17th International Conference on Space Operations, Dubai, United Arab Emirates, 6 - 10 March 2023.

SpaceOps-2023, ID # 622

Towards an AI-enhanced robotic Digital Twin for space exploration assets

E. V. Ntagiou[a]*, K. Kapellos[b], S. Tonetta[c], C. Tranoris[b], R. Bussola[c], M. Cristoforetti[c], A. Micheli[c], J. Ocón[d], V. Reggestad[a]

[a] *Department of Ground Systems Engineering and Innovation, European Space Agency / European Space Operations Centre, Robert-Bosch-str 5, Darmstadt, Germany.* *evridiki.ntagiou@esa.int*, *vemund.reggestad@esa.int*
[b] *Trasys International, Rue d'Arlon, 1040 Bruxelles, Belgium,* *konstantinos.kapellos@trasysinternational.com*, *christos.tranoris@trasysinternational.com*
[c]*Fondazione Bruno Kessler, Via Sommarive, 18, 38123 Povo TN, Italy,* *marco.cristoforetti@fbk.eu*, *rbussola@fbk.eu*; *tonettas@fbk.eu*
[d]*GMV Innovating Solutions, Isaac Newton, 11, 28760 Tres Cantos, Madrid, Spain, jocon@gmv.com*,
* Corresponding Author

Abstract

Simulation technology is well adopted for decades at ESA and European Space industry throughout a space mission lifecycle, during which a few simulators are developed. The concept of Digital Twins is a rather new paradigm. The most accepted definition is that of an appropriately synchronized body of useful information of a physical entity in virtual space, with flows of information that enable convergence between the physical and virtual states. In robotics, digital models of the target systems are traditionally used in all phases of a mission ranging from design to development and operations. These digital systems have shown their limitations to fully support their objectives, when the involved models are not able to capture the complete physical reality. This is particularly true when the operations environment evolves during the mission (e.g., when discovering a new planetary area). In this paper, we present the objectives and achievements of the RoboDT system, which is currently under development by the “Robotic Digital Twin” Technology Development activity at the European Space Operations Centre. The activity proposes a new framework where engineering methods and AI techniques are integrated into a coherent Robotic Digital Twin Framework, in order to allow the on-line update of the system models, planning and what-if analyses, plan monitoring and fault diagnosis. The proposed software architecture to build the Digital Twin system and the corresponding functionalities will be presented, with particular attention to the synergy between engineering methods, symbolic and data-driven AI.
Keywords: Digital Twin, Modelling, Machine Learning, Robotics

Acronyms/Abbreviations

AI	Artificial Intelligence
CDM	Conceptual Data Model
DHS	Data Handling System
DT	Digital Twin
EGS-CC	European Ground Segment Common Core
ESA	European Space Agency
ML	Machine Learning
OBSW	On-Board software
SVF	System Validation Facility
TC	Telecommand
TM	Telemetry
WTI	Wheel Terrain Interaction

1. Digital Twin: an Introduction

In this paper, we present the objectives and early achievements of the “*Robotic Digital Twin*” Technology Development Element activity, funded by the European Space Agency (ESA) and led by TRASYS (BE) in collaboration with Fondazione Bruno Kessler (IT) and GMV Spain (ES).

1.1 Origins and definitions

Simulation technology is well adopted for decades at ESA and European industry throughout a space mission lifecycle. Over the years, European industry and ESA have developed simulation technologies, tools, platforms, harmonisation roadmaps and standards. During the lifecycle of a space mission a number of simulators are developed [1]. Each Simulator is focused in delivering capabilities for specific needs of the project phase. Typical concerns are: feasibility assessment, design optimization and validation, sub-system performance and design assessments (e.g. thermal, power) or detail design verification (e.g. AOCS) and on-board software (OBSW) validation (SVF). In later phases of the mission, the focus is shifted to the training of operations staff, validation of the systems, composing the ground segment, anomaly investigation, etc. The techniques used and the fidelity of each simulator is selected to best deliver to the concern at hand.

The concept of Digital Twins is a rather new paradigm; although simulation technology is an important part in their realisation, they are not the same. The term "digital twin" was first coined by John Vickers in the context of product lifecycle management at NASA. Typically, a digital twin is described as the digital representation of a material or immaterial object or process in the real world, an adaptive model of a complex physical system. The most commonly accepted definition is that of an *appropriately synchronized* body of useful information (structure, function, and behavior) of a physical entity in virtual space, with flows of information that enable convergence between the physical and virtual states. It provides *all relevant properties* and *services* of the represented object or process via interfaces.

Considering the scope of this study, the following definition from [2] (adapted from [3]) is considered as the baseline:

> "*A Digital Twin is defined as a dynamic and self-evolving digital/virtual model or simulation of a physical asset (part, machine, process, human, etc.) representing its exact state at any given point of time enabled through real-time bi-directional data assimilation as well as keeping the historical data, for real-time prediction, monitoring, control and optimization of the asset to improve decision making throughout the life cycle of the asset*."

The scope, hence, of a Digital Twin is much wider than a traditional simulation. In both cases, computer-based digital/virtual models of the physical entity and its environment are utilised. However, traditional simulation is based on models with pre-defined characteristics. The simulation models are developed based on pre-known characteristics of the physical entity and the rules of physics. The behaviour of a simulation model does not change, unless a designer updates it; it is, thus, rather *static*. A Digital Twin may begin its life as such a model, but becomes more powerful when it starts receiving in an automated manner real-time data (e.g. Telemetry (TM)) from its real-world counterpart. The virtual models constituting a Digital Twin then become active, changing as the data are delivered and providing intuitive visualisation of the state of the physical asset to its operators (test engineers, scientists, mission operators, et.c.). Hence, unlike traditional simulators, the DT is continuously kept in sync with its physical counterpart. Because it dynamically adapts its characteristics, state and behaviour, based on this data, the Digital Twin matures through the product lifecycle. AI techniques (mainly Machine/Deep Learning) play an essential role in achieving this characteristic. As a result, the Digital Twin yields valuable information that is not generated by a traditional simulation.

1.2 Functionality and benefits

In order to implement a DT, a number of technologies can be adopted, including Artificial Intelligence (AI), Model Based Systems Engineering (MBSE) tools and technologies, Computer Aided Design (CAD), Augmented/Virtual Reality (AR/VR) and various communication technologies. At earlier phases of the mission, when the physical spacecraft or some of its subsystems are not yet developed, the DT can take MBSE information, design data, simulation models and simulator data to compensate for the lack of real-world sensor data. It provides at these early phases a digital representation of the envisaged future product. Having a digital representation of the envisaged spacecraft at hand, allows enhanced trade-off analysis, design and performance assessments, validation and optimisation, before or during the physical spacecraft is being produced. To give an analogy, it can be compared to driving a car on the road from home to work, before it is even built. Using the combination of the simulated and real-world sensor data, models and analytic techniques, the DT creates output that supports design, production, verification and operations of the spacecraft. The main benefits of a DT are found in the:

- *Continuous synchronisation between the real asset and the DT.* Unlike the traditional simulator that can diverge from the real asset state, a digital twin is kept as close as possible in sync with the state of the real asset. This can be achieved by feeding back the TM as well as environmental data (orbit, surface characteristics, etc.), updating its configuration. Adopting AI and in particular ML/DL techniques allows for a continuous improvement of the quality of the models in an automated and more efficient manner. The combination of collected data with mathematical models is at the heart of any Digital Twin. However, the specific types of models and integration of data strongly depend on the required services the DT is meant to provide. A set of methods have been proposed in the literature to handle various uncertainties and partial observability [4, 5] in case of online model adaptation. Hybrid modelling approaches have also been widely used in scientific applications to embed the knowledge from simplified theories of physics-based models directly into an intermediate layer of the neural network [6]. Within this paradigm, physics-informed learning (PIL) [7, 8] techniques have been proven to be particularly successful. Depending on the phase of the development (e.g. before launch) in absence of sufficient representative data, the models can be improved by means of augmented, synthetic data generated from simulators or AI techniques (e.g. using GANs, Active Learning).

- *Timely prediction of future states.* In an ideal setting, a digital twin will be indistinguishable from the physical asset with the added advantage of acting as a "frontrunner" to the future behavior of its associated physical system by making predictions. Similar to the synchronisation and update of the models, using simulation data and AI techniques, future states and faults in a system can be forecasted much in advance.

- *Assessment of multiple scenarios.* The operator will be assisted in the decision making process by evaluating multiple synthesised unexpected scenarios and assessing their risks. The mission planning process can, thus, benefit greatly and optimal plans can be produced using fitting techniques like reinforcement learning etc [9, 10]. The latter type of algorithms are since recently replacing the traditional model-based planning and control methods. They have been reported to provide accurate real-time dynamical adaptation to new scenarios, scale to large datasets and high-dimensional spaces, and to be robust to observation and model uncertainty. Within the context of Digital Twins, RL algorithms have been applied in the field of manufacturing [11], robotics [12, 13] and autonomous driving [10, 14], to provide services such as real-time model adaptation, anomaly detection, or what-if analysis. In the context of planning, [15] proposes a production system control concept where a digital twin and an automated AI planner are tightly integrated together as one smart production planning and execution system.

- *Predictive Maintenance*. Current predictive maintenance measures can be improved employing ML/DL techniques that will allow for accurate prediction of the time of upcoming system failures. This shall allow for quick reaction on the scheduling of maintenance tasks.

- *Diagnosis facilitation at system and subsystem level*. A DT shall be able to *efficiently* store the full available information on the past states and allow for *optimized* exploration of this information, providing *diagnosis* services to the users.

- *Improved situational awareness*. The most critical difference between satellite and robotic missions lies in the increased need of accurate situational awareness in the latter. Advanced visualizations of a robotic asset's *state* will improve the level of situational awareness for the operator.

At the same time, the DT technology will facilitate the flow back of operational knowledge to the design/production process, which combined with other information will *drive business decisions*, as all the stakeholders (designers, operators, business analysts) are able to virtually "see" the actual and current state of the physical entity and understand how the real product is operating. The concept of the Digital Twin S/C during operation phase of a mission is depicted on a high level in Figure 1 where two main information flow directions are identified:

- *Perception of the physical world*. This is achieved by utilizing simulation models, digitalized asset design information, testing data and received Telemetry. AI algorithms combine all the pieces of information, augment the final dataset if required and produce synchronized models.

- *Operation of the digital world concurrently to the physical asset*. The synchronized models, in turn, allow for an unsurpassed level of monitoring and accurate prediction of the asset's future states, ensuring a safe and optimized commanding sequence.

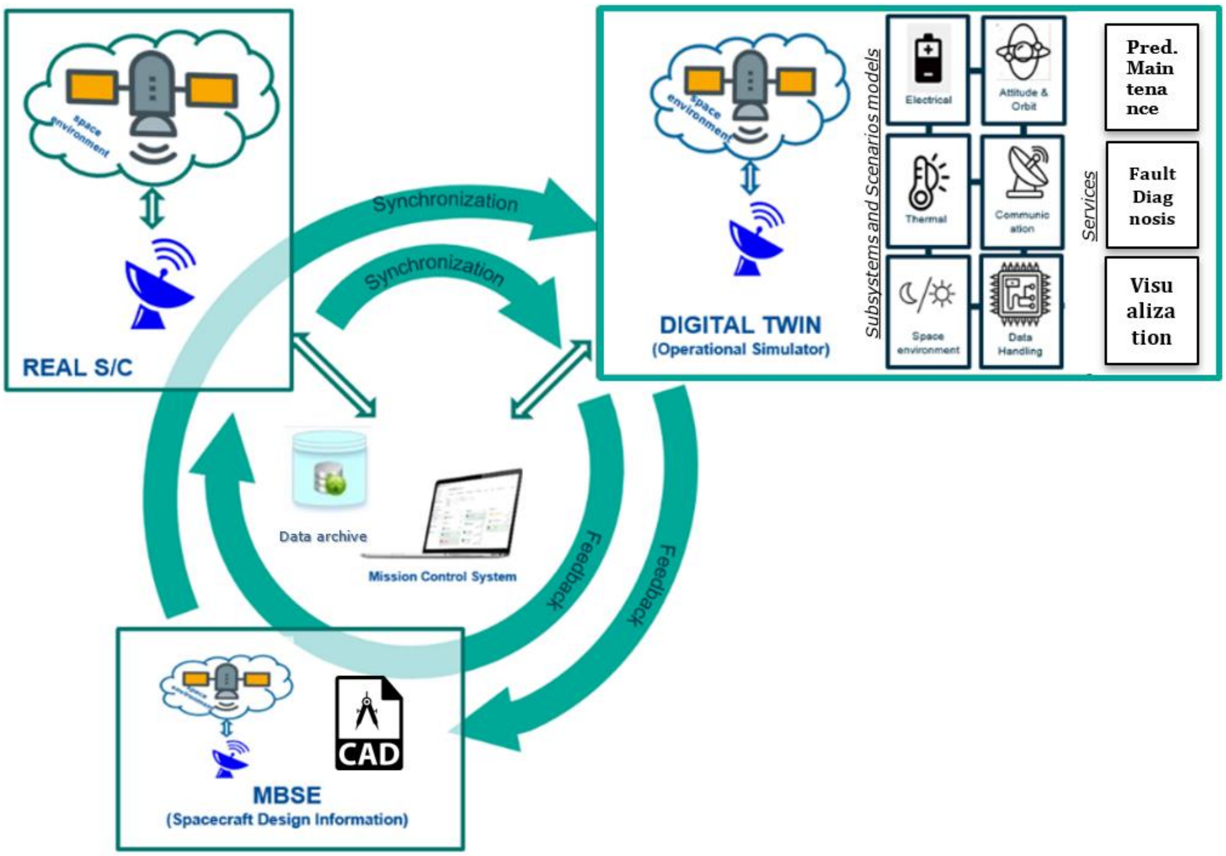

Figure 1. Digital Twin Concept

1.3 DT in Space Mission Operations

More specifically, some of the main challenges in the operations domain that can be addressed by Digital Twin technology are:

- **What is the current state of the asset?:** Understanding the actual current state of the complex robotic asset system at any given point in time can be a non-trivial and effort intensive task. It can involve manual intervention, retrieval of historic data, consultation of the design and conduction of simulations in order to interpret the received telemetry, fill the gaps, determine and understand the current state.

- **What do I observe vs what I can observe:** Information and knowledge of the current state of the sub-systems of a spacecraft is scattered and only understandable in pieces by sub-system experts. The interactions, dependencies and playing of the subsystems together is not easily understandable and not represented in form of easy-to-use information (information is distributed over thousands of individual TM parameters, events, TC history, etc).

- **Time Machine:** Reconstructing the history and understanding the effects of the past operations, events and anomalies on current state of the system is not easy. To give an example, as the complex model of the on-board queue, which is incorporated in the control system may often not be accurate, operators prefer to poll and dump the current state of the on-board queue rather than trusting the information available in the model on the ground.

- **Lost Heritage:** Knowledge of past operations is often only carried forward through individuals rather than supported by the system and processes.

- **What went wrong, why and what will go wrong?**: Diagnostics, root cause analysis and anomaly detection, prediction and predictive maintenance are off-line activities that are performed on retrieved data in order to detect patterns in the Big Data, received from the spacecraft. Although AI/ML as well as Complex Event Processing techniques have shown promising advancements in this direction, there is a big deficiency due to static nature of the models used in operational simulators (no adaptive modelling). Hence simulations are not of a big help in validating and training the data analytic algorithms. In the absence of adaptive models, the gap between reality and simulation grows over the time.

- **Trust the simulator for what-if analysis:** The level of trust for validation of new procedures and performing what-if analysis depends to a large extent on the fidelity, accuracy and representativeness of the simulator. One important element in representativeness of the simulator is its configuration to represent the current state of the spacecraft. The “tuning” of the simulator for this purpose is a manual and time-consuming task. For critical operations, the trust level is not high enough to rely only on the numerical simulator. Hardware and engineering models may then be used in addition to ensure representativeness. Tuning the hardware to replicate the current state of the physical spacecraft at a given point in the mission may however not be an easy task and would require re-programming, replacement of certain hardware with software simulators, etc. to e.g. mimic the orbit environment.

The rest of the paper is organised as follows: Section 2 introduces the specificities of robotic missions operations, the objective of the Robotic Digital Twin activity in this context and the identified use cases. Section 3 covers the use cases selected for implementation whereas Section 4 describes the architecture and selected components of the final prototype DT, as well as the ML-based models developed in the project. Section 5 presents the conclusions and future steps of the project.

2. DT in the operations of Robotics assets

In essence, robotic and rover operations receive and process TM and send TCs similar to spacecraft systems, partly requiring “classical” ground segment architecture for *communicating* with the assets. However, in robotic missions, an operator needs to be provided with information that increase the situational awareness of the robotic asset. This is the key difference with the classic S/C operations; the rover environment will evolve as it traverses a planet’s surface and will trigger ground reaction and commanding that cannot be defined prior to the sufficient analysis of the latest state. The frequency of the ground control cycle is entirely dependent on the mission type, whether telemanipulation is possible (very short communication latency - almost direct) or fully autonomous execution of activities has to be implemented between the time of upload of the TCs and the time of availability of the TMs.

More specifically, in a *lunar environment* time delays between Earth Mission Operators and the lunar surface should range around 5 seconds round trip time, allowing to send commands and observe the results 5 seconds later. Visual awareness would be achieved with a combination of live video feeds, images and sensor readings to the operators. Rover operations would follow initially digital elevation maps of different resolutions that could become more detailed based on the rover in-situ data. Scientists can follow and contribute live to ongoing operations.
Robotic operations on the *Martian environment* results in long communication delays in the order of several minutes or hours. One of the key elements that distinguishes rovers controlled over very long delays (e.g. Mars rovers) with typical spacecraft is the way that they are commanded, using very high-level interpreted commands/activities triggered mostly on event. This is due to the difficulty to accurately predict the rover situation in contrary to orbiters for example. An additional peculiarity of such rover missions is the continuous involvement of the Science Operation Centre (SOC) that could be co-located with the operations team. They could share a number of tools as they have the

same need of understanding the rover executed activities and plan the new ones in the evolving environment, as a team. There is necessity of long-term planning but the execution cannot be left to the control team alone: the scientists provide input on what to do and where to go, the surface operations teams and engineers decide on and support the safe execution of what is possible.

Within the context of a *space exploration mission*, multiple robotic assets are envisioned to be used. In the case of surface operations, a typical mission scenario involves rovers that explore the planet surface with various instruments e.g. robotic arms, drillers, grasping mechanisms, etc. The different operational requirements and constraints e.g. path planning, higher accuracy of situational awareness, variable TM downlink frequency and robotic asset capabilities (grasping, drilling, etc), along with the increased level of environment uncertainty are a few examples that render a space mission involving a rover different from satellite operations. Hence, the need for risk assessment and multiple scenarios evaluations becomes even more relevant, increasing the value of a Digital Twin in such missions.

In robotics, digital models of the target systems are traditionally used in all phases of a mission under the name 'Virtual Flight Segment' ranging from design to development and operations. These digital systems have shown their limitations to fully support their objectives, in particular when the involved models are not able to capture the complete physical reality. This is particularly true when the operations environment evolves during the mission (e.g., when discovering a new planetary area).

The activity proposes a new framework where engineering methods and AI techniques are integrated into a coherent Robotic Digital Twin Framework, in order to allow:

- On-line update of the system models: The appropriate combination of data-based and physics-based simulation models enables the application of online data analytics for adapting at runtime the models of the virtual asset guaranteeing a high-fidelity representation of the physical asset and its environment.
- Planning and what-if analyses: A digital twin enables planning of actions and what-if analyses based on more reliable models. These analyses allow to synthesize unexpected scenarios and study the response of the system as well as the corresponding mitigation strategies. This kind of analysis without jeopardizing the real asset is only possible via a digital twin.
- Plan monitoring and fault diagnosis: Telemetry data are monitored to detect and identify anomalies. Diagnosis is performed to enable a retrospective analysis to extract the root causes of the observed failures. This is essential in order to support timely recovery from problematic situations and/or safely operate the real asset in a degraded mode of operation.

The proposed software architecture to build the Digital Twin system and the corresponding functionalities will be presented, with particular attention to the synergy between engineering methods, symbolic and data-driven AI: e.g., in a planetary exploration mission, the terrain model used for planning is the same as the one used for simulation, as is adapted with machine learning algorithms based on the telemetry data.

2.1 Identified use scenarios and use cases

For the identification of the ROBDT use scenarios to include in the final deployment we analysed the system from two points of view: a) the ground operations support including both engineering and science operations support, and b) the simulator models' candidate to be enhanced using AI/ML technology (see Table 1 and Table 2 respectively).

Operations Support	
Engineering Analysis	
	Health Surveillance process
	Anomaly Detection
	Behaviour Extraction for Time Series Investigation
	ML to identify terrain types and features in orbital and ground-based images
	Awareness in space operations
Science Analysis	
	Automatically analyse data onboard Mars spacecraft and notify scientists when anything noteworthy occurs or changes in order to eliminate bandwidth limitations at large distances

	ML methods for automatic mineral discovery of less common minerals in CRISM acquired data, resulting in mineral discoveries that suggest the existence of water in the Jezero crater landing site and the Northeast Syrtis region
	ML models for Mars weather analysis
	ML for Mars valley network mapping and stream ordering
	ML for Mapping polar-layered deposits and their unconformities
	Analysis of MOMA data

Table 1. Identified DT use cases for Operations Support

Model		
Rover Models		
	Mechanical Dynamics model	
		Learning complex, high-dimensional and novel dynamics
	Power s/s Model	
		ML models for the solar panels power generation
		ML models for the battery charge/discharge cycle estimation
		ML models for the power consumption of the individual power consumption units
	Thermal s/s Model	
		ML models for the thermal evolution of the robotic asset s/s's
	Communications s/s Model	
		ML models for the identification of the adaptable communications rates
	Controller s/s Model	
		ML models for learning control policies
		ML models for Risk- and Resource-aware path planning
		ML models for Drive-By-Science capability
		ML models for autonomous planetary landing
		ML models for advanced manipulation
		ML models for pose estimation of a noncooperative objects
		ML models for advanced object recognition
		ML models for interpreting and anticipating human actions
		ML models for sensor fusion & dimensionality reduction
		ML models for high-level task planning
Environment Models		
	Atmosphere Model	
		Atmospheric features: Fluctuating amount of oxygen, Unknown source of methane, Martian Cloud distribution
		Mars weather analysis
	Terrain Model	
		Valleys network identification and streams ordering, mapping polar-layered deposits

Table 2. Identified DT use cases for model updates (robotic asset and environment).

3. Selected use cases for implementation

For the demonstration of the developed DT framework, we chose to work on a planetary robotic asset provided by TRASYS. Because of the inherent uncertainty of the robot-environment interaction, data-based models are well suited. Moreover, it is an ideal case study for path planning and monitoring. We consider a typical scenario prepared for a 'Sol' execution from the ExoMars planetary exploration mission: the 'Drilling site approach and surface sample acquisition'. The following activities shall be performed by the rover under the constraints of the available *power*, *memory capacity* for data storage, and *duration* (single sol).

- Initially, the rover waits the transition Night to Day to wake-up (when the solar panels generate sufficient power [greater than a predefined threshold]) and is configured accordingly for the upcoming day activities. In particular, the subsystems involved for travelling are warmed-up and moved to a 'standby' state. These steps involve several uncertainties, mainly the exact local Mars time at which the rover wakes-up as well as the warm-up durations, which all depend on the external conditions (e.g., atmospheric temperatures, relative orientation of the rover solar panels with respect to the Sun, etc.).
- After completion of the rover configuration, the rover starts traveling to reach the target location whose position has been identified and communicated to the rover from ground. Even though the duration of the travel depends on the topology and characteristics of the encountered terrain, it can be estimated at planning time.
- Upon arrival at the target location, the rover is re-configured from travelling to drilling operations: travel related units are switched off while the drill box and the drill are warmed-up and moved to the 'standby' state. Similar to the previous re-configuration, the durations of those operations (and therefore the resulting power consumptions) can only be estimated as they depend on the time in the 'Sol' that the rover reached the outcrop.
- The drill box is then deployed, the drill initialised, and reaches the soil to collect the surface sample (10cm depth), before it retracts. The duration of the sampling operation (and therefore the power consumed) depends on the hardness of the soil.
- Finally, images of the environment are acquired and downlinked to ground to guarantee that the operations planning team has a sufficient amount of information for planning for the next cycle/Sol.
- After establishing the communications with the orbiter and transferring the acquired data, the rover waits the transition Day to Night (when the solar panels generate power less than the given threshold), is configured for night and 'sleeps' waiting for the next plan to be uploaded for execution.

In this scenario, the following use cases were found of relevance to a Digital Twin and were selected for implementation:

- *Models update using ML*. Two specific models that can be adapted using ML:
  - the Wheel Terrain Interaction (WTI) model and
  - the Data Handling System (DHS) model that predicts the Actuator Drive Electronics (ADE) warm-up time.

- *Rover activity planning*. For the planning, access to a model of the activities and tasks of the rover was provided, with the objective to automatically synthesize activity plans. However, some of the actions have simulated effects, meaning that the results of applying such actions as well as their durations are not modelled (can only be simulated). Most notably, among these quantities that are possible to evaluate, we have the ADE warm-up timing that is estimated by a learned and evolving ML model as mentioned above. Preliminary results showed that the approach is capable of generating feasible plans quickly.

- *Fault detection*. The TM can provide full information about the components of the state, but not for the state of the task execution. Looking at the case study, the plan provided by the planner requires first to run a task that waits for the rover to warm up and switch on a set of subsystems for traveling. Upon completion of this action, a state component is modified. This state component acts as a precondition for a task that is used for updating the rover heading estimate with a value provided by ground. If the monitor notices that the heating level has just been changed, but the precondition state component of the heating level update task has not been satisfied before, it can decide that the heating update task has been executed violating its preconditions in one of the possible belief states will report this violation to the operator and to the diagnostic component.

We use runtime verification techniques extended with assumptions to detect a failure of the plan execution [16]. The monitor assumes that the rover follows the plan generated by the planner and verifies pre- and post-conditions of actions. If the execution is not consistent with the plan it triggers an error. The monitor is automatically generated from the plan specification by NuRV [17], a tool that supports assumption-based runtime verification.

- Fault diagnosis and root cause analysis. Given the anomalies identified by the monitoring component, the goal of the diagnosis component is to *provide a list of most probable explanations for these anomalies*. The explanations are identified using a fault propagation graph (FPG), which describes how failures of one subsystem or component of the rover can cause failures of other components. In particular, for this activity, we construct the FPG as follows:
    - First, for each task, we use the DT specification to identify the actions of other subsystems that can cause a failure of the given task. For example, the warm-up task that prepares the rover for travel depends on warming up the navigation cameras, localization cameras, actuator drive etc.
    - Second, we use the description of hardware implementation and FMECA tables to describe how failures in the hardware components can cause failures of the higher level subsystems. For example, the actuator drive depends on working hold-down release mechanism, which in turn depends on working motors, motor heaters, etc.
    - We then use efficient techniques rooted in formal methods [18] that for each set of identified failures list all the possible root causes.

    As a result, if the monitoring component reports an anomaly in the warmup task, we can list failure of motors as one of the root causes (among many others). More interestingly, if the monitoring component reports several anomalies, which all transitively depend on the motors, the diagnosis component can report the motor failure as the most probable root cause as it is more probable than multiple separate failures of independent subsystems.

4. Design and implementation

4.1 DT Architecture

The figure below presents the first layer decomposition of the software in several components. This decomposition has been done by identifying the different functional groups of requirements which comprise the software identification.

An initial identification of the components which have emerged is the following (see Figure 2):

- The **Monitoring and Control Station (MCS):** the MCS is based on a prototype of EGOS-CC based Mission Control System [19] extended for robotics needs. Based on the modelling of the assets (using the EGS-CC Conceptual Data Model (CDM)), the MCS monitors and controls the assets (via parameters, products and activities). In addition, the MCS dispatches the same control requests to the DT simulators keeping them always synchronised with the real robotic assets and monitors their evolution. Finally, it allows the distribution and the persistence of all generated telemetry and activity verification statuses.
- **Simulation component (SimAAS):** consists of multiple simulators which simulate the functionality of the robotic assets at different levels of abstraction and concern. This component, organised 'as a service', contributes to the genericity of the proposed architecture supporting simulators of multiple robotic assets with varying capabilities as long as these Simulator components comply with specific requirements regarding their packaging and interfaces. Their configuration and management and the monitoring of their status involves the use of Kubernetes and associated services. The simulation capabilities are provided by the instantiation of the SIMROB multi-asset space robotics simulator [12]. It includes:
    - The Assets models such as the robotic asset mechanical dynamics, power, thermal, DHS and other subsystem models,

- The Environment models including the terrain, the orbital and timekeeping and the Atmosphere model, and finally,
- The Simulation Framework that orchestrates the execution of the simulation models.

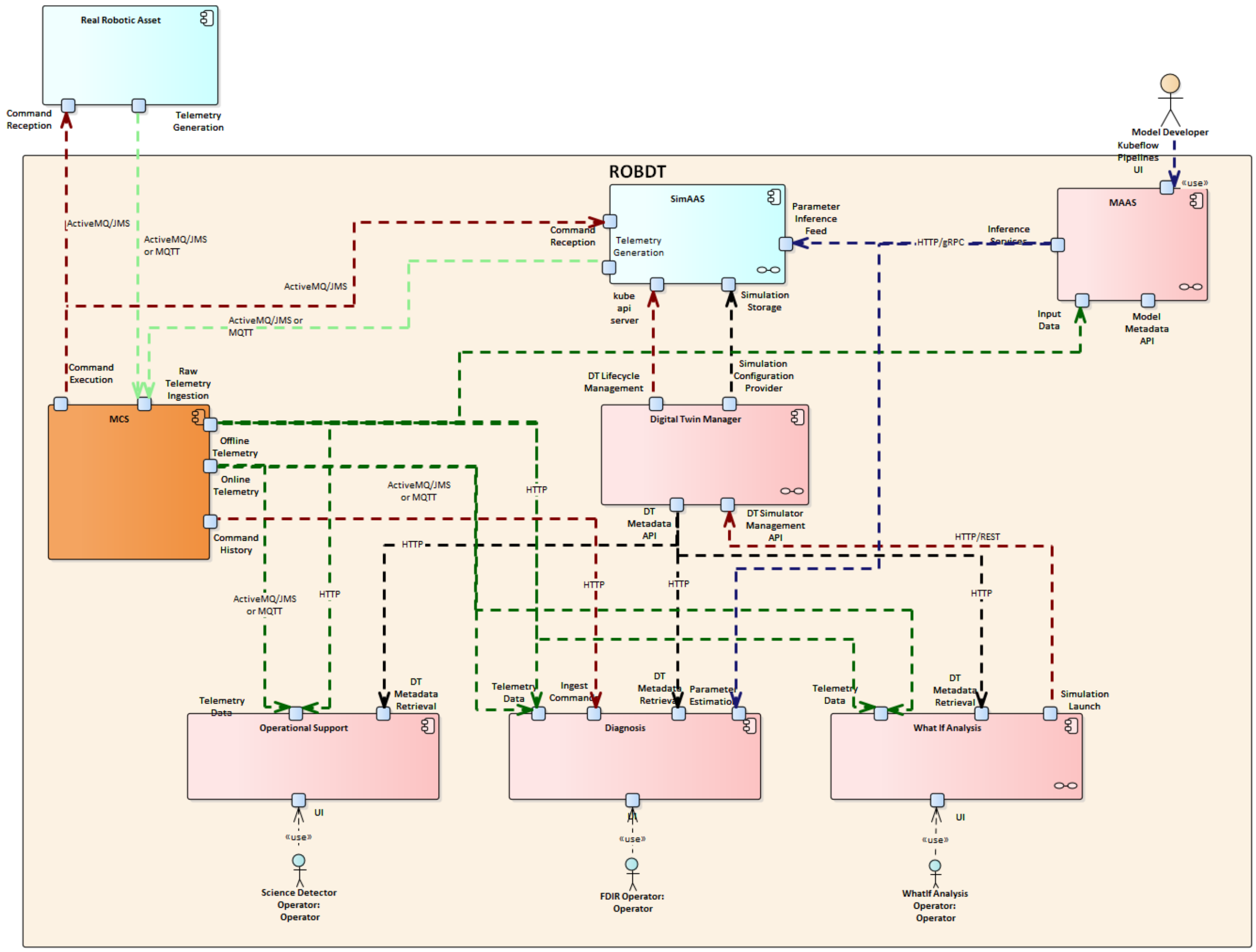

Figure 2. ROBDT first Level decomposition diagram

- > **Model (Updaters) component (MAAS)**: its goal is to provide MLOps (Machine Learning Operations [21]) capabilities to the DT as a service by standardising and facilitating tasks such as data preparation, model training and model serving, while also enabling easy, repeatable, portable deployments on diverse infrastructure. It is based on the Kubeflow open-source project. In the context if this activity two such models and the corresponding pipelines are proposed: the wheel-terrain interaction (WTI) model and the DHS update model. It involves the use of Kubeflow and associated services.
- > **Digital Twin Manager component (DTM):** the main role is first, to manage DT definitions and second, to facilitate operations associated with launching, monitoring and stopping the corresponding simulations by hiding the complexities of Kubernetes' APIs that are used in SimAAS to perform the same operations.
- > **What-if Analysis** (**WIA**): it allows simulating the system from its current state or from a hypothetical state according to a given scenario with the additional possibility to check whether a certain goal condition is satisfied, or it is violated. In the context of this activity the WIA component focuses on the simulation of Activity Plans under various conditions with the additional capability of automatic Activity Plans generation,

> **Diagnosis (FDIR PM)**: it allows detecting faults in the current execution (or on historic data), identifying causes of the faults based on their models and providing the corresponding feedback to the operators. In the context of this activity we propose to focus on the detection of faults during the execution of an Activity Plan and to propose a recovery action by generating an alternative one.

> **Operations Support (SCIDET):** to support engineering or science operations planning and assessment. In this activity, a 'scientific agent' is integrated to detect predefined patterns of interest or novelty on on-line (or historical) images acquired by the robotic asset.

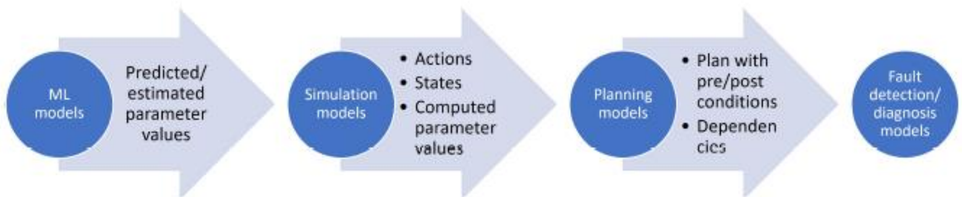

Figure 3. Information exchanged between the RoboDT models

4.2 ML-based modelling

The system utilizes two Deep Learning models to estimate critical operational parameters. Boosted Decision Trees algorithms were also tested with performances comparable to the Deep Learning ones. The final choice of adopting the neural network-based solutions is mostly connected with possible future developments where heterogeneous input data could be available to improve the predictions. In this case, modifying the network to include this different kind of data is simpler than with Boosted Decision Trees.

The WTI model estimates the average drawbar pull (in Newton) generated from the rover starting from four input quantities: the terrain type (class), slip ratio (%), stiffness (%), and the normal-axis load (Newton). The model was trained and validated using a dataset of around 1.000 records of historical data recorded from testbench experiments. The WTI model (Figure 4) is based on data generated during the Adaptable Wheels for Exploration (AWE) ESA project [20] led by the Hellenic Technology of Robotics (HTR). They are generated using the HTR double-corridor testbed where soil conditions, slope, wheel flexibility, axis load and imposed draw bar pull are modified to cover different operational conditions. Crushed basalt has been used as lunar soil simulant with three relative densities 20%, 40% and 60%. The wheel stiffness variation mechanism allowed for a factor of 4x stiffness modification. The slip ratio considered has been the average observed during the duration of the experiments (typically 60 seconds while the slopes varied from 0 to 40deg.

Figure 4. AWE HTR Testbed

The DHS model estimates the actuator driver electronics' warm-up duration (in seconds) that depends on the atmospheric conditions (such as the air temperature) of the area on the planet where the rover operates. The model input consists of the following: longitude (deg), latitude (deg), Mars solar longitude (deg), scenario/season (class). The model's training and evaluation were performed using 821.000 synthetic data of the whole Mars surface generated from simulations based on the engineering estimation of the evolution of the warm-up duration and the

Mars Climate Database* data that provide the atmospheric conditions on Mars. To this end, first, the planet is discretised per 30deg of longitude and latitude and afterwards, at each location, data at specific times in a 'Sol', for all Sols in the martian year and for four scenarios (typical, dust storm, warm and cold) are generated.

The WTI model and DHS model share the same architecture based on Fully Connected Neural Network (FCNN). First, an initial embedding layer maps the class variable to a higher dimensional space. Then, this embedding is merged with the other features forming the input of the first layer of the network. After four additional FCC layers the output is generated. Dropout is used to avoid overfitting.

The detailed structure of the two networks is the following:

- **WTI**: Embedding(3,6), Linear (8,32), Relu(), Dropout(0.5), Linear(32,64), Relu(), Dropout(0.5), Linear(64,128), Relu(), Dropout(0.5), Linear(128,32), Relu(), Dropout(0.5), Linear(32,1). The network was trained for 8000 epochs minimizing the Mean Squared Error loss using Adam optimizer on batches of size 32 and learning rate 1e^{-4}.
- **DSH**: Similar to the WTI case but with slightly different sizes of the hidden layers: Embedding(8,16), Linear(19,128), Relu(), Dropout(0.2), Linear(128,256), Relu(), Dropout(0.2), Linear(256,256), Relu(), Dropout(0.2), Linear(256,128), Relu(), Dropout(0.2), Linear (128,1). The network was trained for 4000 epochs minimizing the Mean Squared Error loss using Adam optimizer on batches of size 128 and learning rate 1e^{-3}.

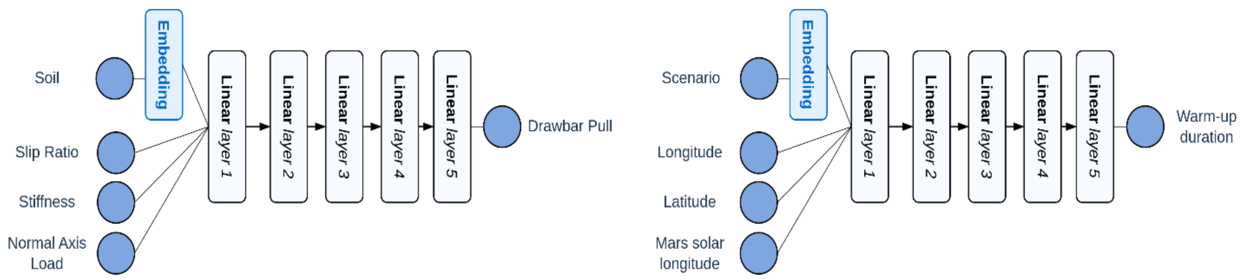

Figure 4: Models' architecture (WTI on the left, DHS on the right)

The Pytorch framework was used for the development. In Figure 5 we can see the performance of the two models. The WTI model achieved an R score of 0.81, DHS model achieved an R score of 0.99.

4.3 Scientific agent

The scientific agent identifies areas of scientific interest in on-line images: During the design phase, the neural networks of the 'scientific agent' are trained by labelled images. During operations, images from the downloaded data are provided:

- Initially, interesting/salient regions are detected in the input image generating region candidates to be processed in subsequent steps; afterwards, a trained variational autoencoder is used to encode and decode the region proposal. The reconstruction error, as well as latent space losses, are used as an indication that the target is novel (unknown).
- Afterwards, a set of VAE-based trained models evaluate each region's proposal to determine its class. The classification is based on a semi-supervised approach in which convolutional neural networks (CNN) are trained on different losses (reconstruction loss, latent space loss, and mixed losses) which result from the VAE.
- Finally, the results of previous steps are consolidated and all known and/or unknown targets are reported in form bounding boxes (including a score and/or a measure of uncertainty) and locations. Outputs of the scientific detector are reported to the MCS as suggestions of areas to be explored.

* http://www-mars.lmd.jussieu.fr/

The classification provided for the scientific agent trained with Public Mars novelty detection Mastcam labelled dataset[†] around 100k images. The Mars novelty detection Mastcam labelled dataset, consists of sub-sampled images (64×64 pixels) of the Mars Science Laboratory (MSL) Analyst's Notebook which were obtained by NASA's Curiosity rover. The Mastcam is a multi-spectral camera and captures images at different wavelengths giving each image six channels. This classification is based on a set of classes: background, broken rock, drill hole, drt, dump pile, and novelty detection.

5. Conclusions and future works

Overall, with respect to the state-of-the-art, the architecture presented demonstrates some novel contributions. On one side, it is focused on space robotic systems, with the relevant peculiarities, as for example the communication delays. On the other side, it provides a unique combination of symbolic automated reasoning, simulation, and machine learning techniques. The activity presented focused on the backbone infrastructure of the robotic digital twin. However, since the project developed some machine learning models for specific use cases, it is always a question whether the models are of good quality. In this respect, the lack of historical data was found to be a major obstacle to the actual training of effective models. Despite the hype on deep learning, classical machine learning methods are also good especially when considering few features and few data points. Finally, the proposed design will be demonstrated by implementing and validating a Robotic Digital Twin prototype targeting a robotic exploration mission scenario. The prototype will be validated applying an 'end-to-end' scenario on a carefully selected 'Exo Mars like' rover and test facility.

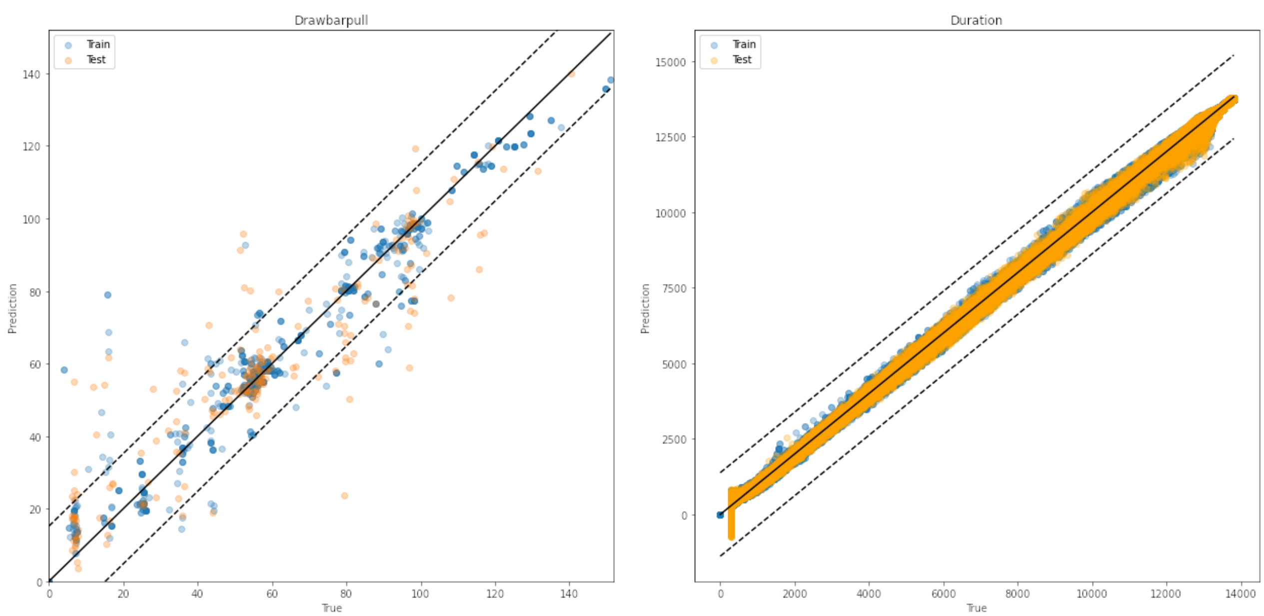

Figure 5. Models performance (predicted vs ground truth). The WTI model is depicted on the left, and the DHS on the right.

References

[1] ECSS-E-TM-10-21A System Modelling and Simulation (April 2010)
[2] Singh, M., Fuenmayor, E., Hinchy, E.P., Qiao, Y., Murray, N., Devine, D.: Digital twin: origin to future. Applied System Innovation 4(2), 36 (2021)
[3] Rasheed, A., San, O., Kvamsdal, T.: Digital twin: Values, challenges and enablers from a modeling perspective. Ieee Access 8, 21980–22012 (2020)
[4] Marmin, S., Filippone, M.: Variational calibration of computer models. arXiv preprint arXiv:1810.12177 (2018)

[†] https://zenodo.org/record/1486196

[5] Damianou, A., Lawrence, N.D.: Deep gaussian processes. In: Artificial intelligence and statistics. pp. 207–215. PMLR (2013)
[6] Chao, M.A., Kulkarni, C., Goebel, K., Fink, O.: Fusing physics-based and deep learning models for prognostics. Reliability Engineering & System Safety 217, 107961 (2022)
[7] Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational physics 378, 686–707 (2019)
[8] Viana, F.A., Nascimento, R.G., Dourado, A., Yucesan, Y.A.: Estimating model inadequacy in ordinary differential equations with physics-informed neural networks. Computers & Structures 245, 106458 (2021)
[9] Tian, Y., Chao, M.A., Kulkarni, C., Goebel, K., Fink, O.: Real-time model calibration with deep reinforcement learning. arXiv preprint arXiv:2006.04001 (2020)
[10] Wu, J., Huang, Z., Hang, P., Huang, C., De Boer, N., Lv, C.: Digital twin-enabled reinforcement learning for end-to-end autonomous driving. In: 2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI). pp. 62–65
(2021). https://doi.org/10.1109/DTPI52967.2021.9540179
[11] Xia, K., Sacco, C., Kirkpatrick, M., Saidy, C., Nguyen, L., Kircaliali, A., Harik, R.:A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence. Journal of Manufacturing Systems
58, 210–230 (2021)
[12] Oh, M.h., Iyengar, G.: Sequential anomaly detection using inverse reinforcement learning. p. 1480–1490. KDD ’19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3292500.3330932, https://doi.org/10.1145/3292500.3330932
[13] Regatti, J.R., Deshmukh, A.A., Cheng, F., Jung, Y.H., Gupta, A., Dogan, U.: Offline RL with resource constrained online deployment (2021). https://doi.org/10.48550/ARXIV.2110.03165, https://arxiv.org/abs/2110.03165
[14] Pires, F., Ahmad, B., Moreira, A.P., Leitão, P.: Recommendation system using reinforcement learning for what-if simulation in digital twin. In: 2021 IEEE 19th International Conference on Industrial Informatics (INDIN). pp. 1–6 (2021). https://doi.org/10.1109/INDIN45523.2021.9557372
[15] Novák, P., Vyskocil, J., Wally, B.: The Digital Twin as a Core Component for Industry 4.0 Smart Production Planning. IFAC-PapersOnLine 53(2), 10803–10809 (2020). https://doi.org/https://doi.org/10.1016/j.ifacol.2020.12.2865,https://www.sciencedirect.com/science/article/pii/S2405896320336314
[16] Alessandro Cimatti, Chun Tian, Stefano Tonetta: Assumption-Based Runtime Verification with Partial Observability and Resets. RV 2019: 165-184
[17] Alessandro Cimatti, Chun Tian, Stefano Tonetta: NuRV: A nuXmv Extension for Runtime Verification. RV 2019: 382-392
[18] Marco Bozzano, Alessandro Cimatti, Anthony Fernandes Pires, Alberto Griggio, Martin Jonás, Greg Kimberly: Efficient SMT-Based Analysis of Failure Propagation. CAV (2) 2021: 209-230
[19] The EGS-CC based Mission Control Infrastructure at ESOC, 2017. https://indico.esa.int/event/180/contributions/1367/attachments/1249/1474/0930_Pecchioli_Presentation.pdf , http://www.egscc.esa.int/architecture.html
[20] Use of flexible variable stiffness wheels for high speed lunar and planetary exploration rovers and their potential impact on the cost of transport. V. Papantoniou, S. Skevakis, A. Katelouzos, A. Papantoniou, K. Kapellos, E. Papadopoulos, and M. van Winnendael, iSAIRAS 2020.
[21] https://ml-ops.org/