attitude of the rocket in quaternions from UEN to Body frame and the rocket angular velocity in Body frame; while $u(t) = [\,\delta_y(t), \delta_p(t), T(t)\,]$ is the input vector containing the thrust vector magnitude and the two TVC gimbal angle about yaw and pitch.

For comparison purposes, the baseline GNC system adopted for the landing phase of the benchmark RLV is a non-AI solution that considers SCVX guidance, a PID controller, and ideal navigation. The selected guidance algorithm is one of the most well-established techniques to solve non-convex optimal control problems with nonlinear dynamics and non-convex state constraints. It is worth to mention that, within the project framework, a simplified version of the algorithm has been adopted both to generate a "one-time" solution, but also to compute re-optimized guidance trajectories in real-time based on the current flight conditions.

4.4. Case studies drivers and definition

After the identification of the methods, from the literature review and the trade-off analyses, that best suits the goals of the project, several case studies were developed and for which specific methods were discussed, implemented and tested by exploiting the benchmark presented above, as follows:

- **Case Study #1:** Reinforcement Learning (RL) based adaptive controller to regulate the attitude of the RLV, in the presence of uncertain dynamics and/or uncertain and strong disturbances.
- **Case Study #2:** cascade control to regulate the position around reference values provided by guidance, with the outer loop acting on the reference to inner loops controlling attitude.
- **Case Study #3:** approximation of an algorithm that is part of a pipeline for state estimation in a navigation system by a NN.
- **Case Study #4:** compressed sparse regression for online system identification.
- **Case Study #5:** learning-base model predictive control (LBMPC) for attitude control, in uncertain environments.
- **Case Study #6:** IQC formalism for NN-based attitude control verification, under large disturbances and uncertain dynamics.
- **Case Study #7:** learning-based Kalman filtering for attitude estimation.
- **Case Study #8:** integrated Guidance & Control (G&C) Deep-RL approach for trajectory tracking for the landing phase.

4.5. Techniques and Results

The goal of this section is to briefly present the techniques analyzed under the case studies and some of the most relevant results, including Monte-Carlo (MC) analyses, for some of the techniques. The results were obtained using the ESA-i4GNC framework.

4.5.1. Reinforcement Learning

This section presents the RL adaptive control algorithms developed within case studies #1 and #2, on the attitude and position tracking problems. Simulations were performed for yaw angle control with RL under nominal conditions, developed in case study #1 for the attitude tracking problem. From the simulation results, it was observed that the tracking performance is very good. In the time span in which the RL controller is acting (after 10 s), the tracking error is smaller then $\pm 0.02^o$.

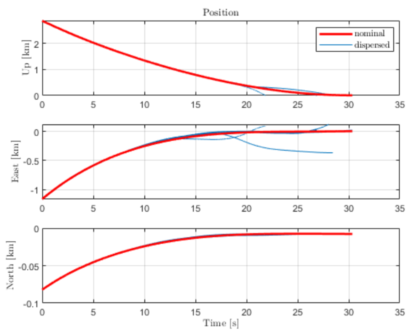

Figure 4-2 - Position profile form the MC analysis

A second controller was also designed for the position tracking error, in case study #2. This corresponds to a cascade control architecture, where the inner loop corresponds to the attitude controller developed in case study #1, and the outer loop correspond to a LQ regulator. It was seen that the position tracking error is withing the requirements, under a nominal scenario. A MC campaign with 50 shots was also executed, and the results in Figure 4-2, showing that 8% (4/50) of the runs resulted in an unstable trajectory, thus not compliant with the requirements.

4.5.2. Compressed Sparse Regression for Online System Identification

The dynamical system identification is based on sparse regression and compressed sensing techniques. In particular, the proposed approach exploits the fact that the dynamics of physical systems, including launch vehicles, are generally defined only by a few non-zero terms, which makes the equations of motion sparse. It is remarked that the system identification task is connected with robust control via LFT theory to identify the uncertainty range of the system parameters. This allows evaluating the state of the system in real-time, which paves the way for on-board system monitoring of failures and performance degradation. A Monte-Carlo Campaign of 1000 shots under no-fault and fault scenarios was performed and the results obtained were satisfactory, as described in Figure 4-3 and Figure 4-4. For the no-fault scenario analysis, overall, the proposed approach successfully captured the variability of both parameters, except for the region around $t = 8s$, due to the sudden increase of thrust magnitude commanded by the guidance function. For the fault scenario, a 30% loss of thrust engine failure is implemented at $t = 15s$. In the second figure, it is seen on the upper plot that the engine thrust failure does not have a strong impact on a_6, which is rather related to the vehicle aerodynamics, with the failure being captured from the estimation of k_1.

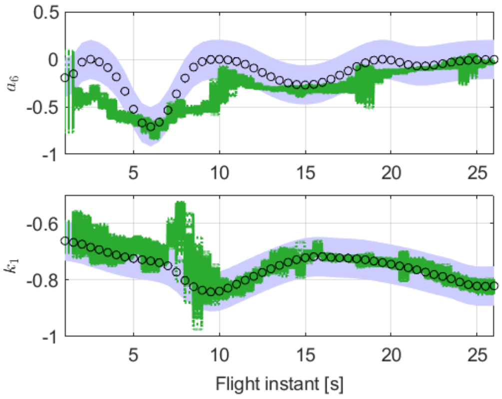

Figure 4-3 - MC campaign with 1000 shots, scenario with no failures

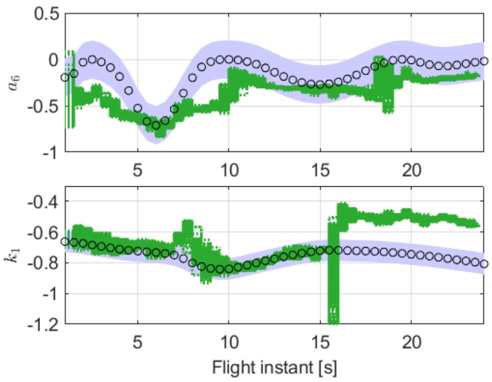

Figure 4-4 - MC campaign with 1000 shots, scenario with engine failures (30% loss of thrust) at t=15s

4.5.3. Learning-Based MPC

The LBMPC algorithm design approach considers the estimation provided by the compressed sparse regression algorithm (defined and implemented in case study #4) to obtain a sequence of linear models based on the time evolution of the parameters a_6 and k_1, called the aerodynamic instability coefficient and the control efficiency parameter, respectively. From the time evolution of the two parameters described above, six linear models were obtained from which the corresponding LBMPC controllers were designed. The state to be controlled corresponds to the error between the attitude angles and the reference attitude angles, and the angular velocity, being the output the gimbal angles for the TVC actuator. As it was implemented, the algorithm is quite general and can handle various types of model errors. The simplest option is to use a linear oracle, and this was deemed sufficient for the attitude control benchmark. Furthermore, LBMPC was shown to result in attitude regulation for RETALT vehicle model at different points along its reference trajectory, as observed in Figure 4-5.

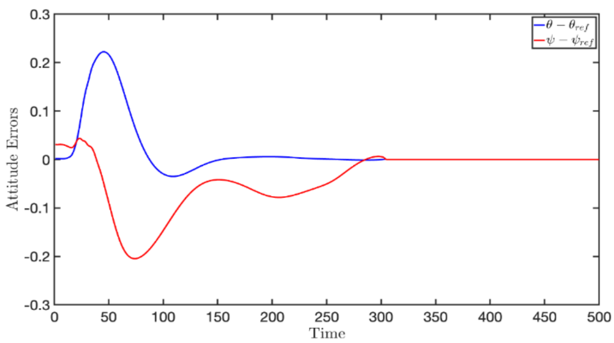

Figure 4-5 - Regulated attitude error states

4.5.4. Learning-Based Kalman Filter

To showcase the LBKF algorithm, the RLV landing problem was slightly modified to generate sensor measurements that are rich enough to clearly see differences between using different estimators. Three techniques were implemented and compared: the standard Kalman Filter (KF), the LBKF with linear regression model and the LBKF with Gaussian process model. Comparing the mean square errors (MSE) of the three techniques, it is possible to conclude that using the LBKF with the linear regression model, the best estimation results are obtained. A 100-shots MC analysis was also performed and, from these results, it was possible to conclude that the LBKF with Gaussian process model is able to satisfy the attitude estimation requirement of 2.5 deg, as observed in Figure 4-6.

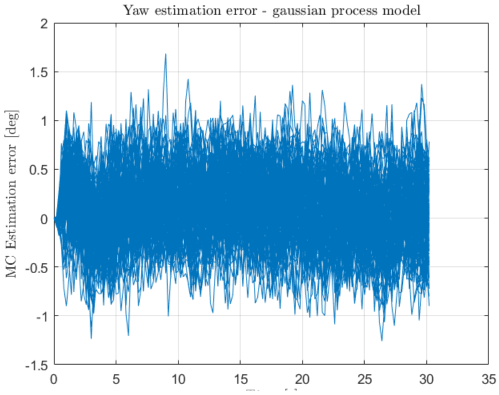

Figure 4-6 - Yaw error for each 100 MC run

4.5.5. Deep Reinforcement Learning

This technique has been adopted in case study #8 to address the Guidance and Control (G&C) problem for the landing phase of the RLV. A NN was trained to directly map the sensor input (observation state) to the control actions (action state), through the RL Coach library. The agent is the DDPG algorithm while the environment is the 3D complete landing scenario. A fine reward function shaping was required to find a suitable reward function able to teach the agent (launcher) to land properly. From the results obtained, it was possible to conclude that the NN was able to make the launcher land successfully and smoothly with a reasonable fuel consumption compared with the baseline non-AI solution. Nevertheless, it is worth to remark that, in the context of case study #8, several different NNs have been trained and tested, including NNs with a lower fuel consumption with respect the baseline solution.

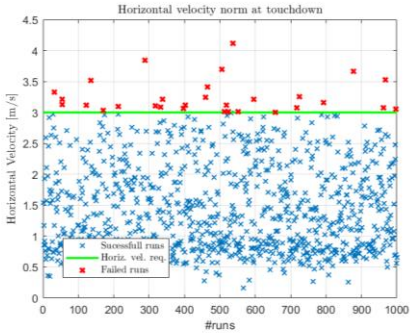

Figure 4-7 - Final horizontal velocity for the MC simulation

Moreover, a 1000-shots MC campaign has been carried out for the NN presented above in order to assess the robustness with respect the landing requirements. Figure 4-7 shows the maximum required velocity at touchdown with the green horizontal line. As observed, only 30/1000 shots (3%) failed. The failures are mainly related to slightly higher horizontal component velocity; however, it is worth to remark that the NN has not been trained with the presence of wind, and in fact it led to a 100% success rate for the scenario without wind. Moreover, the comparison between the NN and the baseline SCVX re-computed each 5 seconds along the trajectory, was done in terms of landing accuracy and fuel consumption. It is interesting to note that, while the baseline SCVX method show an average better velocity accuracy at touchdown, the NN model yields a better position accuracy.

4.5.6. Validation of NN

The main goal of this section is to present the results obtained with the validation with IQC's and the reachability analysis performed by the robustness analysis tool [EHH2021][EHH2021B], which are fully compatible with the ESA-i4GNC framework. Both approaches consider a neural network (NN) trained for the vertical landing (1 DoF) scenario of the RLV benchmark, using the method defined and implemented in case study #8.

4.5.6.1. IQC validation

Figure 4-8 shows the large-scale view, where we selected validation for initial altitudes of 80 to 100 meters, and initial downward speeds between 10 and 20 meters per second. Our blue curve describes a narrow corridor which the trajectories enter, and the second figure shows a closer view of the position at which our corridor begins. The end of the corridor shows that all trajectories are guaranteed to pass through a state with, e.g., an altitude of 22 meters and a downward speed between 8 and 8.3 meters per second. Furthermore, the following points have been identified as relevant for future work on this topic: fuel consumption; higher downwards speeds; attitude dynamics.

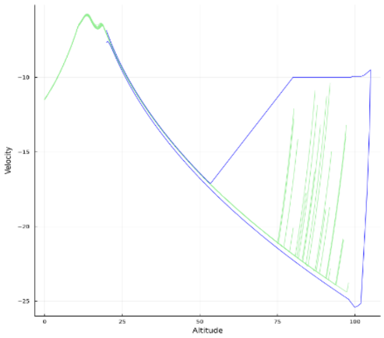

Figure 4-8 - Allowed region

4.5.6.2. Reachability analysis

The analysis [EHH2021][EHH2021B] has been performed considering the NN trained for the vertical 1DoF scenario. The NN receives as input the altitude and the difference between the actual velocity and a target velocity (function of the state) and output the thrust magnitude. In the context of the analysis, a smaller NN was trained and "attached in front" of the main NN, so that the observation state change to simply the altitude and actual velocity. Moreover, in order to tackle the lack of support for 'Tanh' activation function by the tool, another small NN has been trained to approximate the 'Tanh' function present in the last layer of the original NN, with only 'relu' and 'linear' layers. An additional operation, crucial for the convergence of the results, consisted in the training of the smaller NN with a supervised approach, using data coming from the original larger NN. This approach is also called 'teacher-student'. After having the new small NN and the launcher discrete time dynamics defined, the close loop propagation has been performed considering the Greedy Sim Guided partitioner and the CROWN propagator. Setting an initial range for the initial conditions, the propagation of the dynamics and the reachable sets has been carried out showing considerably satisfying results, as showed in Figure 4-9.

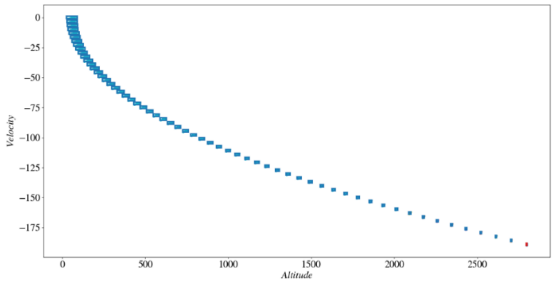

Figure 4-9:Validation results obtained with the smaller NN that mimics the larger NN

5. CONCLUSIONS

This document very briefly presented the main work developed during the entire **AI4GNC** project, starting with the overall context and main objectives of the project. An **extensive literature review** was made, from which a first screening of the most relevant methods was made. From the methods selected, **several case studies were identified, developed, implemented** and their results analyzed in a benchmark that uses the RETALT RLV dynamics. The benchmark also considers realistic aerodynamics and wind models.

In parallel to the implementation of the methods, the **ESA-i4GNC framework** was developed for the design and validation of AI-based GNC system. This framework was developed in **MATLAB/Simulink** considering an **Object-Oriented Programming** approach, allowing to tool to be **flexible and modular**. Besides the integration of several tools, such as CVX, MPT 3 and S-TaLiRo, **an interface between MATLAB and Python** was developed, allowing the extension of the MATLAB functionalities to the Python libraries and toolboxes, that also allowed the **interface to other external frameworks**, such as **Julia**.

After a brief description of the identified case studies, the techniques considered and their results were presented here, from which the following conclusions can be drawn:

- **Learning-based KF:** Although the satisfactory results, it is deemed to have a **low** potential indicator mainly due to the non-existence of formal guarantees for its performance as well as expensive online calculations.
- **Learning-based MPC:** Although the LBMPC controller was able to control the tracking error to zero, the associated tuning process is not straightforward. Therefore, the results indicate the associated potential, with no further research, to be **low**. However, it is stressed the credibility of the approach.
- **Compressed sparse regression for online system identification:** The compressed sparse regression approach showed very satisfying and promising results, as the intense MC campaign stresses. Therefore, the potential indicator for this approach is considered **high**.
- **Reinforcement learning applied to a realistic RLV scenario:** Although the technique provides a way of tackling uncertainty in the dynamics of regulation problems, the resulting closed-loop dynamics is quite complicated and prone to unstable behaviour that is not completely understood. Hence, the potential indicator is considered to be **medium**.
- **Deep reinforcement learning applied to a realistic RLV scenario:** The Deep RL approach yielded remarkable results on the extensive MC campaigns, passing all of the V&V tests with a significantly high level of confidence, being onboard implementable. This was further confirmed by the fact that the results obtained were comparable to the ones from approaches such as SCVX, besides being prone to formal validation. Thus, the potential indicator is considered to be **high**.
- Validation of NNs:
    - **IQC-based analyses:** Since the technique was only used on the 1D scenario without any attitude dynamics, and the expectance of difficulty on extending its use to the 3D scenario, the potential indicator selected is **low**, at the moment, although research on the topic is encouraged.
    - **Reachability analyses:** Similarly to the IQC-based methods, the reachability analysis has been performed only for the 1D scenario NN. Therefore, a **medium** potential indicator has been associated to the study, with further work towards the complete 3D scenario being required but promising.