Techniques of Artificial Intelligence for Space Applications - A Survey

Jan-Gerd Meß
German Aerospace Center (DLR)
Institute of Space Systems
Robert-Hooke-Str. 7
D-28359 Bremen
jan-gerd.mess@dlr.de

Frank Dannemann
German Aerospace Center (DLR)
Institute of Space Systems
Robert-Hooke-Str. 7
D-28359 Bremen
frank.dannemann@dlr.de

Fabian Greif
German Aerospace Center (DLR)
Institute of Space Systems
Robert-Hooke-Str. 7
D-28359 Bremen
fabian.greif@dlr.de

Abstract— **The possibilities to observe and interact with any given spacecraft are naturally limited compared to ground-based systems due to a number of factors. These include but are not limited to the availability and bandwidth of their connection to ground, the availability of staff, communication latencies and power budgets.**
While a minimum level of autonomy is required for every spacecraft, past experiments and missions have shown that introducing more sophisticated autonomy mechanisms can drastically increase the efficiency for many missions in terms of reliability, science output and required operational effort.
Artificial intelligence poses an increasingly popular approach for implementing on-board autonomy. The number of techniques and variants of artificial intelligence available in the literature is, however, just as diversified as their potential field of application. To provide an overview of the current state of the art of artificial intelligence and its application for space systems, this paper provides an extensive survey on existing techniques and algorithms as well as existing and potential applications on board spacecraft and on ground. The survey focuses on autonomous planning and scheduling of operations, self-awareness, anomaly detection and Fault Detection Isolation and Recovery (FDIR), on-board data analysis as well as on-board operations and processing of earth-observation data.

TABLE OF CONTENTS

1. INTRODUCTION

Observability and controllability of spacecraft are naturally limited compared to ground-based systems due to a number of factors. These include but are not limited to the availability and bandwidth of their connection to ground, the availability of staff, communication latencies and power budgets.

Introducing more sophisticated autonomy mechanisms can drastically increase the efficiency for many missions in terms of reliability, science output and required operational effort as past experiments and missions have shown. This autonomy can also result in a significant drop of cost for missions that would otherwise require extensive human operation.

The emerging use of Commercial-Off-The-Shelf (COTS) components and their increased computational power opens the door for more complex mission scenarios including an increasing number of sensors and actuators to assess and influence the current status. This, however, also enlarges the search space for solutions regarding operation scheduling and planning and to estimate the environmental and health status of the spacecraft up to a degree that cannot be handled manually. Hence, there is an increasing need for mechanisms and algorithms to make spacecraft more self-aware and autonomous. This can also enable mission scenarios that require the spacecraft to come to its own decisions in uncertain environments and to operate without or with only limited human intervention.

The number of techniques and variants of artificial intelligence available in the literature is, however, just as diversified as their potential field of application. To provide an overview of the current state of the art of artificial intelligence and its application for space systems, this paper gives a general introduction to spacecraft autonomy, provides an extensive survey on existing techniques and algorithms for anomaly detection and Fault Detection Isolation and Recovery (FDIR) and shows existing applications on board spacecraft and on ground.

The paper unfolds as follows: Section 2 gives a general introduction to the terminology of artificial intelligence and spacecraft autonomy as defined in literature and by the European Space Agency (ESA). In section 3, the purpose of anomaly detection as a foundation for the self-awareness of systems as well as corresponding algorithms are described. Based on these findings, section 4 describes FDIR concepts for spacecraft and the use of Artificial Intelligence (AI) to increase system reliability. Section 5 focuses on autonomous operations of a selection of existing spacecraft. Subsequently, section 6 gives an overview on related work in the literature and section 7 wraps up the paper.

2. TERMINOLOGY

The following sections give an introduction to the terminology used in this paper. These are *autonomy*, *artificial intelligence*, *anomaly detection* and *Fault Detection Isolation and Recovery (FDIR)*.

Autonomy

Autonomy is the capability to make rational, informed, self-determined and self-reliant decisions. In order for a system to be called autonomous, it needs to be able to sense, think and act in the world around it. It requires the capability to sense its surroundings and some consciousness about its own capabilities and their effects on its environment and internal state. From this knowledge about the world and about itself, an autonomous system is able to draw conclusions and make decisions with respect to its own goals and carry out actions to reach these goals.

Furthermore, an autonomous system has to be able to respond to off-nominal situations by adjusting its sequence of actions in order to continue achieving its goal as well as maintain safety. Commanding of an autonomous system is done via sets of goals it shall achieve. The level of off-nominality it can handle and the level of abstraction of its goals determines the degree of autonomy reached by a system [1].

The European Cooperation for Space Standardization (ECSS) defines four levels of such autonomous capabilities with level E4 being the most autonomous. The complete list can be found in Table 1. According to the above definition, only systems compliant with level E4 can be regarded as truly autonomous, whereas levels E1 to E3 deal with manually controlled or automated rather than autonomous systems. In contrast to autonomous systems, automated systems can only handle situations that were explicitly foreseen by its engineers. It will respond to these situations (i.e. events) with so-called On-Board Control Procedures (OBCP, [2]) that are precast sequences of actions.

Level	Description	Functions
E1	Mission execution under ground control with limited on-board capability for safety issues	Ream-time control from ground for nominal operations Execution of time-tagged commands for safety issues
E2	Execution of pre-planned, ground-defined, mission operations on-board	Capability to store time-based commands in an on-board scheduler
E3	Execution of adaptive mission operations on-board	Event-based autonomous operations Execution of on-board operations control procedures
E4	Execution of goal-oriented mission operations on-board	Goal-oriented mission re-planning

Table 1. Mission execution autonomy levels according to ECSS [3]

To achieve autonomy, a layered architecture reflecting the different stages of perception and decision-making from hardware functions such as control-loops to abstract goals is most suitable [1]. Such an architecture consists of three layers:

- The **deliberative layer** or planning layer is concerned with high-level decisions, keeps track of abstract goals and derives long-term sequences of actions to achieve them. These sequences are then delegated to the executive layer.
- The **executive layer** or task sequencing layer is responsible for orchestrating and monitoring tasks and decides - based on its sensing abilities - when to activate or pause the execution of elementary actions. These elementary actions are dealt with by the functional layer.
- The **functional layer** or reactive layer has the fastest response time and is concerned with the execution of elementary actions based on predefined control loops or calculations. These are based on raw sensor readings and direct commanding of the underlying hardware.

Each of these layers encapsulates an infinite loop of perceiving its input variables and internal state, drawing conclusions based on these findings and executing deduced actions. This cycle is also referred to as the sense-think-act cycle [1]. The layers exchange data in the form of feedback and commands from top to bottom as depicted in Figure 1. Thus, the entire system again forms a variation of a sense-think-act cycle in which lower layers provide feedback to higher layers and receive commands after the higher layers have deduced actions according to their given input, system state and their high-level goals.

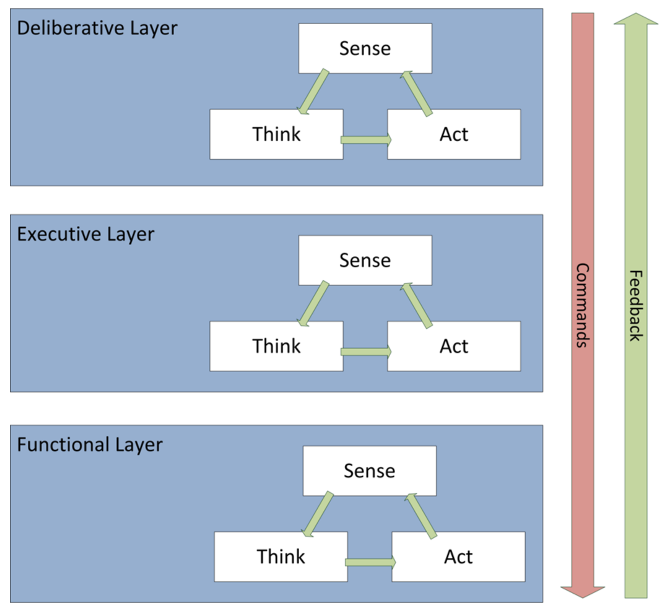

Figure 1. Layered architecture for autonomous systems and the Sense-Think-Act cycle [1]

A detailed survey of autonomous systems in past and present space missions can be found in Section 5.

From the introduction above follows that autonomy describes a set of system functions and capabilities instead of techniques by which they are implemented. Artificial intelligence is, thus, one of many possible approaches to reach autonomy. A brief introduction to artificial intelligence (AI) is given in the following section.

Artificial Intelligence

AI is the study of intelligence as present in computer systems in contrast to natural intelligence to be observed in humans and other living species [4]. More generally, for a computer system to be called *intelligent*, it needs to be able to make rational decisions based on its observings of the world (or a simplified model thereof) and a set of goals it shall achieve. Two different kinds are to be distinguished, strong AI and weak AI. Strong or general AI is concerned with the imitation or outperformance of human intelligence including sentience, consciousness, mind and feelings. Weak or applied AI on the other hand focuses on one narrow task or on solving a specific problem. Since all current research in the space domain is limited to weak AI, this paper focuses solely on this application.

Concerning the problem statements that AI is concerned with, a distinction is drawn between five different categories [4]:

- **Knowledge representation** is concerned with the storage of information about the world (or a model thereof) such that a computer can efficiently process it.
- **Perception** is the ability to deduce aspects of the world given sensor input. Amongst others, this includes anomaly detection, Natural Language Processing (NLP) and Computer Vision.
- **Reasoning and problem solving** generates conclusions from available knowledge using logic and probability theory.
- **Planning and scheduling** finds and realizes strategies for reaching a certain goal or maximizing a given utility function.
- **Machine learning** means the improvement of an algorithms performance through experience.

The process of machine learning is depicted in Figure 2. Based on a data- or knowledge base, a model is trained that can then be queried by an application. Disregarding suitable conditioning, selection of data and overfitting, the model gets better with a growing database and longer periods of training. If the model shall able to learn in the field, the application can add data to the knowledge base during runtime and train the model on that new data.

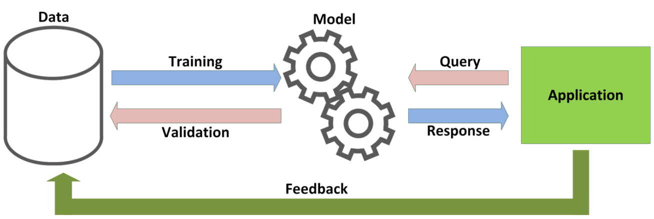

Figure 2. The generalized process of machine learning

All of the above categories are applied in one way or another to modern spacecraft. The techniques that are most widely used include but are not limited to expert systems, (deep) artificial neural networks, fuzzy logic, model-based reasoning and Bayesian networks [5, 6]:

Expert Systems— are used to emulate the judgement of a human operator. To accomplish this, a set of rules is applied to a knowledge base (the expert's insight) containing facts about the current status of the system and its environment. Sophisticated learning expert systems can make additions to either the knowledge base or the set of rules. Expert systems require a lot of a priori knowledge about the system and its environment to accurately model the knowledge base and rule set.

Artificial Neural Networks (ANN)—follow the opposite approach. Instead of using a priori knowledge about the system domain, the network is presented sample data, most preferably from real-world applications, to let the network find patterns in the data, extract important features and find a problem solution by itself. They are mostly used for pattern recognition (in both time-series and multi-dimensional data such as images) and for control of highly nonlinear systems that may need real-time adaption (i.e. learning). With rising computational power on-board, the application of deeper (i.e. consisting of more layers of neurons) and more capable ANNs becomes feasible.

Fuzzy Logic—is an extension of classical set theory to fuzzy sets that can be used to model complex nonlinear systems. Their implementation is typically simpler and requires less computation effort compared to ANNs. They are often used as substitutes for proportional-integral-derivative (PID) controllers since their handling of uncertainty makes them much more robust. Their implementation, however, requires extensive a priori expert knowledge.

Model-based Reasoning—is based on a model of an aspect of the system that predicts the actual system's behaviour. Based on the quality of these predictions, diagnostics can reason about the state of the system and its environment.

Bayesian Networks—can be used to identify the system state based on prior and likelihood beliefs in a set of system variables. They provide a means to model uncertainty and partial knowledge in the observation domain and are mostly implemented as a passive observer system that provides input to a superordinate expert system or ANN. An extension to the model called Dynamic Bayesian Networks (DBN) can express dynamic behaviour over discrete time steps by referencing past values of certain variables [7].

An important aspect of artificial intelligence is the amount of expert knowledge woven into the system. A system that is purely based on data (i.e. the system is unsupervised) may be able to detect patterns and anomalies in the data it has seen, but it is not able to draw semantic conclusions about the underlying system, perform diagnostics or devise actions. Systems that include a lot of expert knowledge (i.e. supervised systems) in their knowledge on the other hand provide deep insight into the systems parameters and behaviours. However, these systems require a lot of effort on the modeling side, deep a priori knowledge about the system and may fail when they find themselves in unforeseen situations. Hence, a combination of data- and expert knowledge-based on-board analysis may be advisable.

One important aspect of autonomy and an open field of research for artificial intelligence is the so-called anomaly detection. Anomaly detection discovers patterns in a stream of data and can identify deviations from these patterns. The capability to detect and classify such anomalies provides important input to superordinate decision-making systems. Anomaly detection is briefly introduced in the next section and covered in depth in Section 3.

Anomaly Detection

Anomaly detection is concerned with the recognition of patterns in some underlying set of datapoints and the discovery of deviations from these patterns. For spacecraft, this is essential for detecting off-nominal situations and responding accordingly. Regarding the management of on-board data perceived by the spacecraft, the ECSS defines two levels of autonomy that are described in Table 2 [3].

Level D1 covers capabilities of storing essential mission data including event reports during ground outage. For autonomous systems and anomaly detection, however, a consistent storage of all mission data is essential to have a broad database for state estimation, pattern recognition and decision-making as covered by D2. This is especially true if machine learning on ground is involved since the ability to downlink data perceived during ground outages is crucial.

Anomaly detection is performed on time-series data like temperature readings over time for detecting off-nominal situations and states, but also on multi-dimensional like images, mostly to detect science opportunities or filter the amount of data selected for downlink.

Level	Description	Functions
D1	Storage on-board of essential mission data following a ground outage or a failure situation	Storage and retrieval of event reports Storage management
D2	Storage on-board of all mission data, i.e. the space segment is independent from the availability of the ground segment	As D1 plus storage and retrieval of all mission data

Table 2. Mission data management autonomy levels according to ECSS [3]

Level	Description	Functions
F1	Establish safe space segment configuration following an on-board failure	Identify anomalies and report to ground segment Reconfigure on-board systems to isolate failed equipment or functions Place space segment in a safe state
F2	Re-establish nominal mission operations following an on-board failure	As F1, plus reconfigure to a nominal operation configuration Resume execution of nominal operations Resume generation of mission products

Table 3. Mission fault management autonomy levels according to ECSS [3]

A very detailed survey on anomaly detection for discrete sequences can be found in [8]. Section 3 focuses on techniques feasible for the application in the space sector and describes an exemplary selection of applications in space missions, mostly health monitoring of spacecraft.

Fault Detection, Isolation and Recovery (FDIR)

For the scope of this paper, it is first necessary to establish a sound definition of fault and failure. A fault is a deviation of at least one system parameter from its desired value. This can be a temperature value that is out of limit, but also a flipped bit in the computer's memory due to a Single Event Effect (SEE). A failure is the manifestation of a fault in terms of system functionality, i.e. the (partly) loss of system services.

In order to guarantee system availability, reliability and performance, the correct handling of faults such that they do not lead to a failure is essential. In spacecraft design, this is called FDIR. Fault detection is the capability of a system to identify that a fault has occurred. It is usually followed by fault isolation to determine the exact location (subsystem, memory area, etc.) of the fault. Ultimately, in the fault recovery step, the system tries to transfer to a safe state of execution in which the fault has been mitigated. This last step is usually implemented in multiple layers such that the system handles faults on the abstraction level at which they occur. Higher levels are involved in the process of fault handling only if strictly necessary.

In cases a fault cannot be handled and leads to a failure, the ECSS defines two levels of autonomy when it comes to FDIR (cf. Table 3, [3]). F1 describes the capability of a system to (partly) transfer to a safe state, report anomalies to ground and essentially wait for further instructions. A system reaching level F2 on the other hand is capable of resuming mission operations after a failure by transferring to a nominal operation configuration through reconfiguration. This may or may not include a decline in performance but the overall generation of mission products is resumed nevertheless.

A detailed survey on existing FDIR techniques and their application in space missions can be found in Section 4.

3. Anomaly Detection

One of the goals of introducing autonomy to spacecraft is to have a cognitive system, that is able to perceive its environment and internal state, draw conclusions based on its goals and act accordingly without ground intervention. In this regard, being able to recognize patterns and anomalies is important to detect off-nominal situations during mission operation.

A distinction is made between *point anomalies*, *contextual anomalies* and *collective anomalies*. For point anomalies, a data point is considered an anomaly if it is different from all the normal data points. For a data point to be considered a contextual anomaly, its surrounding data points (context) have to be taken into account. For example, a high temperature that is nominal during daytime would be considered an anomaly when observed during nighttime. For collective anomalies, a whole sequence of data points is considered anomalous, although single data points, when examined individually, may occur nominal. Obviously, the detection of contextual and especially collective anomalies is significantly harder compared to the detection of point anomalies.

Traditionally, the detection of point anomalies is performed using Out-Of-Limit (OOL) checks for a set of parameters with predetermined upper and lower bounds. Nowadays, hard and soft limits are defined to be able to report events early in the build-up of an anomaly. Furthermore, different limits can be applied according to the current state (orbital period, spacecraft mode, etc.). This also covers simple contextual anomalies. However, even these more sophisticated OOL checks cannot analyze actual patterns in the underlying data and are therefore bound to miss a lot of anomalies. In [9], an example of Venus Express is given in which a reaction wheel showed rising friction which could only be found by manual inspection.

There are three main categories of anomaly detection: Supervised, unsupervised and semi-supervised. In the supervised case, both anomalous as well as nominal sequences of data are available for training and each sequence is labeled one or the other. This has the drawback that, during operations, first-time anomalies might be missed by the system, because it has never seen them before and classifies them as nominal behaviour. In the unsupervised case, the system makes the assumption that anomalies happen far less often then nominal behaviour. Thus, it tries to learn the general pattern of the training data and classifies any deviation as an anomaly. If, however, similar anomalies can frequently be observed, the system may erroneously classify these as nominal behaviour.

In the semi-supervised case, the training data consists only of nominal data. Again, the system learns to find patterns in the underlying data but engineers can now be sure that they do not feed anomalies to the system during training.

In the following, the current state of the art in anomaly detection both inside and outside the space domain is described.

One mathematically and computationally simple approach is to calculate a set of statistical measures (minimum, maximum, average and standard deviation) for a sequence in time of a given parameter and then compute the Euclidean distance to other timesequences that have already been observed [9, 10]. The probability of having observed an outlier or anomaly is then estimated using Local Outlier Probabilities [11]. The technique has successfully been used to perform ground-based analysis of the telemetry of the ESA X-ray space observatory XMM-Newton [12].

Support Vector Machines (SVMs) present another exploitable approach to anomaly detection. SVMs are a mathematical procedure for classification and regression that transforms its input data to higher dimensions under the assumption that the data becomes linearly separable by a hyperplane. This is done using kernels as similarity functions. Typical kernels are linear, polynomial or radial basis function (RBF) kernels. During the backward transformation, this hyperplane may become a non-linear separator based on only a few training examples (support vectors) that have been found necessary. Since there may be infinitely many hyperplanes, in SVMs as opposed to other classifiers, the hyperplane minimizing the L_2 norm and maximizing the minimum margin from any data point to the hyperplane is selected. This results in a simple and robust hyperplane. SVMs present a supervised mean of machine learning. For anomaly detection, this implies that labeled training data for both nominal and off-nominal situations has to be available. The problem of having only a limited number of anomalies, that are by definition rare compared to nominal behaviour, is addressed in [13] by training an SVM in multiple steps with an increasing number of samples. Training is further accelerated by weighting input features before training according to their kernel-based distance. The results have been verified using data from the Interferometry Program Flight Experiment II (IPEX II) [14].

Another possibility is to use only nominal data which eliminates the need for expert knowledge to label the data. The proposed approach also uses kernels but is "knowledge-free" in that it does not need a priori expert knowledge [15]. Here, a timeseries of input data is transformed to a sequence of overlapping windows. These windows are transformed to higher dimensional spaces using a kernel and a Principal Component Analysis (PCA) is applied to resolve this dimensionality to manageable degree. The application of PCA returns linear combinations of the original features that represent trends in the underlying data and are uncorrelated to one another. Subsequently, the distribution of the direction of principal components (the principal component vector) from the PCA is estimated. The principal component vector of newly acquired windows can now be compared to this distribution. From its likelihood, the anomaly score is derived and the window, given that this score surpasses a given threshold, is considered anomalous. The results have been validated using simulation data of a simulated orbital transfer vehicle's thrusters provided by JAXA.

A method called Mixture of Probabilistic Principal Components Analyzers and Categorical Distributions (MPPCACD) [16] builds on dimensionality reduction and clustering. First, the multi-dimensional data is filtered for trivial outliers by using only the αth and $(100-\alpha)$th percentiles of a given timeframe and normalized according to the same percentiles. During training, a low-dimensional statistical model for each cluster of the operational data is learned using previous nominal data. By learning the distribution of both continuous as well as discrete status parameters for a number of modes including the distribution of modes itself, the approach is capable of handling high-dimensional multimodal data. The approach is verified using real-world data from JAXA's Small Demonstration Satellite SDS-4.

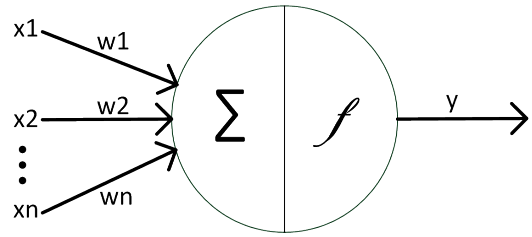

Figure 3. A single neuron as modeled in artificial neural networks [18]

Wavelet-based preprocessing is another possibility to extract frequency features of different timescales [17]. In this approach, the entire mission data is then clustered regarding these features in an unsupervised fashion using Euclidean distances. An expert then classifies a cluster that represents normal operation. For the anomalous clusters, significant features that differentiate these from the nominal cluster are generated. By presenting these significant features and the resulting clusters to an expert, the clusters are labeled for different nominal modes or as true anomalies. However, since the described approach needs access to all mission data that has been acquired in the past, this method is only suitable for ground analysis. The approach has been verified using real-world data of the power system of NASA's Lunar Atmosphere and Dust Environment Explorer (LADEE).

Moving from statistical methods to neural networks, there are several architectures tailored for different purposed. Probably, the best known is the Multi Layer Perceptron (MLP) [18, 19]. This architecture is built from artificial neurons that are supposed to model the way that actual neurons in a brain should work. This model is depicted in Figure 3. For a given input vector x and the neuron's weight vector w, the dot product is calculated which is then fed into the activation function f. The result is the neuron's output y. Typically, the activation function f is a nonlinear function like a sigmoid such that the neural network can learn nonlinear patterns. By stacking layers of parallel neurons, powerful models can be created that learn a given function of the input by comparing their output to some expected value regarding an error function. By propagating the first derivative of this error function with respect to the single weight back (back-propagation) through the whole network, the weights are slowly adjusted to match network output and expected values.

Other types of neural networks include, amongst many more, Self-Organizing Maps (SOM, also called Kohonen networks, [20]), Recurrent Neural Networks (RNN, [21]), Convolutional Neural Networks (CNN, [21]) and Long Short-Term Memories (LSTM, [22, 23]. RNNs allow, in contrast to so-called feed-forward networks like MLPs, recurrent connections such that the output of a neuron is fed back to it in the next time-step. For training, however, RNNs have problems learning long patterns in the time domain, basically because the error function's gradient is multiplied a lot of times during back-propagation, resulting in either vanishing or exploding gradients and weight changes. LSTMs overcome this issue by introducing a memory cell protected by *gates* for input, output and the memory itself (forget gate, [23]) which replaces the multiplications in the back-propagation by additions [22]. Figure 4 shows a single LSTM cell with input, output and forget gates (i,o,f) as well as the new input (in_t).

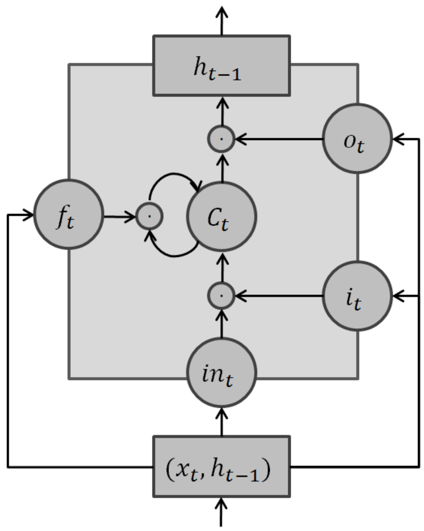

Figure 4. LSTM cell with forget gate [23]

LSTMs can be trained on a time sequence to predict the next upcoming value(s) as realized in [24]. Here, the prediction error is used to rate values as potential anomalies regarding a dynamic threshold. This threshold is set such that the removal of all outliers above it results in the maximum decrease of errors while penalizing by the number of found anomalies. The experiments are validated using data from the Soil Moisture Active Passive (SMAP) satellite and the Mars Science Laboratory (MSL) rover, Curiosity.

Autoencoders, while not a specific type of network by themselves, (cf. Figure 5) present a way of assembling neural cells of any kind such that the network learns a compact representation (i.e. encoding) of the data in one of its hidden layers. At some point in the network, it is forced to reduce the dimensionality of the input data (cf. grey box in Figure 5) and therefore has to extract the most important features of the training data by itself.

In [25], the authors describe an approach using autoencoders to reduce the dimensionality of the telemetry data. Under the assumption that outliers or anomalies are hard to represent in a smaller feature space, a model is trained to represent a the data in a lower dimension while still being able to reconstruct the original value. Figure 5 gives a schematic overview of the approach. In [25], a time-series of features is enriched by extracting overlapping sliding windows to model temporal dependencies and by calculating features from these windows for comparing different sliding windows. The data is then fed to a parallel CNN- and LSTM-based autoencoder that is trained to reconstruct the original data. The magnitude of the reconstruction error is then an indication for the degree to which the input data might be an outlier.

A similar approach of LSTM-based autoencoders is developed in [26] and [27]. The former is very similar to [25], but thresholding for anomalies is done using a normal distribution of the absolute errors and a likelihood estimation for any observed reconstruction error.

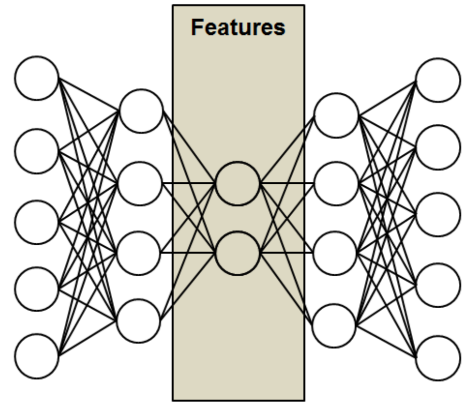

Figure 5. Neural network-based autoencoder for anomaly detection

In [27], a MLP-based autoencoder together with an LSTM are applied to ground-based anomaly detection in satellite telemetry data at the German Space Operations Center (GSOC). The autoencoder is responsible for automatic feature extraction that is combined with statistical features for a given time window. The LSTM is then used to predict the upcoming 4.5 hours of telemetry data. Anomaly detection is performed both on the LSTM-generated data to predict future anomalies as well as on the extracted features of observed telemetry data using statistical clustering and a variant of the Intrinsic Dimensional Outlier Score (IDOS, [28]). The approach is validated using real-world data from the TET-1 satellite.

A summary of approaches for thermal anomaly detection, polar cap edge detection and aerosol opacity estimation for the Mars Odyssey spacecraft are described in [29]. Detection of thermal anomalies on Mars using data of the THEMIS instrument is performed by counting the number of pixels in a thermal image that exceed a certain threshold. If this number is in a certain range, the image is flagged as containing a thermal anomaly. Polar cap edge detection is also performed using image data from the THEMIS instrument. In the temperature histogram, a dip can be observed when in orbit over the polar cap edge. By setting a threshold according to this dip, the spacecraft can reliably determine whether it is over the polar region. The analysis of these histograms also led to more evidence of water ice on Mars since their presence in a picture results in a very specific in the histogram. Ultimately, the detection of high opacity events in the Martian atmosphere was performed by using SVM regression. These events can be indicators for upcoming dust storms and also increase the data quality for surface mineralogy from orbit.

An approach from the domain of robotics towards semantic event or anomaly detection and classification based on time-series shapelets is described in [30]. Time-series shapelets are short, potentially multi-dimensional characteristic sequences of sensor readings that can be used to identify known events in a stream of data.The authors successfully use these shapelets to detect events during wiping actions performed by a robotic arm.

4. FAULT DETECTION, ISOLATION AND RECOVERY

In the last section about anomaly detection, the techniques presented were mostly data driven, thus providing little insight into the space system. While this may be enough to detect whether the system exhibits some strange behaviour or whether *some* fault was observed, for isolating and recovering from the fault (i.e. the upper FDIR layers), a deep understanding of the underlying system, its actions, limitations and behaviour is crucial.

Hence, designing and implementing an FDIR concept is amongst the most complex tasks in a spacecraft development because it involves all subsystems and is very critical for system availability, reliability and performance. Current FDIR processes built on the results of Failure Modes, Effects, and Criticality Analysis (FMECA) and Fault Tree Analysis (FTA). However, these can only applied late in the development process which prohibits FDIR to become an integral part of the system. An extensive survey on the current state of the art of FDIR approaches and emerging techniques, such as model-based methods, to overcome their limitations can be found in [31]. Surveys focusing on the techniques that have the potential to leverage current FDIR designs and make them more autonomous can be found in [32, 33].

In the following paragraphs, a selection of FDIR techniques from the domain of artificial intelligence as well as the FDIR design of ESA's Herschel and Planck space telescopes are discussed.

In [34], an approach for fault detection and isolation (classification) using a combination of PCA, binary and multi-class SVMs is presented. The process of detection and classification is depicted in Figure 6.

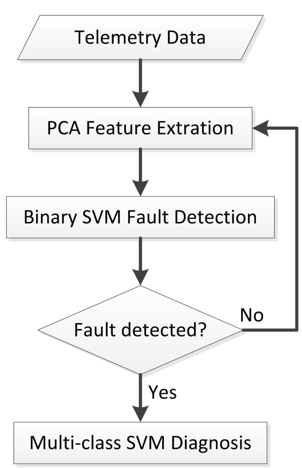

Figure 6. PCA- and SVM-based Fault Detection and Isolation [34]

In a first step, the telemetry data is mapped to a lower dimensional space by PCA. In the next step, the data point is classified by a binary SVM as representing a nominal or fault state. In case a fault is detected, the datapoint is passed to the fault classification performed by a multi-class SVM in a One-Against-All (OAA) fashion. Both SVMs are trained using telemetry data that has been manually labeled with its respective fault state.

Herschel and Planck were two space telescopes jointly developed and launched to Earth's L2 point by ESA in 2009. Planck mapped the cosmic microwave background (CMB) at microwave and infra-red frequencies. Herschel was observing in the far infrared and submillimetre wavebands, mainly focusing on galaxy and star formation regions. Both telescopes remained operational until 2013. To streamline the development, Herschel and Planck shared a high-level FDIR approach based on two different FDIR modes:

- **Autonomous Fail Safe (AFS)**: Events and anomalies are detected and sent to ground. No on-board reconfiguration is performed but the system enters a safe mode when detecting serious problems such as unit failures. This safe mode requires ground assistance for the spacecraft to become operational again.
- **Autonomous Fail Operation** (AFO): On unit failure, the system is allowed to reconfigure autonomously and switch to redundant units, thus increasing the operational time that can be used for generating science.

In the initial phases of the Herschel and Planck missions, the AFS was used to preserve spacecraft safety over operational time. Once the spacecraft were operational, the FDIR mode was switched to AFO to increase science gains and reduce downtime. A complete description of the attitude control computer's FDIR approach can be found in [35].

Bayesian Networks (BN) are graphical models for representing conditional dependencies between variables such as causes and effects. Efficient algorithms for inference in these graphs exist and can even be compiled to arithmetic circuits for fast execution. In FDIR, these nets are commonly used for deducing the cause or origin of a failure given a set of observable variables (symptoms). Dynamic Bayesian Networks (DBN) can also capture probabilistic dependencies between time slices. For example, this can be exploited for modeling the aging of components over time.

In [36], DBNs are used to model and counteract failures in the power generation of a simulated mars rover. Nodes for different failures of the solar arrays and battery strings as well as failure scenarios and recovery plans and their probabilistic dependencies are modeled. Inference on the DBN can then determine whether the current system state is nominal, anomalous or failed. If a failure was detected, the inference can propose a suitable recovery plan. In case of an anomaly, a preventive recovery plan may be advised. Apart from analyzing the current system state, the proposed FDIR mechanism is also capable of executing a prognostic state estimation than can also trigger preventive recovery. The process is depicted in Figure 7.

Dynamic Fault Trees (DFT) or rather their inherently non-deterministic extension (NdDFT) in combination with probabilistic automata can yield another way way of modeling the system's recovery strategy in the presence of faults. In [37], so-called Recovery Automata are synthesized that - when optimized for a specific Markovian Decision Process (MDP) - output the optimal recovery strategy from a reliability point of view. The approach was originally proposed in [38].

Techniques of artificial intelligence can, however, not only assist in fault handling in the form of FDIR, but can also play a vital role during spacecraft's nominal operation, its scheduling and planning. Examples of (partly) autonomous operations in practice are given in the next section.

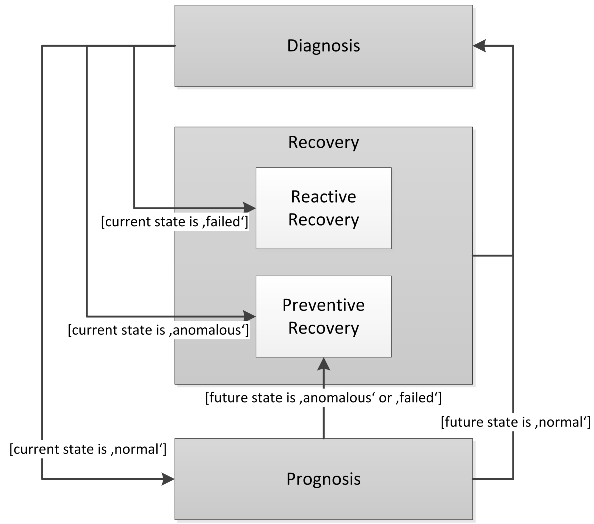

Figure 7. DBN-based Fault Detection and Isolation [36]

5. Autonomous Operations

Especially deep space missions require at least partially autonomous operations to cope with long communication delays. A shining example for this problem is Mars descent that usually takes seven minutes, the so-called "seven minutes of terror". Even with an optimal round-trip time of six minutes for a signal from Mars to Earth and back, there is no way this manoeuver could be remotely controlled.

For orbital systems, cost and efficiency tend to be the driving factors of introducing autonomy. Increasing the time a spacecraft can operate without human intervention greatly reduces its operational cost. At the same time, autonomous science data acquisition can optimize the use of available spacecraft resources (i.e. power, transfer budget, etc.).

The following sections introduces a number of examples of autonomous operations that are applied in past and current space missions.

In a survey from 2018 [39], the use of OBCPs as a means of (simple) closed-loop adaptive control and of Markov decision processes is evaluated. OBCPs are part of the ECSS Packet Utilization Standard (PUS) and are defined in the ECSS-E-ST-70-01C standard [2]. The basic concept - in contrast to traditional telecommands and the time-based Mission Timeline (MTL) - is depicted in Figure 8. In contrast to traditional flight procedures that require a synchronous spacecraft-ground communication for each step, OBCPs enable spacecraft to execute more sophisticated control flows autonomously after being triggered either from ground or by specific on-board events. This introduces a lot of flexibility during mission operations as well as capabilities for on-board autonomy. Examples of the application of such OBCPs on board ESA spacecraft are given in [40] for the Herschel and Planck satellites, in [41] for Rosetta and Venus Express and in [42] for BepiColombo.

While Rosetta (cf. Figure 9) and Venus Express (cf. Figure 10) shared the same OBCP capabilities, the development of OBCPs was introduced in very different stages of development. Rosetta was relying heavily on OBCPs due to signal lead times of up to 100 minutes. For Venus Express, however, OBCPs were only developed after positive experience from Rosetta was gathered. They facilitated mission operations, mainly at the beginning and end of each contact. Additional autonomous FDIR functionality through OBCPs was added later.

Figure 9. Artist's impression of Rosetta and its lander Philae [43]

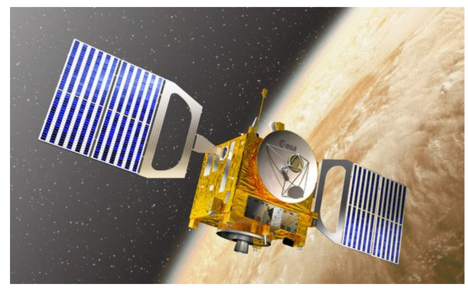

Figure 10. Artist's impression of Venus Express [43]

The Herschel and Planck satellites (cf. Figure 11) added the integration of the system database via XML bridge files allowing easy generation and handling of telemetry and telecommands within the OBCPs. The software for the Spectral and Photometric Imaging Receiver (SPIRE) instrument on board Herschel follows a concept similar to OBCPs [44]. The system uses tables of command sequences that are interpreted during runtime by a Virtual Machine (VM). This allows for easy adaptions and patching of the payload operations without verification, validation and upload of an entire software image.

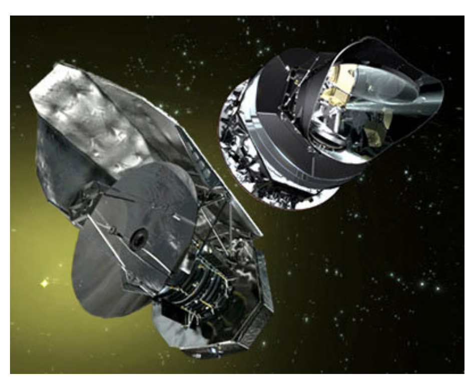

Figure 11. Artist's impression of the Herschel (left) and Planck (right) satellites [43]

BepiColombo integrated a two-stage scheduling for so-called emergency OBCPs whose executing precedes the execution

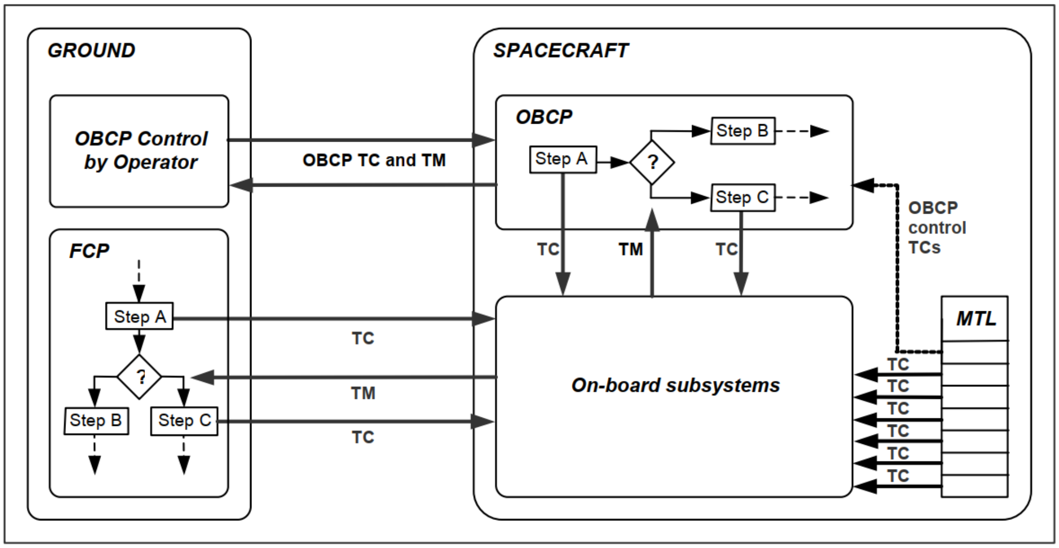

Figure 8. Traditional TC- and MTL-based operations vs. OBCPs [41]

of regular OBCPs. This makes the system more reactive in the presence of mission critical events. An exploded view of the BepiColombo components is displayed in Figure 12.

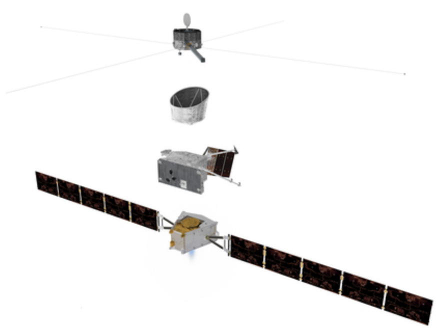

Figure 12. Exploded view of the BepiColombo stack. From top to bottom: Mercury Transfer Module, Mercury Planetary Orbiter, Sunshield and Interface Structure and Mercury Magnetospheric Orbiter [43]

For Rosetta, not only the operational concept involved autonomous behaviour, but also the scheduling of science campaigns for the eleven on-board instruments [45, 46]. The mission is broken down into 16 week long-term plans that are subsequently refined to four week medium-term plans and one week short-term plans. Based on the spacecrafts trajectory, a number of planned observations can be created, resulting in a rough pointing plan that is systematically refined down to instrument timelines (command sequences) and a pointing timeline to follow. By scheduling instrument activations according to windows of opportunity resulting from a number of constraints (pointing, altitude, sun angle, ...), a sequence of pointing and slew commands as well as instrument activations making use of Rosetta's resources can be found in roughly 20 minutes.

On Earth Observing-1 (EO-1, cf. Figure 13), three algorithmic experiments have been conducted [47]. Two of these deal with cloud detection using Random Decision Forests (RDFs) and Bayesian Thresholding (BT), the third provides saliency-based novelty detection. All algorithms rely on a limited number of frequency bands from the on-board hyperspectral imaging data. Using data from previous mission phases, both cloud detection algorithms were trained to drop useless images from the telemetry downstream. Both algorithms reached an accuracy of more than 90% with the RDFs slightly outperforming BT, which was, however, faster in runtime. Based on the intensity histogram of a given window within an image, the novelty detection was able to detect unknown and anomalous objects such as small lakes and buildings in remote locations. Images containing anomalies could then be prioritized for the downlink.

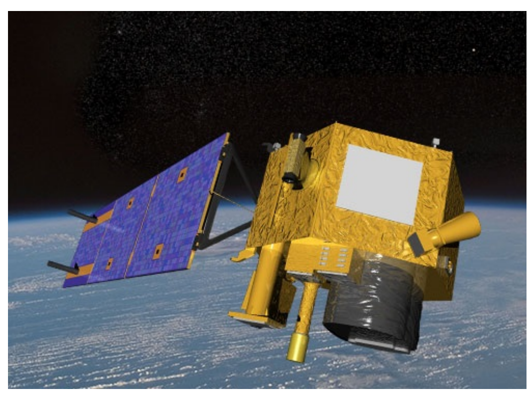

Figure 13. Artist's impression of the Earth Observing-1 satellite [48]

In [49], the development of an optical navigation system (Natural Feature Tracking, NFT) for the Origins Spectral Interpretation Resource Identification Security Regolith Explorer (OSIRIS-REx, cf. Figure 14) mission to the asteroid Bennu is described. The subsystem was introduced late in the development to serve as a backup for the Flash Lidar navigation during the Touch And Go (TAG) manoeuver for collecting an asteroid sample. Based on images acquired during the mission, a catalog of Digital Terrain Maps (DTMs) is created from which the NFT can render expected images including spacecraft and asteroid attitude and sun angle. Correlating these rendered images to actual pictures taken by the on-board cameras results in a precise determination of the spacecrafts position and attidude as well as a prediction of its future trajectory in the complex gravitational field of Bennu.

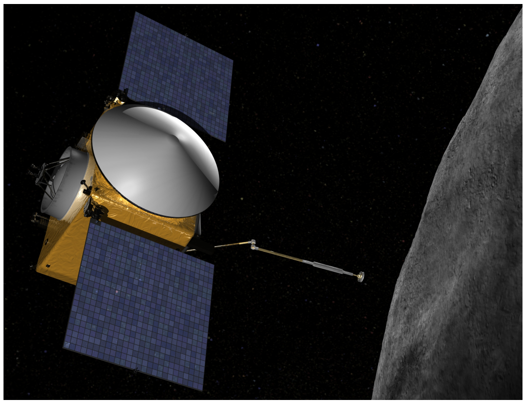

Figure 14. Artist's impression of the OSIRIS-REx spacecraft [48]

The Intelligent Payload Experiment (IPEX, cf. Figure 15) cubesat was a technology demonstrator for autonomous operations and on-board image processing [50]. In addition to the ground-based scheduling system, that had already been used for Rosetta (Automated Scheduling and Planning Environment, ASPEN), for the generation of earth observation data products, IPEX also features an on-board software component called Continuous Activity Scheduler Planner Execution and Re-planner (CASPER). CASPER takes into account real-time information and system resources to adapt the uploaded schedule and also generated observation goals when non were given by its users. Concerning image processing, IPEX demonstrates a variety of algorithms on board:

- Random forest classifiers for cloud and planetary disk detection [50]
- Unsupervised saliency analysis for unknown feature detection based on differences in spatial intensity [50]
- An SVM classifier for autonomous detection of cryospheric changes in lake and sea ice [51]
- Superpixel segmentation and spectral unmixing techniques for endmember detection [52]

Altogether, IPEX autonomously generated 30,000 image products from 450 images based on user requests and schedules that were refined on board.

Figure 15. The IPEX cubesat [48]

In [53], simple three-layer ANNs are proposed for detection of impacts on the simulated Didymos system and plumes around Comet 67P/Churyumov-Gerasimenko. On event detection, the spacecraft could initiate the generation of science products autonomously without any commanding from ground. Due to the rather simplistic models, their weights can directly be visualized for evaluation of the training process. Initial results offered an average detection performance of over 90% for both application scenarios.

The Mobile Asteroid surface SCOuT (MASCOT, cf. Figure 16) was developed by DLR in collaboration with CNES as a compact lander for JAXA's Hayabusa2 mission to Near Earth Asteroid (162173) Ryugu [54]. Due to its limited battery capacity for only 16 hours of operation and long signal run times, the surface operations of the lander's four instruments had to be executed autonomously without ground intervention. As part of the on-board software, the MASCOT Autonomy Manager (MAM) was implemented as a state machine and corresponding transition logic. The MAM's task was to decide whether MASCOT's attitude after touchdown or a relocation manoeuver allowed the execution of science experiments, activate the science instruments according to their predefined order taking into account system resources and states and execute a relocation (hopping) manoeuver after the first landing site had been examined or found infeasible.

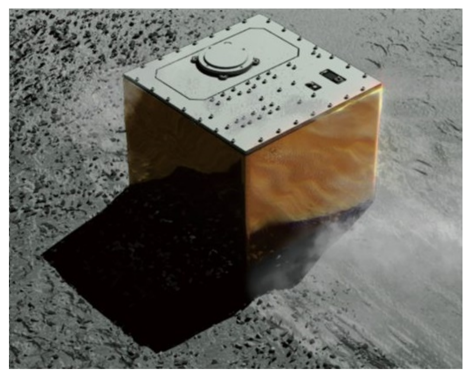

Figure 16. Artist's impression of the MASCOT lander [55]

In ESA's space astrometry mission Gaia (cf. Figure 17), which is situated in the Sun-Earth L2, makes a three-dimensional map of our galaxy containing one billion stars [56]. Gaia's on-board software is capable of autonomously detecting (double) stars, unresolved external galaxies and asteroids and discriminating them from spurious objects like cosmic rays or solar protons. This is done using the spacecraft's various CCD detectors for object detection and confirmation through local maxima in the image data and shape estimation along and across the CCDs' scan direction. Furthermore, the ground segment of Gaia is able to autonomously generate science alerts by comparing new discoveries or diverging star magnitudes to its existing star catalogue [57].

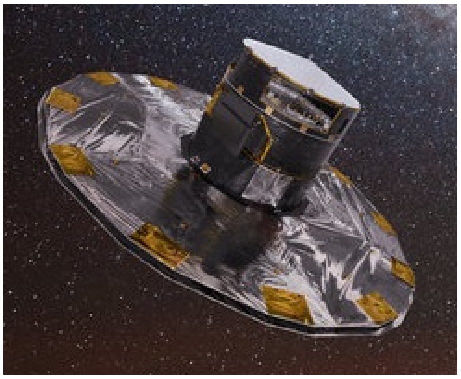

Figure 17. Artist's impression of the Gaia satellite [43]

The Exobiology on Mars (ExoMars) mission is a two-phase ESA/Roskosmos project to search for past or present life on Mars. The first phase, launched in March 2016 and reaching Mars in October 2016, consisted of the Trace Gas Orbiter (TGO) and its accompanying lander Schiaparelli that did, however, not survive the landing due to internal software problems caused by saturated variables for rotation rates during parachute deployment [58]. The second phase, currently planned to launch in 2020, consists of a Russian-built landing platform and a European-built rover (cf. Figure 18).

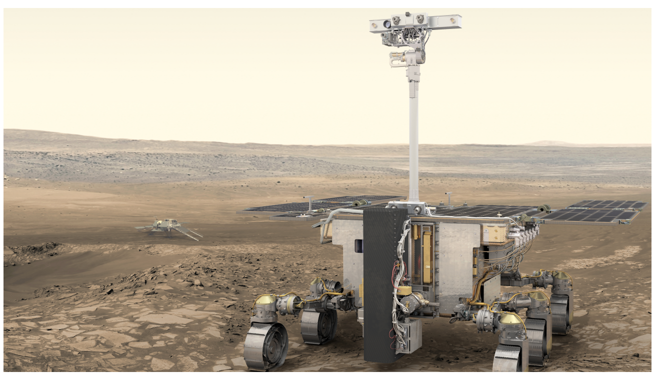

Figure 18. Artist's impression of the ExoMars rover [43]

For navigation and image retrieval, the ExoMars rover features a panning and tilting optical bench with two Wide-Angle Cameras (WAC) and one High-Resolution Camera (HRC). In order to assist the odometry on Mars, the rover has a Visual Data Fusion (VDF) system using a technique called Simultaneous Localization and Mapping (SLAM) and already sensed information from a Geographical Image Database Server (GIDS). This system helps in position determination as well as object detection and path planning. Both object detection and path planning use SOMs and are capable of learning in an online and real-time fashion [59].

6. Related Work

A wide area of application beyond anomaly detection, FDIR, planning and detection of science opportunities are Guidance, Navigation and Control (GNC) and Attitude and Orbit Control (AOC). While an extensive review of these techniques is beyond the scope of this paper, a corresponding survey can be found in [60].

Techniques of artificial intelligence are not only applied in the space domain, but are very famous in other domains as well, especially when a lot of data is involved. One example is the analysis of electrocardiography (ECG) data via LSTM networks [61]. At each timestep t, the current datapoint is used to predict the next l values. An error vector of the predictions and the actual values is then compared to a threshold that is derived from the data and if it is exceeded by the error, the series is marked anomalous.

A current trend in deep learning is using so-called pretrained models such as freely available award winning networks such as the ImageNet Challenge (cf. [62]) and altering them briefly to the actual application's requirements. This eliminates the need for large datasets because it exploits the already-learned features of the pretrained network and leaves the engineer with the task of finding a suitable model for the actual inference. One application of this technique to earth observation data can be found in [63].

The application of techniques of artificial intelligence often comes with an increased need for computational power. This can either be satisfied by dedicated processing hardware as the neural network processor for DLR's BIRD satellite (cf. [64]) or by more powerful all-purpose computers, ideally tailored to the application. One example of such a scalable system is the Scalable On-Board Computing for Space Avionics (ScOSA, cf. [65]) project, currently developed at DLR. It features Reliable Computing Nodes (RCN) for critical system tasks as well as High-Performance Nodes (HPN) to offer a larger amount of computational power.

With the depicted techniques of anomaly detection at hand, another approach developed at DLR is the context-aware compression of spacecraft housekeeping data. The housekeeping data of LEO spacecraft usually follows one or more regular patterns. Given previously gathered data samples, these patterns or models can be learned by techniques of artificial intelligence. Using suitable compression techniques (cf. [66]), the spacecraft's OBC can the decide in which detail housekeeping data should be reported to ground based on the adherence or deviation from the learned models.

An important topic outside the scope of this paper, especially for the safety-driven space domain, is the task of validating autonomous (sub-)systems. A survey of this field can be found in [67].

7. Summary

This survey presents a starting point to understand the concept of artificial intelligence and machine learning, potential, requirements and limitations with a strong focus on the space domain.

We have given an introduction to the terminology of artificial intelligence and machine learning in the space domain. Building upon this terminology, we introduced important techniques suitable for the application on board and on ground in the fields of anomaly detection and FDIR. Finally, we surveyed recent and upcoming space missions for their application of artificial intelligence to show concepts of (partly) autonomous mission operations.

Future missions have a tendency to rely more and more on autonomous systems to meet safety and cost requirements and be as reactive as possible. Techniques of artificial intelligence and machine learning show the potential to not only assist in mission operations, planning and scheduling but also to enable new missions that require immediate action by the spacecraft without the possibility to shift important decisions to ground.

REFERENCES

[1] A. K. Jónsson, R. A. Morris, and L. Pedersen, "Autonomy in space: Current capabilities and future challenge," *AI Magazine*, vol. 28, pp. 27–42, 2007.

[2] "Spacecraft On-Board Control Procedures (ECSS-E-ST-70-01C)," European Cooperation for Space Standardization (ECSS), Tech. Rep., April 2010.

[3] "Space Segment Operability (ECSS-E-ST-70-11C)," European Cooperation for Space Standardization (ECSS), Tech. Rep., July 2008.

[4] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Malaysia; Pearson Education Limited,, 2016.

[5] M. D. Watson, S. B. Johnson, and L. Trevino, "System engineering of autonomous space vehicles," in *Prognostics and Health Management (PHM), 2014 IEEE Conference on*. IEEE, 2014, pp. 1–8.

[6] A. Jones and J. Straub, "The Application of Deep Learning to Space Missions," in *68th International Astronautical Congress (IAC)*, 2017.

[7] P. Dagum, A. Galper, and E. Horvitz, "Dynamic network models for forecasting," in *Proceedings of the Eighth International Conference on Uncertainty in Artificial Intelligence*, ser. UAI'92. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1992, pp. 41–48.

[8] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection for discrete sequences: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 823–839, May 2012.

[9] J.-A. Martínez-Heras and A. Donati, "Enhanced telemetry monitoring with novelty detection," *AI Magazine*, vol. 35, no. 4, pp. 37–46, 2014.

[10] J. Martinez, "New telemetry monitoring paradigm with novelty detection," in *SpaceOps 2012*, 2012.

[11] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "LoOP: local outlier probabilities," in *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 1649–1652.

[12] M. Kirsch, "Cage Instability of XMM-Newton's Reaction Wheels Discovered during the Development of an Early Degradation Warning System," in *SpaceOps 2012*, 2012, p. 1275204.

[13] D. DeCoste and M. B. Levine, "Automated event detection in space instruments: A case study using ipex-2 data and support vector machines," in *SPIE Conference on Astronomical Telescopes and Instrumentation*, 2000.

[14] M. B. Levine, "Interferometry program flight experiments: IPEX I and II," in *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 3350, 1998, pp. 3350 – 3350 – 12.

[15] R. Fujimaki, T. Yairi, and K. Machida, "An approach to spacecraft anomaly detection problem using kernel feature space," in *Proceedings of the* 11^{th} *ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 2005, pp. 401–410.

[16] T. Yairi, N. Takeishi, T. Oda, Y. Nakajima, N. Nishimura, and N. Takata, "A data-driven health monitoring method for satellite housekeeping data based on probabilistic clustering and dimensionality reduction," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 3, pp. 1384–1401, June 2017.

[17] G. Biswas, H. Khorasgani, g. stanje, A. Dubey, S. Deb, and S. Ghoshal, "An approach to mode and anomaly detection with spacecraft telemetry data," *International Journal of Prognostics and Health Management*, vol. 7, p. 18, 01 2016.

[18] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, Dec 1943.

[19] F. Rosenblatt, *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957.

[20] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, Jan 1982.

[21] S. Haykin, *Neural networks*. Prentice hall New York, 1994, vol. 2.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[23] F. A. Gers, J. A. Schmidhuber, and F. A. Cummins, "Learning to forget: Continual prediction with lstm," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000.

[24] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '18. ACM, 2018, pp. 387–395.

[25] T. Kieu, B. Yang, and C. S. Jensen, "Outlier detection for multidimensional time series using deep neural networks," in *19th IEEE International Conference on Mobile Data Management (MDM)*, June 2018, pp. 125–134.

[26] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *23rd ESANN – European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2015.

[27] C. OMeara, L. Schlag, and M. Wickler, "Applications of deep learning neural networks to satellite telemetry monitoring," in *2018 SpaceOps Conference*, 2018, p. 2558.

[28] J. Von Brünken, M. E. Houle, and A. Zimek, "Intrinsic dimensional outlier detection in high-dimensional data," Technical report, National Institute of Informatics, Tokyo, Tech. Rep., 2015.

[29] R. Castano, K. L. Wagstaff, S. Chien, T. M. Stough, and B. Tang, "On-board analysis of uncalibrated data for a spacecraft at mars," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '07. ACM, 2007, pp. 922–930.

[30] S. Stelter, G. Bartels, and M. Beetz, “Multidimensional time series shapelets reliably detect and classify contact events in force measurements of wiping actions,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, Canada, 2017.

[31] M. Tipaldi and B. Bruenjes, “Survey on fault detection, isolation, and recovery strategies in the space domain,” *Journal of Aerospace Information Systems*, vol. 12, no. 2, pp. 235–256, 2015.

[32] A. Guiotto, A. Martelli, and C. Paccagnini, “Smart-fdir - use of artificial intelligence in the implementation of a satellite fdir,” *European Space Agency, (Special Publication) ESA SP*, vol. 532, p. 71, 01 2003.

[33] A. Wander and R. Förstner, “Innovative fault detection, isolation and recovery strategies on-board spacecraft: state of the art and research challenges.” Deutsche Gesellschaft für Luft-und Raumfahrt-Lilienthal-Oberth eV, 2013.

[34] Y. Gao, T. Yang, N. Xing, and M. Xu, “Fault detection and diagnosis for spacecraft using principal component analysis and support vector machines,” in *7th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 2012, pp. 1984–1988.

[35] J. F. T. Bos, D. Zorita, A. Bacchetta, G. Chlewicki, D. Guichon, and I. Rasmussen, “Acms fdir system for the herschel/planck satellites,” vol. 606, p. 36, 12 2005.

[36] D. Codetta-Raiteri and L. Portinale, “Dynamic bayesian networks for fault detection, identification, and recovery in autonomous spacecraft,” *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, vol. 45, pp. 13–24, 01 2015.

[37] S. Müller, A. Gerndt, and T. Noll, “Synthesizing fdir recovery strategies from non-deterministic dynamic fault trees,” in *AIAA SPACE and Astronautics Forum and Exposition*, 2017, p. 5163.

[38] M. Beccuti, G. Franceschinis, D. C. Raiteri, and S. Haddad, “Computing optimal repair strategies by means of ndrft modeling and analysis,” *Comput. J.*, vol. 57, no. 12, pp. 1870–1892, 2014.

[39] M. Tipaldi and L. Glielmo, “A survey on model-based mission planning and execution for autonomous spacecraft,” *IEEE Systems Journal*, vol. 12, no. 4, pp. 3893–3905, Dec 2018.

[40] M. Ferraguto, T. Wittrock, M. Barrenscheen, M. Paakko, and V. Sipinen, “The on-board control procedures subsystem for the herschel and planck satellites,” in *32nd Annual IEEE International Computer Software and Applications Conference*, July 2008, pp. 1366–1371.

[41] C. Steiger, R. Furnell, and J. Morales, “On-Board Control Procedures for ESA’s Deep Space Missions Rosetta and Venus Express,” in *Data Systems in Aerospace (DASIA)*, vol. 602, 07 2005, p. 60.

[42] A. Schwab, “OBCPs - An Integrated Part of Bepi-Colombo Autonomy and Operations Flexibility,” in *SpaceOps Conference*, 2012.

[43] “European space agency,” http://www.esa.int, accessed: 2019-08-26.

[44] S. Liu, S. Molinari, A. Di Giorgio, S. Pezzuto, S. Sidher, P. Cerulli-Irelli, and K. King, “Herschel-spire satellite instrument: configurable on-board software for autonomous and real time operation,” *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 8442, pp. 45–, 09 2012.

[45] S. A. Chien, G. Rabideau, D. Tran, M. Troesch, J. Doubleday, F. Nespoli, M. P. Ayucar, M. C. Sitja, C. Vallat, B. Geiger *et al.*, “Activity-based scheduling of science campaigns for the rosetta orbiter.” in *International Joined Conference on Artificial Intelligence (IJCAI)*, 2015, pp. 4416–4422.

[46] M. Costa, M. Perez-Ayucar, M. Almeida, M. Ashman, R. Hoofs, S. Chien, F. Nespoli, J. Jos Garca, and M. Kueppers, “Rosetta: rapid science operations for a dynamic comet,” in *14th International Conference on Space Operations*, 05 2016.

[47] K. L. Wagstaff, A. Altinok, S. A. Chien, U. Rebbapragada, S. R. Schaffer, D. R. Thompson, and D. Q. Tran, “Cloud filtering and novelty detection using onboard machine learning for the eo-1 spacecraft,” 2017.

[48] “National Aeronautics and Space Administration,” http://www.nasa.gov, accessed: 2019-08-26.

[49] D. A. Lorenz, R. Olds, A. May, C. Mario, M. E. Perry, E. E. Palmer, and M. Daly, “Lessons learned from osiris-rex autonomous navigation using natural feature tracking,” in *2017 IEEE Aerospace Conference*, March 2017.

[50] S. Chien, J. Doubleday, D. R. Thompson, K. L. Wagstaff, J. Bellardo, C. Francis, E. Baumgarten, A. Williams, E. Yee, E. Stanton *et al.*, “Onboard autonomy on the intelligent payload experiment cubesat mission,” *Journal of Aerospace Information Systems*, pp. 307–315, 2016.

[51] T. Doggett, R. Greeley, S. Chien, R. Castano, B. Cichy, A. Davies, G. Rabideau, R. Sherwood, D. Tran, V. Baker, J. Dohm, and F. Ip, “Autonomous detection of cryospheric change with hyperion on-board earth observing-1,” *Remote Sensing of Environment*, vol. 101, no. 4, pp. 447 – 462, 2006.

[52] D. R. Thompson, B. J. Bornstein, S. A. Chien, S. Schaffer, D. Tran, B. D. Bue, R. Castao, D. F. Gleeson, and A. Noell, “Autonomous spectral discovery and mapping onboard the eo-1 spacecraft,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 6, pp. 3567–3579, June 2013.

[53] L. Feruglio, L. Franchi, and S. Corpino, “Deep Learning for Event Detection: Autonomous Operations for Interplanetary Missions,” in *68th International Astronautical Congress (IAC)*, 2017.

[54] T.-M. Ho *et al.*, “Mascot—the mobile asteroid surface scout onboard the hayabusa2 mission,” *Space Science Reviews*, vol. 208, no. 1, pp. 339–374, Jul 2017.

[55] “Deutsches Zentrum für Luft- und Raumfahrt,” http://www.dlr.de, accessed: 2019-08-26.

[56] J. De Bruijne, M. Allen, S. Azaz, A. Krone-Martins, T. Prodhomme, and D. Hestroffer, “Detecting stars, galaxies, and asteroids with gaia,” *Astronomy & Astrophysics*, vol. 576, p. A74, 2015.

[57] L. Wyrzykowski, “First year of the gaia science alerts,” in *37th Meeting of the Polish Astronomical Society*, vol. 3, 2016, pp. 65–68.

[58] “Schiaparelli landing investigation completed,” http://exploration.esa.int/mars/59184-schiaparelli-landing-investigation-completed/, accessed: 2019-08-15.

[59] P. Yuen, Y. Gao, A. Griffiths, A. Coates, J.-P. Muller, A. Smith, D. Walton, C. Leff, B. Hancock, and D. Shin, "ExoMars Rover PanCam: Autonomy and Computational Intelligence," *Computational Intelligence Magazine, IEEE*, vol. 8, pp. 52–61, 11 2013.

[60] D. Izzo, M. Mrtens, and B. Pan, "A survey on artificial intelligence trends in spacecraft guidance dynamics and control," 12 2018.

[61] S. Chauhan and L. Vig, "Anomaly detection in ecg time signals via deep long short-term memory networks," in *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Oct 2015, pp. 1–7.

[62] "ImageNet Challenge," http://www.image-net.org, accessed: 2019-08-26.

[63] D. Marmanis, M. Datcu, T. Esch, and U. Stilla, "Deep learning earth observation classification using imagenet pretrained networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 1, pp. 105–109, 2016.

[64] W. Halle, H. Venus, and W. Skrbek, "Thematic data processing on board the satellite bird," in *SPIE's 45th Annual Meeting , 30 July - 4 August 2000, San Diego, California, USA*, ser. Proceedings of SPIE, M. R. Descour and S. S. Shen, Eds., vol. 4132. SPIE, 2000, pp. 118–125.

[65] T. Gärtner, C. J. Treudler, F. Dannemann, and M. Jetzschmann, "Scalable Avionics for the DLR Micro- and Minisatellite Platforms S2TEP and COMPSAT," in *DASIA – Data Systems In Aerospace*, 2017.

[66] J.-G. Meß, R. Schmidt, and G. Fey, "Adaptive Compression Schemes for Housekeeping Data," in *2017 IEEE Aerospace Conference*, March 2017, pp. 1–12.

[67] C. Ebert and M. Weyrich, "Validation of Autonomous Systems," *IEEE Software*, vol. 36, no. 5, pp. 15–23, Sep. 2019.