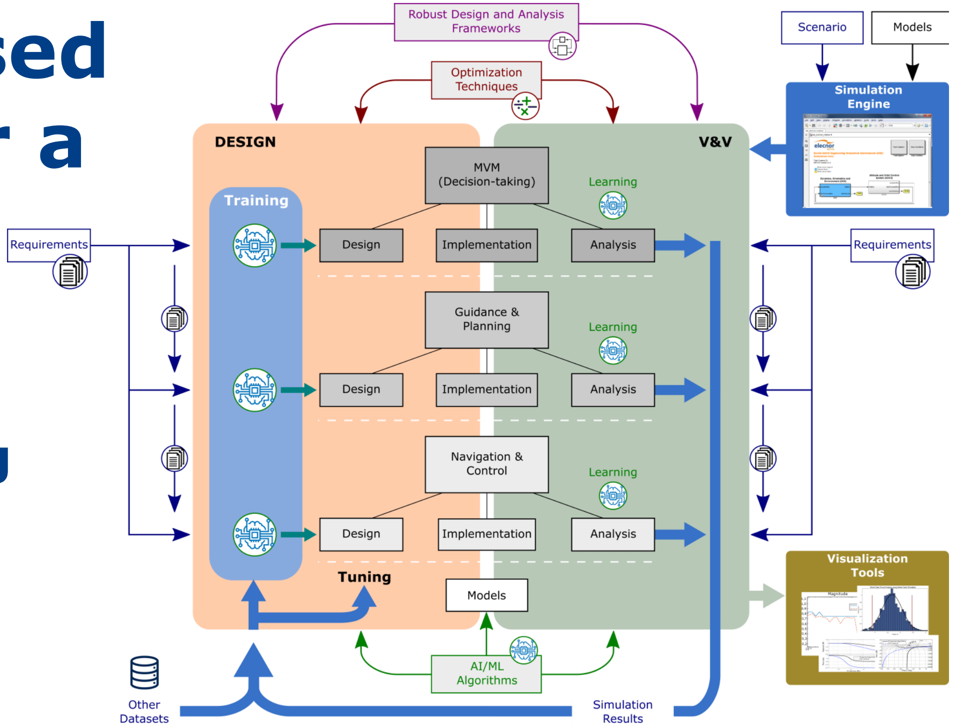
Deep Reinforcement Learning-based Integrated Guidance & Control for a Launcher Landing Problem



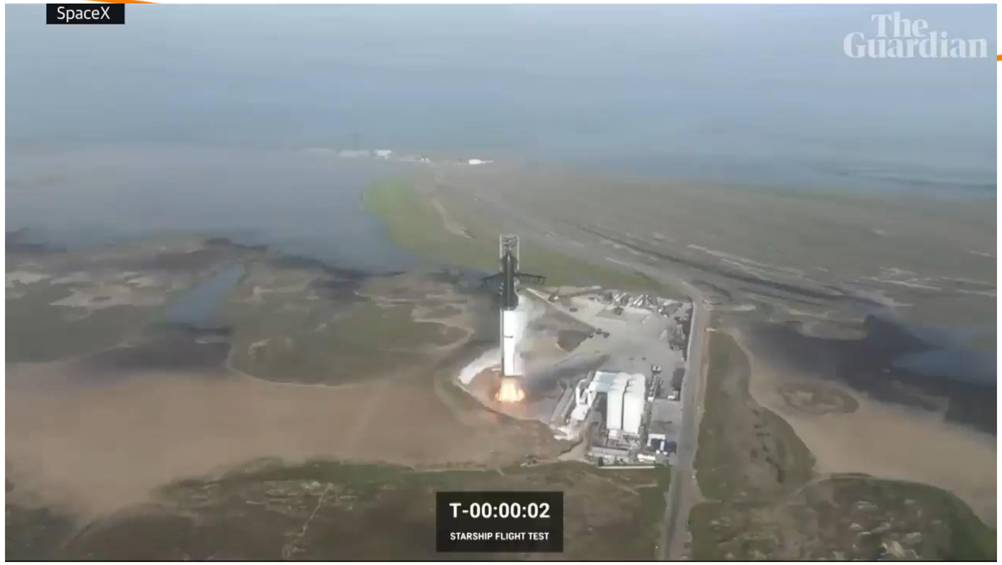
Seminar in Mathematics, Physics & Machine Learning

April 27, 2023

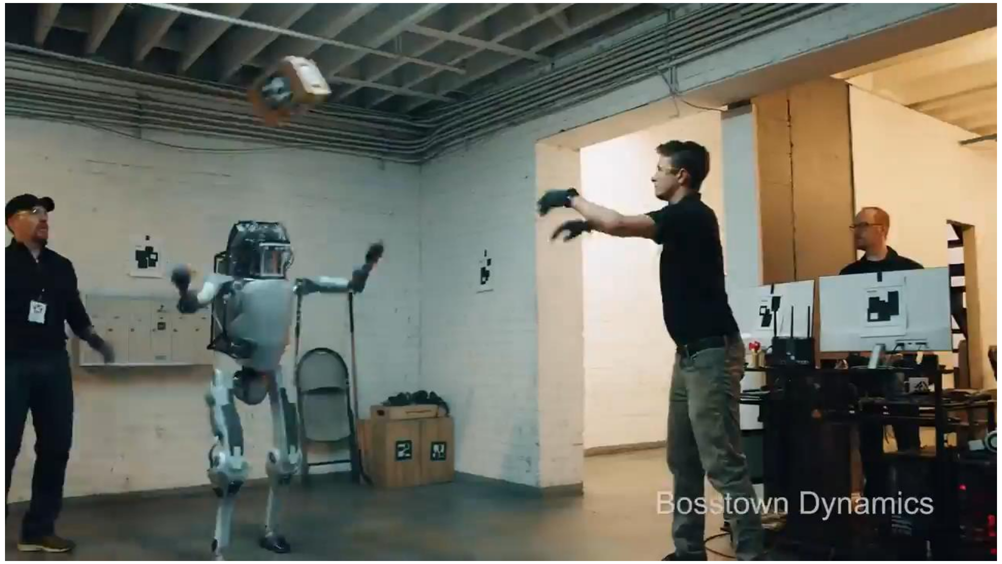
Paulo Rosa (paulo.rosa@deimos.com.pt)

Introduction
Challenges Addressed

- How can we cope with the level of uncertainty in a reusable launch mission?



- Can we really trust AI for that task?



deimos

EXPANDING FRONTIERS

elecnor

A LEADING GLOBAL PLAYER

ENGINEERING

INFRASTRUCTURE

RENEWABLE ENERGY

ENERGY & ENVIRONMENT

NEW TECHNOLOGIES

2450 M€ of Turnover in 2020

More tan 60 years of expertise

More than 50 Countries

More than 14,000 Employees around the world

TECHNOLOGY COMPANY OF THE ELECNOR GROUP

SPACE

AERONAUTICS

MARITIME

TRANSPORT

INDUSTRY & UTILITIES

TELECOM & MEDIA

More than 500 High-tech projects

More than 20 Years of expertise

5 Countries

>400 Highly-qualified staff

FOUNDED IN SPAIN IN 2001
NATIONAL AND INTERNATIONAL EXPANSION

Shareholder of ORBEX, private launch services company

deimos

SPACE

Capabilities to lead a complete space mission

SPACE AREAS

SPACE SCIENCE & EXPLORATION

SATELLITE NAVIGATION

EARTH OBSERVATION

SPACE SITUATIONAL AWARENESS

LAUNCHERS

Expertise across the entire value chain in satellite systems

Deimos technology is present in more than 60 satellites

PHASE 0 ▷ PHASE A ▷ PHASE B ▷ PHASE C/D ▷ PHASE E ▷ USER

DEFINITION	DESIGN	DEVELOPMENT	LAUNCH	APPLICATIONS

deimos

MORE THAN 500 CONTRACTS

esa

EUMETSAT

NASA

cnes
CENTRE NATIONAL D'ÉTUDES SPATIALES

UK SPACE AGENCY

asi
agenzia spaziale italiana

THE WORLD BANK

DLR

INTA

ThalesAlenia Space
a Thales / Leonardo company

AIRBUS

OHB

SENER

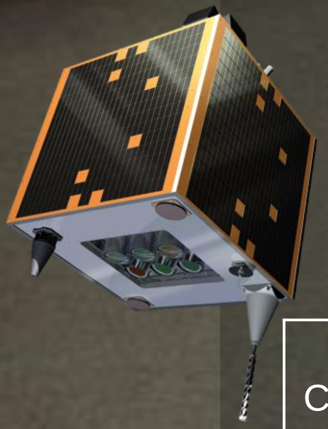
urthecast

SI SATREC INITIATIVE

SURREY
SATELLITE TECHNOLOGY LTD

SATELLITE SYSTEMS

DEIMOS-1



First Spanish Earth Observation Satellite

Copernicus contributing mission

Operated by Deimos Imaging

CHARACTERISTICS	Optical Resolution: 22 m Coverage: 600 km
APPLICATIONS	Crop yield prediction Emergency response Maritime surveillance
LAUNCH	July 2009

DEIMOS-2

Integrated and tested at Deimos Satellite Systems premises

Operated by Deimos Imaging

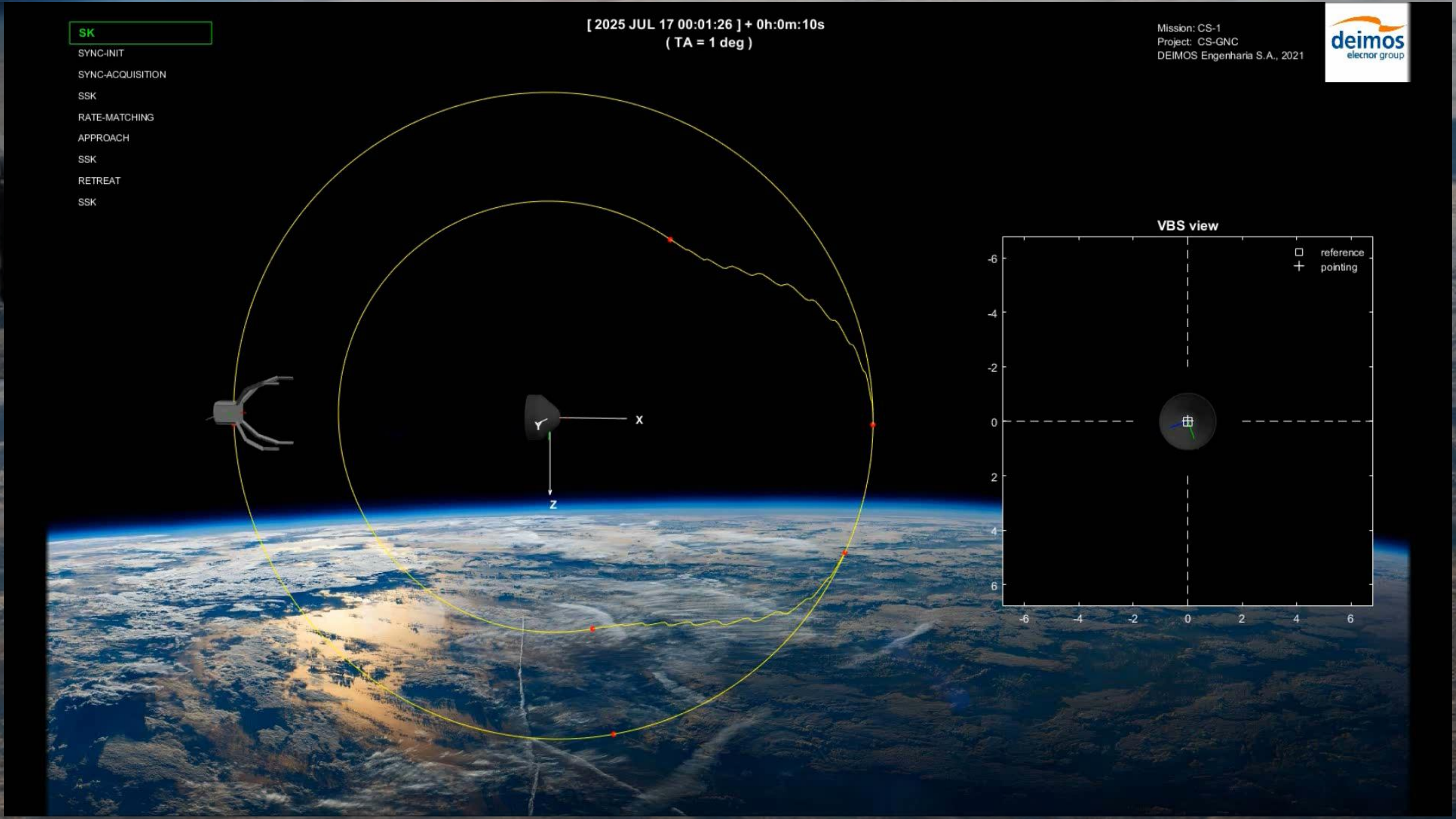
CHARACTERISTICS	Optical Multispectral Resolution: 75 cm Coverage: 12 km
APPLICATIONS	Intelligence Emergency response Urban planning
LAUNCH	June 2014

Example Projects

ADRIOS/ClearSpace-1
DEIMOS is the GNC SS Responsible

GNC Subsystem and Mission Analysis responsibility

Includes all the phases of the mission, i.e. orbital, rendezvous & capture, and de-orbiting



Orbex
Prime Launcher GNC

ORBEX

deimos
elecnor group
Leading the Space systems of tomorrow

DEIMOS Flight Systems responsible for

- Mission Analysis
- Guidance, Navigation, and Control for all flight phases
    - Ascent phase (up to 80 km)
    - Orbital phase (including circularization burn maneuvers, safe modes implementation, etc.)
- GNC Failure, Detection, Isolation, and Recovery (FDIR)

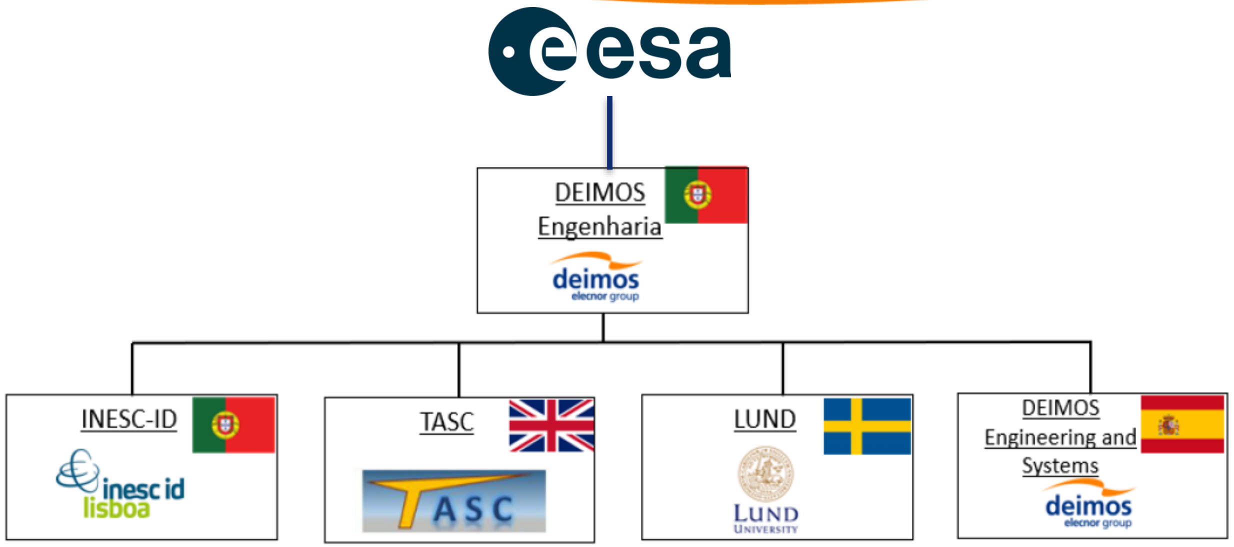
Image credits: Orbex

deimos

- How can we cope with the level of uncertainty in a reusable launch mission?
- Can we really trust AI for that task?

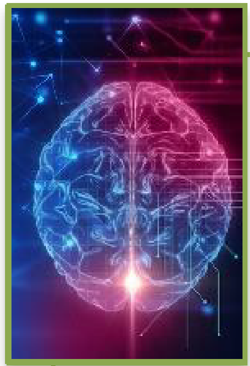
AI4GNC: Artificial intelligence techniques for GNC design, implementation, and verification

AI4GNC
Consortium



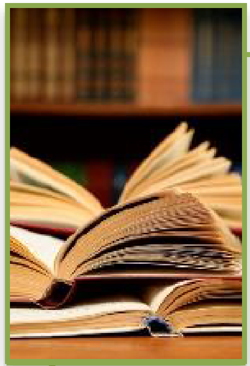
- **DEIMOS Engenharia**
  - Overall project coordination
  - ESA-i4GNC framework development
  - Application to the DRL case study
  - Overall software implementation of the tool
- **INESC-ID**
  - Literature review, trade-off analysis and AI-based GNC design support
  - Contribution for topics such as adaptive control and reinforcement learning control
- **TASC**
  - Responsible for the robust+ML modelling and V&V techniques, inc. the study cases consolidation and test plan
  - Support the selection of the study cases, as well as the implementation of the algorithms
- **LUND**
  - Contribute with state-of-the-art knowledge on analytic robustness and convergence guarantees (e.g. Integral Quadratic Constraints (IQCs)), especially in the presence on nonlinearities, e.g. NNs
- **DEIMOS Engineering Systems**
  - ML/AI techniques development for embedded GNC systems, inc. the learning and simulation approach and associated simulator

Scope and Goals of the Activity



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

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



Goal 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

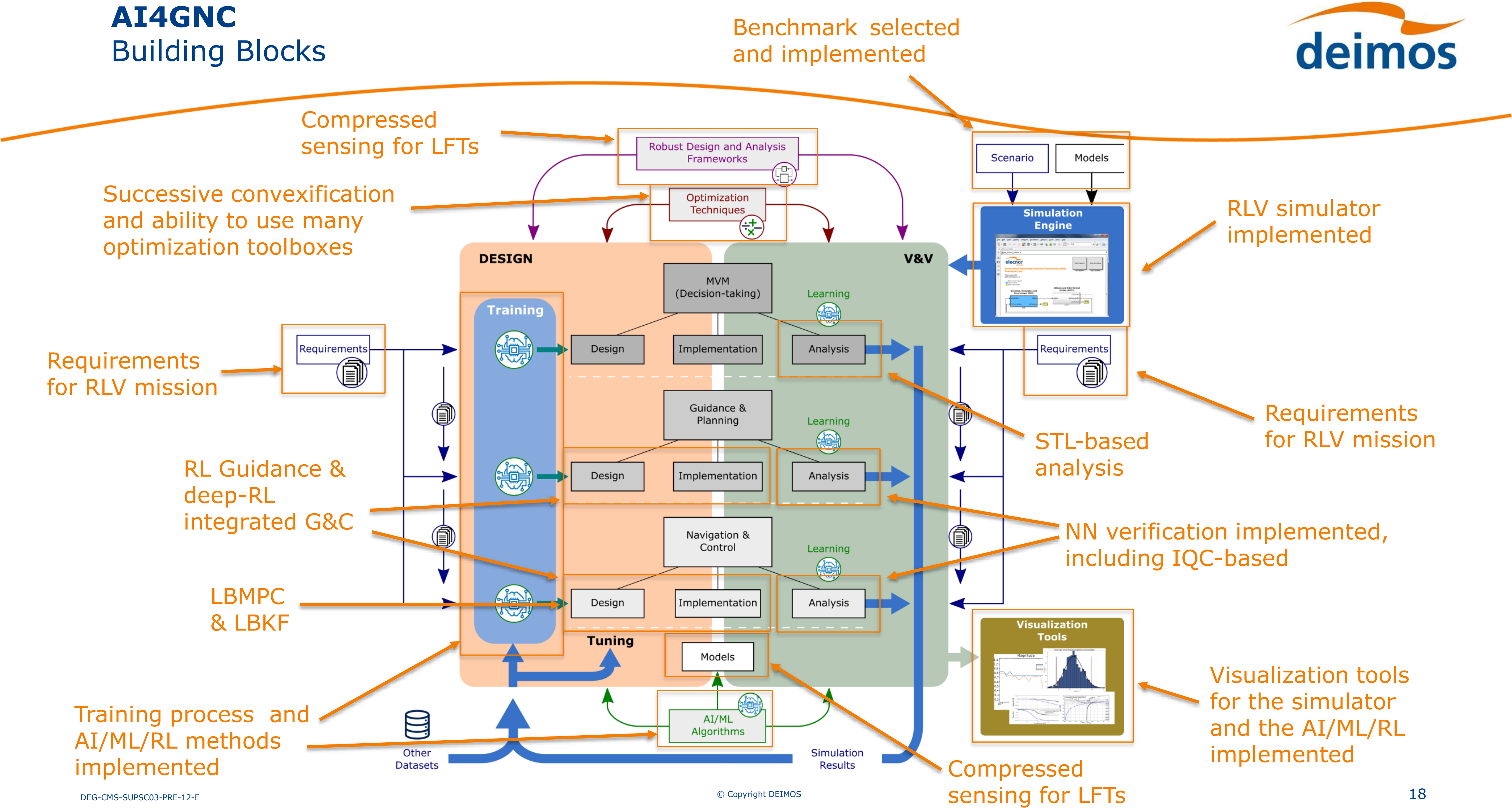
Goal 3: Perform Trade-off analyses

- 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

Goal 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

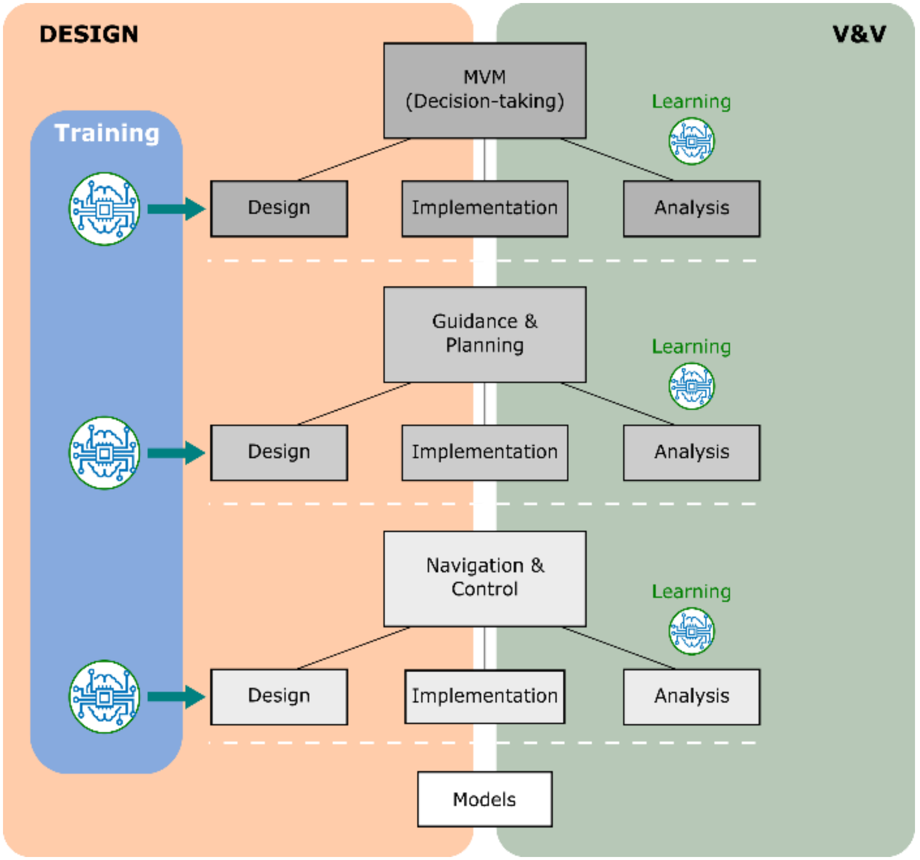
AI4GNC
Building Blocks



ESA-i4GNC (Enhanced Safe AI for GNC) Framework

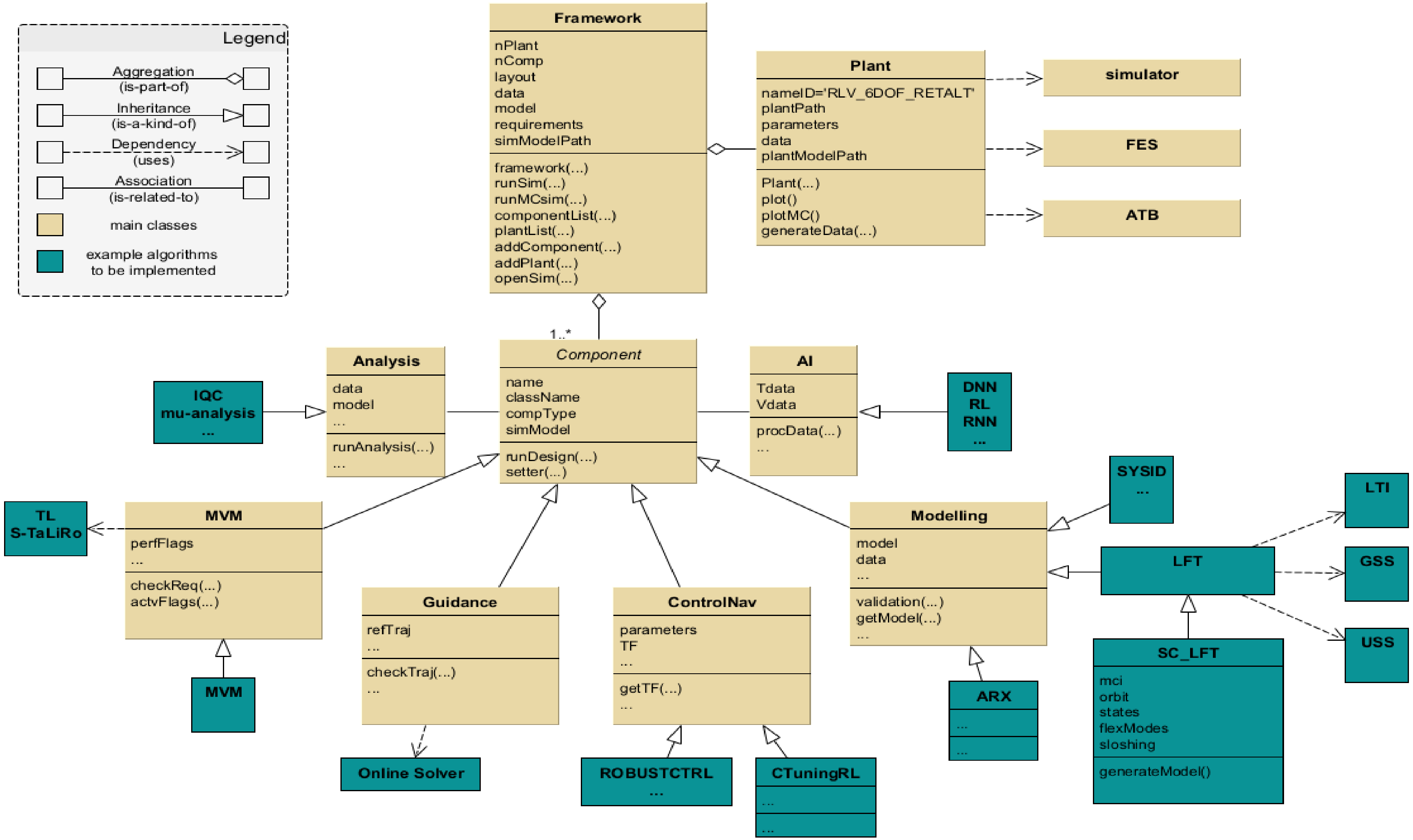
ESA-i4GNC tool

- Implement the architecture in a systematic and structured manner
- Allow the implementation of diverse algorithms: model-based & data-driven
- Requirements satisfaction
- Support and manage models with different levels of fidelity/complexity
- Object-Oriented Programming (OOP)

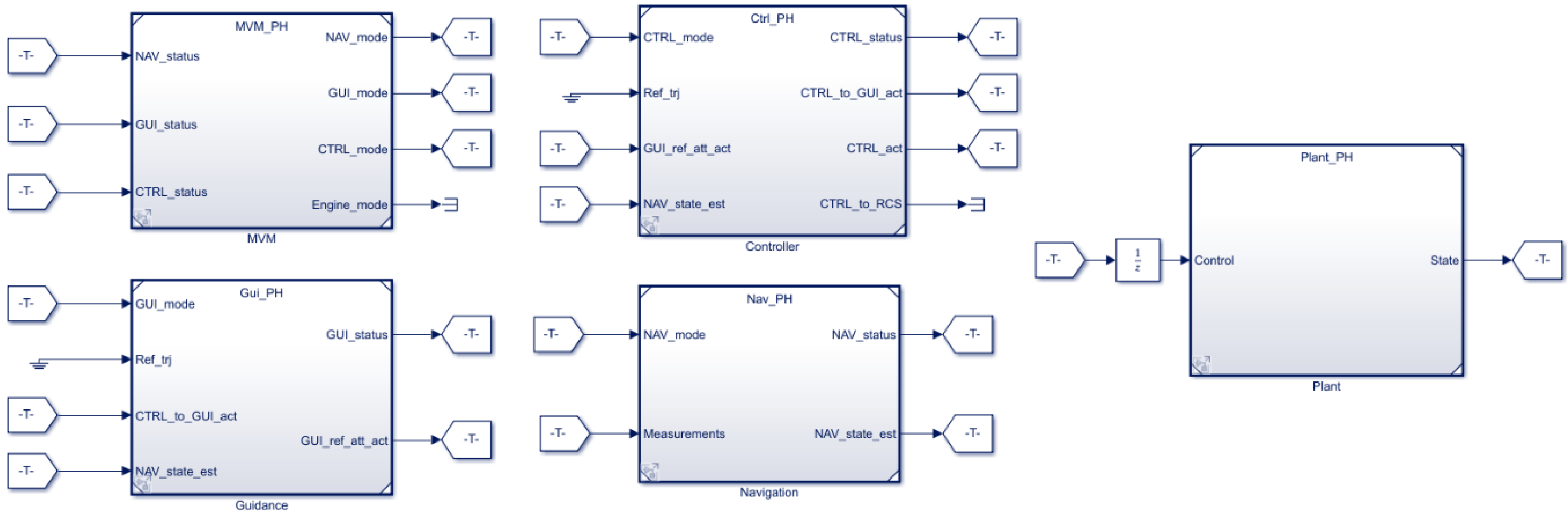


ESA-i4GNC - Design Framework

Framework definition



ESA-i4GNC - Design Framework
Framework definition



Additional functionalities:

- **esai4gnc_install**: installs the tool by adding the necessary files to the MATLAB path
- **esai4gnc_clean**: cleans the tool by removing all the unnecessary files generated while running the tool (cache, slprj,…)
- **autoDoc_ESAi4GNC**: automatically documents the tool using the M2HTML tool, generating HTML files
- **Profiler**: provides the profile execution time when initializing tool and running any simulation

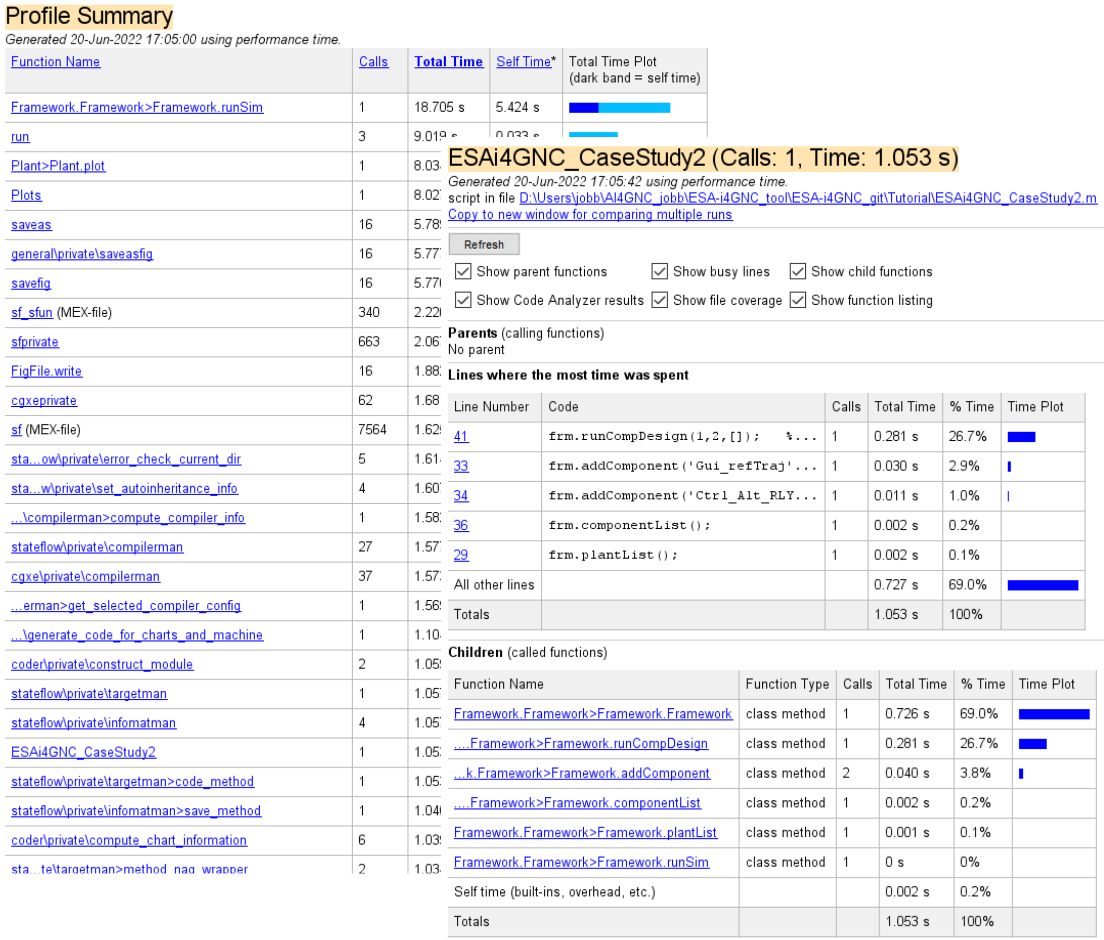
Libraries:

- **CVX**
- **MPT 3.0**
- **S-TaLiRo Runtime Verification**
- **M2HTML**
- **RL Coach**
- Other Python libraries, e.g. **juliacall**

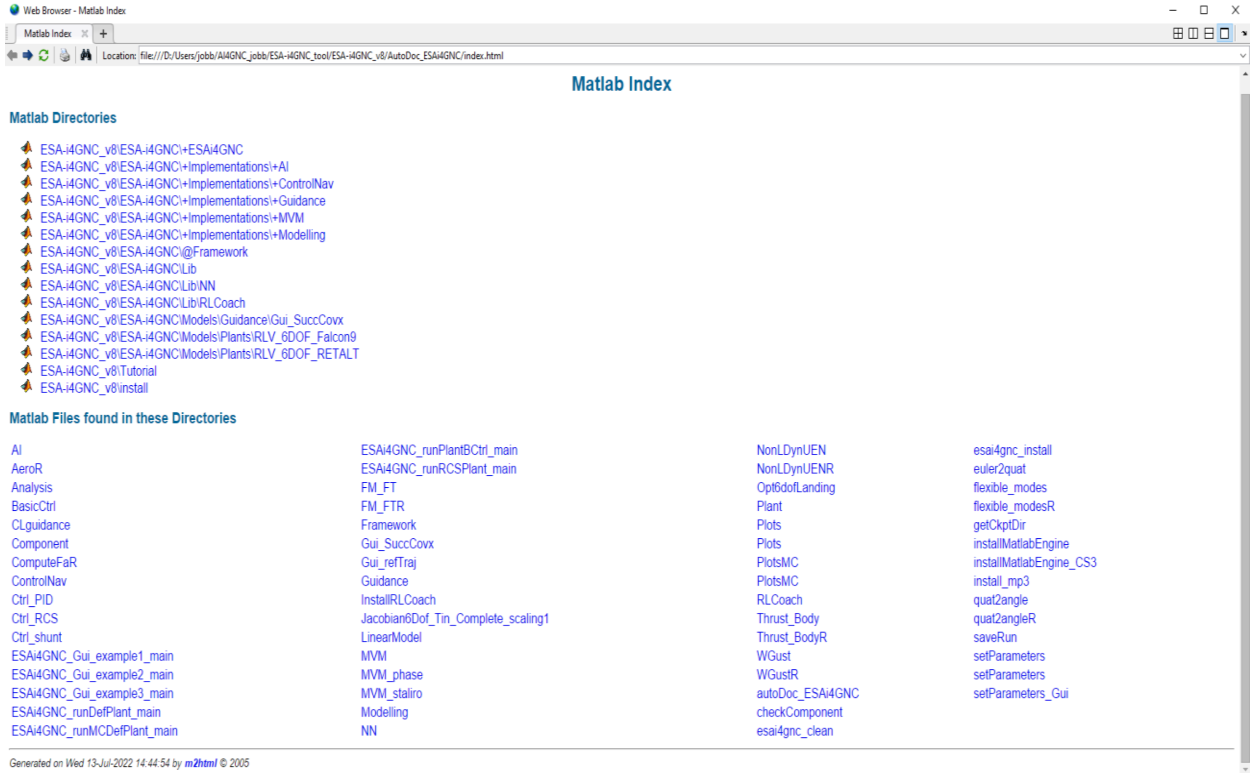
ESA-i4GNC - Design Framework

Examples

Profiler

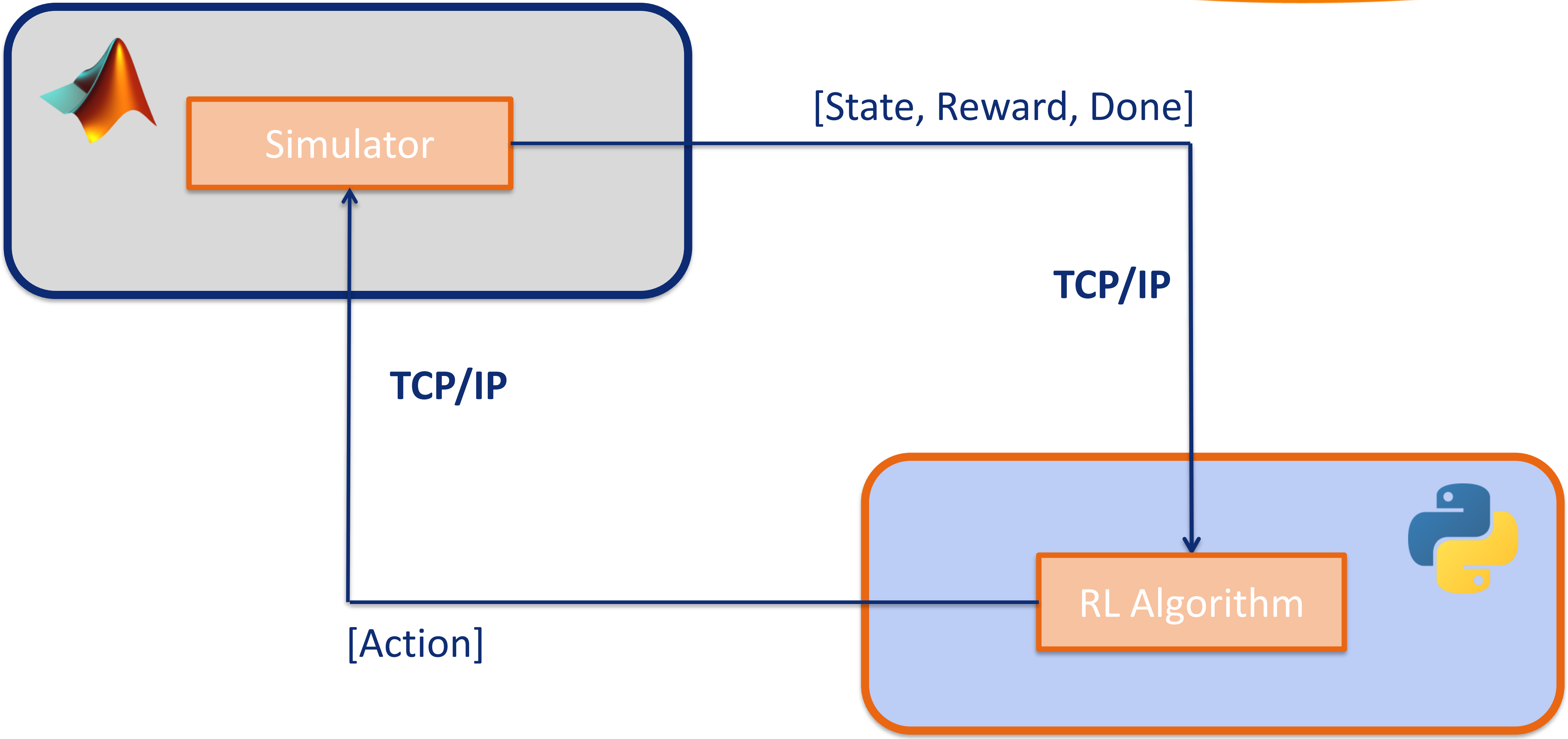


Auto-documentation



ESA-i4GNC - Design Framework

MATLAB-Python interface – RL Python tool

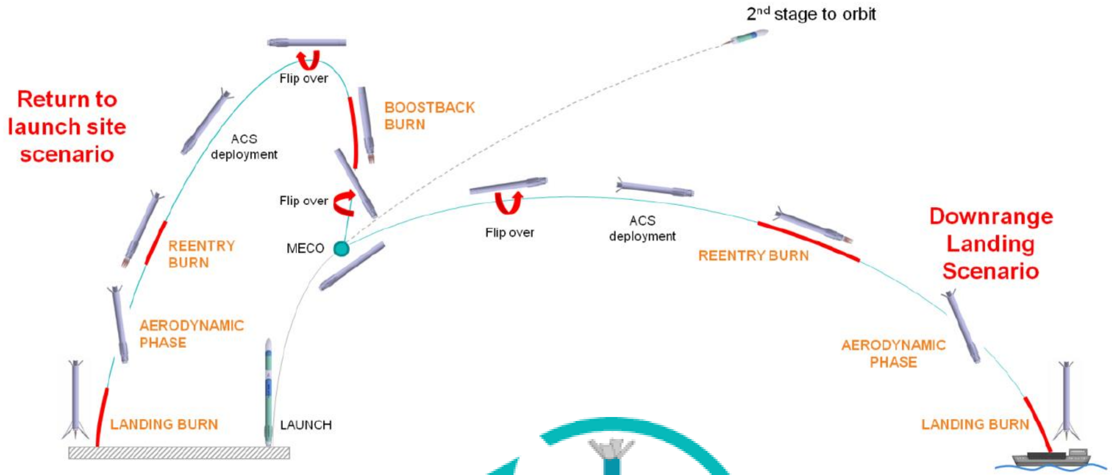


Benchmark

Reusable Launch Vehicle

- Phases of interest:
  - First stage entry
  - **Descent and precision landing**

- Focus was given to the demonstration of the techniques and the ESA-i4GNC tool
- Parameters from Falcon 9 and RETALT RLV, although the techniques are applied to RETALT RLV:
  - **Wind model**
  - **Aerodynamic model**



RETALT
RETro propulsion Assisted Landing Technologies

https://www.retalt.eu/

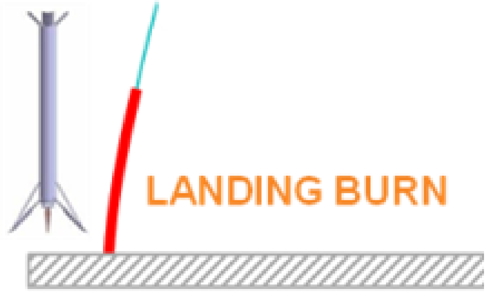
Benchmark
Selected Benchmark

- 6 DOF **Landing Burn Scenario** of a Reusable Launcher Vehicle (**RETALT**)
  - Realistic Aerodynamics DB
  - Actuator (TVC) model
  - Wind model
  - Flexible modes

- **Baseline GNC**:
  - SCVX guidance
  - Ideal navigation
  - PID controller

Initial Conditions

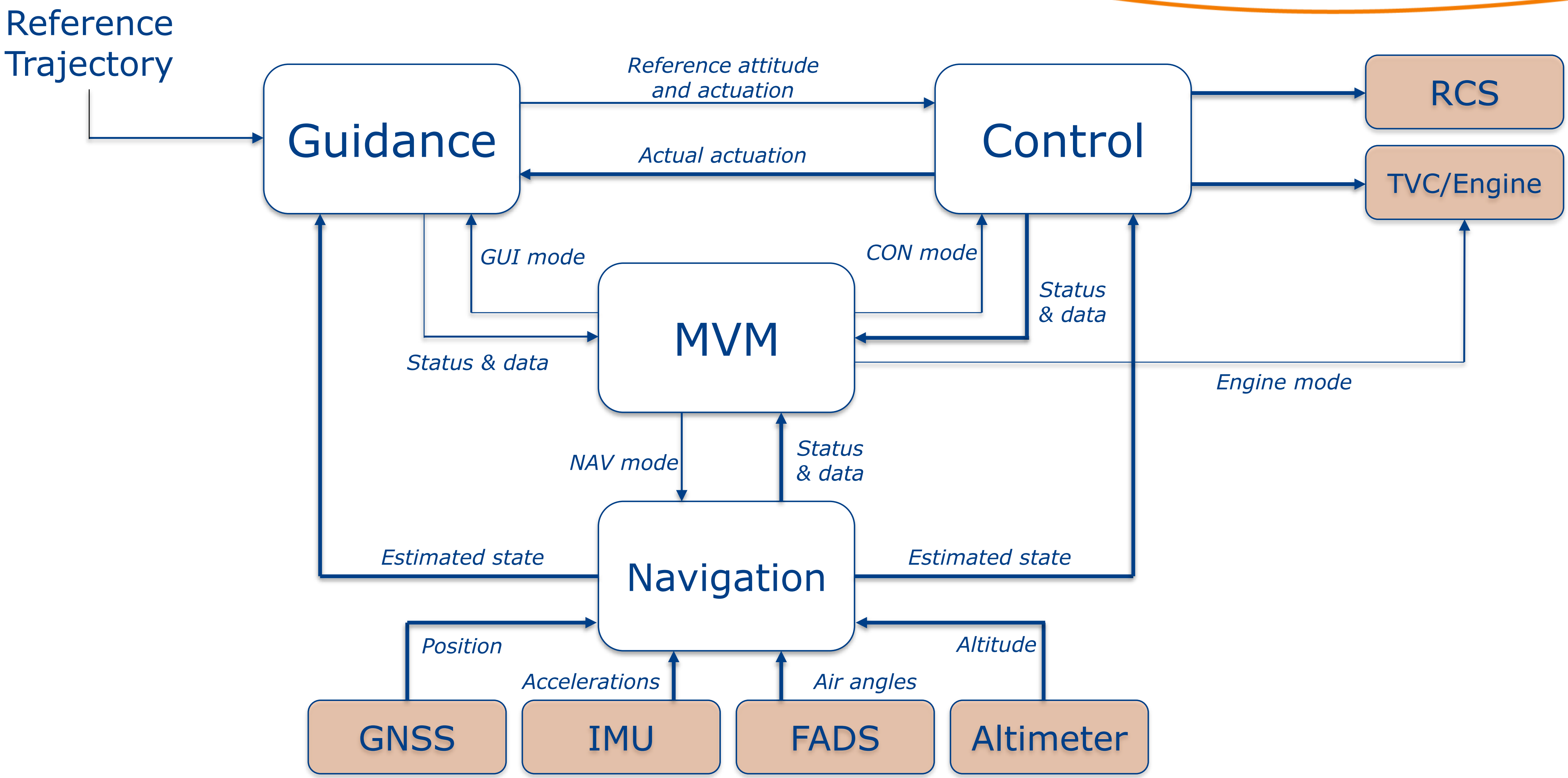
Mass [kg]	Position [m]	Velocity [m/s]	Attitude [q]	Angular vel. [rad/s]
80334	[2874,-1288, -82.2]	[-189.9, 151.3, 9.6]	[0.943, 0.006, 0.018, -0.329]	[0, 0, 0]



1st Stage features	Value
Height [m]	71.2
Diameter [m]	6
Dry mass [kg]	59300
Propellant mass (incl. descent propellant) [kg]	621500
Specific Impulse SL [s]	401.6
Thrust SL [kN]	11453

RETALT
RETro propulsion Assisted Landing Technologies

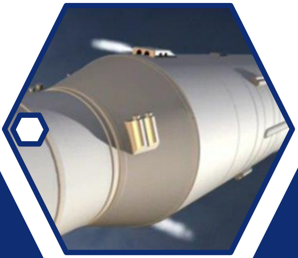
Benchmark
Proposed GNC Architecture



RLVs Challenges

To Support the Case Studies Definition

deimos

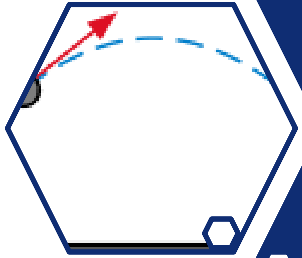


RCS failure tolerance

Main engine re-ignition

Center of gravity estimation

Aerodynamic fin failure detection



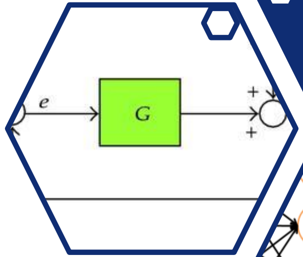
Attitude control

Trajectory generation for the landing phase

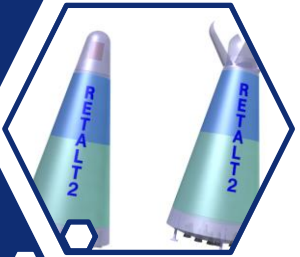
Aerodynamic fin failure tolerance

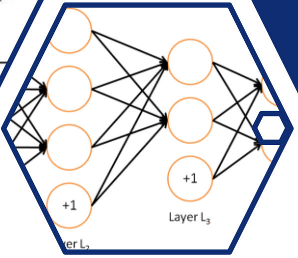
MCI estimation

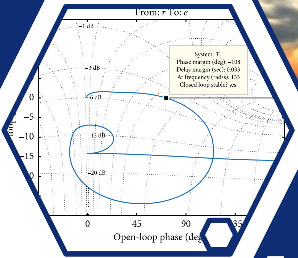
Thruster misalignment



Trajectory tracking for the aerodynamic phase







Fuel sloshing

Inertia estimation during the flip over before the boostback burn

Case Studies

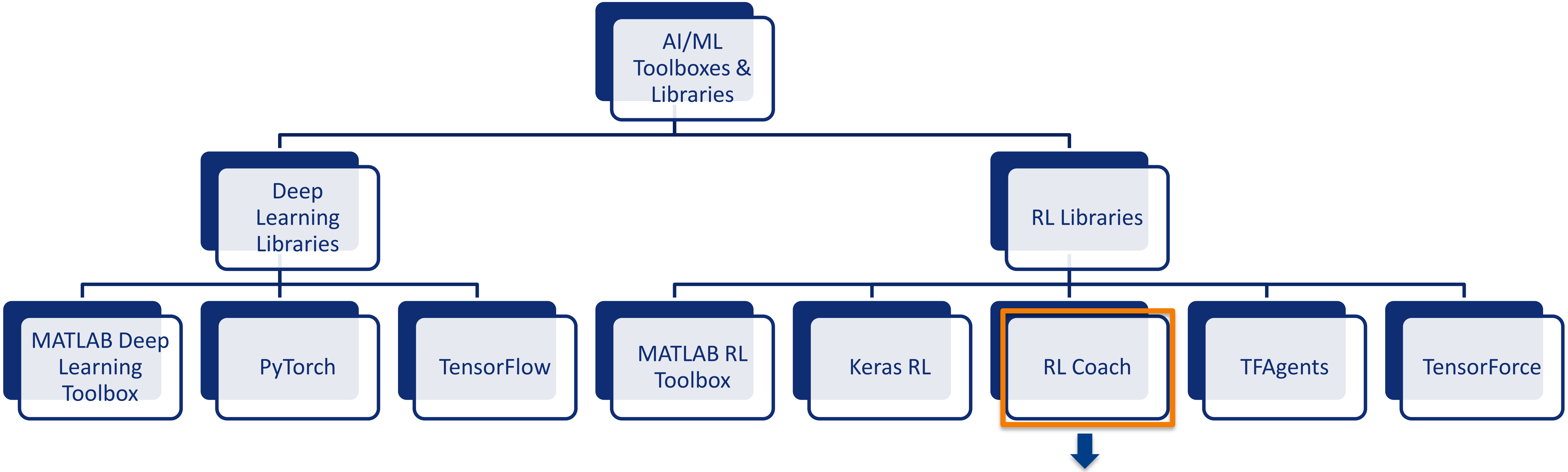
Case Studies
Definition of Baseline Case Studies

Case Study #	Description
1	RL-based adaptive control to regulate the attitude in response to disturbances
2	RL-based adaptive control to regulate the trajectory in the aerodynamic phase with respect to the reference trajectory
3	NN approximation of the QUEST algorithm for three axis attitude estimation from vector observation data
4	Sparse regression, compressed sensing, compressed learning and potential connections with LFT modelling
5	Learning-based model predictive control (LBMPC) for attitude control
6	IQC formalism for NN-based attitude control verification
7	Learning-based Kalman filtering for attitude estimation
8	Deep RL for trajectory tracking

In this presentation…

Deep RL for trajectory tracking (Case Study #8)

Trade-offs
Learning approaches

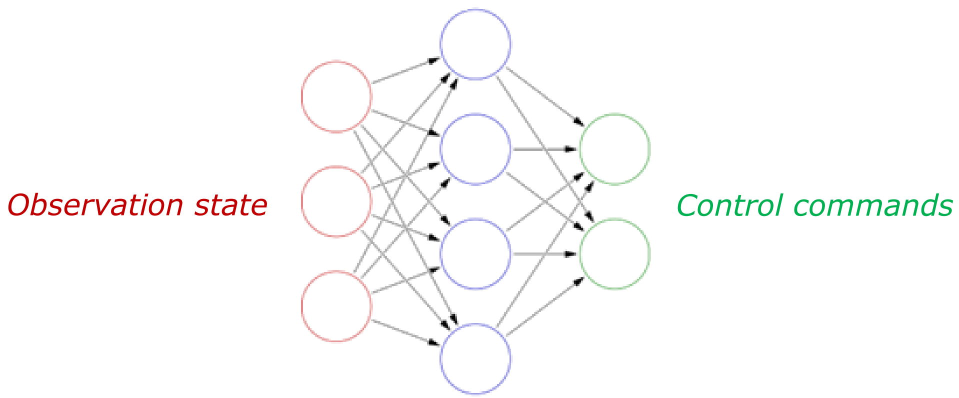


- Several algorithms available
- Well documented
- **Open Source**

Case Study #8
Overview

<u>GOAL</u>: Implementation of a **Reinforcement Learning** technique to address the **G&C** problem for the landing phase of a **RLV**

- On-board solution
- Address non linearities of the RLV dynamics
- Map sensors measurements to action commands

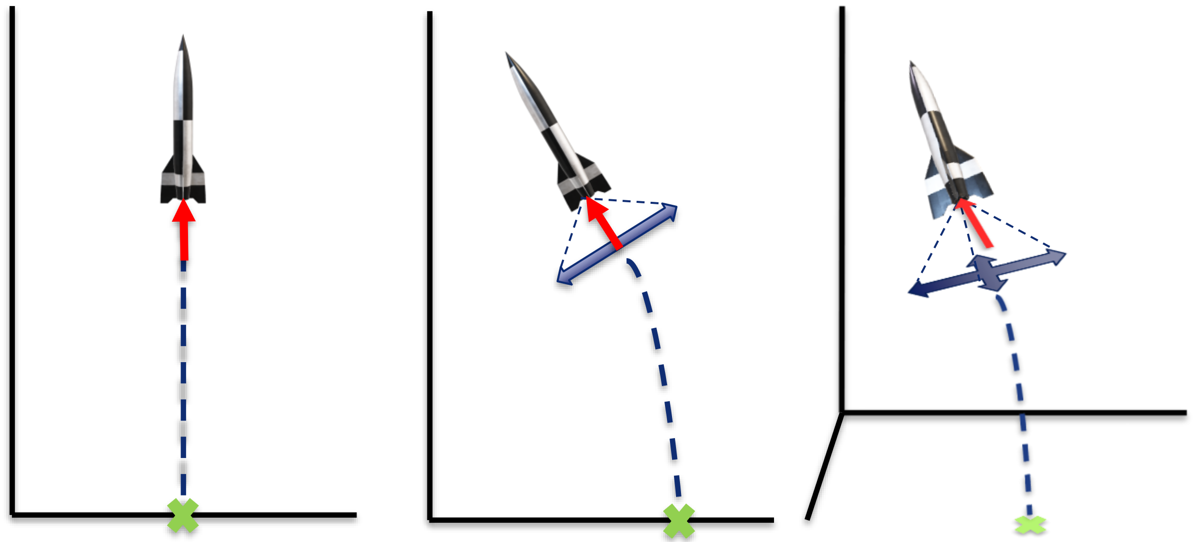


<u>Implementation</u>:

- RL Coach toolbox
- Agent: DDPG (Actor – Critic)
- Environment: RETALT landing simulator

<u>Approach</u>:

- Incremental 1D ➞ 2D ➞ 3D



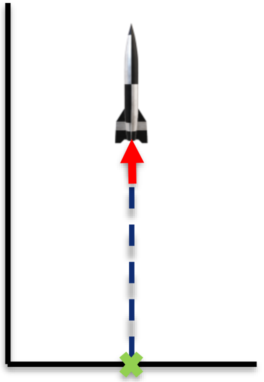
Case Study #8

1 DoF scenario, simple case

1D vertical landing problem →

Action = [Thrust]

Observation = [altitude, velocity]



Reward 1 : $-|v_x - v_t| - 1e^{-10} \cdot Thrust + 30 \cdot (x < 0.1 \,\&\&\, |v_x| < 2) + 80 \cdot (x < 0.1 \,\&\&|v_x| < \; 0.5) - 1000 \cdot (x > x_0) - 50 \cdot (v_x > 3)$

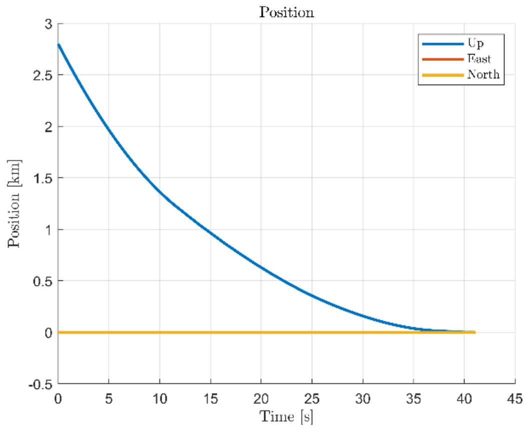
Penalization on target velocity

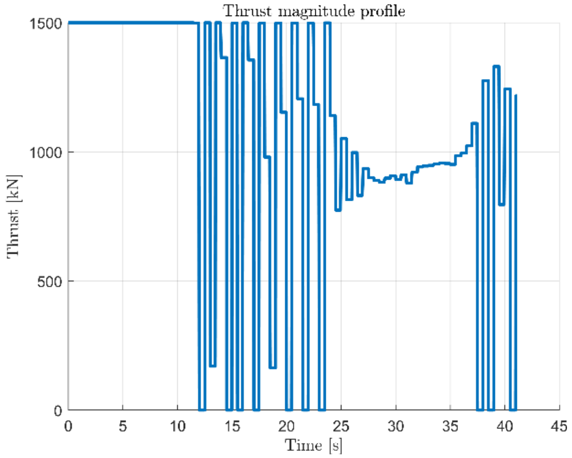
Penalization on thrust

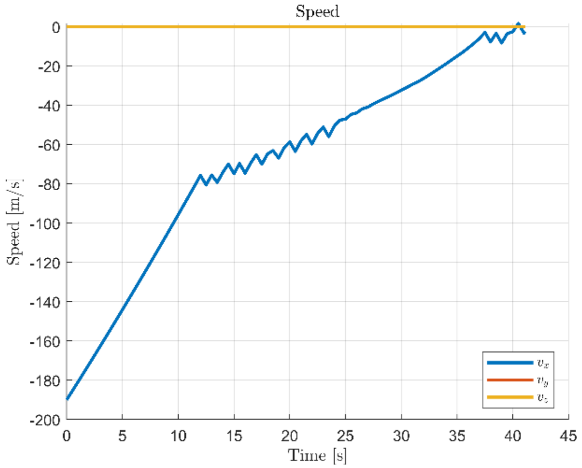
Boolean conditions on landing conditions for positive reward

$$\mathrm{v_t} = v_0\left(1 - e^{-t_{go}/\varepsilon}\right)$$

$$t_{go} = x/v_x$$



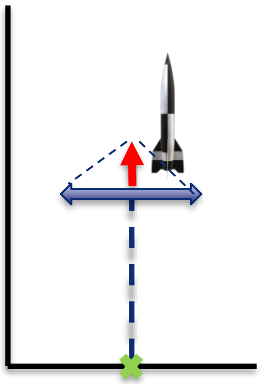




Case Study #8
2 DoF vertical scenario

2D vertical landing problem → Action = [Thrust, gimbal angle]

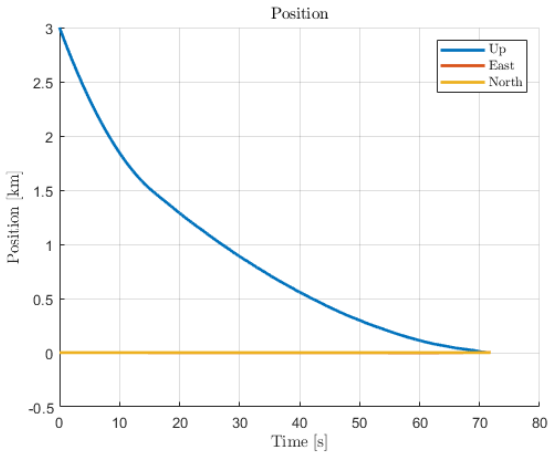
observation = [altitude, $(v-v_t)_{vert}$, $(v-v_t)_{horiz}$, yaw, $omega_{yaw}$]

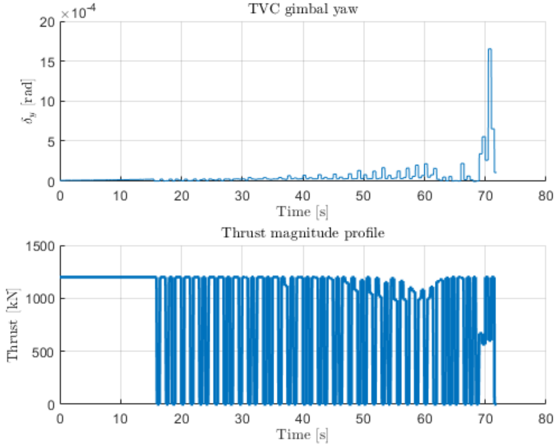


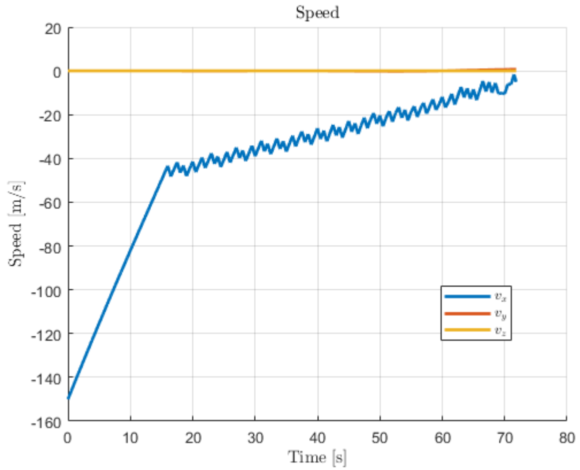
- Hyperparameters setting

Heat-up steps	Training steps	Steps btw evaluations	Actor/Critic Learning rate	Actor/Critic Batch size	γ	τ/Cw
5000	120000	10	0.0005/0.005	68/68	0.9	0.1/1000

Reward : $\underbrace{-|v_x - v_t|}_{\text{Penalization on target velocity}} \underbrace{- 1e^{-7} \cdot Thrust}_{\text{Penalization on thrust}} + \underbrace{100 \cdot (x < 0.1 \,\&\&\, |v_x| < 2) + 200 \cdot (x < 0.1 \,\&\&|v_x| < 0.5)}_{\text{Boolean conditions for positive reward}} \underbrace{- 100 \cdot (x > x_0) - 100 \cdot \left(|yaw| > \frac{\pi}{2}\right) - |gimbal_{deg}|}_{\text{Boolean conditions for negative reward}}$

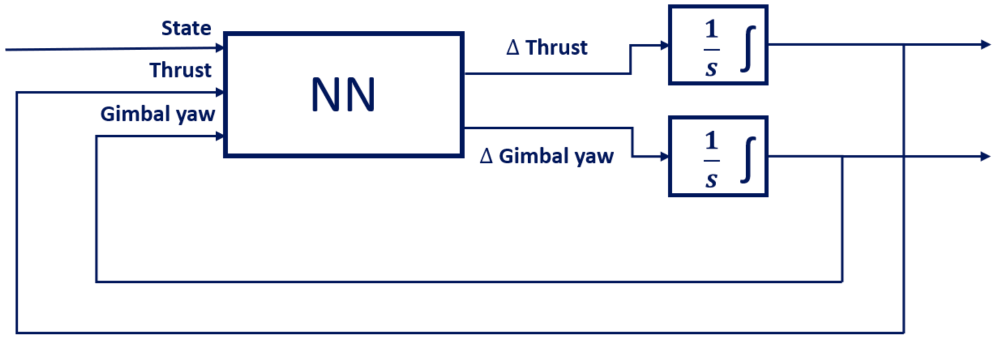


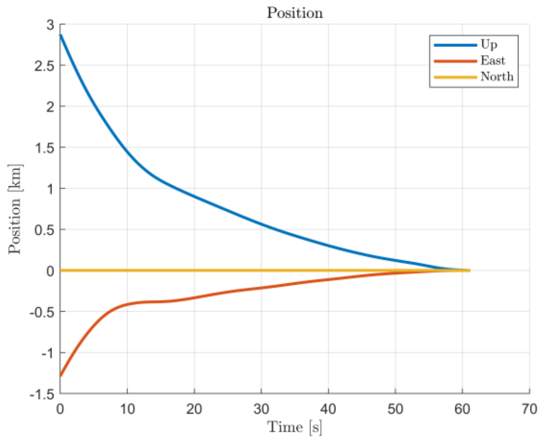


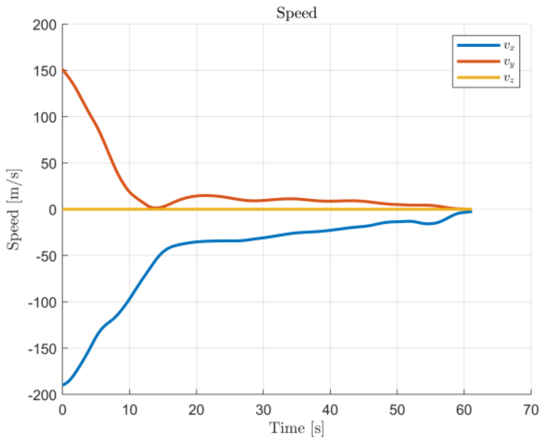


Case Study #8

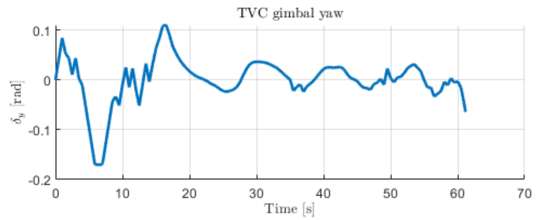
Application of the Techniques – 2D with **rate limiters**

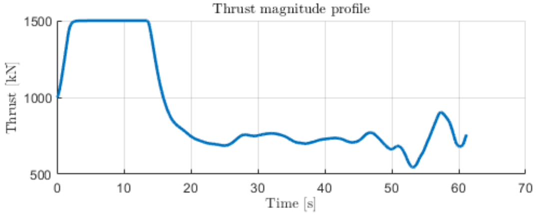


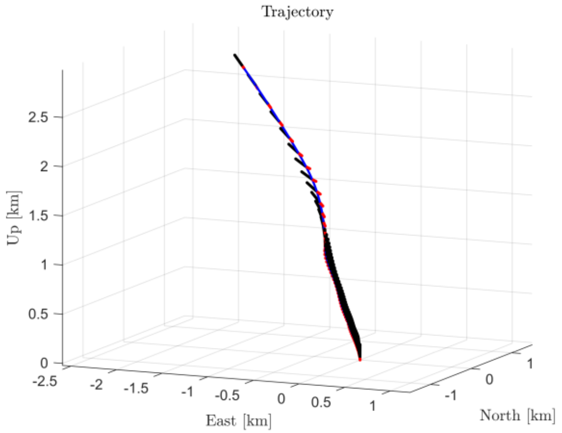




- Control the action rates instead of the action values
- The rates are integrated and feedback as input of the NN
- Initial conditions, upper and lower bounds for the integrators
- Penalization when the rate leads to violation of the action bounds



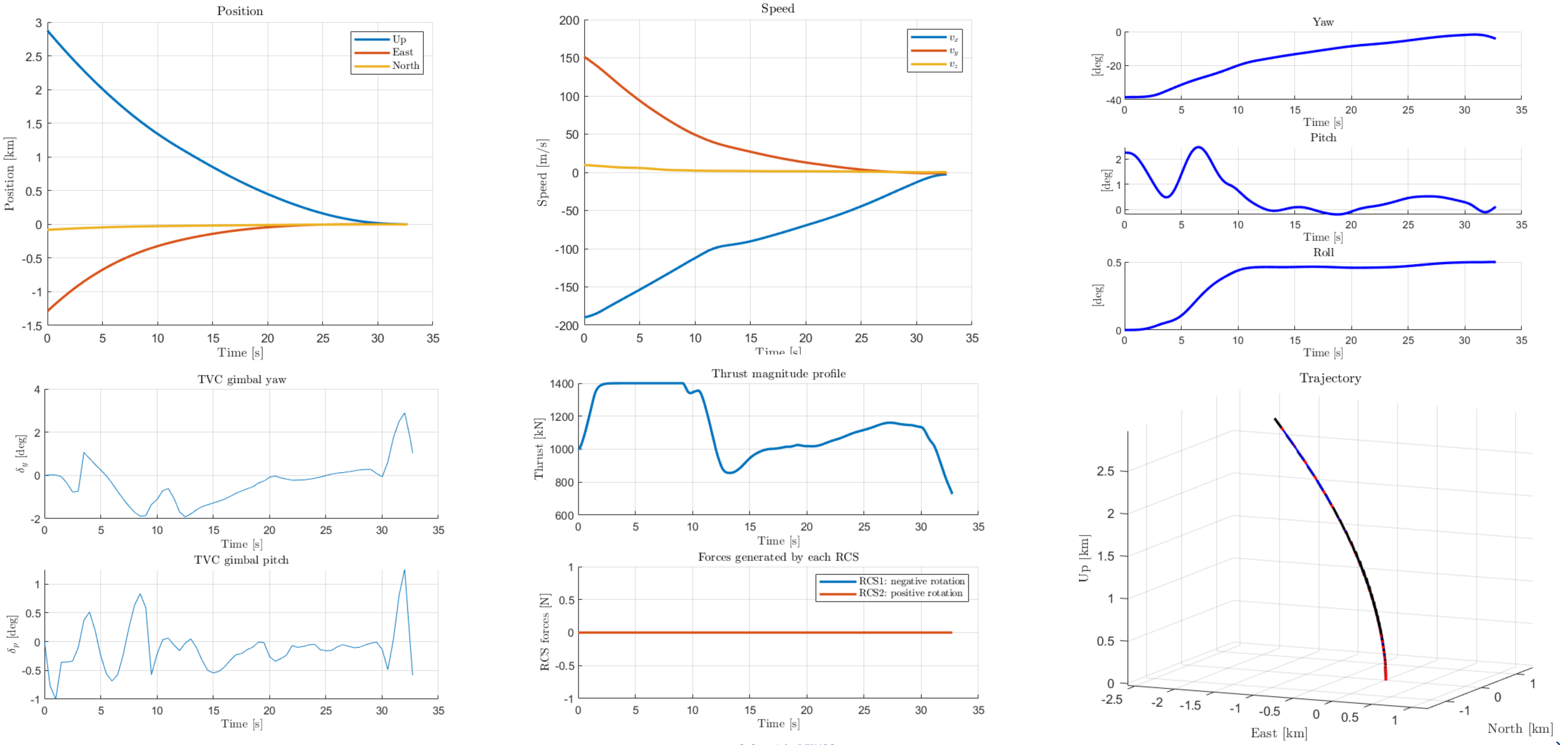




$$R = -|v - v_t| - 0.1 \cdot \left((T_c + \Delta t \cdot \Delta T) > T_{max}\ or\ (T_c + \Delta t \cdot \Delta T) < 0\right)\ \dots$$

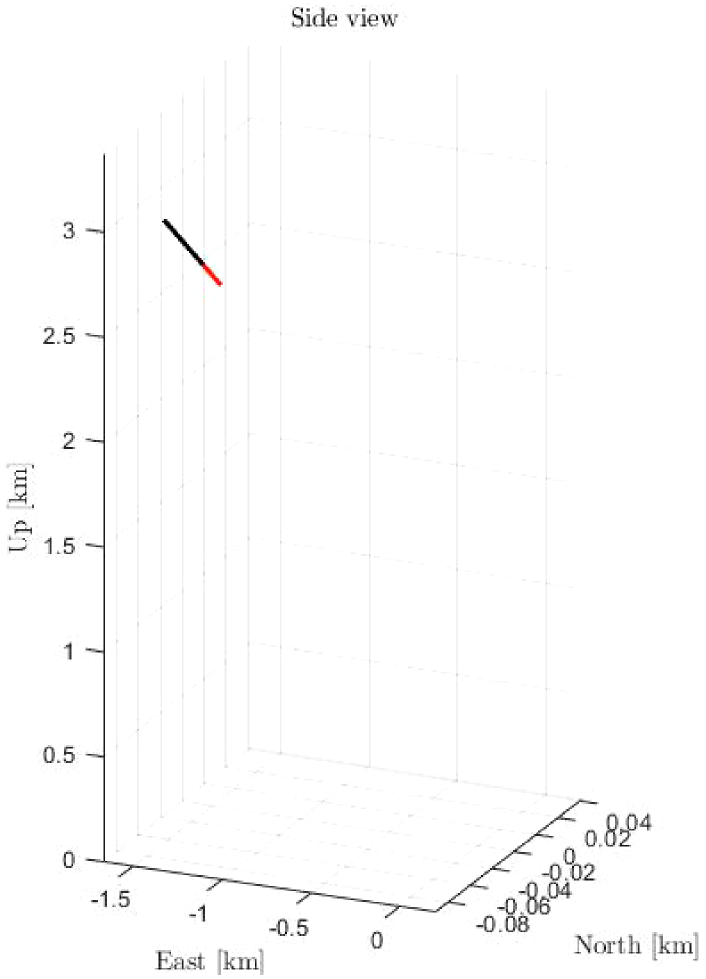
Case Study #8

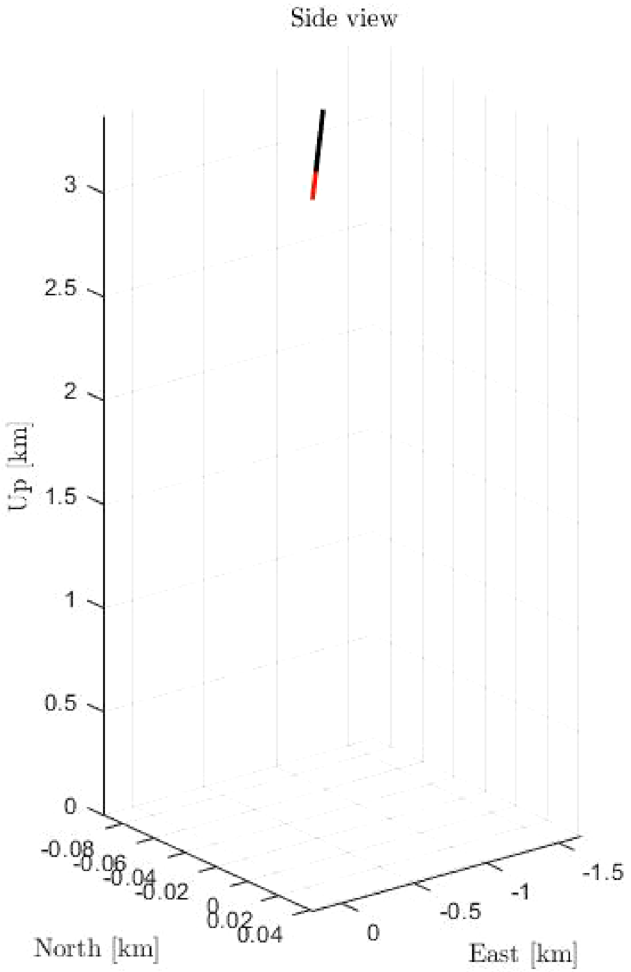
Application of the Techniques – 3D with **rate limiters**

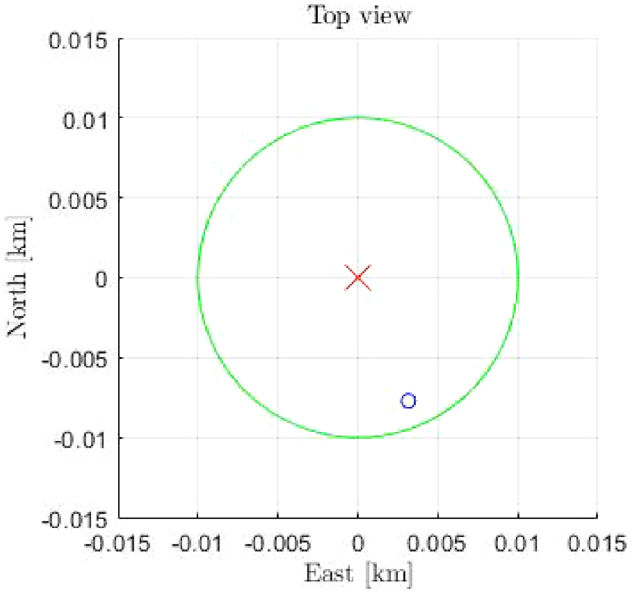


Case Study #8
Animation comparison

Performance animation when considering the SCVX guidance:

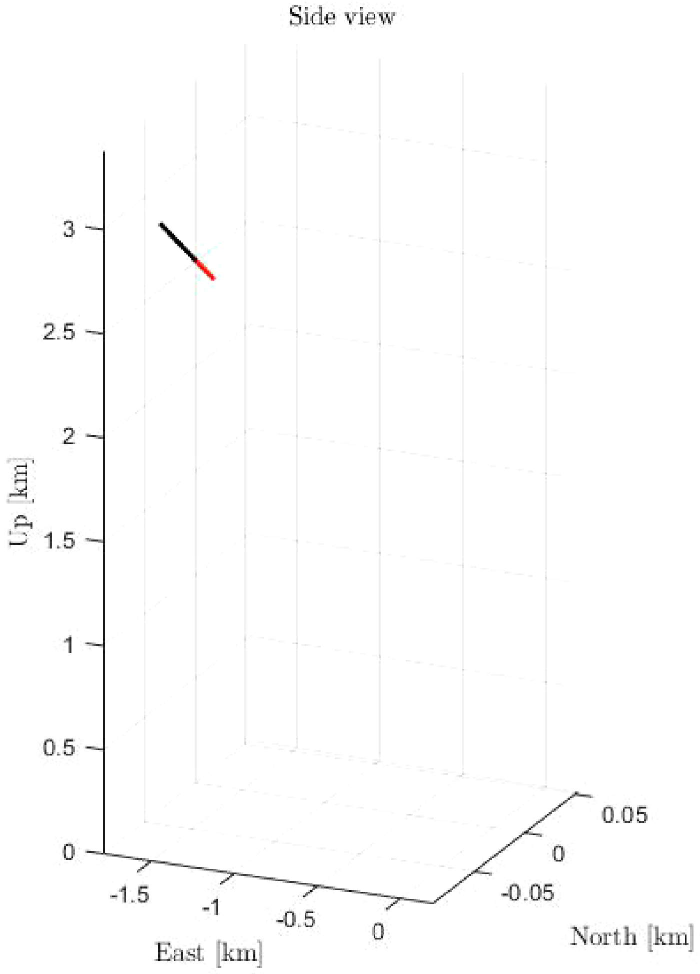


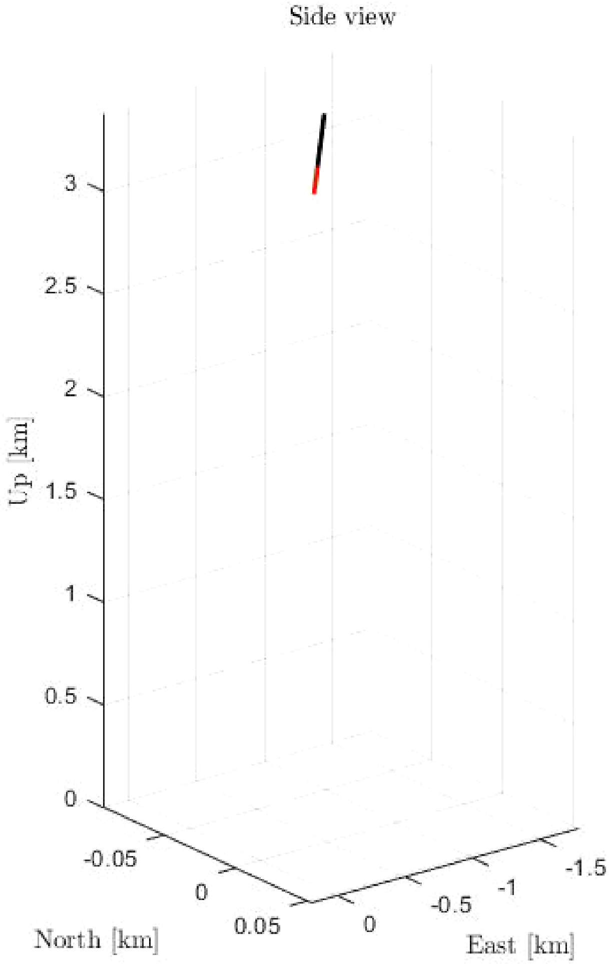


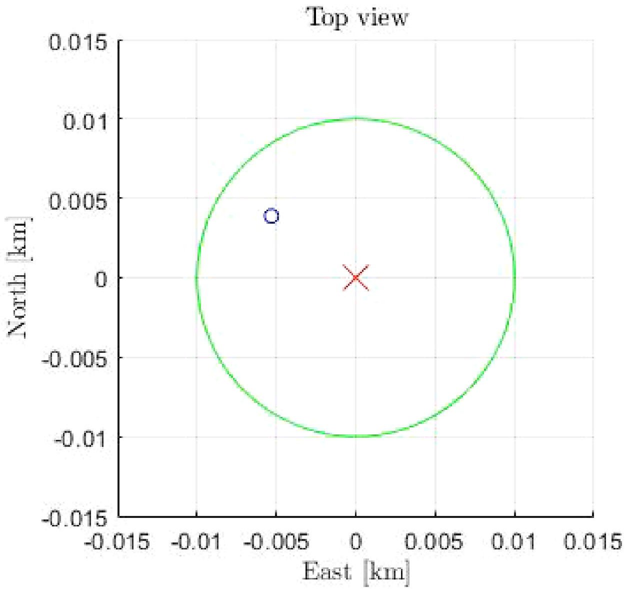


Case Study #8
Animation comparison

Performance animation for the initial NN obtained:

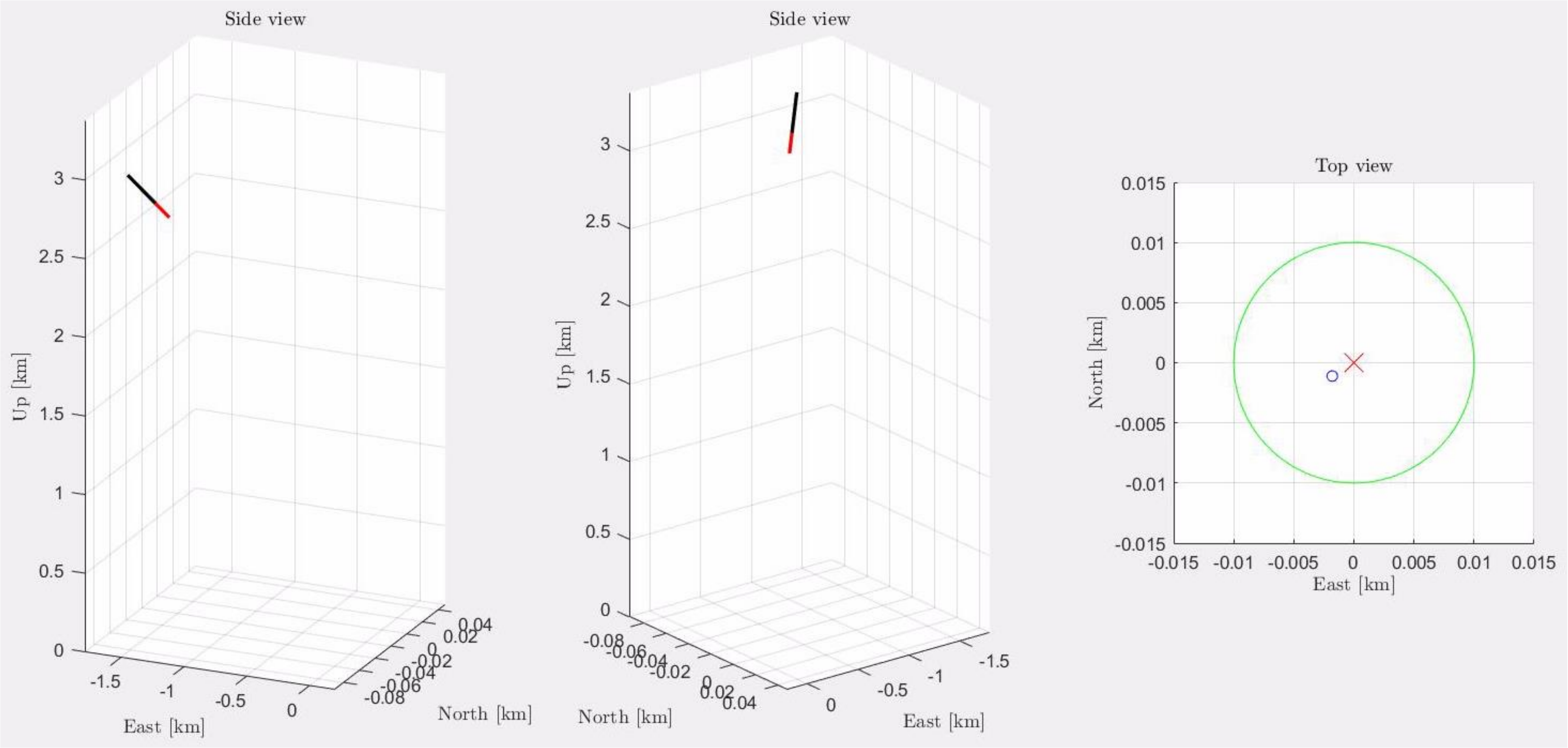




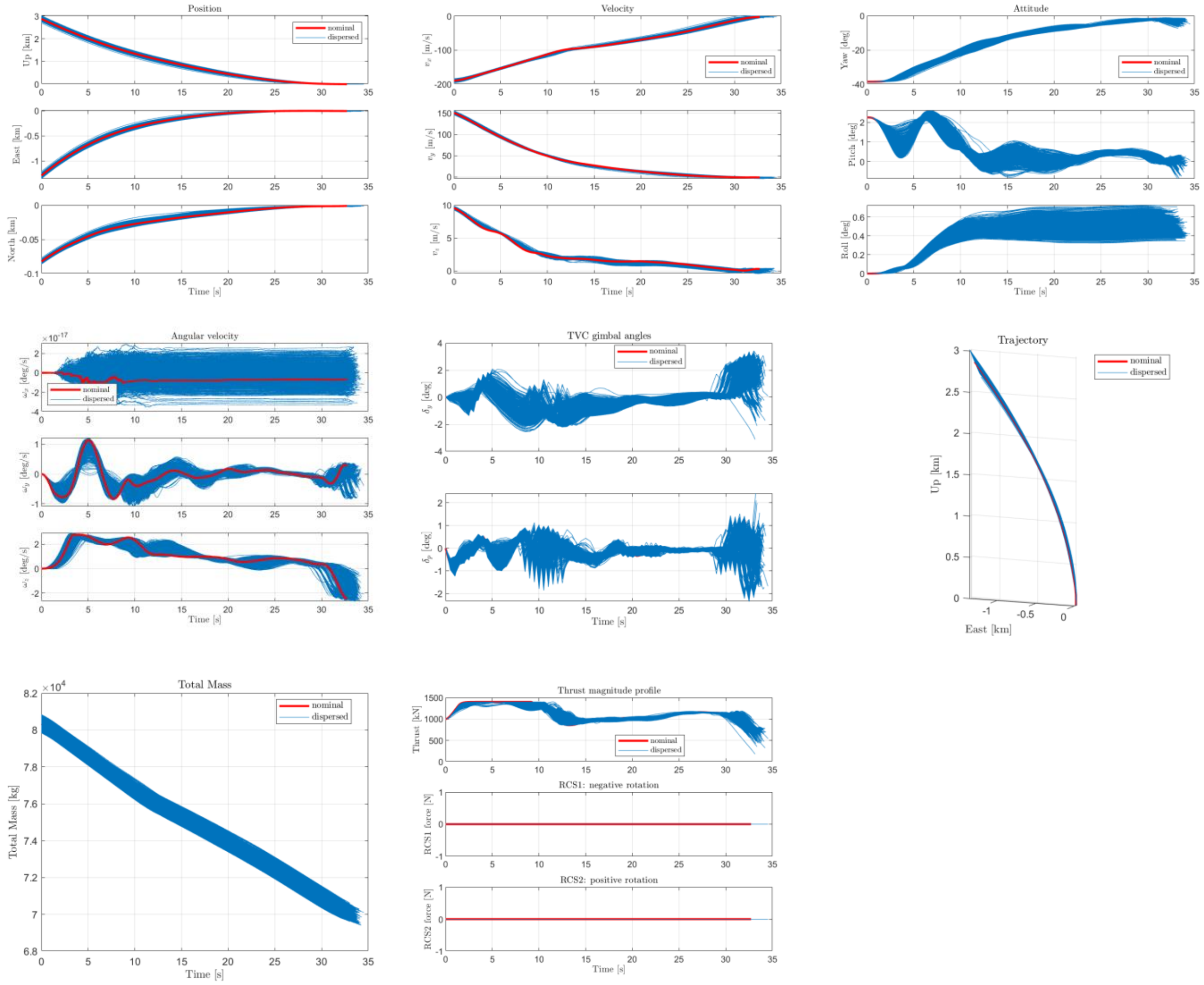


Case Study #8
Animation comparison

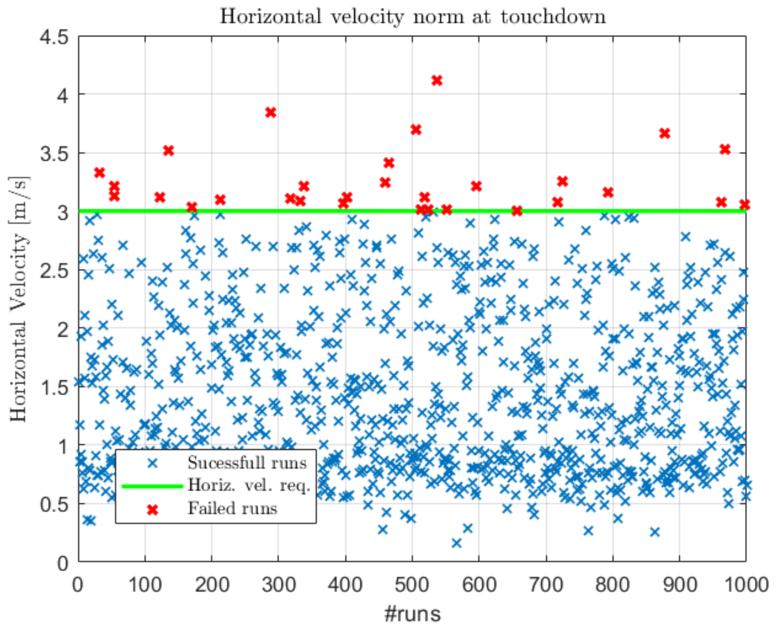
Performance animation for the final NN obtained:



Case Study #8
Verification results



- **97 %** success rate (970 out of 1000)
- All the failed runs are related to the horizontal velocity
- Error always below 1.5 m/s



Case Study #8
MC Comparison Analysis

- **NN** yields better **position accuracy**
- **SCVX** guidance re-computed each 5 sec better **velocity accuracy**

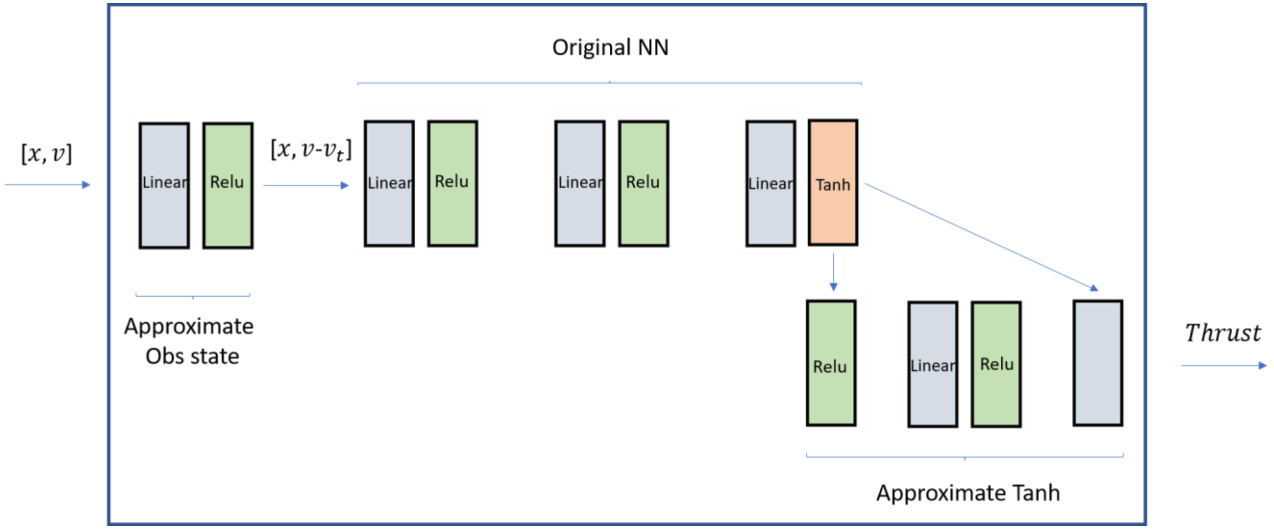
This indicates we are reaching a Pareto optimal set, with Deep-RL being a remarkable solution to the problem

MC Simulation	Quantity	Mean	Standard Deviation
Neural Network	Fuel consumption [kg]	10304.019	158.48
	Position accuracy [m]	3.957	1.4181
	Velocity accuracy [m/s]	2.89694	0.44976
SCVX, 5s	Fuel consumption [kg]	9763.39	69.13
	Position accuracy [m]	8.7947	0.40740
	Velocity accuracy [m/s]	1.54723	0.36812

Verification of the NN

Case Study #8
Neural Network V&V with nn_robustness_analysis

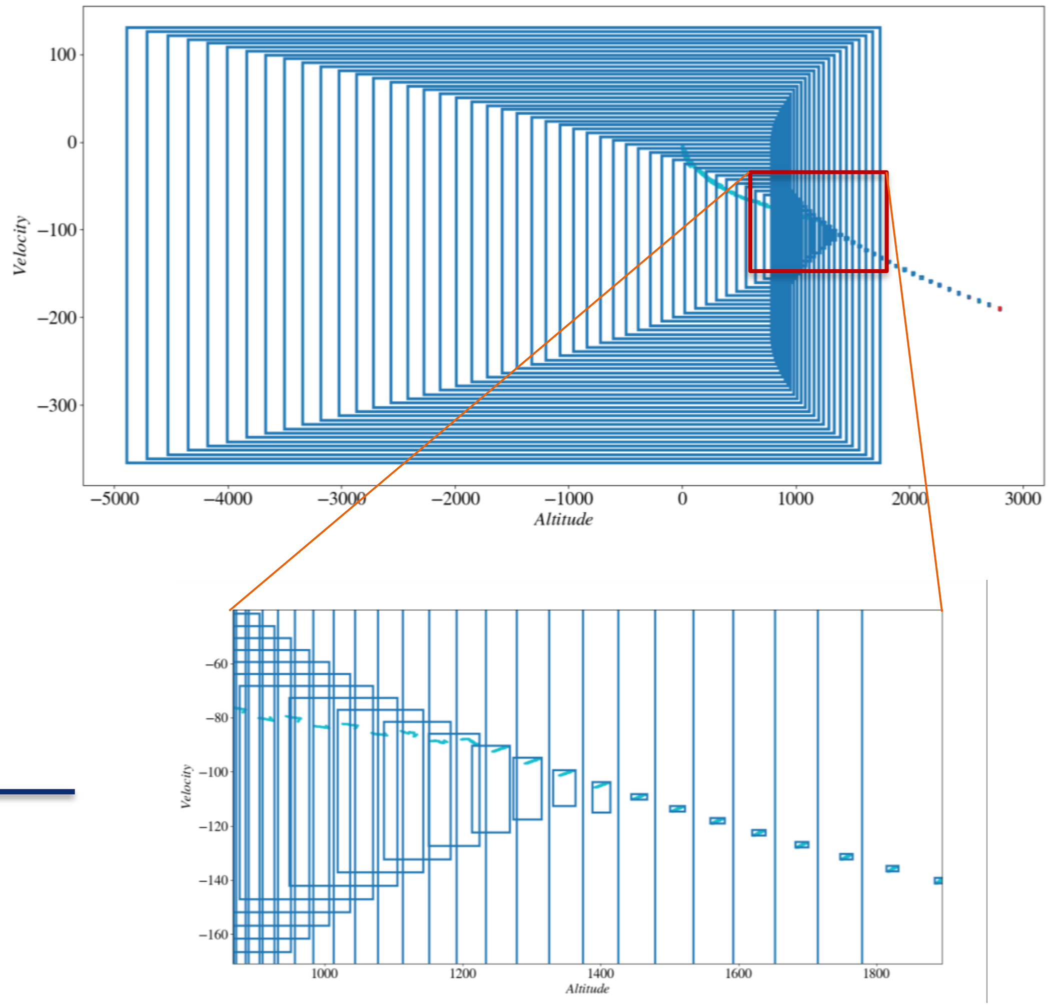
- NN trained for the 1D scenario
- Original NN presented: $[x.v - v_t] \longrightarrow [Thrust]$
  - NN trained to learn the observation state
  - NN trained to learn *tanh* activation fun.
- Discrete time dynamics implemented
  - $s(t+1) = \boldsymbol{A}\, s(t) + \boldsymbol{B}\, u(t) + \boldsymbol{C}$
- Greedy Simulator Guided partitioner
- CROWN propagator

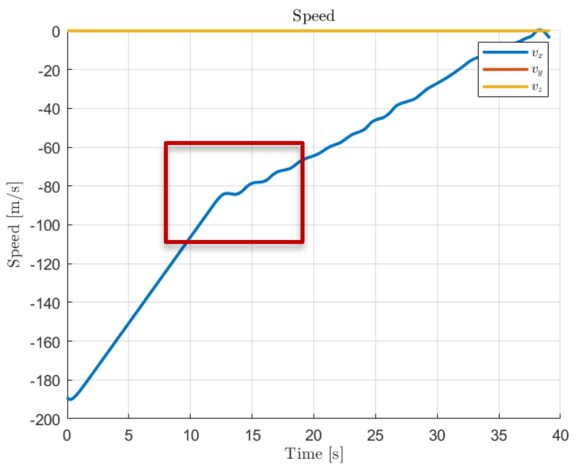


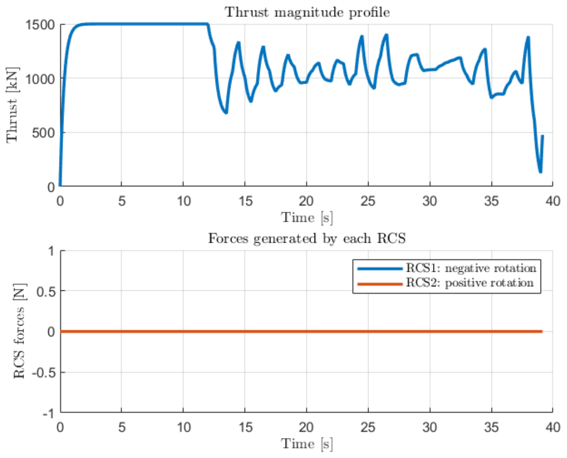
Concatenated NN

Case Study #8
Neural Network V&V: Preliminary Results

- Blue dots: Dynamic propagation
- Boxes: Reachable set
- 200 propagator calls



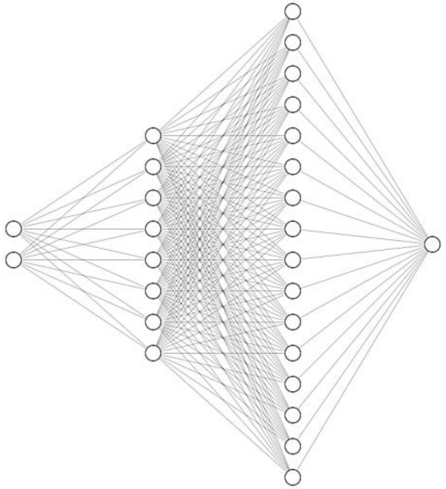


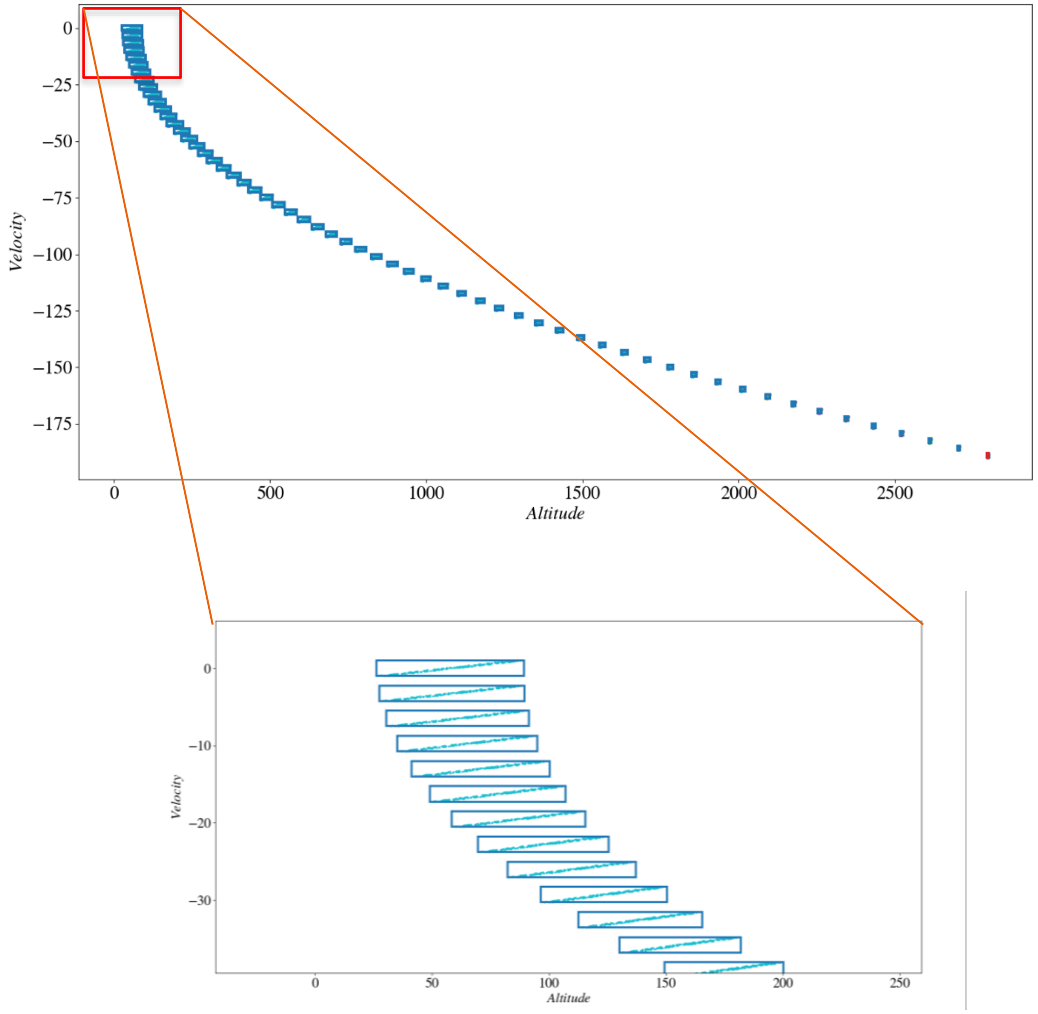


Case Study #8
Neural Network V&V: Final Results

- Smaller NN trained through supervised learning with the previous concatenated NN

Layer Name	Layer Type	Activation Function	Input size	Output Size
Input Layer	Linear	ReLU	**2**	**16**
Hidden Layer	Linear	ReLU	**16**	**32**
Output Layer	Linear	None	**32**	**1**



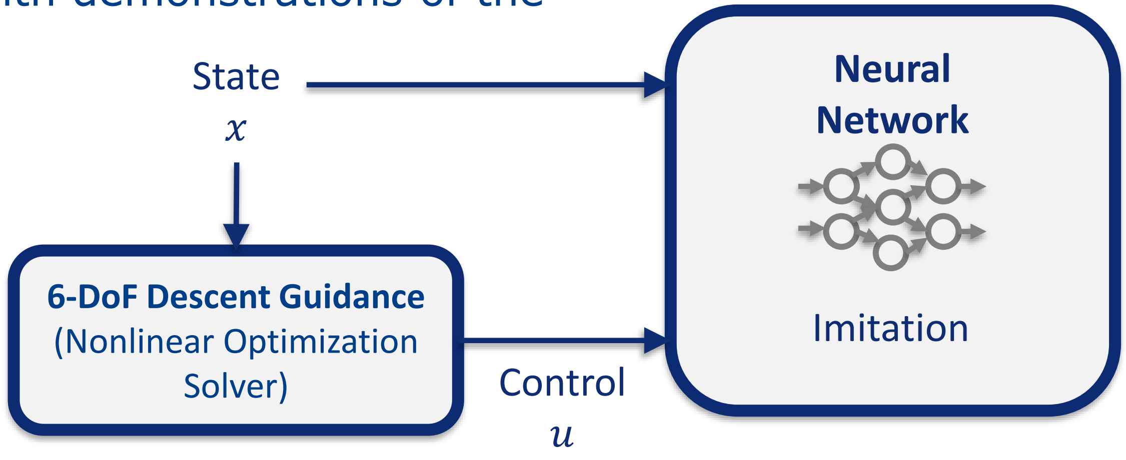


Alternative to Deep RL:

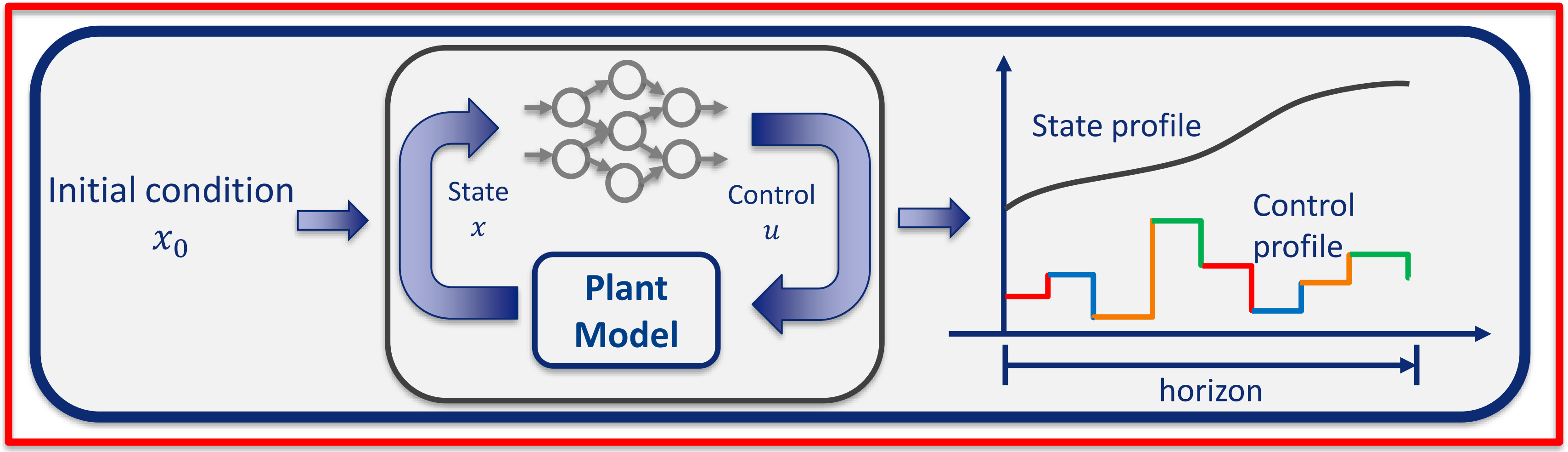
Supervised Learning

Supervised Learning Approach

- **Description**: Supervised Learning approach to train a NN with demonstrations of the optimal guidance

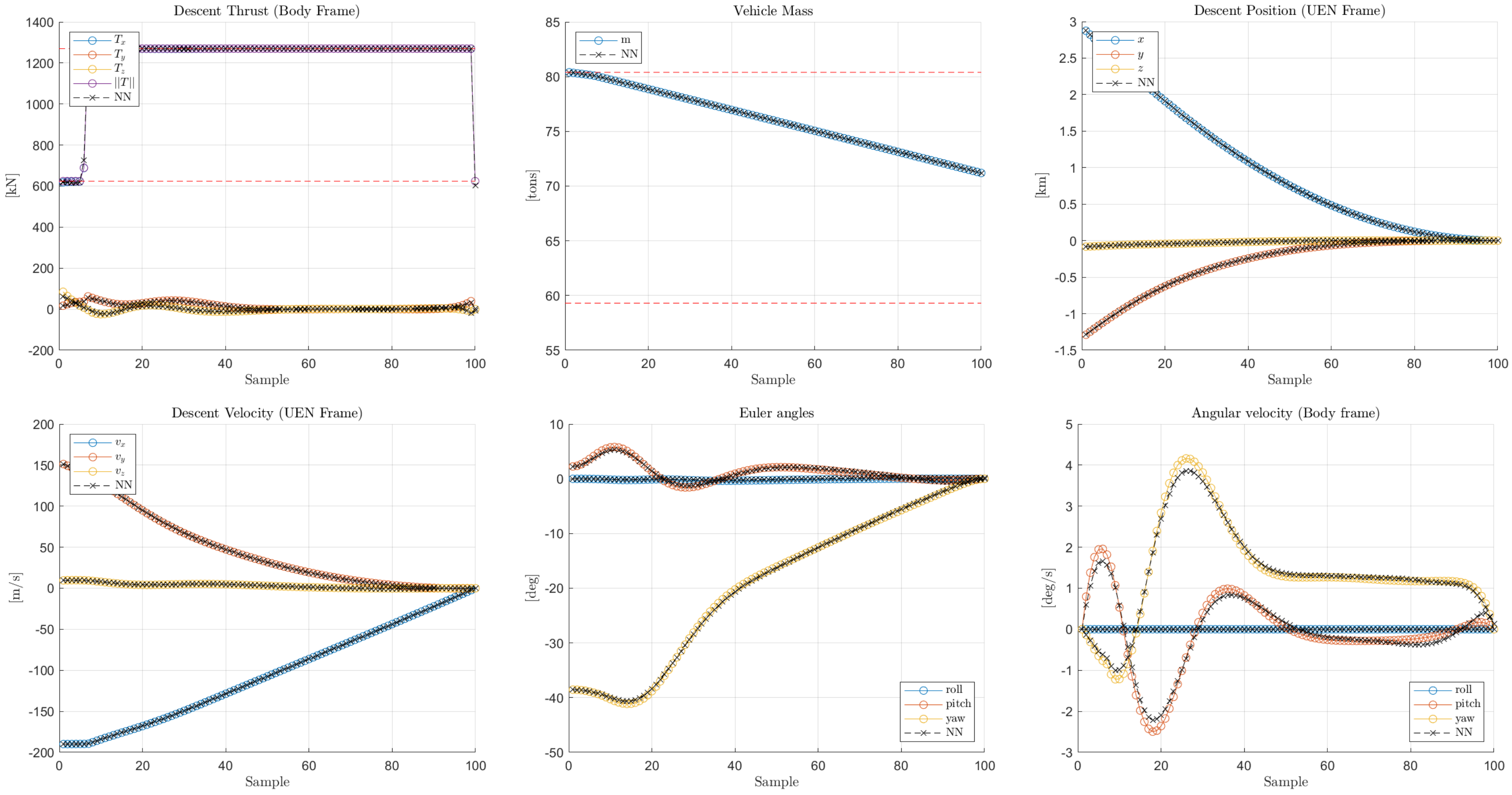


- 3 hidden layers of 50 neurons each
- Hyperbolic tangent sigmoid activation
- 10 thousand expert demonstrations
- 1 million datapoints
- Training with DAgger algorithm



Case Study #8
Supervised Learning Approach - Results

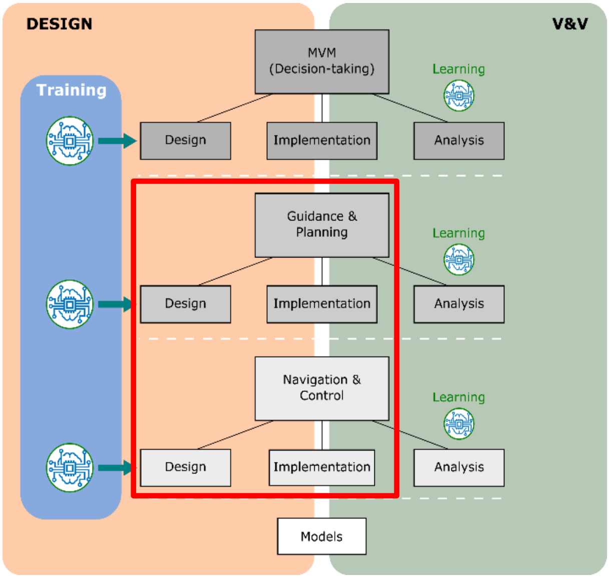
Comparison between expert guidance and NN output:



Way Forward

Way Forward - Maturation Plan & Roadmap
Deep Reinforcement Learning G&C

Points to improve	How to improve
Robustness to Wind	• Include a sensor to estimate wind to provide measurements to the NN
Robustness to initial conditions	• Increase the dispersion considered during RL training
Fuel Consumption	• Modify the reward such that the consumption is reduced maintaining a good landing accuracy
RL hyperparameters fine tuning	• Manual fine tuning • Optimization problem to optimize the hyperparameters
Extension of the convergence analysis to the 3D scenario	• Adapt the implementation of the robustness tool to handle the 3D scenario



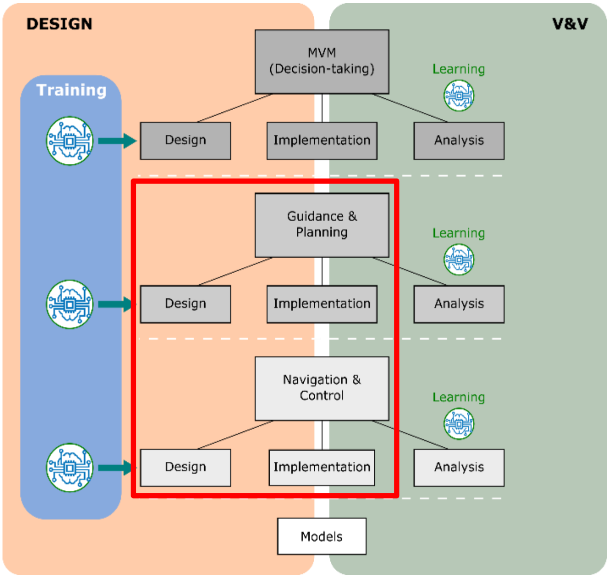
- **Application scenarios**
  - In-Orbit Servicing (IOS)
  - Active Debris Removal (ADR)
  - Entry, Descent and Precision Landing (EDL):
    - Reusable Launch Vehicle (RLV); Re-entry vehicles with Inflatable Heat-Shields (IHS)

- **Potential indicator: HIGH**
  - Remarkable results of the Monte-Carlo campaign
  - The NN training can be repeated whenever the dynamics change
  - Non-iterative algorithm with guaranteed computational time
  - NN validation approaches exist

Way Forward - Maturation Plan & Roadmap

ML-based Guidance Optimization Surrogates

Points to improve	How to improve
Training performance	• Test other optimizers in Keras • Test other open-source libraries and tools
Expert guidance optimizer	• Consider a different sub-problem solver, such as ECOS • Consider other external and open-source tools (SCP by Danylo Malyuta et al.)
Assess performance in simulation	• After training, test the NN in the high-fidelity simulator; • Iterative design process may be necessary for tuning the expert guidance
Validation	• The validation tools used in other case studies may be used to validate the resulting NN



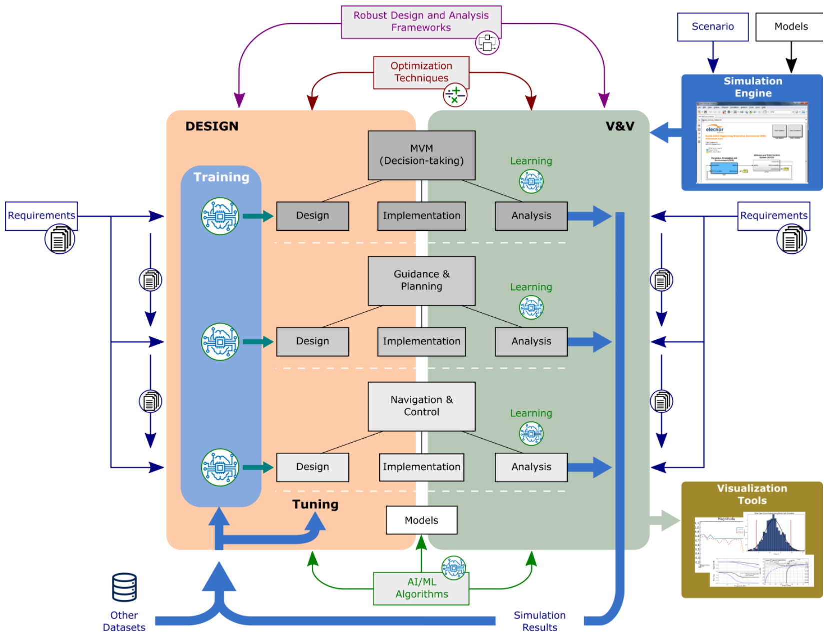
- **Application scenarios**
  - In-Orbit Servicing (IOS)
  - Active Debris Removal (ADR)
  - Entry, Descent and Precision Landing (EDL):
    - Reusable Launch Vehicle (RLV); Re-entry vehicles with Inflatable Heat-Shields (IHS)

- **Potential indicator: HIGH**
  - Good results in the approximation of an online optimization algorithm
  - The training process is typically easier than the Deep RL method, although it requires a very high number of expert demonstration
  - NN validation approaches exist

Conclusions

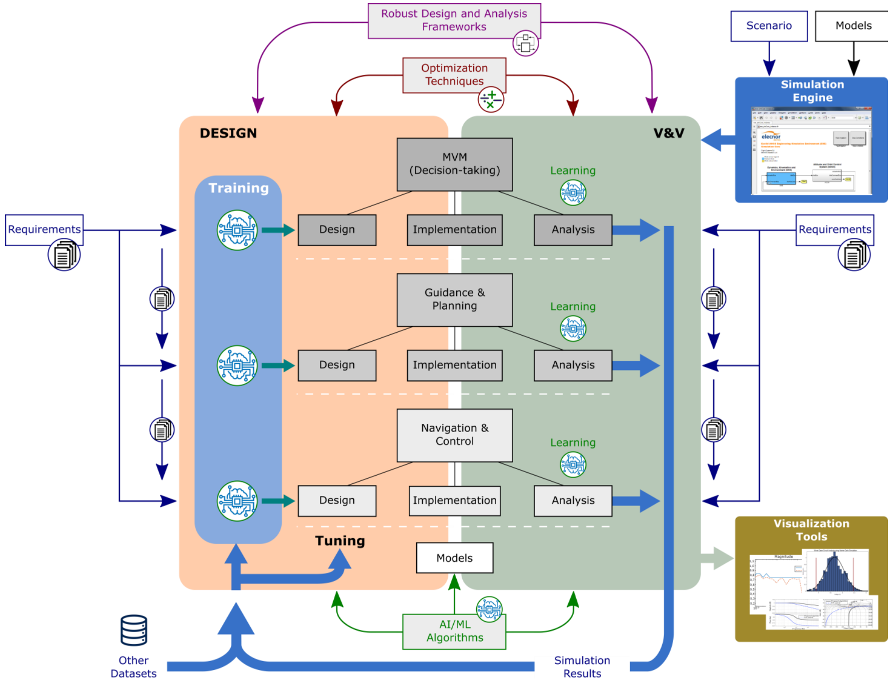
Conclusions

- **Very promising results** in the use of AI/ML/RL for complex GNC problems
- **AI4GNC** addresses **8 Case Studies** considering an overall **realistic RETALT RLV dynamics** benchmark
- **ESA-i4GNC framework** developed in MATLAB/Simulink using an OOP approach and exploiting libraries in AI community
  - The framework **SW will be published soon in GitHub**



Stay tuned!!

deimos



Acknowledgements

The results presented have been achieved under funding by the ESA TDE programme with ESA contract No. 4000134108/21/NL/CRS. The view expressed in this presentation can in no way be taken to reflect the official opinion of ESA.

ESA Contact Point: joris.belhadj@esa.int
DEIMOS Contact Point: paulo.rosa@deimos.com.pt