ERGO

INTERFACE CONTROL DOCUMENT

ERGO_D1.3

Due date of deliverable:	31-07-2017
Start date of project:	01-11-2016
Duration:	27 months
Topic:	COMPET-4-2016 Building Block b) Autonomy framework Time/Space/Resources planning and scheduling
Work package:	1200 – System Requirements
Lead partner for this deliverable:	GMV

This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement No 730086.

Dissemination Level		
PU	Public	**X**
CO-1	Confidential, restricted under conditions set out in Model Grant Agreement. Version providing the PSA will all the information required to perform its assessment.	
CO-2	Confidential, restricted under conditions set out in Model Grant Agreement. Version providing the PSA and the other operational grant the information required for the integration of all the building blocks and the continuity of the Strategic Research Cluster	

Prepared by	ERGO Team
Approved by:	GMV
Authorized by:	Jorge Ocón
Code:	ERGO_D1.3
Version:	2.2
Date:	15/09/17
Internal code:	GMV 23814/16 V5/17

DOCUMENT STATUS SHEET

Version	Date	Pages	Changes
1.0	30/01/17	21	First published version for ERGO SRR+ICD Workshop.
1.1	28/02/17	39	Second published version for ERGO SRR+ICD Workshop Close-out including: - Section 4. Reference Frames Definition. - Section 5.2. Ground Control Interfaces - Section 5.3. OG2 ←→ OG3 Interfaces. - Section 5.4. OG2 ←→ OG4 Interfaces.
2.0	27/06/17	45	Third version, published for PDR
2.1	31/07/17	47	Document updated as consequence of PDR's RID implementation: - Fixed typos en row number 3 of Table 5-1. - **OG2-100**: Modified corresponding cell in Table 5-1. - **OG2-101**: Removed affected row in Table 5-1. - **ICD-MAZ-04**: Corrected typo in second paragraph of section 4.1. - **ICD-MAZ-06**: Corrected error in second row of Table 5-1. - **ICD-MAZ-07**: Corrected error in second row of Table 5-1. - **ICD-MAZ-08**: Corrected error in second row of Table 5-1. - **ICD-MAZ-11**: Removed affected row in Table 5-1 (same as OG2-101). - **ICD-MAZ-12**: Added CDFF and DFPCs acronyms to section 3.2.
2.2	15/09/17	47	Document updated as consequence of PDR's RID implementation: - **OG2-95:** Added new section 5.4.4 containing the conclusion of the OG1_OG2_OG5 splinter: Control & intelligent behind SIROMs, APM & modular Satellite meeting maintained at 11th and 12th of July.

NOTICE

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

1. INTRODUCTION

1.1. PURPOSE

ERGO (OG2 activity of COMPET 04 2016 call), has as main challenge to realise a software framework for the development of highly autonomous space robotics missions. In these a robot system, given a high level goal, will (re)plan, schedule and oversee the execution of elementary actions to attain the goal, considering Time/ Space/Resources constraints.

This document describes the major interfaces between ERGO subsystems and with the rest of OGs.

1.2. SCOPE

This document is part of the outcome of the WP 1200 and WP 2100 to WP 2500 of the ERGO activity "Preliminary Design and Modelling". A preliminary version (v1) was already produced as outcome of the WP 1200 "System Requirements", which is now completed in this current version.

1.3. CONTENTS

This document is structured as follows:

- Section 1. Introduction: this section; presents the purpose, scope and structure of the document.
- Section 2. Reference and Applicable Documents: lists other documents that complement or are needed to understand this document.
- Section 3. Terms Definitions and Abbreviated Terms: defines terms and acronyms used in the document.
- Section 4. Reference Frames Definition: definition of reference frames related to the planetary exploration and the orbital scenarios.
- Section 5. Interface Control Document: describes the major internal and external interfaces identified as needed in ERGO project.

2. REFERENCE AND APPLICABLE DOCUMENTS

2.1. APPLICABLE DOCUMENTS

The following is the set of documents that are applicable:

Table 2-1: Applicable documents

Ref.	Title	Date
[AD.1]	European Robotics Forum, "The PERASPERA Roadmap"	March 11th, 2015
[AD.2]	"Master Plan of SRC activities", PRSPOG1-R-ESA-T3 1-TN-D3.4. PERASPERA Consortium	2015
[AD.3]	PERASPERA consortium, "Compendium of SRC activities (for call 1)", version 1.8, PRSPOG1-R-ESA-T3 1-TN-D3.1. PERASPERA Consortium.	2015
[AD.4]	Guidelines for strategic research cluster on space robotics technologies horizon 2020 space call 2016	October 30th, 2015
[AD.5]	ERGO (European Robotic Goal-Oriented Autonomous Controller) HORIZON 2020 COMPET-04-2016 Proposal: PART B, issue 1.2	Sept. 19th, 2016
[AD.6]	ERGO System Requirements. ERGO_D1.2. issue 1.0	
[AD.7]	ERGO – Preliminary Design Document	28/06/17
[AD.8]	ERGO – Unitary and integration test plan	

2.2. REFERENCE DOCUMENTS

The following is the set of documents referenced:

Table 2-2: Reference documents

Ref.	Title
[RD.1]	T. De Laet, S. Bellens, R. Smits, E. Aertbelien, H. Bruynickx, J. De Schutter, Geometric Relations between Rigid Bodies: Semantics for Standardization", IEEE Robotics and Automation Magazine, 2012. https://people.mech.kuleuven.be/~tdelaet/geometric_relations_semantics/geometric_relations_semantics_theory.pdf
[RD.2]	ALL_Integrated_ICD (compilation of all OG's ICDs) – 02/06/17

3. TERMS DEFINITIONS AND ABBREVIATED TERMS

3.1. DEFINITIONS

Concepts and terms used in this document and needing a definition are included in the following table:

Table 3-1: Definitions

Concept / Term	Definition
Digital Elevation Map (DEM)	2.5D representation of a terrain's surface in Cartesian coordinates, also known as height-map. DEM is a grid map: a collection of squared cells organized into a grid structure with associated height (i.e. elevation). While man-man objects (e.g. lander platform) are included in this map, the rover itself shall be excluded (e.g. wheels, solar panels, which could be visible in raw sensor data). *Note: this map contains solely height information, other type of information are excluded.*
Domain	Formal description of the system to be controlled from the planning point of view. It consists of a set of components and a set of relevant physical constraints that influence the possible temporal evolution of such components (e.g., possible state transitions over time of the component, coordination constraints among different components, maximal capacity of resources, etc). A component can be non-controllable in case it presents a predefined temporal behaviour (for example the visibility of an observation target is given as a predefined data from Flight Dynamics). Such components act as constraining factors for controllable components which the planner is able to influence. It is an input to the deliberative layer.
Fused Rover Map	Map produced with information gathered by sensors on the rover itself at the last and previous sensing captures.
Fused Total Map	Map produced with information from any sensing sources at any capturing time, e.g. rover, orbital, other mobile or static devices on the surface.
Goal	A goal specifies an action or state desired to be achieved by the target agent in the future. The planner's task is to find a valid sequence of actions/states (the plan) that achieves those goals from a given initial state. There are four main properties that characterize goals: - Formal representation: Both classical and timelines-based planning use predicate logic to represent goals. - Temporal scope: In classical planning, goals must be satisfied at the end of the plan. On the other hand, goals in timeline planning can be defined to be achieved at any time within the temporal scope of the problem. - Hierarchy: In those systems modelled hierarchically, goals are classified as complex (high-level) and primitive (low-level). Complex goals must be decomposed (at planning or execution time) before they can be executed. - Hard/Soft: Hard goals must be achieved in the plan, while soft goals represent preferences that might be disregarded in the plan.
Guidance	Process of guiding the rover through a terrain, this includes creation of a navigation map from Digital Elevation Map (DEM), planning a path on this map, estimating the required resources to achieve this path and reacting to the environment (hazards) while executing the path.
Local, Regional or Global Map	Refer to the geographical extent and spatial resolution of the map: - Local map is high resolution with small geographical extent. - Regional map is medium resolution with medium geographical extent. - Global map is low resolution with large geographical extent.
Navigation Map	2D map onto which the rover path can be planned. This is generally a traversability map and any additional information regarding other aspects that can be taken into account to plan the path, like areas of scientific interest or shadows impacting illumination of the solar panels.
Orbital 3D-Model	3D generated model acquired from orbital sensors (e.g. from stereo-images, TOF/Lidar or radar).
Orbital Map	Map generated from orbital sensing data. *Note: this terminology can be used in combination with other definitions, e.g. "orbital navigation map".*

Concept / Term	Definition
Planning	Planning is the reasoning side of acting that aims to organize actions according to their expected outcomes in order to achieve some given goal. Automated Planning is the area of Artificial Intelligence (AI) that studies this process. One informal division in the field of automated planning can be done between classical-based and timelines-based planning, even though there are other approaches. Classical planning focus to a great extent on performance and is based on broadly accepted standards such as PDDL. On the other hand, timeline-based planners are more expressive, especially in terms of temporal representation, but they don't share any common standard as they are intended to be proprietary products.
Planning Component	A system is traditionally organized in subsystems (e.g. rover locomotion, image acquisition, pan and tilt unit motion). Each subsystem is represented as a component from the planning perspective. There are two types of planning components: - State Variable: Represented as a state-machine containing states and transitions. - Resource: Represented as a list of parameters indicating the maximum/minimum capacity of the resource and the current amount available. There are two kind of resources. Reusable are those which are borrowed but cannot be produced, such as a communication window. Consumable are those that are produced/consumed such as energy.
Problem	Formal description of a planning task for a given domain. It defines the initial state of the world (indicating the current state for each component of the domain), a list of goals (defined as states expected to be achieved by some of the components in the future) and a metric used to determine the plan quality. It is an input to the deliberative layer.
Reactor	Separate part of the control architecture in charge of a control loop. Each control loop is embodied in a reactor that encapsulates all details of how to accomplish its control objectives. Deliberative reactors embed planning capabilities while reactive reactors are purely reactive.
Rover Map	Map produced with information gathered by sensors on the rover itself at the last sensing capture.
Soil Type Map	Map associated to a DEM, describing the type of soil associated to each cell of the DEM. *Note: this map does not contain the height information, as that is contained in the DEM.*
Timeline	A timeline is a sequence of tokens
Traversability Map	2D map identifying which area of the terrain is traversable by a specific locomotion system, including level of difficulty to traverse. This includes information regarding to the locomotion traverse capability: cost function (e.g. ability to climb rocks and drive up slopes). *Note: this excludes any other factors related to navigating through a terrain, e.g. this excludes energy.*
Token	Temporally qualified assertions expressed as a predicate with start and end time bounds defining the temporal scope over which they hold.
Uncertainty Map	Map associated to a DEM, describing the height uncertainty associated to each cell of the DEM. *Note: this map does not contain the height information, as that is contained in the DEM.*

3.2. ACRONYMS

Acronyms used in this document and needing a definition are included in the following table:

Table 3-2: Acronyms

Acronym	Definition
APSI	Advanced Planning and Scheduling Initiative
BIP	Behaviours, Interactions and Priorities
CDFF	Common Data Fusion Framework
CDR	Critical Design Review
CFI	Customer Furnished Item
DDL	Domain Description Language

Acronym	Definition
DFPCs	Data Fusion Process Compound
ECSS	European Cooperation for Space Standardization
FDIR	Fault, Diagnosis, Isolation and Recovery
GOAC	Goal Oriented Autonomous Controller
GOTCHA	GOAC TRL Increase Convenience Enhancements Hardening and Application Extension
GRR	GOTCHA Readiness Review
KOM	Kick-off Meeting
MDA	Model-driven Architecture
MER	Mars Exploration Rovers
MMPOS	Mars Mission On-board Planner and Scheduler
PDDL	Planning Domain Definition Language
PSM	Platform Specific Model
PUS	Packet Utilization Standards
RCOS	Robot Control Operating System
RTEMS	Real Time Executive for Multiprocessor System
RTOS	Real Time Operating System
ROS	Robot Operating System
SARGON	Space Automation & Robotics General Controller
SCR	Sampling Catching Rover
SFR	Sampling Fetching Rover
TASTE	The Assert Set of Tools for Engineering
T-REX	Teleo-Reactive Executive
TRF	Timeline-based Representation Framework
UML	Unified Modelling Language
V&V	Validation & Verification
WBS	Work Breakdown Structure
WPD	Work Package Description

4. REFERENCE FRAMES DEFINITION

Hereafter they are presented a first iteration in the definition of reference frames related to the planetary exploration and the orbital scenarios. First of all it is needed to provide several basic definitions related to rigid bodies and frames. Afterwards they are presented some generic definitions for the reference frames and later on specific implementations for the planetary exploration and on-orbit servicing ERGO scenarios.

4.1. INTRODUCTION TO RIGID BODIES AND FRAMES

A rigid body is an idealization of a solid body of infinite or finite size in which deformation is neglected. A body in three-dimensional space has six degrees of freedom: three degrees of freedom in translation and three in rotation. The subspace of all body displacements that involve only changes in the orientation is often denoted by SO(3) (the Special Orthogonal group in three-dimensional space). It forms a group under the operation of composition of relative displacements. The space of all body displacements, including translations, is denoted by SE(3) (the Special Euclidean group in three-dimensional space). A general six-dimensional displacement between two bodies is called a (relative) pose: it contains both the position and orientation. Remark that the position, orientation, and pose of a body are not absolute concepts, since they imply a second body with respect to which they are defined. Hence, only the relative position, orientation, and pose between two bodies are relevant geometric relations. The geometric relations between bodies are described using a set of geometric primitives (point, vector, orientation frame and displacement frame).

A reference or orientation frame represents an orientation, by means of three orthonormal vectors indicating the frame's X-axis X, Y-axis Y, and Z-axis Z. Next Figure 4-1 shows the geometric primitives that are useful to define the position, orientation, pose, linear velocity, angular velocity, and twist (relative angular and linear velocity) of body C with respect to body D: an orientation frame [a], a point e, and frame {g} fixed to body C, an orientation frame [b], a point f, and frame {h} fixed to body D, and a coordinate frame [r], considered instantaneously fixed to body D, in which the coordinates are expressed.

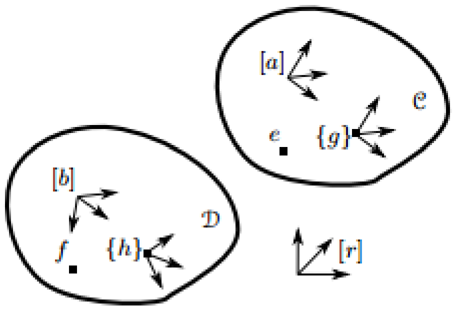

Figure 4-1: Geometric relations between rigid bodies [RD.1].

The coordinate system are right-hand Cartesian and the positive rotational direction is defined counter clockwise around the selected axis, when this axis points toward the observer. As pointed by classical literature there are many Euler angles representation [1]; by default it is proposed to be used the yaw, pitch and roll sequence (3-2-1) around the Z, Y, and X axes as the most used sequence for robots. Units must be expressed in the International System.

More information can be found within the following reference related to "Geometrics Relations between Rigid Bodies: Semantics for Standardization" [RD.1].

[1] https://en.wikipedia.org/wiki/Euler_angles

4.2. GENERIC REFERENCE FRAMES

This section covers some basic definitions of generic reference frames. Each application should take them as starting point and customize them for its own implementation. Hereafter they are presented as follows:

- **PLANETARY**. Its origin is placed in the centre of mass of the planet with the following axes:
  - X-axis permanently fixed towards the vernal equinox;
  - Y-axis in the planet's equatorial plane, right-hand side orthogonal to axis 1 and 3;
  - Z-axis lies at a 90° angle to the equatorial plane and extends through the North Pole.
- **WORLD.** This frame is the local reference frame for the robotics operations. It is placed at any salient and stable feature on the scenario at the initial time of operations. It is common to both planetary exploration and in-orbit robotics servicing scenarios. It defines the following axes:
  - X-axis: along the most significant direction of the scenario.
  - Y-axis: transversal to the previous defined direction of the scenario.
  - Z-axis: normal to the plane upwards pointing to the sky and completing the triad.
- **BODY**. This frame is placed in a stable and salient feature (e.g. the robot guiding point as the kinematic middle point of the robot chassis). It defines the following axes:
  - X-axis: It is parallel to the forward motion direction and points towards the rover forwards motion in straight line.
  - Y-axis: It is transversal to the forward motion direction and points towards the rover left wheel.
  - Z-axis: It is vertical to the forward motion direction and points up to the sky.
- **CALIBRATION**. This frame or set of frames are placed in an external and visible part of the robot and allows to calibrate the BODY frame. It is expressed w.r.t. the BODY frame. It defines the following axes:
  - X-axis: It is parallel to the typical forward motion direction of the robot.
  - Y-axis: It is transversal to the forward motion direction.
  - Z-axis: It is vertical to the forward motion direction and points up to the sky.
- **ARM**. This frame is associated is in the first joint of the robotic arm origin of the robotic arm w.r.t. to the BODY frame. It is located at the intersection of the first joint axis with the robot plate. It is expressed w.r.t. BODY frame. It defines the following axes:
  - X-axis is normal to the base plate of arm pointing outwards.
  - Y-axis is orthogonal to the XZ plane.
  - Z-axis is parallel to the Z-axis of the BODY frame.
- **ARM TIP**. This frame is associated with the tip at the end of the robotic arm (there could be more than one associated to different tips). It is located at the centre of the latest joint axis (with the Z collinear to the axis). Its actual position is given by the direct kinematic model of the robotic arm. It is expressed w.r.t. ARM frame. It defines the following axes:
  - X-axis: It is parallel to the last robotic arm link.
  - Y-axis: It is transversal to the X-axis.
  - Z-axis: It is perpendicular to the arm tip plane (normal to the XY plane) pointing outwards.
- **ARM END-EFFECTOR**. This frame (or set of frames) is associated with an end-effector placed at the tip of the robotic arm. It is expressed w.r.t. ARM TIP frame. It defines the following axes:
  - X-axis: It is parallel to the last robotic arm link.
  - Y-axis: It is transversal to the X-axis.
  - Z-axis: It is perpendicular to the arm tip plane (normal to the XY plane) pointing outwards.
- **TARGET**. This frame is placed in the target centre of masses. It is expressed w.r.t. WORLD frame. It defines the following axes:
  - X-axis: It is parallel to the most significant direction of the target.
  - Y-axis: It is transversal to the X-axis.
  - Z-axis: It is normal to the XY plane pointing outwards.

- **TARGET DOCK**. This frame is placed in the geometrical centre of the docking port. It is expressed w.r.t. TARGET frame. It defines the following axes:
  - X-axis: It is parallel to the most significant direction of the target dock.
  - Y-axis: It is transversal to the X-axis.
  - Z-axis: It is normal to the XY plane pointing outwards.

Please note the reader that relative position and orientation between rigid bodies can be represented also in other ways than only a reference frame. For example, any set of three or more points on significant parts of the bodies, or combination of lines or other geometric primitives. OG1-ESROCOS will support the above-described frame definitions, but also allow 'add ons' through which one or more of the just mentioned alternative representations can also be used.

There is a second, complementary, addition that makes sense, and that is to give "**uncertainty**" on relative body positions and orientations an explicit place in the representations. As with the mentioned add-ons, uncertainty (and other relevant aspects, such as numerical and digital representations) will also be realised first via frames-based definitions, and secondly through "add ons" on the other geometric primitives.

4.3. PLANETARY SURFACE EXPLORATION SCENARIO.

Hereafter they are presented the specific reference frames defined by the ERGO Rover Guidance within the planetary surface exploration scenario.

The definition of these reference frames uses a quaternion convention. The quaternion q is defined as a 4 element vector $[q_1 \quad q_2 \quad q_3 \quad q_4]^T$ which can be written in complex number notation as:

$$q \equiv q_4 + q_1 i + q_2 j + q_3 k$$

where i, j, and k are hyper-imaginary numbers

4.3.1.ROVER BODY (RB)

The **Rover Body** frame is fixed with the Rover and thus moves as the Rover moves (see Figure 4-2).

- The x-axis, X_{RB}, lies towards the front of the Rover in the nominal direction of travel.
- The z-axis, Z_{RB}, lies vertically upwards, antiparallel to the gravity vector when the RV is on flat, horizontal terrain.
- The y-axis, Y_{RB}, completes the orthogonal right-handed set, and will lie to the left of the Rover.

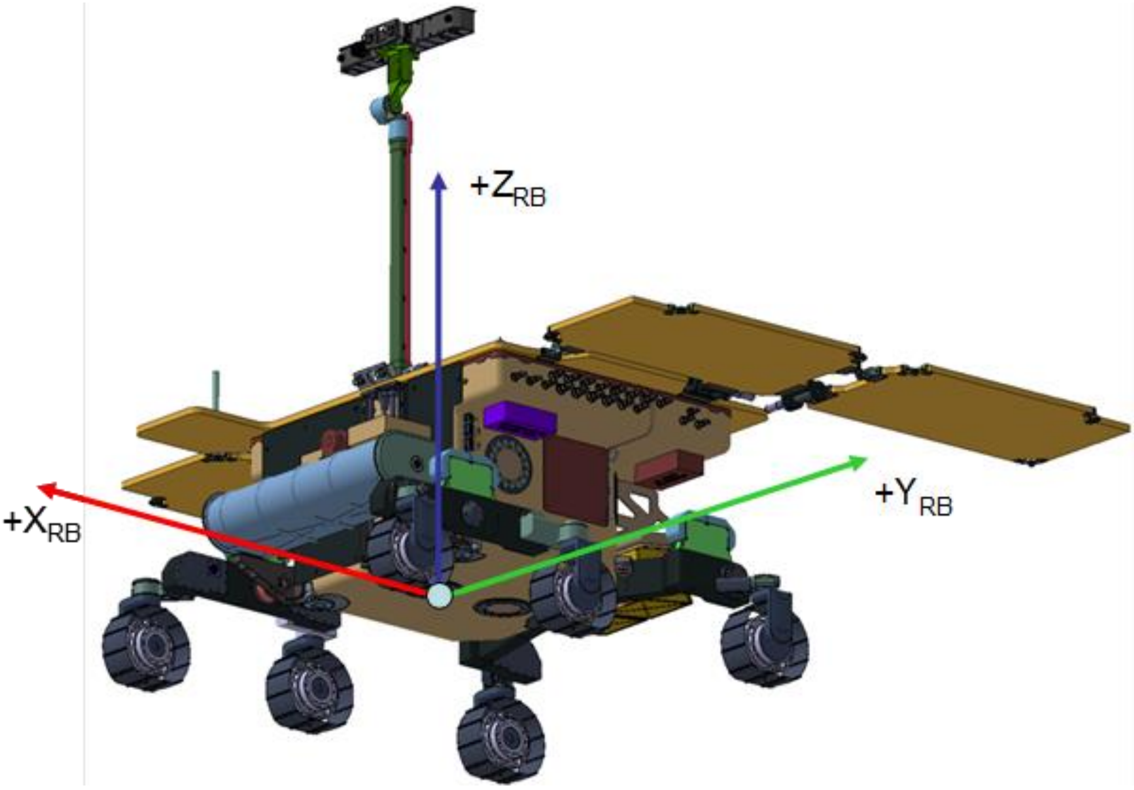

Figure 4-2: Rover Body Reference Frame.

4.3.1.1. ROVER LOCAL GEODETIC (RLG)

The **Rover Local Geodetic** frame is a special case of a frame linking the local position with the Rover Body. It is used to enable commanding of targets in a frame with respect to the RV.

- The origin O_{RLG} is coincident with O_{RB}.
- The z-axis, Z_{RLG}, lies in the direction of the geodetic vertical i.e. approximately the negative gravity vector.
- The x-axis, X_{RLG}, is the projection of the RB frame X_{RB} onto a horizontal plane passing through O_{RLG} with Z_{RLG} as normal vector to the plane.
- The y-axis, Y_{RLG}, completes the orthogonal right-handed set, and will lie approximately along the projection of the RB frame Y_{RB} onto the horizontal plane.

4.3.2. ROVER ATTITUDE DEFINITION

The attitude of the Rover is given as the rotation from the reference frame, i.e. Mars Local Geodetic (MLG) frame to the Rover Body (RB) frame. This attitude or the rover can be described by:

- A quaternion from MLG frame to RB frame
- A DCM matrix from MLG frame to RB frame
- The yaw, pitch and roll (321) Euler angles from MLG frame to RB frame

The following sections provide a description of the yaw, pitch and roll (321) Euler angle from MLG frame to RB frame.

4.3.2.1. EULER ANGLES DEFINITION

4.3.2.2. YAW / HEADING ANGLE DEFINITION

The **yaw** or **heading** angle is defined as the angle between the X_{MLG}-axis and the projection of the X_{RB}-axis onto the $X_{MLG}Y_{MLG}$ plane.

Notes:

1. At ±90 degrees pitch the X_{RB}-axis becomes coincident with the Z_{MLG}-axis and hence does not project onto the $X_{MLG}Y_{MLG}$ plane. However this does not detract from the definition as it is consistent with the 321 Euler angle singularity that occurs at pitches of ±90 degrees.
2. With the exception of the ±90 degrees pitch case, this definition remains valid for small and large angles irrespective of the rover's roll or pitch angle.

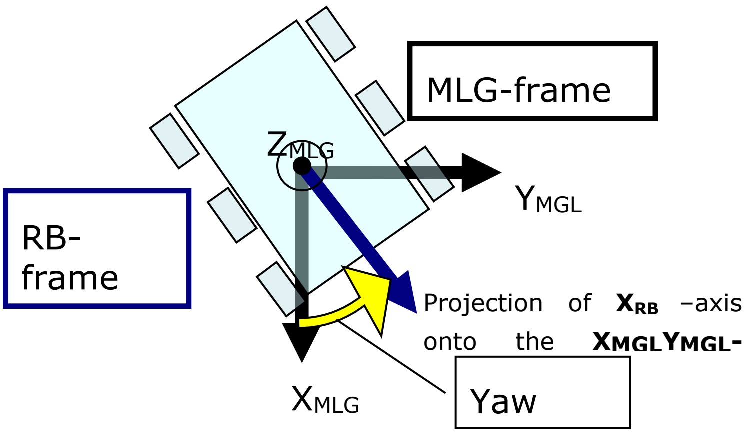

Figure 4-3: Yaw / Heading Angle Definition

4.3.2.3. PITCH ANGLE DEFINITION

The **pitch** angle is defined as the angle between the $X_{MLG}Y_{MLG}$ –plane and the X_{RB}-axis.

This definition remains valid for small and large angles irrespective of the rover's roll or yaw angle.

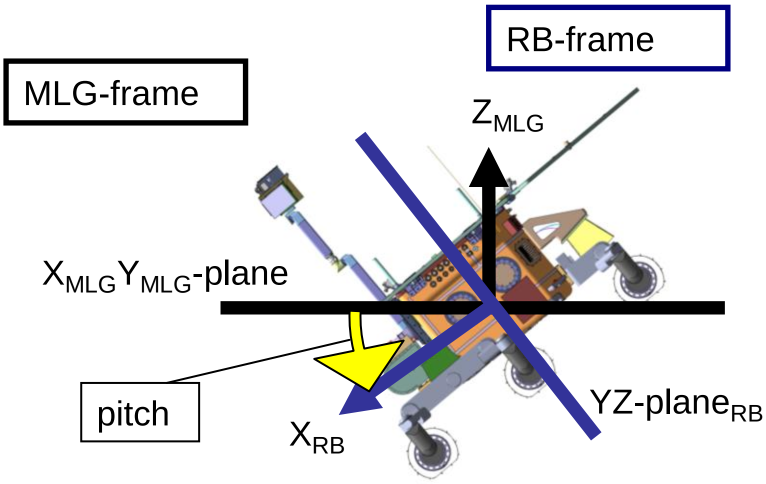

Figure 4.3-4 Pitch Angle definition

4.3.2.4. ROLL ANGLE DEFINITION

The **roll** angle is defined as the angle between the $X_{RB}Z_{MLG}$ –plane and the Z_{RB}-axis.

Notes:

1. The $X_{RB}Z_{MLG}$–plane becomes undefined at pitches of ±90 degrees. This is because X_{RB} becomes coincident with Z_{MLG} and the pair is hence insufficient to define a plane. However this does not detract from the definition as it is consistent with the 321 Euler angle singularity that occurs at pitches of ±90 degrees.
2. With the exception of the ±90 degrees pitch case, this definition remains valid for small and large angles irrespective of the rover's pitch or yaw angle.

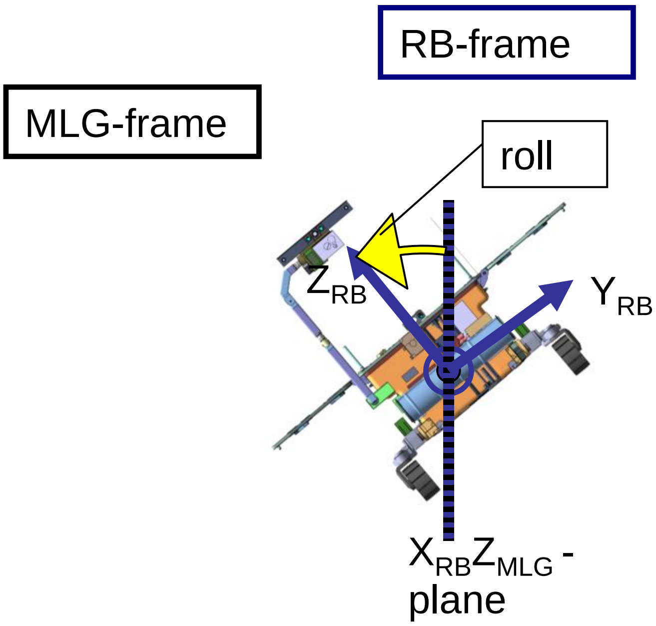

Figure 4-5: Roll Angle definition

4.3.3.ROVER SLIP RATIO

The **slip ratio** defines the relationship between the actual (effective) and drive (commanded) speed of the rover. It is usually expressed as a percentage.

$$slipRatio = \frac{V_{drive} - V_{actual}}{V_{drive}}$$ **Equation 4.3-1**

4.3.4.ROVER SLIP ANGLE DEFINITION

The **slip angle** is the angle between the rover's actual direction of travel and the direction towards which the wheels' configuration is pointing at. See Figure 4-6 below.

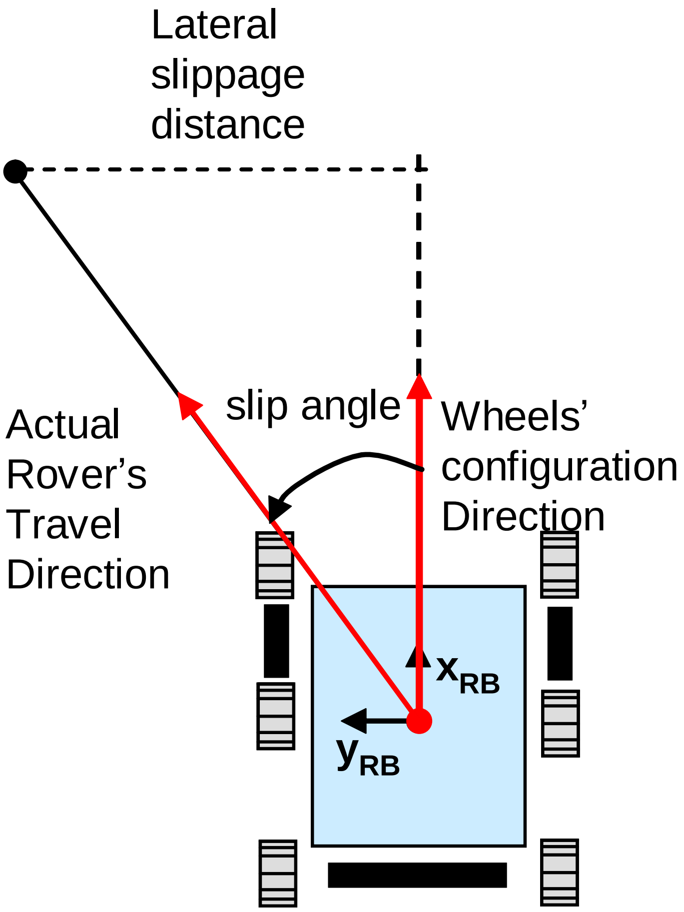

Figure 4-6: Slip Angle Definition

4.3.5.MARS FRAMES

The main Mars frames are: **Mars Local Geodetic** - a local Mars reference frame.

4.3.5.1. MARS LOCAL GEODETIC (MLG)

During a travel sequence, this frame remains fixed with Mars, but becomes reset at the start of a new travel sequence. The rationale for this is that a travel sequence requires a stationary reference frame in which to measure position and coordinate navigation maps.

The MLG frame is related to the local position of a point on, or close to, the Martian reference ellipsoid (See Figure 4-7). The Martian ellipsoid is an oblate spheroid, given by 2 parameters; the mars mean equatorial radius (the semi-major axis) and the polar radius (the semi-minor axis). It is assumed that the Martian reference ellipsoid is a good approximation of the equipotential surface of the Martian gravity field.

- The origin O_{MLG} is defined below.
- The x-axis, X_{MLG}, lies tangential to the local geodetic horizontal in an eastern direction (i.e. parallel to lines of latitude).
- The z-axis, Z_{MLG}, lies in the direction of the geodetic vertical i.e. approximately the negative gravity vector (assuming that the small local variation can be ignored).
- The y-axis, Y_{MLG}, completes the orthogonal right-handed set, and will lie northwards.

For the purposes of the RV, the origin of the frame is initially defined to be coincident with the origin of the Rover Body frame, O_{RB}, prior to the start of travel.

This frame is selected so that the Z axes of the MLG and Rover Body frame are parallel when the RV is on horizontal ground. Headings are then represented as rotations about the Z-axis of the MLG. When the RV is pointing East, both reference frames are co-aligned.

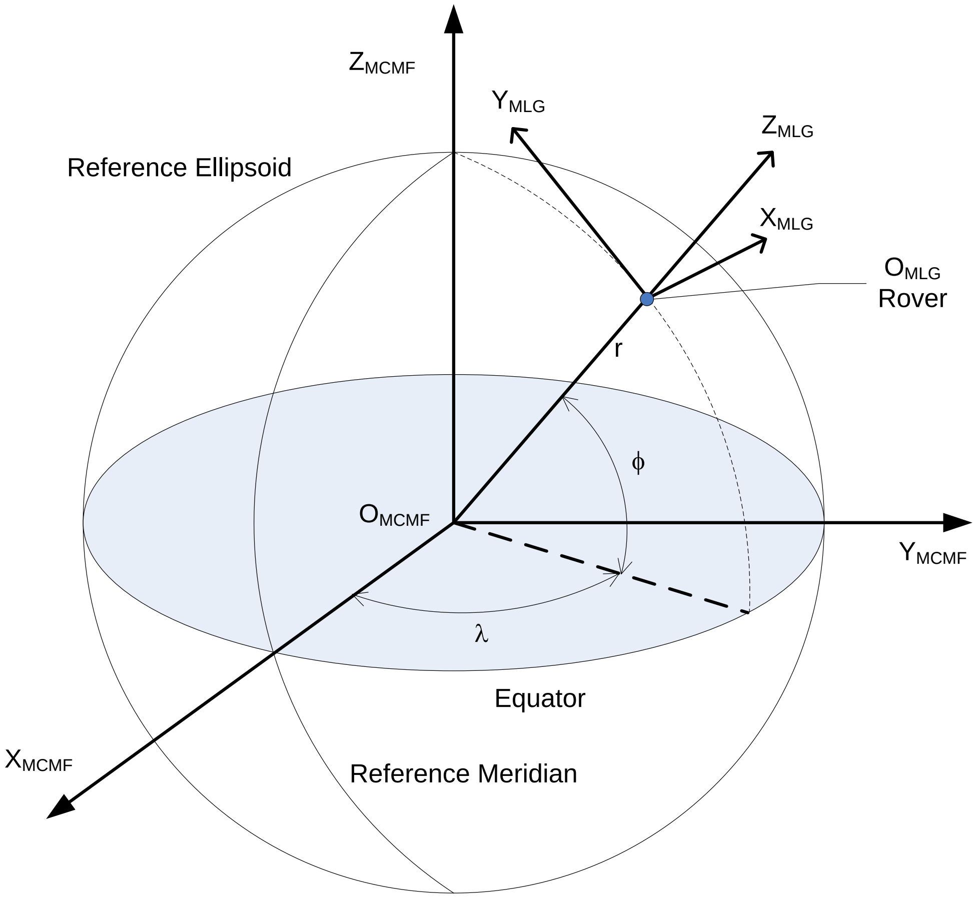

Figure 4-7: Mars Local Geodetic Reference Frame.

4.4. ON-ORBIT SERVICING SCENARIO

This scenario defines a servicing spacecraft (chaser) with a robotic arm performing operations around a serviced object in Earth's orbit (target). In the validation scenario, the target is composed by several modules which will be reconfigured by the chaser.

All frames defined below are right–handed cartesian frames. The following reference frames are defined for the scenario:

- **Earth Centered Inertial Frame (ECI):** Reference frame centred in the Earth:
  - x-axis is permanently fixed towards the vernal equinox point;
  - z-axis lies at a 90° angle to the equatorial plane and extends through the North Pole;
  - y-axis in the Earth equatorial plane, right-hand side orthogonal to axis x and z.
- **Local Orbital Coordinate System of Target (LOT):** This LVLH frame is the local reference frame for the servicing. It is placed at the centre of mass of the target:
  - z-axis lies along the geocentric radius vector to the vehicle and is positive toward the center of the Earth.
  - y-axis is normal to the orbit plane, opposite of the orbit momentum vector
  - x-axis completes the right–handed orthogonal system and positive in the direction of the vehicle motion.
- **Local Orbital Coordinate System of Chaser (LOCH)**: This LVLH frame fixed with the chaser used when not servicing. It is placed at the centre of mass of the target:
  - z-axis lies along the geocentric radius vector to the vehicle and is positive toward the center of the Earth.
  - y-axis is normal to the orbit plane, opposite of the orbit momentum vector
  - x-axis completes the right–handed orthogonal system and positive in the direction of the vehicle motion.
- **Chaser Base Frame (CBC)** and **Target Base Frame (TBF)**: The CBC and TBF frames are geometrical frame for each chaser and target involved in the on-orbit servicing manoeuvres. It has its origin at the geometrical centre of the S/C and the axes are oriented as follow:
  - x-axis is the principal axis of the S/C, it is positive in forward servicing-motion direction.
  - y-axis is parallel with the geometrical centre line of the solar panels (or similar representative elements).
  - z-axis completes the right–handed orthogonal system.
- **Arm Base:** The origin of this frame is in the first joint of the robotic arm origin of the robotic arm w.r.t. to CBC frame. It is defined w.r.t to CBC frame with the following axes:
  - x-axis is normal to the base plate of arm pointing outwards.
  - z-axis is parallel to the z-axis of the chaser.
  - z-axis and y-axis define the base plate of the arm and are orthogonal.
- **Arm Tip:** Frame associated with the robotic arm tip. It is located at the centre of the joint 6th axis (with the Z collinear to the axis as defined by the DH parameters). Its actual position is given by the direct kinematic model of the robotic arm. It is expressed w.r.t. Arm frame.
  - x-axis: It is parallel to the last robotic arm link.
  - y-axis: It is transversal to the X-axis.
  - z-axis: It is perpendicular to the arm tip plane (normal to the XY plane) pointing outwards.
- **Arm End-Effector**. This frame is associated with an end-effector placed at the tip of the robotic arm. It is expressed w.r.t. Arm Tip frame. It defines the following axes:
  - x-axis: It is parallel to the last robotic arm link.
  - y-axis: It is transversal to the x-axis.
  - z-axis: It is perpendicular to the arm tip plane (normal to the XY plane) pointing outwards.

- **Module frame**. Each module of the target has associated a frame. It is expressed w.r.t. TBF frame. The origin of the frame is in the centre of mass of the module. The axes are parallel to the TBF when in the default position.
- **Dock frame**. Each module can be grasped by the robotic arm. The origin of dock frame is in the capture point geometrical center, i.e. possible goal for the robotic arm end-effector placement. The axes are parallel with the Module frame in the default position.

Next Figure 4-8 shows the defined local frames within the On-Orbit Servicing scenario. The chaser spacecraft is set-up in the range of few metres around the target spacecraft.

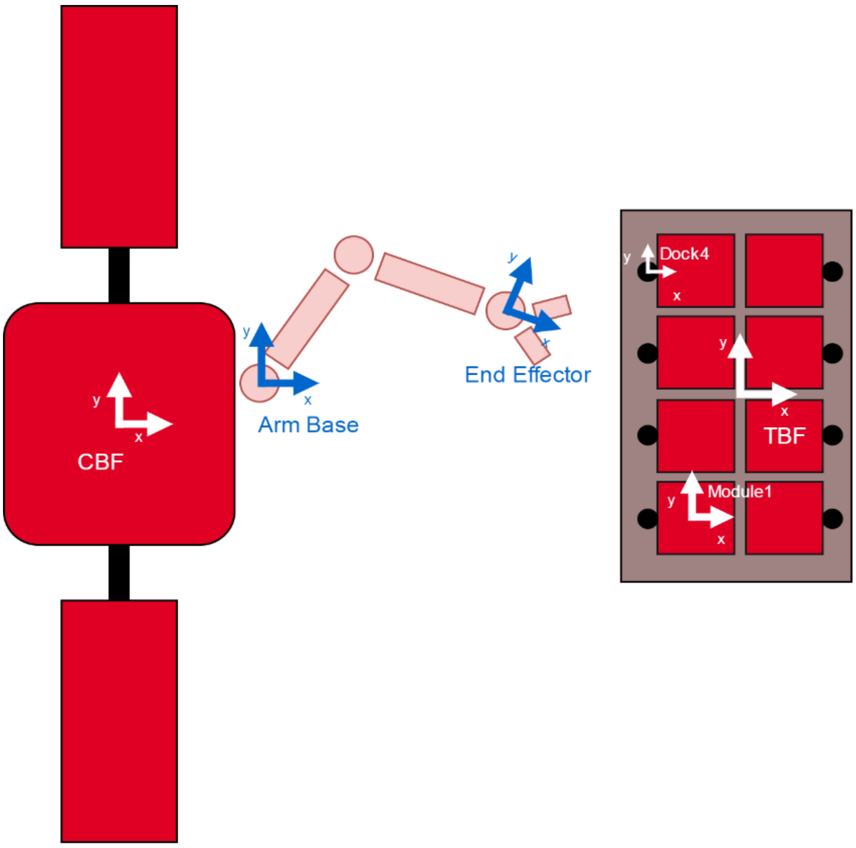

Figure 4-8: Local frames within the On-Orbit Servicing scenario. CBF is the Chaser Base Frame and TBF is the Target Base Frame.

5. INTERFACE CONTROL DOCUMENT

5.1. SUBSYSTEM ACRONYMS AND FIELDS DEFINITION

Subsystem Name	Subsystem Code
Command Dispatcher	CD
Controller	CNTLR
Failure Detection, Isolation and Recovery	FDIR
Functional Layer	FL
Goal Orientated Data Analysis	GODA
Ground Control Interface	GCI
Ground Segment	GS
Mission Planner	MP
Planner Reactor	PR
OG6 Robotic Arm On-Board SW	RASW
OG6 Rover On-Board SW	RSW
Robotic Arm Motion Planner	RAMP
Rover Guidance	RG
System Data Manager	SDM

The following fields will be used within the ICD definition:

- **Component**: Component origin and component destination.
- **Interface Identifier**: Acronym composed using the following syntax: ERG_Origin_Target-DataAcronym (example: ERG_GS_GCI-CF describing that a Configuration File (CF) goes from GS to GCI Subsystem)
- **Interface Title**: Title of the interface.
- **Comments**: Brief description of the Interface (text)
- **Origin**: Origin Entity Acronym.
- **Target**: Target Entity Acronym.
- **Frequency**: Frequency (daily, hourly, under request,…)
- **Format**: Data Format (ASCII, XML, binary)

5.2. INTERNAL INTERFACES

5.2.1.DATA INTERFACES

Table 5-1: ERGO Internal interfaces (data)

Components	Interface Identifier	Interface Title	Comments	Origin	Target	Frequency	Format
Controller Mission Planner	ERG_CNTLR_MP-CF	Configuration File	Mission planner configuration file loaded upon initialization by the controller	CNTLR	MP	On request (from origin)	ASCII
Controller Rover Guidance	ERG_CNTLR_RG-CF	Configuration File	Rover guidance configuration file loaded upon initialization by the controller	CNTLR	RG	On request (from origin)	ASCII
Controller Robotic Arm Motion Planner	ERG_CNTLR_RAMP-CF	Configuration File	Robotic Arm Motion Planner configuration file loaded upon initialization by the controller	CNTLR	RAMP	On request (from origin)	ASCII
Controller Goal Orientated Data Analysis	ERG_CNTLR_GODA-CF	Configuration File	Goal Orientated Data Analysis configuration file loaded upon initialization by the controller	CNTLR	GODA	On request (from origin)	ASCII
Controller Mission Planner	ERG_CNTLR_MP-DOM	Domain	Describes the system to be controlled and the environment. The domain is part of the information required during the configuration of the MP reactor	CNTLR	MP	On request (from origin)	Binary: Domains shall remain unchange d along the mission. Therefore, the parser can stay on-ground

Components	Interface Identifier	Interface Title	Comments	Origin	Target	Frequency	Format
GCI Mission Planner	ERG_GCI_MP-Token	Token	A token (goal) that contains a partial problem with one or several goals to be achieved plus some additional information such as invariant values related to the mission or a metric to evaluate the plan quality	GCI	MP	On request (from origin)	Binary
Mission Planner Rover Guidance	ERG_MP_RG-G_RGPath	Token	A token (goal) that contains as a goal the computation of a path between waypoints	MP	RG	On request (from origin)	Binary
Rover Guidance Mission Planner	ERG_RG_MP-O_RGPath	Token	A token (observation) that contains the path and its related time and resources required	RG	MP	When Changed	Binary
Mission Planner Robotic Arm Motion Planner	ERG_MP_RAMP-G_RAMPPlan	Token	A token (goal) that contains as a goal the computation of the pick/drop/move operation and its related time and resources required	MP	RAMP	On request (from origin)	Binary
Robotic Arm Motion Planner Mission Planner	ERG_RAMP_MP-O_RAMPPlanObs	Token	An token (observation) that contains the planned movement and its related time and resources required	RAMP	MP	When Changed	Binary
Mission Planner Robotic Arm Motion Planner	ERG_MP_RAMP-G_RAMPExecGoal	Token	A token (goal) that contains as a goal the execution of the pick/drop/move operation and its related time and resources required	MP	RAMP	On request (from origin)	Binary
Robotic Arm Motion Planner Mission Planner	ERG_RAMP_MP-O_RAMPExecObs	Token	An token (observation) that informs the result of the executuion and its related time and resources required	RAMP	MP	When Changed	Binary
Ground Control Interface Goal Orientated Data Analysis	ERG_GCI_GODA-G_Ident	Token	A token (goal) that contains as a goal the identification of novel targets in a picture	GCI	GODA	On request (from origin)	Binary

Components	Interface Identifier	Interface Title	Comments	Origin	Target	Frequency	Format
Goal Orientated Data Analysis Ground Control Interface	ERG_GODA_GCI-O_Ident	Token	A token (observation) containing a new observation, a ranking of its relevance and the region of the picture it has been detected	GODA	GCI	When Changed	Binary
Ground Control Interface Command Dispatcher	ERG_GCI_CD-G_E1_E3	Token	A token (goal) containing a low-level goal value desired for an internal timeline of the command dispatcher. While working in E3,E2,E1 mode, the GCI will be able to post goals to any internal timeline of the CD	GCI	CD	On request (from origin)	Binary
Mission Planner Command Dispatcher	ERG_MP_CD-G_E4	Token	A token (goal) containing a low-level goal value desired for an internal timeline of the command dispatcher. While working in E4 mode, the MP will be able to post goals to any internal timeline of the CD	MP	CD	On request (from origin)	Binary
Command Dispatcher MissionPlanner	ERG_CD_MP_OBS	Token	A token (observation) containing an observation on an internal timeline of the command dispatcher: robotbase, communications, etc… all changes in the functional are reported by these observations	CD	MP	When changed	Binary
Command Dispatcher GCI	ERG_CD_GCI_OBS	Token	A token (observation) containing an observation on an internal timeline of the command dispatcher: robotbase, communications, etc… all changes in the functional are reported by these observations	CD	GCI	When changed	Binary
System Data Manager Goal Orientated Data Analysis	ERG_SDM_GODA-IR	Image Reference	Reference to an image used by GODA to detect science targets or space assets.	SDM	GODA	On request	Binary

Components	Interface Identifier	Interface Title	Comments	Origin	Target	Frequency	Format
Goal Orientated Data Analysis System Data Manager	ERG_GODA_SDM-GGOAL	Grounded Goal	A grounded goal (concept from planning terminology) is a goal which parameters have been assigned specific values. A GODA goal is a specific instantiation of the Goal presented above. As a result of the detection process, GODA can return a list of goals, each containing the following information: Target class; Relative coordinates with respect to the image; Metric value.	GODA	SDM	On request	Goal
System Data Manager Rover Guidance	ERG_SDM_RG-T	Target	Coordinate of the target location (2D) and desired heading to which the Rover Guidance shall plan a path.	SDM	RG	On request	
Rover Guidance System Data Manager	ERG_RG_SDM-GGOAL	Grounded Goal	A grounded goal is a goal which parameters have been assigned specific values. A Rover Guidance goal is a specific instantiation of the Goal presented above. As a result of the path planning process, RG can return a goal containing the following information: Path from current location to the target; estimated resources required to achieve the planned path.	RG	SDM	On request Update every time a short term path is planned or updated for hazard avoidance.	Goal
Rover On-Board Software Rover Guidance	ERG_RSW_RG_RoverMap	Rover Map	Refer to OG2-OG3 corresponding interface. This interface will be inspired and will be instantiated adapting it to the OG6 platform capabilities.	RSW	RG	On request (from origin)	Medium frequency, value TBD
Rover On-Board Software Rover Guidance	ERG_RSW_RG_FusedRoverMap	Fused Rover Map	Refer to OG2-OG3 corresponding interface. This interface will be inspired and will be instantiated adapting it to the OG6 platform capabilities.	RSW	RG	On request (from origin)	Medium frequency, value TBD

Components	Interface Identifier	Interface Title	Comments	Origin	Target	Frequency	Format
Rover On-Board Software Rover Guidance	ERG_RSW_RG_FusedTotal Map	Fused Total Map	Refer to OG2-OG3 corresponding interface. This interface will inspired and will be instantiated adapting it to the OG6 platform capabilities.	RSW	RG	On request (from origin)	Low frequency, value TBD
Rover On-Board Software Rover Guidance	ERG_RSW_RG_RoverPose	Rover position and attitude	Refer to OG2-OG3 corresponding interface. This interface will be inspired and will be instantiated adapting it to the OG6 platform capabilities.	RSW	RG	10 Hz TBC	10Hz while driving
Rover Guidance Rover On-Board Software	ERG_RG_RSV_RoverCmd	Rover commanding	Rover level commands, e.g. Ackerman Turn (radius of turn and rover speed) or Point Turn parameters (rover turn rate).	RG	RSW	Commanding expected every 1cm travelled, i.e. for maximal speed of 10cm/s, this means commanding at 10Hz.	TBD with OG6
Robotic Arm On-Board Software Robotic Arm Motion Planner	ERG_RASW_RAMP_Target 3DModel	Target 3D Model	Refer to OG2-OG3 corresponding interface. This interface will be inspired and will be instantiated adapting it to the OG6 platform capabilities.	RASW	RAMP	On Request from OG2-RAMP	Medium frequency, value TBD
Robotic Arm On-Board Software Robotic Arm Motion Planner	ERG_RASW_RAMP_Target Pose	Target position and attitude	Refer to OG2-OG3 corresponding interface. This interface will be inspired and will be instantiated adapting it to the OG6 platform capabilities.	RASW	RAMP	10 Hz	TBD
System Data Manager Robotic Arm Motion Planner	ERG_SDM_RAMP-T	Desisted Target Location	Robotic arm XYZ tip position.	SDM	RG	On request (from MPPre or CD)	TBD

Components	Interface Identifier	Interface Title	Comments	Origin	Target	Frequency	Format
Robotic Arm Motion Planner System Data Manager	ERG_RAMP_SDM-GGOAL	Grounded Goal	A grounded goal is a goal which parameters have been assigned specific values. A Motion Planner goal is a specific instantiation of the Goal presented above. As a result of the motion planning process, RAMP can return a goal containing the following information: sequence of robotic arm XYZ position and joints angles; estimated resources required to achieve the planned path.	RG	SDM	On request (from MPPre or CD) Update every time a short term path is planned or updated for hazard avoidance.	Goal

5.2.2.COMMAND INTERFACES

Table 5-2: ERGO internal Interfaces (commands)

Component	Interface Identifier	Interface Title	Comments	Origin	Target	Frequency	Format
Planner Reactor Mission Planner	ERG_PR_MP-PLAN	Planning	Given a domain and problem, computes a new plan.	PR	MP	On request (from origin)	Domain; Problem
Planner Reactor Mission Planner	ERG_PR_MP-REPLAN	Re-Planning	Given a domain, an existing plan and potentially a new list of goals, updates the current plan.	PR	MP	On request (from origin)	Domain; Problem
Mission Planner Preprocessor Rover Guidance	ERG_MPPREP_RG-CALCPATH ERG_CD_RG-CALCPATH	Calculate a path	Command requesting a path planning activity. It will not start moving the rover.	MPPrep CD	RG RG	On request (from origin)	Target
Command Dispatcher Rover Guidance	ERG_CD_RG-DRIVE	Drive Path	Command requesting to drive the planned path.	CD	RG	On request (from origin)	Path

Component	Interface Identifier	Interface Title	Comments	Origin	Target	Frequency	Format
Command Dispatcher Rover Guidance	ERG_CD_RG-STOP	Stop path driving	Command requesting stopping of path driving.	CD	RG	On request (from origin)	None
Mission Planner Preprocessor Command Dispatcher Robotic Arm Motion Planner	ERG_GCI_RAMP-CALCPATH ERG_MP_RAMP-CALCPATH	Plan arm operation (pick/drop/ move)	Command requesting a motion planning activity. It will not start moving the robotic arm.	PR GCI	RAMP RAMP	On request (from origin)	Initial state Goal
Mission Planner/GCI Robotic Arm Motion Planner	ERG_GCI_RAMP-OP ERG_MP_RAMP-OP	Perform Operation	Command requesting to perform pick/drop/move the robotic arm.	GCI/MP	RAMP	On request (from origin)	Operation, coordinates
Mission Planner/GCI Robotic Arm Motion Planner	ERG_GCI_RAMP-STOP ERG_MP_RAMP-STOP	Stop the arm	Command requesting stopping of robotic arm	GCI/MP	RAMP	On request (from origin)	Operation, coordinates
Planner Reactor Command Dispatcher Goal Orientated Data Analysis	ERG_PR_GODA-ProcessImage ERG_CD_GODA-ProcessImage	GODA Process Image	The input is an image, or reference to image file, and the output is a list of image co-ordinates, combined with suggested goals and a metric to assess their importance.	MPPrep CD	GODA	On request (from origin)	Image Reference
Command Dispatcher Functional Layer	ERG_CD_FL-G_TasteF_XX	Taste Function	TASTE function (list of TASTE functions of the functional layer TBD) required by the Command Dispatcher that is used to send commands to the functional layer	CD	FL	On request (from origin)	Binary
Functional Layer Command Dispatcher	ERG_FL_CD-O_TasteF_XX	Taste Function	TASTE function provided by the CD that receives an observation from the functional layer modules or an ERGO Commandable Subsystem. It is an action populated with additional information received from sensors (complete list TBD)	FL	CD	On request (from origin)	Binary

Component	Interface Identifier	Interface Title	Comments	Origin	Targe t	Frequency	Format
Functional Layer Failure Detection, Isolation and Recovery	ERG_FL_FDIR-Detect	Detect Faults	Command requesting the FDIR TASTE function to detect, identify, and react to faults.	FL	FDIR	On request from monitoring	Observable State
Functional Layer Failure Detection, Isolation and Recovery	ERG_FDIR_FL-Recover	Recover	Command sent by the FDIR functional layer TASTE function to the functional layer to recover from a fault.	FDIR	FL	On request (from origin)	Recovery Sequence
Functional Layer Failure Detection, Isolation and Recovery	ERG_FDIR_FL-Isolate	Isolate	Command sent by the FDIR module to the functional layer to isolate faulty components and to prevent from error propagation in the system. It contains the set of isolations actions to be executed.	FDIR	FL	On request (from origin)	Isolation Actions

5.3. GROUND SEGMENT INTERFACES

This section covers the foreseen interfaces between the Ground-Segment and the ERGO system related to Telemetry and Telecommands. More details about each of the contents of the telemetry and telecommand messages will be provided during next ICD release (PDR milestone).

5.3.1. TELEMETRY

ERGO will acquire on-board data and will convert it into telemetry. The following on-board data are possible data to be sent via telemetry:

- On-board time, current tick and autonomy level.
- Robot basic information and data status.
- State of resources and health status of sensors.
- Current plan of each deliberative reactor.
- Current state of all timelines.
- Active OBCPs
- Requests sent by the command-dispatcher reactor to the functional layer, and replies received from the functional layer.

The header (spacecraft id, timestamp and tick) is mandatory. Other parameters are optional.

5.3.2. TELECOMMANDS

The ERGO system will mimic the PUS Services that would be traditionally used in a real space mission (Service 144, Service 13, etc.). We will assume that a PUS file service is able to put a file on a recipient directory. On every tick, ERGO will check if there is a new telecommand file to be processed in that directory. When the file is processed, it will be removed from the recipient directory. Different files are used for commanding different autonomy levels. A special telecommand file will be used for the selection of the operational mode, i.e. for the autonomy level.

To summarize, ERGO will accept the following telecommands:

- **Configuration Telecommand:** This telecommand is embedded within the file TC_autonomy.dat. This file is accepted at any time and its contents are executed immediately, and before any other telecommand. It indicates the level of autonomy to be used from now on.
- **ECSS E1 Level Telecommand**: This telecommand is embedded within the file TC_E1.dat. This file is accepted when the autonomy level is E1, E2 or E3. It contains commands for immediate execution. It will be executed before TC_E2.dat, TC_E3_*.dat.
- **ECSS E2 Level Telecommand**: This telecommand is embedded within the file TC_E2.dat: This file is accepted when the autonomy level is E2 or E3. It contains time-tagged commands. It will be executed before TC_E3_*.dat
- **ECSS E3 Level Telecommand**: This telecommand is embedded within the file TC_E3_*.dat: These files are accepted when the autonomy level is E3. They contain events and actions associations and OBCPs.
- **ECSS E4 Level Telecommand**: This telecommand is embedded within the file TC_E4.dat. This file is accepted when the autonomy level is E4.

5.3.2.1. CONFIGURATION TELECOMMAND

The Configuration Telecommand (TC_autonomy.dat) contains information about the operational mode, that is, the autonomy level that must be running and its associated configuration:

- Controller parameters
- Mission planner parameters

- Rover guidance parameters
- Robotic Arm Motion planner parameters
- GODA parameters
- Planning Domain
- Planning Problem
- Flexible timeline plan

It can be requested as an immediate command or as a time-tagged command. In case of E4, optionally it is possible to specify some reactors to be disabled. In the latter case, a reactor will be actually disabled only if all reactors depending on it are also disabled.

The format of TC_autonomy.dat is as follows:

<sc-id><exec-type><timetag><tc-id>

DIS <reactor-name>

...

DIS <reactor-name>

The following conventions are adopted:

- <**exec-type**> Execution type: 'I' for immediate execution; 'T' for scheduled execution, in the latter case, the next line must be a timetag time.
- <**timetag**> Execution time in case <exec-type> = 'T'. It is assumed that it can be directly used as an on-board time. Format: YYYY.DDD.hh.mm.ss
- <**tc-id**> Telecommand identifier, uniquely identifying a telecommand. Allowed telecommands and on-board behaviour:
  - TCE1: It selects E1 level of autonomy
  - TCE2: It selects E2 level of autonomy
  - TCE3: It selects E3 level of autonomy
  - TCE4: It selects E4 level of autonomy
- <reactor-name> An ASCII string representing the name of an ERGO reactor to be disabled.
- DIS <reactor-name> commands are optional and only applicable if <tc-id> = TCE4.

5.3.2.2. ECSS E4 LEVEL TELECOMMAND

When the operational mode is goal-oriented, i.e. E4, ERGO will accept commands with the following contents:

- **Problem**. Describes the initial conditions and the goals to be achieved by the plan
- **Goals**. Structure containing a PDDL Predicate, its related parameters and meta-information associated to the goal such as a metric, constraints, etc. Goals defined on-ground can be hard (mandatory) or soft (optional).
- **Flexible timeline plans**. Timeline-based plan containing for each subsystem (timeline) a sequence of durative actions which starting time and duration are time intervals.

5.3.2.3. ECSS E1 LEVEL TELECOMMAND

The ECSS E1 level telecommand is embedded within the TC_E1.dat file. It contains commands for immediate execution. The format of TC_E1.dat is as follows:

<sc-id><tc-id> <tc-parameters>

The following conventions are adopted:

- <**tc-id**>: It is the telecommand identifier, uniquely identifying a telecommands. There is a one-to-one relationship between E1 telecommands and primitives of the functional layer.

- <**tc-parameters**>: There are the telecommands parameters, which depend on the specific telecommands.

5.3.2.4. ECSS E2 LEVEL TELECOMMANDS

The ECSS E2 level telecommand is embedded within the TC_E2.dat file. This file contains commands for scheduled execution. This file plays the role of a mission timeline. The format of TC_E2.dat is as follows:

<sc-id><timetag><tc-id><tc-parameters>

The following conventions (including previous ones from E1 telecommands) are adopted:

- **<timetag>**. It provides the schedule execution time of the telecommand. It is assumed that it can be directly used as an absolute on-board time. Format: YYYY.DDD.hh.mm.ss

5.3.2.5. ECSS E3 LEVEL TELECOMMANDS

The ECSS E3 level telecommand are embedded within the TC_E3_*.dat file. Two different aspects are characteristic of E3 operations: the on-board capability of event-driven execution of commands and the on-board capability of executing control procedures (OBCPs).

The table of events and actions can be changed by command, but the set of events that can trigger actions is predefined and cannot be changed. It might be changed by modifying the on-board software, but this is out of the scope of the prototype.

Several types of telecommand files will be used for E3:

- TC_E3_OBCP_xx.dat: A file of this type contains an OBCP (syntax TBD). 'xx' uniquely identifies the OBCP.
- TC_E3_EAT.dat. It contains the events and actions table, the whole table. An action can be any E1/E2 command or an E3 control command.
- TC_E3_ctrl.dat. This file follows the same format as TC_E2.dat, but the allowed telecommands are the so called E3 control commands:
  - COPSA <xx> Start OBCP 'xx'.
  - COPSO <xx> Stop OBCP 'xx'.
  - COPLD <xx> Load OBCP 'xx', which copies it to the area of OBCPs
  - COPDE <xx> Delete OBCP 'xx' from the area of OBCPs

  This file plays the role of a mission timeline.
  - TC_E3_MD.dat It contains the metadata, such as communication windows.

5.3.2.6. GROUND-SEGMENT TM/TC ICD

Table 5-3: Ground segment TM/TC ICD

Component	Interface Identifier	Interface Title	Comments	Origin	Target	Frequency	Format
Ground Segment Controller	ERG_GS_CNTLR-File	Config File	Transfer of files required by the architecture such as configuration files and the domain model for the planner	GS	CNTRL	On request (from ground)	ASCII/binary
Controller Ground Segment	ERG_CNTLR_GS-File	File	GS uses the TM_IF interface to download TM files via PUS.	CNTRL	GS	On request (from ground)	ASCII
Ground Segment Ground Control Interface	ERG_GS_GCI-TC	Telecommand File	GS use the TC_IF interface to upload TC files via PUS. ERGO will accept the following telecommands: TC_E1, TC_E2, TC_E3 and level of autonomy TC_Autonomy	GS	GCI	On request (from origin)	ASCII
GCI-> Ground Segment	ERG_GCI_GS-TM	Telemetry	The following on-board data are possible data to be sent via telemetry: On-board time and current tick, autonomy level, timeline states, current plan and execution status, functional layer subsystems status and data, etc...	GCI	GS	Hourly (TBC)	ASCII
Ground Segment Ground Control Interface	ERG_GS_CNTLR-CF	Configuration File	Controller parameters,	GS	CNTLR	On request (from ground)	ASCII
Controller Mission Planner	ERG_CNTLR_MP-DOM	Domain	Describes the system to be controlled and the environment. The domain is part of the information required during the configuration of the MP reactor	CNTLR	MP	On request (from origin)	Binary: Domains shall remain unchanged along the mission. Therefore, the parser can stay on-ground

Component	Interface Identifier	Interface Title	Comments	Origin	Target	Frequency	Format
Ground Segment Ground Control Interface	ERG_GS_GCI-Problem	Problem	A file that contains a partial problem including one or several goals to be achieved plus some additional information such as invariant values related to the mission or a metric to evaluate the plan quality	GS	GCI	On request (from origin)	ASCII: As problems might be created on the fly, their processing into binary format must be done on-board

5.4. EXTERNAL INTERFACES WITH OTHER OG,S

The following tables shows the interface requirements with OG1,OG3,OG4 and OG6 as described in [RD.2].

The format of these tables is based on the following columns:

- The TYPE literal defines the nature of interfaces and is defined as:
  - D data
  - F functional
  - P power
  - O operational/procedural
  - S software
  - M mechanical
  - T thermal
- The two sequence numbers (Major and Minor) are meant to provide further categorization of the interface in 2 levels.
- The version.revision field is to provide for the ability to evolve the interface.
- Title of the interface.

5.4.1.OG2 ←→ OG1 INTERFACES

The only OG1 interfaces are off-line tools that will be used during the development. It is understood that OG1 will be responsible of the correct behaviour of the automatically generated code when using these tools in OG2.

Table 5-4: OG1-OG2 Interfaces

Type	Major sequence	Minor sequence	Version	Title	Description
D	1	0	V0.1	ASN.1 Types	OG1 shall provide TBD[1] types defined in ASN.1 to represent motion and plan actions data.
F	1	1	V0.1	TASTE IV	OG1 will provide the TASTE IV for the definition of the interfaces between the agent and the functional layer, and generate the corresponding code based on it.
F	2	1	V0.1	TASTE Deployment View	OG1 will provide the TASTE Deployment view for the definition of the executables (nodes) and the way they communicate, and generate the corresponding code for the interfaces, being the automatically generated code responsible of handling properly the low-level protocols associated (CAN bus; EtherCAT).
F	3	1	V0.1	IMA Modelling	OG1 will provide to OG2 the possibility to define partitions of different criticality, to be deployed using ESROCOS Deployment view and to be run under AIR.
F	4	1	V0.1	Constraint definition	OG1 will provide to OG2 the possibility to detect violations of constrains at the design phase using BIP restrictions.
F	5	1	V0.1	Taste Dataview	OG1 will provide to OG2 the TASTE DataView to define different datatypes used among different functions in ASN.1 (independently of the language)
F	6	1	V0.1	Interface properties	OG1 will provide the capability to detail the following characteristics of any interface: mutual exclusion of the interface function w.r.t other interface functions, periodicity of the interface, and whether the interface is synchronous or asynchronous
F	7	1	V0.1	PUS Services	OG1 will provide as a minimum the following PUS services to be used by OG2: time-tag commanding, Event-action service, On-board control procedures

5.4.2.OG2 ←→ OG3 INTERFACES

The following are the interfaces between OG2 and OG3 [RD.2].

Table 5-5: OG2-OG3 Interfaces

Type	Major sequence	Minor sequence	Version	Title	Description
D	1	1	V0.1	Rover Map	OG3 will provide to OG2 the Map produced with information gathered by sensors on the rover itself at the last sensing capture. This includes a DEM (describing the height associated to each cell in a grid).
D	2	1	V0.1	Fused Rover Map	OG3 will provide to OG2 the Map produced with information gathered by sensors on the rover itself at the last and previous sensing captures. This is used by the rover guidance to perform short term path planning. This includes a DEM and Uncertainty map.
D	3	1	V0.1	Fused Total Map	OG3 will provide to OG2 the Map produced with information from any sensing sources at any capturing time, e.g. rover, orbital, other mobile or static devices on the surface. This is used by RG to perform long term path planning. This will include a DEM and Uncertainty map.
D	4	1	V0.1	Attitude and position	OG3 will provide to OG2 the 3-axes position and 3-axes attitude using data structures defined in OG-1
D	5	1	V0.1	Target 3D model	OG3 will provide to OG2 the 3D Model of the target Spacecraft as perceived by the chaser S/C
D	6	1	V0.1	Target position and attitude	OG3 will provide to OG2 the estimated target 3-axes position and 3-axes attitude
D	4	2	V0.1	Planetary - Rover	This interface produces the Rover Map.
D	4	3	V0.1	Planetary - Rover	This interface produces the Fused Rover Map
D	4	4	V0.1	Planetary - Model	This interface produces the Fused Total Map
D	4	5	V0.1	Planetary – Local Pose	This interface produces the LocalPose Pose of the BodyFrame in the LocalTerrainFrame
D	4	6	V0.1	Planetary – Global Pose	This interface produces the GlobalPose Pose of the BodyFrame in the GlobalTerrainFrame
D	4	7	V0.1	Planetary – Absolute Pose	This interface produces the AbsolutePose Pose of the BodyFrame in the AbsoluteFrame
D	7	1	V0.1	Orbital - Target	This interface produces the 3D model of the target spacecraft.
D	8	1	V0.1	Planetary – Relative Target Pose	This interface is the relative pose (3 axes position and 3 axes attitude) of the target Body Frame expressed in the chaser Body Frame, with associated uncertainties (format to be defined)

Type	Major sequence	Minor sequence	Version	Title	Description
D	9	1	V0.1	Planetary – Relative Target Speed	This interface is the relative speed (3 axes translation speeds and 3 axes rotation speeds) of the target Body Frame expressed in the chaser Body Frame, with associated uncertainties (format to be defined)
F	1	1	V0.1	Fused data retrieval generic interface.	OG3 will provide a generic API that allows OG2 to ask OG3 to return the data in case it has computed it by means of a set of primitives .
S	1	1	V0.1	Fused data request generic interface.	OG3 will provide a generic API that allows OG2 to request under demand different fused data by means of the primitive requestFusedData.
S	1	2	V0.1	Initialization of OG3	OG3 provides an interface to OG2 to initialize, reset or put the CDFF framework into an idle state (no processing). This can be used by OG2 to command the runtime internal state of the data fusion framework (not to be confused with the OG1 run time states). The specific parameters could be desired state of system:
S	2	1	V0.1	Querying OG3	OG3 provides an interface for OG2 to query and get a notification on the current state of the CDFF framework to OG2. This can be used by OG2 to check the state before posting a new request that influences the internal runtime state of OG3 as indicated in OG2-OG3
S	3	1	V0.1		"OG3 provides an interface for OG2 to query the CDFF orchestrator based on a set of criteria of parameters for activating DFPCs. OG3 internally maintains a list of possible CDFFs that can be selected indirectly by OG2 based on a specific set of parameters such as: 1. List of sensors activated by OG2 2. Default configurations or operational modes of sensors 3. Type of fused maps (rover map, fused rover map and fused total mapl) 4. Resolution of fused data (maps) 5.Update frequency (localization and map updates) 6.Area coverage (map) 7.Time bounds for providing specific fused data for aperiodic requests 8.Priority of request (to manage request queue in OG3) 9.The orchestrator software component within CDFF has the task to map the criteria to a specific DFPC that can optimally satisfy the above requirements.
S	3	2	V0.1		This interface can be used by OG2 to get an update regarding the runtime status of the DFPC in terms of the quality of data inputs from OG4 (raw and pre-processed) and the quality of fused data products in terms of maps and localization data.

Type	Major sequence	Minor sequence	Version	Title	Description
S	4	1	V0.1		Interface provided by OG2 to OG3 for providing fused data updates. There are 2 mechanisms that are foreseen: 1. For aperiodic requests from OG2 (Ex. absolute localization, global fused map etc.), OG3 will notify OG2 regarding the availability of fused data and provide access to the data buffer with corresponding information on the data type. The notification is foreseen to be asynchronous due the inherent delay in producing fused data for a specific query from OG2. 2. For OG2 requests that require OG3 to generate periodic data (local rover map, high frequency local pose estimates etc.), a predefined data stream for specific data types will be used to send the fused data periodically to OG2.
O	2	0	v0.1	Power	OG2 is responsible for initialising, power up and power down of OG3 data fusion and OG4 sensors suite in a coordinated way.

5.4.3. OG2 ←→ OG4 INTERFACES

The aim of theses interfaces is to standardize the exchange of information between OG4 and OG2 for future integration (not foreseen within the current activity). These interfaces are inspired and will be adapted to the OG6 platform capabilities according to the validation scenario (e.g. contact and FT sensors are not foreseen to be interfaced but emulated with other sensors).

The following are the interfaces between OG2 and OG4 [RD.2].

Table 5-6: OG2-OG4 Interfaces

Type	Major sequence	Minor sequence	Version	Title	Description
D	1	1	V0.1	Image	OG4 will be able to provide to OG2 images from the cameras
D	2	1	V0.1	Contact Sensor	OG4 will provide to OG2 performed contact status
D	3	1	V0.1	Contact Sensor	OG4 will provide to OG2 Force (Fx, Fy, Fz) and Torque (Tx, Ty, Tz) validated measurements
D	4	1	V0.1	Torque forces	OG4 will provide to OG2 Force (Fx, Fy, Fz) and Torque (Tx, Ty, Tz) validated measurements
D	5	1	V0.1	Sensor Status.	OG4 will provide to OG2 the status of all sensors
D	1	0	V0.1	ASN1 Messages	OG2 supports ASN.1 messages defined by OG4 following the OG1 specification for all communication with OG4.
F	1	1	V0.1	Sensor configuration interface	OG4 will provide an API, that allows OG2 to request under demand a specific configuration of a given sensor by means of the primitive requestSensorConfiguration, which would require the following information configParams List of parameters that helps OG4 to process the request.
F	2	1	V0.1	Sensor data request generic interface	OG4 will provide an API that allows OG2 to request under demand data by means of the primitive requestSensorData, which would require the following information requestData List of parameters that helps OG4 to process the request. priority Indicates the priority of the request. preprocessed Indicates whether the sensor should pre-process or not the data.
F	3	1	V0.1	Sensor diagnostics	OG4 will provide to OG2 interfaces for determining the status and diagnostics of sensors.
F	1	0	V0.1		OG2 receives the same sensor data as OG3 (refer to their functional requirements section) through the OG1 middleware, and can therefore use raw measurement data without sensor fusion if needed.

Type	Major sequence	Minor sequence	Version	Title	Description
F	2	0	V0.1		OG4 provides proprioceptive sensors (FT, IMU, STR) raw data to OG2.
F	3	0	V0.1		OG4 provides an interface selected individual sensor configuration that could be used by OG2. This is complementary to the operational modes. The exact sensor configurations supported are TBD.
O	1	0	V0.1		OG4 does not check the sender of command messages, therefore OG2 and OG3 must coordinate commanding of OG4 to avoid conflicting commands.
O	2	0	V0.1		OG4 does not store measurements for later retrieval, therefore OG2 receives the data at the specified rate.
O	3	0	V0.1		OG2 is responsible for initialising and power up of OG4 sensors suite in a coordinated way.

5.4.4.OG2 ←→OG5 INTERFACES

OG2 is a framework for autonomy. As such, it provides a template for the development of autonomous systems. Although it is not foreseen in this call to perform the interconnection of OG2 with OG5, the framework provided by OG2 can be extended to interact with OG5 (SIROM) devices. This should be done by creating a functional layer that includes the primitives required to interact with SIROM devices. This functional layer will be implemented in the form of TASTE functions using OG1 framework as modelling tool environment by the application developer. This is work to be done in the frame of the next call.

- It is understood that the functional layer implemented via OG1 primitives shall communicate with SIROM through two data and control interfaces
- This functional layer will implement primitives that through the CAN bus interface can discover and identify all SIROM interconnects attached to the CAN bus using the CANOpen protocol.
- Both telecommanding and polling for telemetry will be accomplished through the corresponding TASTE functions that will communicate with the SIROM device.

5.4.5.OG2 ←→ OG6 INTERFACES

These interfaces describe the needs from the point of view of ERGO w.r.t the needs from the use cases hardware.

These requirements are commented in the preliminary design document.

The following are the interfaces between OG2 and OG6 [RD.2].

Table 5-7: OG2-OG6 interfaces

Type	Major sequence	Minor sequence	Version	Title	Description
D	1	1	V0.1	Planetary - Rover - 2D Position and Heading interface	OG6 will provide an API that allows OG2 to request under demand 2D rover position and heading.
D	2	1	V0.1	Planetary – Rover - On-board camera images	OG6 will provide an API that allows OG2 to request under demand images (stereo or monocular) from the on-board cameras (panoramic and navigation).
D	3	1	V0.1	Planetary – Rover – Sensor information.	OG6 will provide an API that allows OG2 to request under demand sensoring information from the platform (e.g. rover odometry, IMU, etc.).
D	4	1	V0.1	Planetary – Orbiter - Orbital map and images	OG6 will provide an API that allows OG2 to request under demand orbital maps and images of the area to be traversed.
D	5	1	V0.1	Planetary – Rover - DEM - DTM map	OG6 will provide an API that allows OG2 to request under demand DEM-DTM maps.

Type	Major sequence	Minor sequence	Version	Title	Description
D	6	1	V0.1	Planetary – Rover – Ground data truth	OG6 will provide an API that allows OG2 to request ground data truth in terms of true rover position and attitude.
D	7	1	V0.1	Planetary – Simulator – Environment and Rover models	OG6 will provide a model of the environment and rover to so that OG2 can conduct virtual tests.
D	8	1	V0.1	Planetary – Rover – Ground data truth	OG6 will provide an API that allows OG2 to request ground data truth in terms of true S/C attitude and orbit.
D	9	1	V0.1	Orbital - Chaser – Attitude and orbit interface	OG6 will provide an API that allows OG2 to request under demand attitude and position from the robotic platform.
D	10	1	V0.1	Orbital – Chaser – Other Sensor information	OG6 will provide an API that allows OG2 to request under demand sensoring information from the robotic platform (e.g. imagery)
F	1	1	V0.1	Planetary – Rover - 2D Motion command interface	OG6 will provide an API that allows OG2 to request to move the rover platform. This is provided with rover level commands, e.g. Ackerman Turn (radius of turn and rover speed) or Point Turn parameters (rover turn rate). Commanding is expected every 1cm travelled, i.e. for maximal speed of 10cm/s, this means commanding at 10Hz.
F	2	1	V0.1	Planetary – Rover – Robotic arm joint control interface	OG6 will provide an API that allows OG2 to control the manipulator position.
F	3	1	V0.1	Orbital – Chaser – Attitude and orbit command interface	OG6 will provide an API that allows OG2 to request under demand AOCS commanding.
F	5	1	V0.1	Orbital – Chaser – Robotic arm joint control interface	OG6 will provide an API that allows OG2 to control the manipulator position.
F	1	0	V0.1	Orbital – Computer	OG6 provides, for the orbital scenario validation, a computer that hosts ERGO which commands a manipulator in an orbital facility that mimicks a servicing scenario for a modular client satellite.
F	2	0	V0.1	Orbital – ATMs	OG6 provides to OG2, for the orbital scenario validation, a number of modular elements on the client satellite that can be reconfigured, substituted and replaced by the orbital manipulator commanded by ERGO.
F	3	0	V0.1		OG6 provides to OG2, for the orbital scenario validation, the relative position between the manipulator end-effector and the modular elements on the client satellite. This will be done by visual means to be implemented by OG6.

Type	Major sequence	Minor sequence	Version	Title	Description
F	5	0	V0.1		OG6 provides, for the planetary scenario validation, a computer that hosts ERGO which commands a mobile robotic platform (rover) that mimicks a planetary exploration scenario.
F	4	0	V0.1		OG6 provides, for the planetary scenario validation, a rover that is capable to mimick the behavior of an EXOMARS type rover.
M	1	0	V0.1		OG6 provides, for the orbital scenario validation, means to facilitate the grappling between the manipulator and the modular elements commanded by ERGO.
D	1	0	V0.1		OG6 provides, for the orbital scenario validation, the realised trajectory computed by the robots telemetry.
D	2	0	V0.1		OG6 provides to OG2, for the orbital scenario, a 3D model of the mockup including the position of all the elements under testing.
D	3	0	V0.1		OG6 provides, for the planetary scenario validation, data on the rover position and pose to be used by ERGO as the basis of path planning.
D	4	0	V0.1		OG6 provides, for the planetary scenario validation, detailed DTMs of the earth-analogue planetary surface.
P	1	0	V0.1		OG6 provides, for the planetary scenario validation, a rover with sufficient power to complete the proposed exploration task.

END OF DOCUMENT

