INSPIRe

Spec3.1 - Functional and Software Design and Test Specifications – DFMC M(G)RAIM

Prepared for:

Prepared by:	GMV NSL INSPIRe TEAM
Approved by:	M Pattinson (GMV NSL)
Authorized by:	T Richardson (GMV NSL)
Code:	INSPIRe-GMVNSL-Spec3.1-v1.0
Version:	1.0
Date:	May- 2023

In accordance with the provisions of the above Contract 4000138525/22/NL/RR, Taylor Airey Limited hereby certifies both on its own behalf and on behalf of its consortium/Sub-Contractor(s), that no Intellectual Property Rights (as defined in Annex IV of the General Clauses and Conditions for ESA Contracts, ref. ESA/REG/002, Rev. 2, the "GCC") have been generated in the course of or resulting from work undertaken for the purpose of this Contract.

Change Record

Issue / Rev	Date	Change Record	Authors
v1.0	05/05/2023	The first version of the deliverable submitted ESA	GMV INSPIRe Team

Table of Contents

© GMV 2023
The copyright in this document is vested in GMV.

This document may only be reproduced in whole or in part, or stored in a retrieval system, or transmitted in any form, or by any means electronic, mechanical, photocopying or otherwise, either with the prior permission of GMV or in accordance with the terms of Contract No. 4000138525/22/NL/RR.

List of Tables and Figures

1 INTRODUCTION

1.1 Purpose

This document is deliverable Spec3.1 of the INSPIRe project, titled 'Functional and Software design and test specification DFMC M(G)RAIM. This deliverable document is an output for WP3 providing which includes the

- Functional specification and associated test specification for single frequency DFMC MRAIM and MGRAIM
- Software design specification for implementing the GPS MRAIM and MGRAIM algorithms
- Software test specifications for each module for integration and at the system level

1.2 Scope

Following the background information presented in Section 1, the layout of the remainder of the document is as follows:

- Section 2 contains a list of applicable and reference documents
- Section 3 provides an overview of the algorithm design
- Section 4 provides an overview of the functional and software requirements
- Section 5 provides a description of the test plan to be executed to validate the systems implementing the Architecture and the resulting outcomes

1.3 Definitions and Acronyms

1.3.1 Definitions

Concepts and terms used in this document and need defining are included in the following table:

Table 1-1 Definitions

Concept / Term	Definition
MG-RAIM	Maritime General-RAIM: integrity algorithm designed for the configuration of single. constellation, single frequency and without augmentation
MRAIM	MRAIM: integrity algorithm designed for a configuration of potential multi-constellation, multiple frequencies, and with/without augmentation

1.3.2 Acronyms

Acronyms used in this document and need defining are included in the following table:

Table 1-2 Acronyms

Acronym	Definition
AL	Alert Limits
ARAIM	Advanced Receiver Autonomous Integrity Monitoring
CDF	Cumulative distribution function
DFMC	Dual Frequency Multi constellation
DGNSS	Differential GNSS
DGPS	Differential GPS
DOP	Dilution of Precision
ECAC	European Civil Aviation Conference
EGNOS	European Geostationary Navigation Overlay Service

ESA	European Space Agency
FD	Fault Detection
FDE	Fault Detection and Exclusion
GBAS	Ground-Based Augmentation System
GEAS	GNSS Evolutionary Architecture Study
GLONASS	GLObal NAvigation Satellite System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GRAD	GLA Research and Development
GSA	European GNSS Agency
HAL	Horizontal alarm Limit
HDOP	Horizontal Dilution of Precision
HMI	Hazardous Misleading Information
HPE	Horizontal Position Error
HPL	Horizontal Protection Level
IALA	International Association of Marine Aids to Navigation and Lighthouse Authorities
ICAO	International Civil Aviation Organisation
IEC	International Electrotechnical Commission
INSPIRe	Integrated Navigation System-of-Systems PNT Integrity for Resilience
IMO	International Maritime Organisation
IR	Integrity Risk
ISM	Integrity Support Message
LPV	Localizer Performance with Vertical guidance
MHSS	Multiple Hypothesis Solution Separation
MOPS	Minimum Operational Performance Standards
MGRAIM	Maritime General RAIM
MRAIM	Maritime RAIM
MSC	Maritime Safety Committee
MSR	Multi-system shipborne receiver
N/A	Not Applicable
NLOS	Non-Line of sight
NPA	Non-Precision Approach
PFA	Probability of False Alarm
PL	Protection Level
PHMI	Probability of Hazardously Misleading Information
PMD	Probability of Miss detection
PNT	Positioning Navigation and Timing
PVT	Position, Velocity and Time
RAIM	Receiver Autonomous Integrity Monitoring
RTCA	Radio Technical Commission for Aeronautics
RTK	Real-time kinematic positioning
SARPS	Standards and Recommended Practices
SBAS	Satellite Based Augmentation System
SIS	Signal in Space
SOLAS	Safety at Life at Sea
TBC	To Be Confirmed
TTA	Time to Arrival
UL	Uncertainty Level
VAL	Vertical alarm Limit
VHF	Very High Frequency

2 REFERENCE

2.1 Applicable Documents

The following documents, of the exact issue shown, form part of this document to the extent specified herein. Applicable documents are those referenced in the Contract or approved by the Approval Authority. They are referenced in this document in the form [AD.x]:

Table 2-1 Applicable Documents

Ref.	Title	Code	Version	Date
1.	INSPIRe Technical Proposal, Taylor Airey	T-062-001-02 Part 1	-	June 2022
2.	INSPIRe Management Proposal, Taylor Airey	T-062-001-02 Part 2	-	June 2022
3.	INSPIRe Proposal GMV	GMV 10842/21	v2/21	

2.2 Reference Documents

Although not part of this document, the following documents amplify or clarify its contents. Reference documents are those not applicable and referenced within this document. They are referenced in this document in the form [RD.x]:

Table 2-2 Reference Documents

Ref.	Title	Code	Version	Date
[RD.1]	INSPIRe Alg3.1 - Algorithm documentation (GPS M(G)RAIM)	INSPIRe-GMVNSL-Alg3.1-v1.0	1.0	05/05/2023
[RD.2]	IEC, ‘Maritime navigation equipment - GNSS, part 1: GPS’	*IEC 61108-1*	-	2003

3 ALGORITHM DESIGN OVERVIEW

The algorithm design functional and software architecture is described in [RD.1]. A high-level description is presented here for completeness for both M(G) RAIM and MRAIM.

3.1 MG-RAIM General Description

The maritime integrity algorithm proposed for M(G) RAIM is based on a Classical RAIM algorithm, used for decades in maritime receivers, that is applied to the overall solution and to measurement subsets to ensure the fault detection (and exclusion if needed) capabilities.

The algorithm is based on two sequential steps once the position is calculated with all the available satellites.

- Availability check: A set of simple checks are applied to determine whether the derived all-in-view solution is suitable for navigation as defined by the maritime receiver specifications [RD.2]. If any of these tests fail, then an integrity alarm is raised.
- Fault detection: If all the previous test meets the defined conditions, a Chi-Squared test is performed to identify a fault in the positioning estimation.
- Geometry screening: For each potentially faulty element of the navigation solution, a subset navigation solution is formed by eliminating the faulty element from the all-in-view solution. Then, availability checks are performed to check if the remaining solution would pass, to determine if the detection capability of the solution is sufficient.

Figure 3-1 and Figure 3-2 provides a high-level view of the rationale, inputs outputs and relationships of each of the functions detailed.

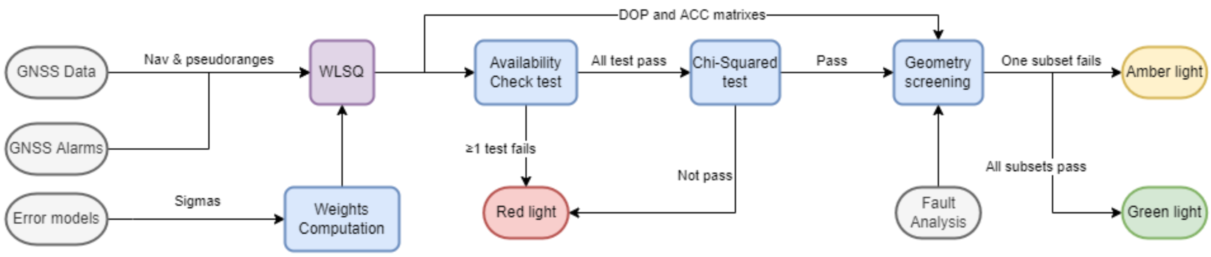

Figure 3-1. MG-RAIM conceptual flowchart without augmentation

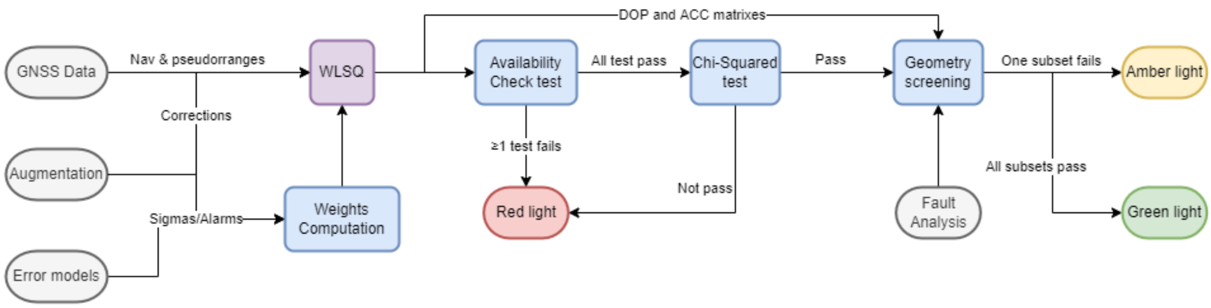

Figure 3-2. MG-RAIM conceptual flowchart with augmentation

Where:

- Grey bubbles are the required inputs
- Purple box is the PVT engine which computes the user estimated state vector
- Blue boxes are the functions involved in the integrity algorithm
  - Weights computation: computes satellite error models considered as nominal for the maritime environment.

- Availability Check Test: the function that checks if the current solution is suitable for maritime in terms of expected accuracy.
- Fault Detection, Chi-Squared test: function that detects if solution complies with the nominal error models.
- Geometry screening: function that identifies the potential failure and performs additional tests to measurements subsets.

- Red, amber and green bubbles are output information provided to mariners

NB Fault Detection and Exclusion (FDE) may be used by some future refinement of this process and may be a user-set option, therefore the scheme will include an iteration loop after a red light is raised, excluding the identified fault from the solution, and starting the WLSQ module again without that measurement.

3.2 MRAIM General Description

The maritime integrity algorithm proposed, is a maritime-specific implementation of the aviation ARAIM concept and performs a multiple-hypothesis solution-separation process, then computes a protection level and iteratively optimises this PL through re-allocation of integrity risk. Solution separation methods are characterised by the use of the distance (separation) between position estimates obtained with subsets of the available satellites.

ARAIM is an evolution of the RAIM techniques and has been developed considering DFMC environment with the goal to protect multi-constellation users by means of a robust user integrity algorithm. Being Civil Aviation applications the driver for the development of ARAIM concept, ARAIM intends to provide a service for stringent aviation operations: LPV-200.

The algorithm is based on the following main principles.

- Ground monitoring system: Different ARAIM concepts might have different levels of ground monitoring, implying different levels of fault detection by the ground segment. This would have a significant impact on the user ARAIM architectures in terms of their performance, and on the design of the algorithm itself.
  - **Horizontal ARAIM** (H-ARAIM) is a minor extension of today's RAIM architecture adding multi-constellation and dual-frequency capabilities. It is based on a static or quasi-static ISD to support horizontal navigation. ISD content is based upon Constellation Service Provider commitments to maintain certain level of performance and observational history.
  - **Offline ARAIM** to support horizontal and vertical navigation based on a monthly ISD from the ground to ensure that ISD parameters are consistent with up-to-date monitoring results. ISD parameters receive greater scrutiny than in the horizontal architecture due to the more stringent operation targets proposed for the offline architecture.
  - **Online ARAIM** to support horizontal and vertical navigation based on an hourly ISD from the ground. In this way, Service Providers are given a larger control over GNSS performance.
- ARAIM assumptions and Feared Events: The ARAIM user algorithm needs to make certain assumptions about errors and threats and requires certain information to be provided by a specific ground segment to generate protection levels and provide integrity. In particular, the user ARAIM algorithm requires values for failure probabilities, the standard deviation of a distribution that bounds the orbit/clock error in the fault-free case and the nominal and maximum biases in fault-free conditions.
- User algorithm:

- Fault Detection: Each failure mode defines a reduced-subset solution, which excludes the potentially faulty measurements. The physical separation distance between the all-in-view (AIV) solution and the subset solution is used as a fault detection test. Only if all separation tests are passed (i.e., all lie below the detection threshold) is the AIV solution accepted. As a sanity check, a conventional chi-square test of the residuals is also performed for the AIV solution.
- Protection Level calculation: The horizontal protection level is computed that considers all monitored failure modes, nominal errors, and nominal biases.

As aforementioned, MRAIM proposed in this document is an adaptation of the aviation ARAIM to capture the particularities of the maritime domain. The processing and the architecture remain the same, but adaptation is required in the ISD parameters and in the error models used to represent the maritime environment. In addition, this project will consider:

- **Offline architecture** for **non-augmented scenarios**, since the ISD parameters shall be conservative and validated enough for any operation.
- **Online architecture** for **augmented scenarios**. Augmentation messages will be used to fine tune the measurement error models and the ISD parameters related to the probability of satellite and constellation failure.

Figure 3-3 and Figure 3-4 provide a high-level view of the rationale, inputs outputs and relationships of each of the functions detailed.

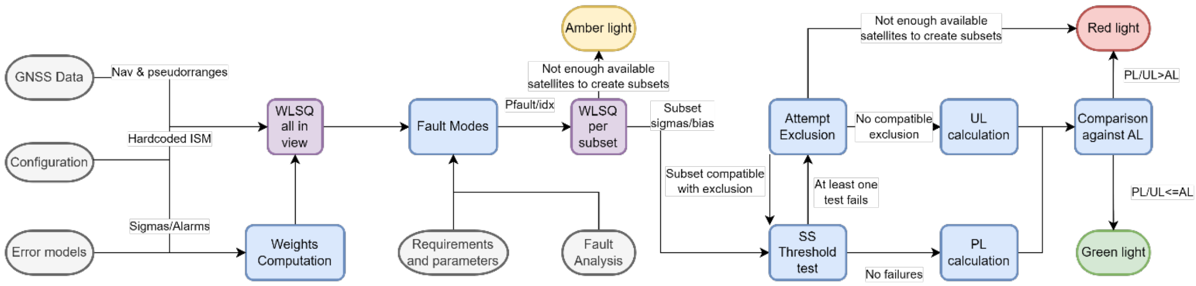

Figure 3-3. M-RAIM offline conceptual flowchart

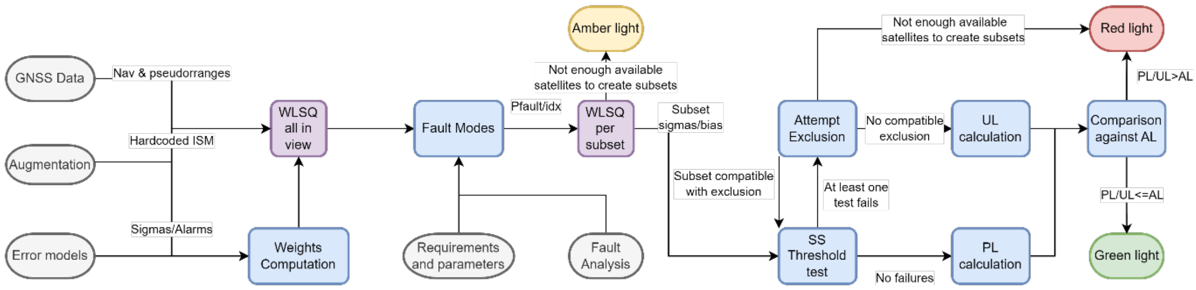

Figure 3-4. M-RAIM online conceptual flowchart

Where:

- Grey bubbles are the required inputs.
  - Particularly relevant are the ISD parameters required for the suitable tuning of the MRAIM algorithm
- Purple box is the PVT engine which computes the user estimated state vector.
- Blue boxes are the functions involved in the integrity algorithm

- Weights computation: computes satellite error models considered as nominal for the maritime environment.
- Faults Mode calculation: the function that computes the required subsets to provide the desired level of integrity. According to the fault analysis, this function classifies the error that needs to be monitored to be included in the calculation.
- Solution Separation Threshold test: the function that performs a threshold test for each subset and analyses if their separation is compatible with a failure. In that case, the faulty could be excluded to provide a safe positioning.
- Attempt Exclusion: Fault exclusion could be performed if one of the Solution Separation Threshold tests fails. This function computes the compatible potential subsets and provides the compatible subset again to the Solution Separation Threshold test module as the subset is the new “all in view”.
- Protection/Uncertainty Level calculation: the function that computes the Protection/Uncertainty Level according to the previously computed information.

■ Red, amber and green bubbles are output information provided to mariners

4 SPECIFICATION OF REQUIREMENTS

This section presents functional, non-functional, and safety requirements for the main elements of the PVT Algorithm architecture. The algorithm design functional and software architecture is described in [RD.1].

4.1 Requirement nomenclature

For each requirement the following characteristics are specified:

- ID — A unique identifier: it is composed of a string and a numeric part.
- Description — The content of the requirement.
- Trace-back — Points to source material from which the requirement has been derived.

Note: All specified requirements are to be considered mandatory unless otherwise stated.

The unique requirement ID complies with the following format:

[Category].[Group].[###]

Where:

- [Category] can be
  - FUN - functional requirements.
  - INT - interface requirements
- [Group] can be
  - ALG - requirements related to the Algorithm
  - SW - requirements related to the Software

- [###] is a numeric incremental identifier for requirements which have the same [Category]. [Group] prefix.

The requirement characteristics are presented in paragraph format, as demonstrated in Table 4-1.

ID	[Category].[Group].[###]
Description	A concise description of the requirement

Table 4-1: Requirement specification format

4.2 Algorithm Functional Specification

4.2.1 General Functional Requirements

FUN. ALG.001

The algorithm shall process the GNSS input data, as specified in the table below, prior to position and integrity calculations.

Input data	Guidance
GNSS constellation ID	GPS, GALILEO
GNSS frequency ID	L1/L5, E1/E5a
Code Observation	Measurement
Carrier phase Observation	Measurement
Navigation Data	Decoded messages

FUN.ALG.002

The algorithm shall Generate a snapshot weighted least squares 3D position solution using code measurements:

Req ID	Guidance
FUN.ALG. 002.01	Be able to do standalone solution for GPS L1/L5 and GALILEO E1/E5a individually and together (GPS+Galileo)
FUN.ALG. 002.02	Be able to do an SBAS corrected GPS L1/L5 and GALILEO E1/E5a solution individually and and together (GPS+Galileo)

FUN.ALG.003

The algorithm shall reject satellites that are flagged as ‘do not use’, while in SBAS Mode.

FUN.ALG.004

The algorithm shall use appropriate values from SBAS messages to weight orbit and clock to error budgets, while in SBAS Mode

FUN.ALG.005

The algorithm shall use DFREI values from SBAS messages to Ionosphere contribution to error budgets, while in SBAS Mode.

FUN.ALG.006

The algorithm shall use the GPS navigation message broadcast URA value to weight orbit and clock contribution to error budgets, while in GNSS Mode.

FUN.ALG.007

The algorithm shall use the GALILEO navigation message broadcast SISA value to weight orbit and clock contribution to error budgets, while in GNSS Mode.

FUN.ALG.008

The algorithm shall use Configurable values for other contributions to the error budget.

FUN.ALG.009

The algorithm shall use set weights using error budget values.

FUN.ALG.010

The algorithm shall allow the usage selection of M(G)RAIM or MRAIM by configuration.

INT. ALG.011

The algorithm shall provide the following output data:

Req. ID	Output data	Guidance
INT. ALG. 011.01	3D position	WGS84 coordinates
INT. ALG. 011.02	GNSS timestamp	
INT. ALG.011.03	Indicator Flags	

FUN. ALG.012

The algorithm shall set the output integrity flag to red, if red light integrity indicator was raised at any point in the previous applicable period (configurable).

FUN. ALG.013

The algorithm shall set the current output integrity flag to the integrity indicator for the current epoch if red light integrity indicator was not raised at any point in the previous applicable period (configurable).

4.2.2 M(G)RAIM Functional Requirements

FUN.ALG.014

The algorithm shall perform an availability check of the solution to check HDOP, GDOP, 95% horizontal accuracy and the number of satellites against thresholds.

FUN. ALG.015

The algorithm shall raise a RED flag integrity indicator for that epoch if the availability check fails.

FUN. ALG.016

The algorithm shall perform fault detection for the solution using measurement residuals and check against the threshold.

FUN. ALG.017

The algorithm when in using M(G)RAIM mode shall perform geometry screening on selected subsets of satellites to check HDOP, GDOP, 95% horizontal accuracy and number of satellites against thresholds.

FUN. ALG.018

The algorithm shall raise traffic light indicators based on the following:

	Traffic light Flag Indicator	Guidance
FUN.ALG. 018.01	RED	An indicator flag is raised for that epoch if fault is detected And also if the solution cannot meet navigation requirements (DOP or horizontal accuracy > threshold)
FUN.ALG. 018.02	AMBER	An indicator flag is raised if geometry screening fails for any subset
FUN.ALG. 018.03	GREEN	An indicator flag is raised if availability check, fault detection and geometry screening do not raise any issues.

4.2.3 MRAIM Functional Requirements

FUN.ALG.019

The algorithm shall use Configurable values for the ISM parameters.

FUN.ALG.020

The algorithm shall monitor at least single failure modes and perform the Solution Separation Threshold test as discussed in [RD.1].

FUN.ALG.021

The algorithm shall perform the exclusion algorithm, when configured, as discussed in [RD.1].

FUN. ALG.022

For the 3D position solutions, the algorithm when in using MRAIM mode shall calculate Horizontal Protection Level (HPL) based on the MRAIM concept discussed in [RD.1] when no failure is detected in the final satellite subset.

FUN. ALG.023

The algorithm shall calculate Uncertainty Level (UL) based on the MRAIM concept discussed in [RD.1] or raise a red flag when failure is detected and is not possible to exclude it.

FUN. ALG.024

For the 3D position solutions, the algorithm shall calculate Horizontal Protection Level (HPL) or Uncertainty Level (UL) based on the SBAS integrity concept adapted.

FUN. ALG.025

The algorithm shall raise traffic light indicators based on the following:

	Traffic light Flag Indicator	Guidance
FUN.ALG. 025.01	RED	An indicator flag is raised for that epoch if fault is detected and not excluded or the PL/UL>AL

FUN.ALG. 025.02	AMBER	An indicator flag is raised if fault modes cannot be monitored from the all-in-view solution
FUN.ALG. 025.03	GREEN	An indicator flag is raised if fault detection does not raise any issues and PL≤AL.

4.3 Software Functional Specification

FUN. SW.001

The software shall Implement algorithm with functional requirements listed above.

FUN. SW.002

The software shall run in post-processing and not a Real Time data stream.

FUN. SW.003

The software shall read the configuration settings from the configuration file.

FUN. SW.004

The software shall read the RINEX Observation file.

FUN. SW.005

The software shall read the RINEX Navigation file.

FUN. SW.006

The software shall read the SBAS message.

FUN. SW.007

The software shall apply carrier smoothing of the code for single frequency measurements (the smoothing constant shall be configurable).

FUN. SW.008

The software shall read the smoothing constant from the configuration parameters.

FUN. SW.009

The software shall read the elevation mask from the configuration parameters, removing observation for the satellite below the mask.

FUN. SW.010

The software shall apply RTCA tropospheric correction model.

FUN. SW.011

The software shall form ionosphere-free pseudo-range measurement, in DFMC mode

FUN. SW.012

The software shall make the SBAS mode configurable to be enabled or disabled.

FUN. SW.013

The software when operated in SBAS mode shall:

Req ID	Guidance
FUN. SW.013.01	Remove satellites that have 'do not use' condition in DFREI

FUN. SW.014

The software shall make the following configurable:

Req ID	Guidance
FUN. SW.014.01	Thresholds for the integrity level required configurable
FUN. SW.014.02	False alarm rates the integrity level required configurable

FUN. SW.015

The software shall have allow a configurable time period to keep the RED flag.

FUN.SW.016

The software shall be capable of computing GNSS DFMC only and SBAS + DFMC solutions.

INT.SW.017

The software shall process one epoch at a time of observations and save the results. The Position Results saved should include but not be limited to the following:

Position Results Output
▪ Time ▪ Estimated Position (XYZ) ▪ Estimated position (LLH) ▪ Integrity flag (Green, Amber, Red) ▪ HDOP ▪ GDOP ▪ Number of Satellites used ▪ Computed test statistics ▪ Horizontal Protection Levels / Uncertainty Levels

5 TEST PLAN

This section contains the test cases that have been created to test the operation of INSPIRe algorithm architecture described in the deliverable Alg3.1. Within describes which tests are to be executed and provide a high-level description of the purpose of the test along with required data and test tools.

All test cases shall be independent; no test case will rely on the output of another test case. This has the benefit that the tests can be executed in any order, except for the Installation test procedure which will be run first, and that the failure of a given test will not prevent the running of other tests.

5.1 Test Data

The data and configuration settings at all processing stages are stored in files. Sample data sets and configurations for testing will be created during the development of the algorithm. These files will be archived with documented filenames.

Test data consists of:

- GNSS RINEX version 3.04 format
  - Observation File
  - Navigation File

5.2 Test Environment

The test will be carried out standalone from the GMV network and will use a PC or laptop as the “diagnostics” machine running Windows OS

5.3 List of Test Cases

Each test case will be defined based on a Case ID with next format:

TC - Prefix to indicate this is a Test Case Identifier.

[Function] - is the name of the software component under testing

[ID#] – sequential numbering for each case.

Table 5-2 below identifies a set of test cases proposed for both the MGRAIM and MRAIM algorithms. To demonstrate that the algorithm is compliant with its system, it is necessary to provide a trace of the system requirements in the test activities. Each requirement will be traced to the test that verifies it. In some instances, requirements may need more than one test to demonstrate compliance (e.g., if the requirement has two or more ‘parts’ to it). In such cases, a comment will be included in the traceability column to identify the aspects of the requirement that each test is responsible for demonstrating.

Not all functional specifications will be traceable to a functional test execution; these may require inspection of hardware and/or technical documentation.

Table 5-1 Test Cases

Test CaseID	Objective/Test sets	Description	Success Criteria	Traceability Requirements
TC.SW.01	To check that files are read, and a solution is generated as expected with MGRAIM algorithm	Execution of the positioning software using a nominal configuration which includes: • Archived real data (GPS L1/L5 Galileo E1/E5 - RINEX Observation(*.obs) and Navigation (*.nav)), • Faults: = NONE, • Smoothing constant: 100s	• Population of the output parameter • Manually check the consistency of loaded data • a valid solution, green flag, horizontal position errors < 10m • Solution output can be compared with commercial/open sources PVT processing software to check that the position is as expected, (e.g., solution, number of satellites, DOP values, etc)	FUN. ALG.001 FUN. ALG.002.01 FUN. ALG.006 to FUN. ALG.013 FUN. ALG.014 to FUN. ALG.018 FUN. SW.001 FUN. SW.002 FUN. SW.003 FUN. SW.004 FUN. SW.005 FUN. SW.010 FUN. SW.011 FUN.SW.016 FUN.SW.017
TC.SW.01.1	To check that files are read, and a solution is generated as expected with MRAIM algorithm	Execution of the positioning software using a nominal configuration which includes: • Archived real data (GPS L1/L5 Galileo E1/E5 - RINEX Observation(*.obs) and Navigation (*.nav)), • Faults: = NONE, Smoothing constant: 100s	• Population of the output parameter • Manually check the consistency of loaded data • a valid solution, green flag, horizontal position errors < 10m • HPL < AL (25m) • Solution output can be compared with commercial/open sources PVT processing software to check that the position is as expected, (e.g., solution, number of satellites, DOP values, etc)	FUN. ALG.002.01 FUN. ALG.006 to FUN. ALG.013 FUN. ALG.019 to FUN. ALG.025 FUN. SW.001 FUN. SW.002 FUN. SW.003 FUN. SW.004 FUN. SW.005 FUN. SW.010 FUN. SW.011 FUN.SW.016 FUN.SW.017
TC.SW.02	To demonstrate that the availability check is working and HDOP threshold is configurable to the MGRAIM algorithm.	Same as TC.SW.01 with the HDOP threshold configuration set to very small value	The output integrity flag is red	FUN.ALG.014 FUN. ALG.015 FUN. SW.014.01
TC.SW.03	To demonstrate that the availability check is working and GDOP threshold is configurable to the MGRAIM algorithm	Same as TC.SW.01 with the GDOP threshold configuration set to very small value	The output integrity flag is red	FUN.ALG.014 FUN.ALG. 018.01 FUN. SW.014.01
TC.SW.04	To demonstrate that the availability check is working, and the Horizontal accuracy threshold is configurable to the MGRAIM algorithm	Same as TC.SW.01 with the Horizontal accuracy threshold configuration set to a very small value	The output integrity flag is red	FUN.ALG.014 FUN.ALG. 018.01 FUN. SW.014.01
TC.SW.05	To demonstrate that the availability check is working, and MGRAIM and elevation mask is configurable. RED FLAG raised if number satellites in used less than 6 for DFMC case	Same as TC.SW.01, but increase configured elevation mask so that have a minimum of 6 satellites in view	• The number of satellites used is lower because low-elevation satellites are removed • The output integrity flag is red when 6 satellites are in view	FUN.ALG.014 FUN.ALG. 018.01 FUN. SW.009

TC.SW.05.1	To demonstrate that the availability check is working, and elevation mask is configurable to MRAIM algorithm	Same as TC.SW.05 to check the number of the satellites in view affect the integrity check	• The number of satellites used is lower because low-elevation satellites are removed • The output integrity flag is red/amber when 6 satellites are in view	• FUN.ALG. 025.01 • FUN.ALG. 025.02 • FUN. SW.009
TC.SW.06	To demonstrate the geometry screening check is working and HDOP threshold is configurable	Same as TC.SW.01 but set HDOP threshold to only just larger than the maximum all-in-view HDOP from any epoch in the test data	The output on the epoch that has the highest all-in-view, HDOP should see an amber flag	FUN. ALG.016 FUN.ALG. 018.02 FUN. SW.014.01
TC.SW.07	To demonstrate the geometry screening check is working and GDOP threshold is configurable to the MGRAIM algorithm	Same as TC.SW.01 but set GDOP threshold to only just larger than the maximum all-in-view GDOP from any epoch in the test data	The output on the epoch that has the highest all-in-view GDOP should see an amber flag	FUN. ALG.016 FUN.ALG. 018.02 FUN. SW.014.01
TC.SW.08	To demonstrate the geometry screening check is working and horizontal accuracy threshold is configurable to the MGRAIM algorithm	Same as TC.SW.01 but set the horizontal accuracy threshold to only just larger than the maximum all-in-view horizontal accuracy from any epoch in the test data	The output on the epoch that has the highest all-in-view Horizontal accuracy should see an amber flag	FUN. ALG.016 FUN.ALG. 018.02
TC.SW.09	To check that the fault detection function in the MGRAIM algorithm can detect the fault and that the red integrity flag is raised	Same as TC.SW.01 but inject a large bias fault (100m) on one satellite for one second, and again 10 seconds later. Assuming that the time configuration for the red flag is 6 seconds	• Position errors same as in TC.SW.01 except on the two affected epochs when they will be much larger. • The integrity flag is green on all epochs except on the epochs where the fault is added and for 6 seconds after each fault when it is red	FUN. ALG.012 FUN. ALG.013 FUN. ALG.015 FUN.ALG. 018.01
TC.SW.09.1	To check that the MRAIM fault detection and exclusion function can detect and exclude the fault. All Green flag is expected	Same as TC.SW.09	• Position errors at faults injected epoch shall significantly reduce • Number of satellites in used shall be reduced 1 if the faulty satellite is removed • The Integrity flag is green on all epoch	FUN. ALG.012 FUN. ALG.013 FUN. ALG.021 FUN. ALG.022 FUN.ALG. 025.03
TC.SW.09.2	To check that the fault detection and exclusion function can detect and exclude the dual GPS faults. Whole GPS satellites are removed, the number of satellites in used reduce to 9. Green flag raised as fault free results are estimated.	Same as TC.SW.09	• Position errors at faults injected epoch shall significantly reduce • All satellites from the constellation shall be removed. • The Integrity flag is green on all epoch	FUN. ALG.012 FUN. ALG.013 FUN. ALG.021 FUN. ALG.022 FUN.ALG. 025.03

TC.SW.09.3	To check that the fault detection and exclusion function can detect the dual faults from different constellation and RED flag shall raise at epoch 6th and 17th.	Same as TC.SW.09	• Position errors at faults injected epoch shall significantly reduce • The Integrity flag is RED on the fault epoch if the protection level or uncertainty level is higher than the predefined AL	FUN. ALG.012 FUN. ALG.013 FUN. ALG.021 FUN. ALG.022 FUN.ALG. 025.01
TC.SW.09.4	To check that the greater AL value allow all the epoch to have a Green Flag under	Same as TC.SW.09	• All positioning results and HPL and UL shall be identical to TC.SW.09.3	FUN. ALG.012 FUN. ALG.013 FUN. ALG.021 FUN. ALG.022 FUN.ALG. 025.01
TC.SW.09.5	To check that the fault detection and exclusion function in the MRAIM algorithm can detect and remove the dual fault from the different constellations and that the GREEN integrity flag is raised with a lower probability of Threshold value	Same as TC.SW.09	• Position errors at faults injected epoch shall significantly reduce • Number of satellites in use shall be reduced to 2 if the faulty satellite is removed • The Integrity flag is green on all epoch	FUN. ALG.012 FUN. ALG.013 FUN. ALG.021 FUN. ALG.022 FUN.ALG. 025.03
TC.SW.10	To check that the URA and SISA is being considered in weighting in the MGRAIM algorithm	Same as TC.SW.01 but change URA and SISA values for all navigation messages to be bigger	• The position solution should be identical to test TC.SW.01 • The test statistic on each epoch should be smaller and the horizontal accuracy value bigger	FUN.ALG.006 to FUN.ALG.009
TC.SW.11	To check that the error budgets are configurable and are used in the MGRAIM algorithm	Same as TC.SW.01 but increase error budget values at all elevations	• The position solution should be identical to test TC.SW.01 • The test statistic on each epoch should be smaller and the horizontal accuracy value bigger	FUN.ALG.006 to FUN.ALG.009
TC.SW.12	To demonstrate that the Software can produce a DFMC SBAS solution	Same as TC.SW.01 but now generating an DFMC SBAS solution	• The output should produce a valid solution, green flag errors < 10m • The test statistic and accuracy value should be different to the GPS case (even for the same satellites) • Solution output can be compared with commercial/open sources PVT processing software to check that the position is as expected • Manually check the consistency of loaded EGNOS data	FUN.ALG. 020.02 FUN.ALG.003 to FUN.ALG.005 FUN.ALG.005 FUN.ALG.008 FUN. SW.006 FUN. SW.012 FUN. SW.013.01 FUN.SW.016

TC.SW.13	To demonstrates that satellites marked “do no use” by EGNOS are removed	Same as TC.SW.13 but modify one of the SBAS messages for one satellite identified as 'do not use'	When the satellite is set 'do not use': • Manually check the consistency of Satellite information i.e. – the number of satellites is reduced by 1 • The test statistic and accuracy value should be different to the	FUN. SW.013.01

End of Document