IAC-18-B6.3.10.47050

Simulation for goal-based mission continuation on-board interplanetary spacecraft

Alexandra Wander[a,b], Roger Förstner[a,c]

[a]*Institute of Space Technology and Space Applications (ISTA), Universität der Bundeswehr München, Germany*
[b]*alexandra.wander@unibw.de*
[c]*roger.foerstner@unibw.de*

Abstract

ESAs 5th cornerstone mission BepiColombo to the innermost planet, Mercury, serves as an example and case study for an interplanetary spacecraft that has to operate in a particularly harsh environment. After decades of development and seven years of ight time, there will only be one year of nominal scientic mission, as is common for highly complex, scientific space probes.

To ensure the maximum scientic benefit within this limited time frame in the future, a high level of on-board autonomy would be beneficial to avoid unnecessary safe mode events and ensure safe mission continuation even in case of no link to ground. By means of this simulation and testing concept, the increasing resilience by enhanced self-awareness of the spacecraft in the fault management domain on system level is demonstrated. The work to be presented introduces the respective simulation concept, its set-up and demonstrates its possibilities. For simulation purposes, a test concept is introduced that features a goal-based mission continuation strategy with eight levels of increasing complexity and fault/failure scenarios implemented in a cognitive recovery unit. The simulation consists of three parts, each of which will be presented: the system simulation of all relevant spacecraft subsystems, the independent goal-based mission continuation knowledge base and processing as well as the graphical user interface to introduce faults and failures and control the simulation process. First achievements of the simulation and considerations for future developments conclude the paper.

Keywords: simulation, spacecraft systems, goal-based mission continuation

1. Introduction

The motivation for this paper derives from the challenge of designing and operating unique, highly complex one-of-a-kind spacecraft on scientific missions in hardly known environments of other planets within the solar system. There are numerous examples of great science missions, that involve a decade of design and technology development in order to meet their scientific goals in only a fraction of that time. For future missions, today's technology enables an enhanced implementation of on-board autonomy, especially in system level fault management applications.

The developed simulation environment shall support the testing of a goal-based mission continuation approach on system level. This paper presents its software architecture, mathematical models, possibilities and limitations.

First, general space mission operations and goal-based mission continuation are introduced. Section 3 gives an overview of the mission to Mercury and its planetary environment. Section 4 describes spacecraft system simulation tool with its mathematical models and explains the graphical user interface as well as the implementation of the cognitive recovery unit. Section 5 provides an overview of first achievements and future developments conclude the paper.

2. Space Mission operations

The operational task of space missions is divided between the ground segment (control systems and operators) and the space segment (spacecraft). To cite [1], traditional space mission operations utilizes time-based command sequencing, where commands are planned to be executed at pre-defined instances in time and where telemetry is retrieved for operators to determine if the planned activities were accomplished. Today, cosidering the increasing computing power for both segments and the increasing experience of operator's from previous missions, more and more capabilites are transferred to the space segment for automated execution on-board even if no link to ground is available.

2.1. Fault management on-board spacecraft

Interplanetary space mission operations seldom include a known and predictable environment, change and unforeseen events are given in numerous situations. The spacecraft experiences natuarally limited on-board resources,

Table 1: Mission execution autonomy levels as defined by the European ECSS space segment operability standard in [2].

level	description	functions
E1	mission execution under ground control; limited on-board capability for safety issues	Real-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
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 function 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 operational configuration resume execution of nominal operations resume generation of mission products

limited communications with the ground operator and highly complex operations with highest criticality at the same time. To deal with these situations, some very effective fault management techniques have been implemented that enabled successful deep space mission for decades like hardware redundancy, majority voting and a pre-defined safe mode, see [3] for an overview.

Given the progress in computing power and techniques, research in numerous fields of modern computing techniques that aim at enhancing either the ground segment planning activities (like machine learning methods and automated command sequence generation), fault detection and identification techniques on ground to ease the time consuming process of telemetry analysis (e.g. by outlier detection methods) or the on-board capabilities of the space probes is conducted.

The research leading to this paper aims at an evolution of the fault detection, identification and recovery (FDIR) system on-board deep space probes to increase their operational time and hence, their scientific output.

2.2. Goal-based mission continuation

The concept of a goal-based mission continuation system was chosen based on an analysis of available systems, their capabilites and maturity presented in [3], the plans for implementation follow [4], [5], [6]. As of today, the concept of goal-based mission continuation as such still is attractive and seems to offer the correct solution for the intended application - to enhance the knowledge of the system and its operational capabilites on-board.

The system under development aims at a central knowledge base on-board the spacecraft that includes a-priori knowledge collected before launch and additional situational knowledge gathered in orbit. This knowledge base shall be accessible for all subsystem and system levels, and shall provide the spacecraft with knowledge about its mission goals and its own capabilities. Thus, the spacecraft shall be enabled to react to unforeseen situations in a way other than waiting weeks in safe mode for fault resolution by ground loosing valuable scientific time and therefore, reach the mission execution levels E4, F1 and F2 that are defined by ECSS space segment operability standard, see table 1 and [2].

Table 2: Major data of the orbits of Mercury and Earth for comparison.

	Mercury	Earth
mass ratio M/M_E	0.055	1
eccentricity e	0.205	0.017
mean distance to sun R (AU)	0.39	1
inclination i (deg)	7.01	0
period T(years)	0.24	1

3. Mission to Mercury

As an example and case study, any deep space mission could have been chosen for their unique requirements and challenges. For this research, the BepiColombo mission to Mercury is considered: The feasibility of the mission was assessed in the year 2004, the launch is expected in October 2018. After seven years of flight, the spacecraft will arrive at Mercury (see Fig. 1) in 2025. Europe's contribution, the Mercury Planetary Orbiter, is scheduled for one year nominal operational phase and an optional year extension (refer to [7] for details).

As major design driver for BepiColombo, the thermal environment was identified: the mission faces an extremely harsh environment visiting the innermost planet of our solar system. Since Mercury's distance to the Sun is only one third of the Earth's, (see table 2), the incoming solar

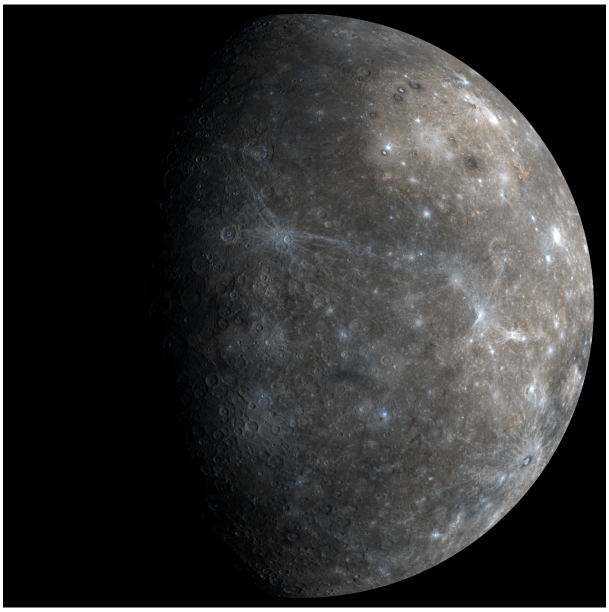


Figure 1: Planet Mercury, image credit: NASA.

flux equals almost ten solar constants. In addition, Mercury's slow rotational rate causes the day side of its surface to heat up and reach temperatures up to 427 °C, whereas the night side is as cold as 180°C. The BepiColombo Mercury Plantary Orbiter aims at a 480 x 1500 km polar orbit with a period of 2.3 h.

A respective set of SPICE kernels is available for download from the BepiColombo SPICE repository.

4. Simulation tool

For the demonstration purposes of the case study, a simulation tool is developed and described within this section. It comprises three parts: the environmental and spacecraft subsystems simulation, the cognitive recovery unit and the graphical user interface for control and visualisation.

4.1. Motivation & specification

The motivation for creating yet another tool for spacecraft simulation derives from the need of a high level system simulation of the spacecraft. It shall not provide a high-fidelity simulation of each spacecraft subsystem rather than represent the inter-subsystem relationships and their interaction with the environment. At the same time, it shall be easy to use and offer the interface to a the cognitive recovery unit under development for application and testing of the approach. Table 3 provides the resulting list of requirements.

The simulation tool is realized in Matlab and is therefore easy to use and transferrable to other systems.

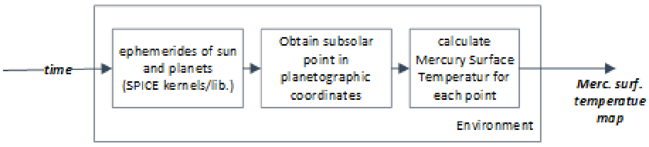

time
ephemerides of sun and planets (SPICE kernels/lib.)
Obtain subsolar point in planetographic coordinates
calculate Mercury Surface Temperatur for each point
Environment
Merc. surf. temperatue map

Figure 2: Mercury surface temperature calculation.

4.2. Environmental simulation

Fig. 3 shows an early software architecture of the spacecraft environmental and subsystem simulation. As described in the previous section, the thermal environment is a crucial aspect to model for the chosen Mercury mission since the incoming heat fluxes from both the Sun and the planetary surface onto the spacecraft are very high and results in a high radiation load onto the spacecraft.

4.2.1. Planetary Ephemerides

The environmental simulation implements the SPICE Toolkit in the current version N0066 [8] for Matlab along with latest generic spice kernels for ephemerides of planets (SPK), planetary constants (CK) and leapseconds kernel (LSK) from the NAIF repository. For convenience, the ECLIPJ2000 reference frame is used for visualisation of Mercury's orbit, whereas the J2000 frame is used for calculations.

4.2.2. Mercury surface temperature

As explained earlier, the heat flux the spacecraft receives from the planetary surface is considerably high. Therefore, is has to be modelled and influences the thermal modelling of the spacecraft. The surface temperature of Mercury is based on data collected by the MESSENGER spacecraft, see [9] for reference:

$$T(\theta) = T_{min} + (T_{max} - T_{min}) \cdot (\cos\theta)^{1/3} \tag{1}$$

$$T_{max} = T_{max}(\nu = 0) \cdot \sqrt{\frac{1 + e \cdot \cos\nu}{1 + e}} \tag{2}$$

$$T_{max}(\nu = 0) = 720K \tag{3}$$

$$T_{min} = 100K \tag{4}$$

T_{min} is the minimum temperature representing the average temperature on Mercurys nightside ($\theta \geq 90°$ and $\theta \leq 270°$), T_{max} is a maximum temperature at the subsolar point, and θ is the longitude relative to the subsolar point. The maximum subsolar temperature varies with the solar distance which is accounted for by Mercury's true anomaly ν and its eccentricity e.

4.2.3. Spacecraft state (position and velocity)

Similar as the planetary ephemerides, the spacecraft ephemerides are obtained from the respective BepiColombo SPICE kernel set, which are available for download from ESA's SPICE repository [10], and displayed for the user's convenience in the Mercury body-fixed frame.

Table 3: Specifications of simulation tool under development.

requirements	current implementation	future development
modular	yes	add further modules, enhance others
transferable	yes, due to Matlab implementation	n.a.
generic	no, specific	yes, using spice kernel and modify respective modules
high-fidelity	no	improvement of models possible with medium effort
interface to CRU	currently software implementation	enable hardware application in the loop
user fault introduction	yes, predefined errors	enhanced fault introduction features
easy to use	yes	n.a.

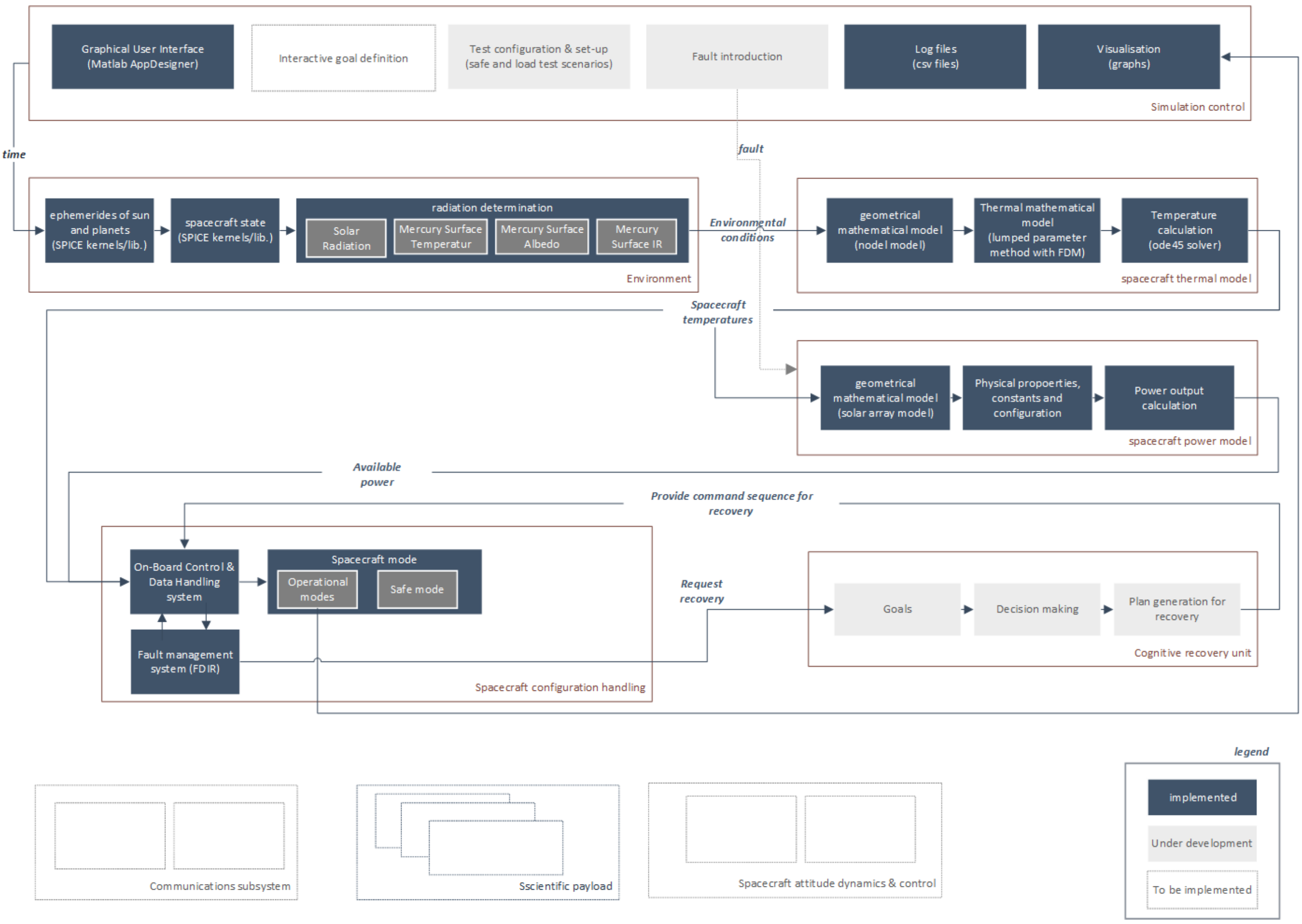

Graphical User Interface (Matlab AppDesigner)
Interactive goal definition
Test configuration & set-up (safe and load test scenarios)
Fault introduction
Log files (csv files)
Visualisation (graphs)
Simulation control
time
fault
ephemerides of sun and planets (SPICE kernels/lib.)
spacecraft state (SPICE kernels/lib.)
radiation determination
Solar Radiation
Mercury Surface Temperatur
Mercury Surface Albedo
Mercury Surface IR
Environment
Environmental conditions
geometrical mathematical model (nodel model)
Thermal mathematical model (lumped parameter method with FDM)
Temperature calculation (ode45 solver)
spacecraft thermal model
Spacecraft temperatures
geometrical mathematical model (solar array model)
Physical propoerties, constants and configuration
Power output calculation
spacecraft power model
Available power
Provide command sequence for recovery
On-Board Control & Data Handling system
Spacecraft mode
Operational modes
Safe mode
Fault management system (FDIR)
Spacecraft configuration handling
Request recovery
Goals
Decision making
Plan generation for recovery
Cognitive recovery unit
Communications subsystem
Sscientific payload
Spacecraft attitude dynamics & control
legend
implemented
Under development
To be implemented

Figure 3: Early simulation tool architecture and development status. Note, that every box is connected to the simulation control for input of values to the log file and visualisation.

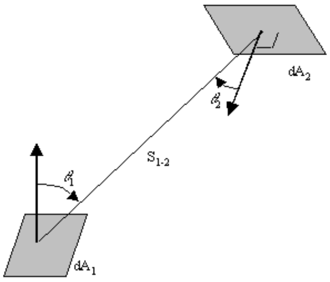


Figure 4: Geometry for view factor calculation between two surface elements.

4.2.4. Thermal environment: radiation determination

The radiation that is received by the spacecraft consists of three external parts: the solar radiation $\dot{Q}_{sol}$, calculated based on the distance to the sun and the angle ψ between the surface normal vector and the direction of solar rays; the infrared heat flux $\dot{Q}_{IR,merc}$ received from Mercury's surface due to its surface temperature T and the reflected incident sunlight from Mercury's surface (Albedo, $\dot{Q}_{al}$). Using the geometry and equations for field of view calculation given in [11] to estimate whether or not a surface element of Mercury contributes to the spacecraft's heat flux, the following heat fluxes are received by the spacecraft at any given moment:

$$SC(AU) = \frac{SC(1AU)}{r_{AU}^2} \tag{5}$$

$$\dot{Q}_{IR,merc} = F_{1,2} \cdot \sigma T^4 \tag{6}$$

$$F_{1,2} = \frac{\cos\theta_1 \cos\theta_2}{\pi S_{1-2}^2} dA_1 dA_2 \tag{7}$$

$$\dot{Q}_{sol} = SC(AU) \cdot \cos\psi \tag{8}$$

$$\dot{Q}_{al} = SC(AU) \cdot F_{1,2} \cdot AF \tag{9}$$

$F_{1,2}$ is the view factor between the surface element and the spacecraft, σ the Stefan Boltzmann constant, AF is albedo factor of the surface. $F_{1,2}$ is calculated under the assumption that each Mercury surface element is small enough to be approximated by a flat surface, see fig. 4.

4.3. Spacecraft subsystems

The spacecraft subsystem's models are highly interconnected, e.g. the temperature is based on the orbital geometries of any given time and result in a generated power of the soar array. All of these interdependencies are considered in the models.

4.3.1. thermal subsystem

The thermal subsystem is a nodal model, see table 4 that calculates temperatures based on incoming and outgoing heat fluxes. As external heat sources, the planet's surface and the sun are considered, see figure 5 the space environment provides a heat sink for infrared radiation exchange from the boundary nodes.

Table 4: Nodal breakdown of solar array model.

	node	description
1	SC	panel front side
2	OSR	panel back side
3	SA	panel structure

Let R_{ij} be the radiative coupling coefficient from node i to node j in m^2 and C_{ij} the conductive coupling coefficient in W/K, then the energy balance for each of the three nodes can be written as follows:

$$\sum \dot{Q}_{out,i} + P_i = \sum_{j=1}^{m} \sigma R_{ij}(T_j^4 - T_i^4) + \sum_{j=1}^{m} C_{ij}(T_j - T_i) \tag{10}$$

$$C_{ij} = \frac{A}{l} \cdot \lambda \tag{11}$$

$$\sum \dot{Q}_{in,i} = A_i \cdot (\alpha_i \cdot \dot{q}_{sol,i} + \alpha_i \cdot \dot{q}_{Al,i} + \epsilon_i \cdot \dot{q}_{IR,i}) \tag{12}$$

$$\sum \dot{Q}_{out,i} = A_i \cdot \epsilon_i \cdot T_i^4 \tag{13}$$

With the cross sectional area of the node A, its length l and its thermal conductivity λ.

Note, that for the temperature of the solar panels the generated electrical power is considered as

$$P_i = P_{el} = P_{op}. \tag{14}$$

The temperature T_{n+1} of timestep n+1 can then be calculated from the temperature T_n at the time n by using

$$\Delta T = T_n + T_{n+1} \tag{15}$$

to yield

$$T_{n+1} = T_n + \frac{\Delta t}{C_w}(\sum \dot{Q}_{in} - \sum \dot{Q}_{out}). \tag{16}$$

for each node. The resulting system of equations can then be solved numerically.

4.3.2. power subsystem

The generated power $\mathrm{P_{SA}}$ of the overall solar array that consists of n=3 panels is calculated using by the operating current $\mathrm{I_{op}}$ (produced by $\mathrm{N_p}$ strings per panel in parallel) multiplied the operating voltage $\mathrm{V_{op}}$ (generated by $\mathrm{N_s}$ solar cells in series per string).

$$P_{SA} = n \cdot P_{op} \tag{17}$$

$$P_{op} = I_{op} \cdot (V_{op} - V_s - V_c) \cdot K_p \tag{18}$$

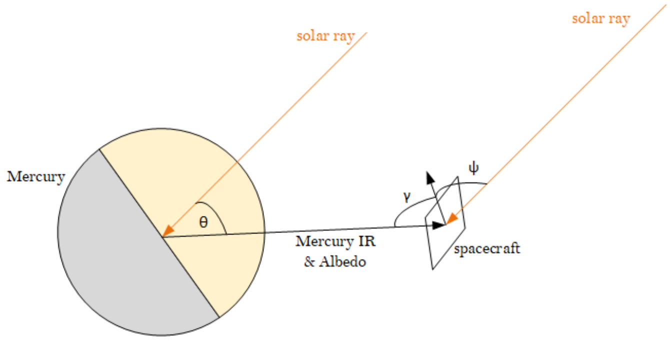


Figure 5: Geometrical model of heat fluxes to and from Mercury and the spacecraft. Adapted from [12].

The voltage drop V_s due to the resistance of cell interconnectors is calculated by

$$V_s = I_{op} \cdot R_s \tag{19}$$

with

$$R_s = \frac{1}{N_p} \cdot \left(\frac{V_{mp}(T)}{I_{mp}(T)} + R_{int} \cdot N_s \right) \tag{20}$$

where R_{int} is the resistance of cell interconnectors. The voltage drop due to harness resistance as function of cable length l_c with $n_{c,p}$ cables in parallel at $T_{0,c}$=20° can be calculated by

$$V_c = I_{op} \cdot R_c \tag{21}$$

$$R_c = R_{c,0} \cdot \frac{l_c}{n_{c,p}} \cdot (1 + 0.0043 \cdot (T - T_{0,c})) \tag{22}$$

The operating current is calculated as follows:

$$\begin{aligned} I_{op}(T) =& I_{sc}(T) \cdot N_p \\ & \cdot \left(1 - C_b \cdot \left(\exp\left(\frac{-V_{op}}{C_a \cdot V_{oc}(T) \cdot N_s} - 1 \right)\right)\right) \\ & \cdot K_I \cdot B \end{aligned} \tag{23}$$

with

$$C_a = \frac{\frac{V_{mp}(T)}{V_{oc}(T)} - 1}{\ln\left(1 - \frac{I_{mp}(T)}{I_{sc}(T)}\right)} \tag{24}$$

$$C_b = \left(1 - \frac{I_{mp}(T)}{I_{sc}(T)}\right) \cdot \exp\left(\frac{-V_{mp}(T)}{C_a \cdot V_{oc}(T)}\right) \tag{25}$$

The operating voltage V_{op} can then be calculated

$$V_{op} = (N_s \cdot V_{mp}(T) - N_{d,s} \cdot V_{diode}) \cdot K_V \tag{26}$$

with

$$V_{diode} = 0.03 \cdot \log\left(\frac{I_{sc}(T) \cdot N_p}{N_{d,p} \cdot 10^{-13}} + 1 \right) \tag{27}$$

where $N_{d,s}$ is the numer of diodes in series per string and $N_{d,p}$ is the number of diodes in parallel per string. The loss factors are considered for several degradations throughout the mission in three application categories: applied to operating current K_I=0.6885, to operating voltage K_V=0.931 and to operating power K_P=0.8717.

the characteristic voltages and currents at maximum power points (index mp), open circuit (oc) and short circuit (sc) are calculated according to equations 28 to 31 taking into account

- the respective remaining factors R after the arrival at Mercury,
- the temperature gradients,
- the solar cell area A_{sc}
- the electrical data at the reference temperature T_0 and
- the reference temperature $T_0 = 28\,°C$.

$$V_{mp}(T) = V_{mp}(T_0) \cdot R(V_{mp}) + \frac{\mathrm{d}V_{mp}}{\mathrm{d}T} \cdot (T - T_0) \tag{28}$$

$$V_{oc}(T) = V_{oc}(T_0) \cdot R(V_{oc}) + \frac{\mathrm{d}V_{oc}}{\mathrm{d}T} \cdot (T - T_0) \tag{29}$$

$$I_{mp}(T) = \left(I_{mp}(T_0) \cdot R(I_{mp}) + \frac{\mathrm{d}I_{mp}}{\mathrm{d}T} \cdot (T - T_0) \right) \cdot A_{sc} \tag{30}$$

$$I_{sc}(T) = \left(I_{sc}(T_0) \cdot R(I_{sc}) + \frac{\mathrm{d}I_{sc}}{\mathrm{d}T} \cdot (T - T_0) \right) \cdot A_{sc} \tag{31}$$

The defined models follow [13].

4.4. Cognitive Recovery Unit

The favoured solution and formerly considered Cognitive System Architecture (COSA) [14] is not developed further and documentation can hardly be found. A successful implementation for automation of space mission operations could not be achieved.

Instead, a costumized version of the classical three tier architecture as depicted in fig. 6 is developed following a proposal of Peters [15] and implemented using the terminology and also the concepts of behavioural architectures in general and especially of the cognitive process developed with COSA.

It features three layers like many robotic architectures:

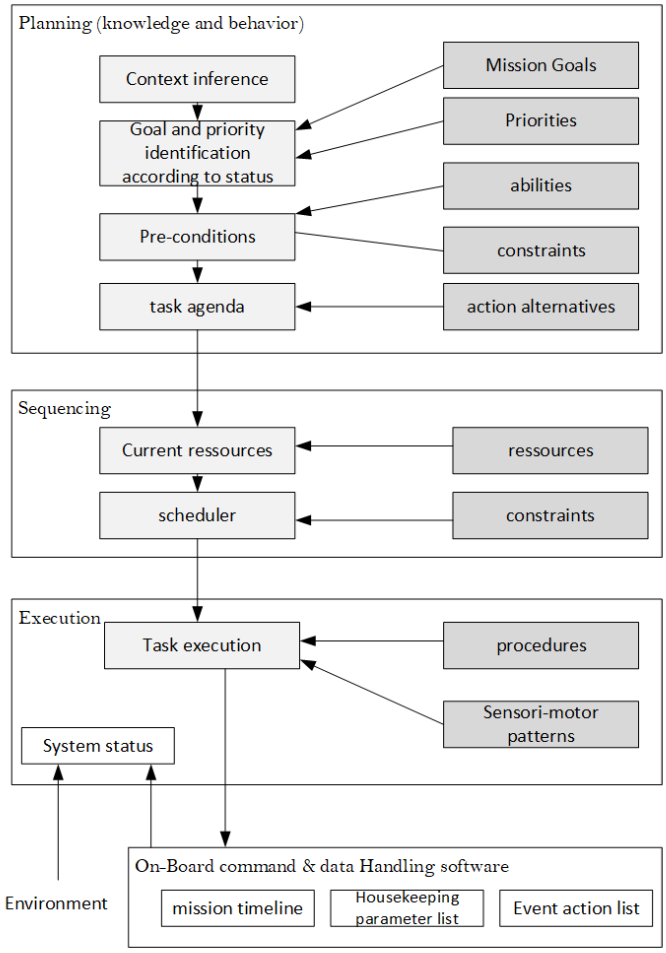


Figure 6: Three tier architecture for goal-based mission continuation; light grey boxes represent aquired situational knowledge and dark grey boxes.

- planning. the system status is assessed, mission goals and their priorities are compared and a rough task agenda is compiled.
- sequencing. ressources and constraints of the system confirm or change the task agenda and schedule activies accordingly.
- execution. task execution commands are sent to the spacecraft on-board command and data handling software which again commands the respective subsystems.

4.5. *Graphical User Interface*

The graphical user interface is developed to fulfill the following requirements:

- Set-up of the simulation by safing/loading a spacecraft configuration, i.e. time, state vector of spacecraft (position, velocity, attitude), physical properties of subsystems

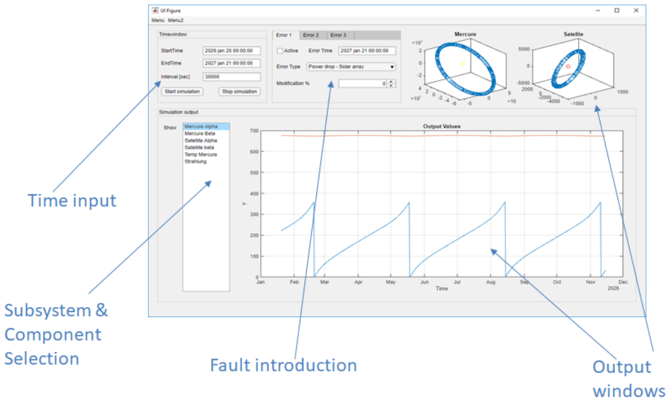

StartTime 2026 jan 20 00:00:00
EndTime 2027 jan 21 00:00:00
Interval [sec] 36000
Error Time 2027 jan 21 00:00:00
Error Type Power drop - Solar array
Modification % 0

Figure 7: Graphical User Interface of the simulation tool developed in Matlab App Designer.

- control of simulation inlcuding insertion of faults either ad-hoc, at a pre-defined time or random
- visualisation of the simulation by viewgraphs for each subsystem and parameters of interest and thus,
- ad-hoc assessment of the results by the user as well as
- output of a storable log-file for post-simulation evaluation of the results.

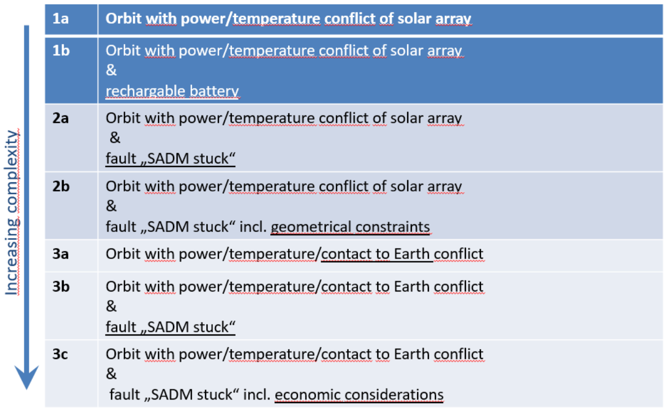
The first version of the graphical user interface (GUI) was presented in [5] and was written in C# only for the reason that the visualisation using a .NET environment was more intuitive. It turned out that the exported functional libraries from Mathworks *Matlab* were executed in a sandbox environment and could not be used with Naif SPICE kernels. Therefore, the GUI was updated and migrated using Matlab App Designer.

4.5.1. *Visualisation*

The result is depicted in figure 7, where all calculation results can be displayed by the user's choosing from the menu on the left hand side. In addition, the positions of Mercury and the spacecraft are shown for every time step in the upper right output graphs.

4.5.2. *Control, interface & fault introduction*

Control of the simulation is enabled in the upper left corner where the user has to specify the start and end time as well the desired time step for the simulation. Fault introduction is possible in this state by choice amongst three pre-defined errors which will then be injected into the system simulation and trigger a response by the cognitive recovery unit, which again results in an update of simulation results.


Increasing complexity

1a	Orbit with power/temperature conflict of solar array
1b	Orbit with power/temperature conflict of solar array & rechargable battery
2a	Orbit with power/temperature conflict of solar array & fault „SADM stuck“
2b	Orbit with power/temperature conflict of solar array & fault „SADM stuck“ incl. geometrical constraints
3a	Orbit with power/temperature/contact to Earth conflict
3b	Orbit with power/temperature/contact to Earth conflict & fault „SADM stuck“
3c	Orbit with power/temperature/contact to Earth conflict & fault „SADM stuck“ incl. economic considerations

Figure 8: Test scenario definition with increasing complexity.

5. Tool development status and first simulation results

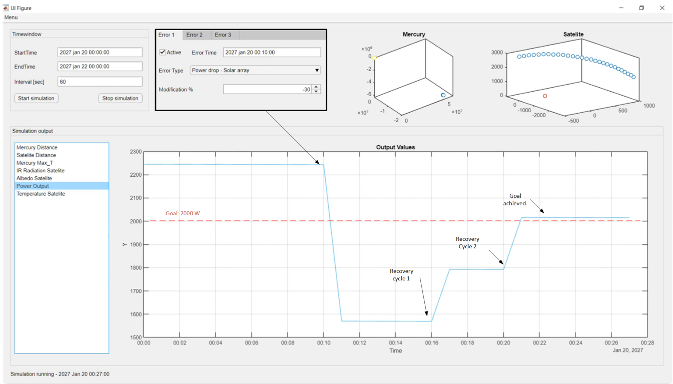
Figure 3 shows the architecture of the simulation tool as well as the development status, grey boxes represent modules under development, blue modules are implemented, white ones need future development and implementation. For example, the thermal and power subsystems are modelled each using simple subsystem models, the communications subsystem and the attitude dynamics of the spacecraft still need to be implemented.

5.1. Test Scenario.

The test scenario that could be applied is called scenario 1a of the test sequence in fig. 8. It implies a conflict of required power for a given operational mode, a fault that somehow prevents the system from providing that power and/or is heating up the solar array beyond its operational temperature. Both options are not acceptable, since the value of scientific time is rated high. Thus, the cognitive recovery unit would be triggered by a simulated power drop, the goal *provide sufficient power* would be instantiated, necessary pre-conditions like array temperature would be checked and an action suggested to increase the solar aspect angle in order to increase the power output.

5.2. Results

Figure 9 shows a first simulation result of test scenario 1a. The power dops at a pre-defined time by 30 % which represents the loss of 1 out of 3 sections of the solar array. The resulting power output is below the required power of the system, therefore the goal-based mission continuation system is activated. The goal *provide sufficient power* combined with the required power for a specific task (here: *2000W*) leads to the action selection *increase solar aspect angle by 5* °. The result is an increased power output which is still below the desired power output. A similar second cycle leads to repetition of the action and



StartTime 2027 jan 20 00:00:00
EndTime 2027 jan 22 00:00:00
Interval [sec] 60

☑ Active Error Time 2027 jan 20 00:10:00
Error Type Power drop - Solar array
Modification % -30

Simulation running - 2027 Jan 20 00:27:00

Figure 9: A simulation result: Error is introduced, power output drops the specified amount and is recovered in two cycles by increasing the solar aspect angle onto the solar array whilst maintaining the temperature within operational limits.

5.3. Validation

Any new tool developed needs to be validated against existing models. Some of the included models like reference frames, orbits and Mercury surface temperature maps are already validated, the validation of the thermal and power model as well as validation of the integrated software is in progess. Since the models are based on commonly used equations, only minor deviations due to approximations in use are expected.

6. Conclusions & future developments

This paper presents a simulation tool that can be used for future evaluation of goal-based mission continuation systems. A test concept that might lead to a proof-of-concept is explained. The tool's modular nature allows for substitution of models and might even be modified for hardware-in-the-loop test.

Future developments of the cognitive architecture will explore the implementation of hierarchical task network techniques (HTN) as part of constraint satisfaction problem (CSP) techniques (an example can be found in [16]) and the soar cognitive architecture [17] itself (which is the underlying functionality of COSA, but is still being developed and maintained with tutorial material available) amongst others.

References

[1] D. D. Dvorak, M. D. Ingham, J. R. Morris, J. Gersh, Goal-based operations: An overview, Journal of Aerospace Computing, Information, and Communication 6 (3) (2009) 123–141. `doi:10.2514/1.36314`.

[2] ECSS Secretariat, ESA-ESTEC Requirements & Standards Division, Space engineering: space segment operability (31 July 2008).

[3] A. Wander, R. Förstner, Innovative fault detection, isolation and recovery strategies on-board spacecraft: State of the art and research challenges, in: Deutscher Luft- und Raumfahrtkongress 2012, Berlin, 2012.

[4] A. Wander, R. Förstner, Feasibility of innovative fault detection, isolation and recovery onboard spacecraft using cognitive automation, 63rd International Astronautical Congress (IAC), Naples, Italy, 2012.

[5] A. Wander, R. Förstner, Concept, architecture and simulation considerations for a cognitive recovery unit on-board interplanetary spacecraft, iac-17-b6.2.11, 68th International Astronautical Congress (IAC), Adelaide, Australia, 25-29 September 2017, 2017.

[6] A. Wander, R. Förstner, Innovative fault detection, isolation and recovery on-board spacecraft: Study and implementation using cognitive automation, in: Conference on Control and Fault-Tolerant Systems (SysTol), Nice, 2013, p. 336.

[7] Esa's bepicolombo mission fact sheet.
URL `http://sci.esa.int/bepicolombo/47346-fact-sheet/`

[8] The spice toolkit.
URL `https://naif.jpl.nasa.gov/naif/toolkit.html`

[9] D. R. Stanbridge, K. E. Williams, A. H. Taylor, B. R. Page, C. G. Bryan, D. W. Dunham, P. Wolff, B. G. Williams, J. V. McAdams, D. P. Moessner, Achievable force model accuracies for messenger in mercury orbit: Aas 11-548., in: H. Schaub (Ed.), Astrodynamics 2011, Published for the American Astronautical Society by Univelt, San Diego and Calif, 2012.

[10] Bepicolombo spice kernels set.
URL `https://www.cosmos.esa.int/web/spice/spice-for-bepicolombo`

[11] W. McClain, D. Vallado, Fundamentals of Astrodynamics and Applications, Space Technology Library, Springer Netherlands, 2001.
URL `https://books.google.de/books?id=PJLlWzMBKjkC`

[12] Cassandra Belle VanOutryve, A thermal analysis and design tool for small spacecraft, Master's thesis, San Jose State University (December 2008).
URL `http://scholarworks.sjsu.edu/etd_theses/3619`

[13] K. Wittmann, W. Ley, W. Hallmann, Handbuch der Raumfahrttechnik, Carl Hanser Verlag, 2008.
URL `http://elib.dlr.de/53447/`

[14] R. Onken, A. Schulte, System-ergonomic design of cognitive automation: Dual-mode cognitive design of vehicle guidance and control work systems, Springer-Verlag, Berlin and Heidelberg, 2010.

[15] S. Peters, C. Pirzkall, H. Fiedler, R. Frstner, Mission concept and autonomy considerations for active debris removal, Acta Astronautica 129 (2016) 410 – 418. `doi:https://doi.org/10.1016/j.actaastro.2016.10.006`.
URL `http://www.sciencedirect.com/science/article/pii/S0094576516306786`

[16] F. d. N. Kucinskis, M. G. V. Ferreira, On-board satellite software architecture for the goal-based brazilian mission operations, Aerospace and Electronic Systems Magazine, IEEE 28 (8) (2013) 32–45. `doi:10.1109/MAES.2013.6575409`.

[17] J. Laird, The Soar cognitive architecture, MIT Press, Cambridge and Mass. and and London and England, 2012.