



HAL
open science

Deliverable 3.2 - GATE4RAIL - Models for Fail-Safe positioning components w.r.t. Faults

Juliette Marais, Syed Ali Kazim, Giuseppe Rotondo, Cosimo Stallo, Alessia Vennarini, Andrea Coluccia, Olivier Desenfans

► **To cite this version:**

Juliette Marais, Syed Ali Kazim, Giuseppe Rotondo, Cosimo Stallo, Alessia Vennarini, et al.. Deliverable 3.2 - GATE4RAIL - Models for Fail-Safe positioning components w.r.t. Faults. [Research Report] IFSTTAR - Institut Français des Sciences et Technologies des Transports, de l'Aménagement et des Réseaux. 2020, 42p. hal-03159118

HAL Id: hal-03159118

<https://hal.science/hal-03159118v1>

Submitted on 4 Mar 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



GNSS Automated Virtualized Test Environment for RAIL (GATE4Rail)

Deliverable 3.2 – Models for Fail-Safe positioning components w.r.t. Faults

Document manager:	Juliette MARAIS		UGE
-------------------	-----------------	--	-----

Programme:	S2R-OC-IP2-02-2018- Research and Innovation Action		
Project Name:	GNSS Automated Virtualized Test Environment for RAIL		
Project Acronym:	GATE4RAIL		
Contract Number:	826324		
Project Coordinator:	RADIOLABS		
WP leader:	UGE/Ifsttar		

Document Id. N°:	GATE4RAIL-003-002-000	Revision	01
Deliverable:	D3.2	Date:	01/07/2020

Document classification	Public
-------------------------	--------

Approval Status	
Prepared by:	Juliette Marais, Syed Ali Kazim, Cosimo Stallo, Giuseppe Rotondo; Alessia Vennarini, Andrea Coluccia
Approved by (WP Leader):	Juliette Marais (UGE)
Approved by (Coordinator):	Alessandro NERI (RDL)

CONTRIBUTING PARTNERS

Name	Company/Organization	Role/Title
Juliette Marais	UGE/Ifsttar	WP Leader
Syed Ali Kazim	UGE/Ifsttar	Contributor
Giuseppe Rotondo	GUIDE	Contributor
Cosimo Stallo	RDL	Contributor
Alessia Vennarini	RDL	Contributor
Andrea Coluccia	RDL	Contributor
Olivier Desenfans	M3S	Contributor



DISTRIBUTION LIST

Name	Company/Organization	Role/Title
Alessandro NERI	RDL	Project Coordinator
Lea PATIES	Shif2Rail	Shif2Rail Programme Officer

This project has received funding from the Shift2Rail Joint Undertaking (JU) under grant agreement No 826324. The JU receives support from the European Union's Horizon 2020 research and innovation programme and the Shift2Rail JU members other than the Union. Any dissemination of results reflects only the author's view and JU is not responsible for any use that may be made of the information it contains.

REVISION TABLE

Version	Date	Modified pages	Modified Sections	Comments
00	01/07/2020	ALL	ALL	First Delivery
01	09/11/2020	See modifications at Paragraph 1.6		Second delivery after S2R comments implementation

Executive summary

This deliverable is the second report delivered by GATE4Rail WP3. The general objective of WP3 was to define a comprehensive methodology and the associated tools able to characterise the Global Navigation Satellite System (GNSS) performance into the railways application scenarios defined in WP2 in the framework of a VB detection in ERTMS (European Railway Traffic Management System)/ETCS (European Train Control System) context.

The developed methodology must provide the capability to evaluate the effects of global and local GNSS faults on the properties of some Fail-Safe Train Positioning components. Indeed, GNSS based navigation solutions are affected by global as well as local hazards.

By definition, global hazards have impact at global system level. As a consequence, they could affect a large number of users. Global hazards cover effects such as satellite clock errors or ephemeris failure.

On the contrary, local hazards are due to the impact of the local environment on the GNSS receiver. They are caused by infrastructure or vegetation in the surrounding of the receiver that creates blockages or multipath. Intentional and non-intentional interferences are also considered as local hazard. Local hazard mainly result in degradation of the accuracy, the availability and/or the integrity of the GNSS PVT (Position, Velocity, Time).

The project relies on the state of the art and experience from the past and ongoing projects related to GNSS application in the rail domain (in particular GSA H2020 ERSAT GGC, S2R VITE, GSA H2020 STARS and GSA H2020 RHINOS). In these projects, some experimentation have been performed in order to evaluate availability or accuracy of GNSS positioning in railway environments. In particular, STARS performed extensive measurement campaigns.

Based on this state of the art, WP2 defined the primary GNSS-ETCS operational scenarios including Hazardous and System Fault situations, the use cases in the railway environment (D2.1 document [ReD-17]) as well as the test cases for the performance evaluation (D2.2 document [ReD-16]). Test cases were defined for ERTMS/ETCS, covering Virtual Balises cases of study in the ERTMS field, and for GNSS, covering GNSS phenomenon cases in the railway field. In addition, the main railway environments that affect and/or impact the GNSS overall performance within a typical operational scenario were identified.

The work performed in WP3 is focused on the definition of the main railway environments that will impact GNSS reception, thus PVT performance. This deliverable D3.2 in particular, defines which are the key hazards (both local and global) that have to be taken into account and simulated in order to cover the main events that can occur during a railway operational scenario and how error models will be introduced in the simulation framework defined by the project.

In absence of existing error models for local effects and in particular effects caused by masking or reflecting obstacles, two models have been developed and implemented by the partners that aim to simulate them. This report details the process used for model generation. Two families of local errors have been selected and models: common errors observed in typical environments (forest, urban area, open sky...) and errors generated by local obstacles such as tunnels or bridges, causing partial or total loss of GNSS signals. These two models are complementary as they will apply at different points of the trajectory. Errors will be added along a reference track in order to generate local errors all along the track used as input for the chain.

The error models presented are part of Module #3 described in D4.1 document ([ReD-2]) of WP4 concerning GNSS Signal Generation. This module is responsible for generating GNSS raw data both in nominal and fault conditions taking into account different environments crossed by train in the considered railway track.

D3.1 ([ReD-1]) and 3.2 both describe the work performed in WP3. The main results of WP3 are the definition of the structure of a standardised database detailed in D3.1 and the development of the tools for GNSS characterization in different rail operational scenarios and GNSS environments described in this report.

Contents

Executive summary.....	4
1. Scope	8
1.1. Context	8
1.2. Reference Documents	8
1.3. Related Documents	8
1.4. Acronyms and Abbreviations.....	9
1.5. Disclaimer excluding JU responsibility.....	11
1.6. Description of Changes from the Previous Revision	11
2. Introduction.....	12
3. State of the Art of GNSS feared events and their occurrence or error model.....	12
3.1. Multipath and NLOS effects	14
3.2. Interferences	15
4. Simulation framework and tools definition.....	16
4.1. Architecture description.....	16
4.2. Introduction of GNSS errors and their models into the GATE4Rail simulation chain	18
5. Errors, faulty events and vulnerabilities modelling for Fail-Safe Positioning Components	20
5.1. Error model representation and methodology	20
5.1.1. Taking into account the estimator	20
5.1.2. Position error computation	20
5.1.3. Different approaches investigated	21
5.1.3.1. Typical Environment	21
5.1.3.2. Local Obstacles approach	22
5.2. Use of the error models for the estimation of positioning errors on tracks	23
5.3. Error models: Results and analysis with real data.....	25
5.3.1. For typical environments.....	25
5.3.1.1. Computation of error distributions	25
5.3.1.2. Model Validation	27
5.3.2. Specific obstacles (GUIDE).....	30
5.3.2.1. Regional Line Bridge	30
5.3.2.2. Single Track Tunnel.....	35
5.3.2.3. Model Validation	39
6. Conclusions and perspectives.....	41

INDEX OF FIGURES

Figure 1: GNSS errors classification	13
Figure 2: GATE4Rail architecture	17
Figure 3: How GNSS error on track reference will be used by module#3.....	18
Figure 4: Module#3 input trajectory computation based on Local error models and ground truth.	19
Figure 5: Module#3 input trajectory based on Local error models and ground truth.....	19
Figure 6: An illustration to differentiate LS and KF estimator.....	20
Figure 7: An example of Track error classification without a priori knowledge of the environment of the track.	21
Figure 8: Schematic process of first approach.	21
Figure 9: An example of a priori classification of the environment before error computation.....	22
Figure 10: Schematic process of second approach.	22
Figure 11: Single PVT track (cyan/blue) after a bridge crossing.....	23
Figure 12: Multiple PVT tracks (cyan/blue) after bridge crossing	23
Figure 13: Use of error models to simulate errors on track reference.	24
Figure 14: Environment identification with Google maps.....	25
Figure 15: Error classified in typical environment.....	26
Figure 16: Histogram and error distribution.	26
Figure 17: Comparison between simulated and real trajectories in Forest/Foliage.....	28
Figure 18: Comparison between simulated and real trajectories in Urban Environment	29
Figure 19: Comparison between simulated and real trajectories in Open sky environment	30
Figure 20: Railway bridge present on regional lines	31
Figure 21: Bridge errors profile for along and cross track.....	31
Figure 22: bridge error peak model.....	32
Figure 23: bridge error peak model with 2 seconds under bridge example	32
Figure 24: Bridge error model for positive/negative errors.....	33
Figure 25: Bridge error model for all possible cases	34
Figure 26: Comparison between real errors and modelled ones on cross-track	34
Figure 27: Vertical error analysis for bridge crossing.	35
Figure 28: Railway tunnel present on regional lines	35
Figure 29: Longitude behaviour in presence of a tunnel.....	36
Figure 30: Latitude behaviour in presence of a tunnel	36
Figure 31: Latitude errors for tunnel pass.....	37
Figure 32: Simulated cross and along track errors in presence of a tunnel	39
Figure 33: Comparison between simulated and real trajectories for a bridge crossing	40
Figure 34: Comparison between simulated and real trajectories for a tunnel crossing.....	41

INDEX OF TABLES

Table 1. Reference Documents	8
Table 2. Related Documents.....	8
Table 3. Acronyms and Abbreviation	9
Table 4: Available signals for interference simulation	15
Table 6. No of samples for each class.....	25
Table 7. Parameters of error distribution.....	27

1. Scope

1.1. Context

GATE4Rail aims to evaluate the GNSS performances in the railway environment and to contribute to the GNSS introduction in the railway.

As part of this objective, it is key to define a methodology for GNSS quality reception characterization along the track. In particular, GATE4Rail addresses several important topics in the WP3:

- the characterization of the GNSS performance in the railway local environment: the description of scenarios and error models in link with railway surroundings will help evaluating the performances and properties of some fail-safe train positioning components strongly influenced by global and local effects. It aims at defining a proper methodology for GNSS simulation ;
- this characterization will be part of the development of a dedicated system architecture for the lab tests in order to simulate much more faults in the system as ever could be tested in the field. By this, the probability of outages in the field due to errors will be reduced, leading to significantly increased availability of infrastructure.

1.2. Reference Documents

Table 1. Reference Documents

Document Number	Document Description
RD-1	GATE4Rail Grant Agreement Number 826324 — IP/ITD/CCA1 — IP2
RD-2	GATE4Rail Consortium Agreement v1.1
RD-3	GATE4Rail Proposal #826324 Technical annex 1-3

1.3. Related Documents

In Table 2 are listed the documents related to GATE4Rail project that have been used to develop this document.

Table 2. Related Documents

Document Number	Document Description
ReD-1	Deliverable 3.1 - GNSS Characterization in the Railway Domains
ReD-2	Deliverable 4.1 - Geo-distributed Simulation and Verification Infrastructure Modules and Interfaces, Functional and Operational Requirements
ReD-3	STARS project www.stars-gnss.eu
ReD-4	Khairol Amali Bin Ahmad, Mohamed Sahmoudi, Christophe Macabiau. Characterization of GNSS Receiver Position Errors for User Integrity Monitoring in Urban Environments. ENC-GNSS 2014, European Navigation Conference, Apr 2014, Rotterdam, Netherlands
ReD-5	Stallo, C., "An Anti-Jamming and Anti-Spoofing Digital Beamforming Platform for the GNSS-Based ERTMS Train Control System", Proceedings of 30th International Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2017), Portland, OR, September 2017.

Document Number	Document Description
ReD-6	Stallo, C., "Intelligent Antennas for Mitigating GNSS Jamming & Spoofing Hazards on the ERTMS Train Control," Proceedings of the ION 2019 Pacific PNT Meeting, Honolulu, Hawaii, April 2019
ReD-7	Deliverable 4.2 - Geo-distributed Simulation and Verification Infrastructure Modules and Interfaces Detailed Design
ReD-8	UNISIG, SUBSET-026 v3.6.0, "ERTMS/ETCS System Requirements Specification"
ReD-9	Marais, Juliette, Zhu, Ni, Bétaille, David, Berbineau, Marion, Safe satellite-based localization of the train thanks to a combination of accuracy enhancement and fault detection and exclusion schemes, WCRR 2019, Tokyo, Japan
ReD-10	Borio, D., Dosis, F., Kuusniemi, H., & Presti, L. L. (2016). Impact and detection of GNSS jammers on consumer grade satellite navigation receivers. Proceedings of the IEEE, 104(6), 1233-1245.
ReD-11	Dosis, F. (Ed.). (2015). GNSS Interference Threats and Countermeasures. Artech House.
ReD-12	Grejner-Brzezinska, D. A., Toth, C. K., Moore, T., Raquet, J. F., Miller, M. M., & Kealy, A. (2016). Multisensor navigation systems: A remedy for GNSS vulnerabilities?. Proceedings of the IEEE, 104(6), 1339-1353.
ReD-13	Ferrara, N. G., Bhuiyan, M. Z. H., Söderholm, S., Ruotsalainen, L., & Kuusniemi, H. (2018). A new implementation of narrowband interference detection, characterization, and mitigation technique for a software-defined multi-GNSS receiver. GPS Solutions, 22(4), 106.
ReD-14	Marais, J., Beugin, J., & Berbineau, M. (2017). A survey of GNSS-based research and developments for the European railway signaling. IEEE Transactions on Intelligent Transportation Systems, 18(10), 2602-2618.
ReD-15	Marais, J., Berbineau, M., & Heddebaut, M. (2005). Land mobile GNSS availability and multipath evaluation tool. IEEE transactions on vehicular technology, 54(5), 1697-1704.
ReD-16	Deliverable 2.2 – GNSS and Rail Test Cases
ReD-17	Deliverable 2.1 – Railway Scenarios and Requirements
ReD-18	Kazim, S.A., Ait Tmazirte, N., Marais, J. (2020). Realistic position error models for GNSS simulation in railway environments. ENC 2020, Dresde, Nov.

1.4. Acronyms and Abbreviations

In Table 3 are listed all Acronyms and Abbreviation used inside this document.

Table 3. Acronyms and Abbreviation

Terms, Acronyms and Abbreviation	Description
AIMN	Augmentation and Integrity Monitoring Network
CNO	Carrier to Noise Ratio
DBP	Digital Beamforming Platform
DOP	Dilution of Precision
ECEF	Earth Centred Earth Fixed

Terms, Acronyms and Abbreviation	Description
ERTMS	European Railway Traffic Management System
ETCS	European Train Control System
EVC	European Vital Computer
GNSS	Global Navigation Satellite System
GSM-R	Global System for Mobile communications - Railways
GUI	Graphic User Interface
KF	Kalman Filter
(N)LOS	(Non) Line Of Sight
NRTK	Network Real Time Kinematic
RBC	Radio Block Centre
TST	Track Simulation Tool
OBU	On-Board Unit
PPP	Precise Point Positioning
R&R	Record and Replay
RS	Reference Station
VB	Virtual Balise
VBR	Virtual Balise Reader
VIRGILIO	Virtual InstRuments for GNSS Augmentatlon and LocalizatIOn
WP	Work Package

1.5. Disclaimer excluding JU responsibility

Any dissemination of results must indicate that it reflects only the author's view and that the JU is not responsible for any use that may be made of the information it contains (see RD-1, § 29.5).

1.6. Description of Changes from the Previous Revision

Revision 00 – N/A.

Revision 01 – List of Modifications

1	Information about funding and disclaimer has been added page 3.
2	Executive summary has been extended in order to include relation of the work with previous work/other work done and the next steps within the GATE4Rail project
3	Explanation on the use of the state of the art added in section 3.
4	An explanation on how the environment has been characterised has been added in section 4.2 as well as perspectives on this topic and a reference.

2. Introduction

The objective of WP3 is to define a comprehensive methodology and the associated tools that are able to characterize the GNSS performance in the railway application scenarios defined in WP2 in the framework of a VB detection in ETCS/ERTMS context.

The developed methodology must allow to evaluate the effects of global and local GNSS faults on the properties of some Fail-Safe Train Positioning components (e.g. due to local effects such as multipath, radio frequency interference, spoofing or global effects as satellite clock/ephemeris failure or multiple ones).

GNSS technologies are influenced by global as well as local hazards. By definition, global hazards are shared by a large panel of users and are today well defined by global systems. Local hazards are caused by the infrastructure or vegetation around the track that creates blockages or multipath but also by interferers (intentional as well as non-intentional) that can degrade accuracy, availability and integrity of the GNSS solution.

Based on the state of the art, the WP3 focused on the identification of the railway environments and parameters that have to be taken into account (local and global) and simulated in order to cover the main events that can occur during an operational scenario. This is the content of D3.1.

The present deliverable relates the work performed in tasks 3.2 and 3.3 in particular.

Tasks 3.2 was performed in a close connection with WP4 in order to define the main GATE4Rail simulation blocks and their relative tools from a functional point of view and the high-level interfaces between them. For each block, inputs and outputs have been described (content and format). Moreover, each block has been defined with a set of parameters that will allow to tune the simulation (for example, in order to evaluate the impact of different AIMN architectures, OBU positioning algorithms and railway RAIM, GNSS receiver correlation schemes, jamming and spoofing detection and exclusion techniques, etc...). These results will be presented in section 5.

Task 3.3 identified faulty events and defined error **models** for fail-safe positioning components focusing in particular on GNSS global hazards (satellite clock and ephemeris error, rare constellation fault and anomalous ionosphere conditions) and local ones (depending on the selected railway scenario and local environment as severe multipath, attenuation foliage, orographic obstacles and presence of buildings or train stations) to be synthetically generated and simulated at the very top level of the chain (local and global hazards) with respective occurrence probabilities. In particular this deliverable will concentrate on the progress of GATE4Rail beyond the state of the art, that is to say, the provision of new error models for local effects.

The following of this deliverable is composed of 3 main sections and a conclusion. Section 3 reminds the main GNSS feared events and their modelling. Section 4 presents the GATE4Rail simulation framework and explains how GNSS errors will be introduced in the simulation chain. Section 5 presents how errors models have been developed in the project.

3. State of the Art of GNSS feared events and their occurrence or error model

The project and its consortium strongly benefit from the experience of past projects already mentioned, in particular STARS, ERSAT GGC and RHINOS that all focused on the integration of GNSS in the railway domain with a special attention to the local propagation effects that degrade performance.

However, as presented in [ReD-1], previous projects focused either on the identification of the effects that impact GNSS signals along a railway lines (STARS) or on their detection in order to localize the effects (ERSAT GGC). In these two mentioned projects, several acquisition campaigns were performed in order to record GNSS measurements in both nominal and harsh environments. STARS illustrated the effects of multipath, NLOS and interferences in different areas (urban, mountain...). ERSAT GGC demonstrated that some technics could detect the presence of such events. RHINOS focused on the definition of an integrity concept in order to ensure integrity of the position even in the presence of these effects. RHINOS also investigated some mitigation techniques but didn't provide either any error modelling.

These projects highlighted the presence of local events along railway lines and described some of the effects but none of them modelled their effects on positioning as occurrence probabilities or error distribution per type of event or type of area.

A major interest of these projects consist in the availability of large experimental databases. Some GATE4Rail consortium members being partners of these projects, they can access them and continue the process started in the previous projects in order to fully benefit of the richness of these data. The databases have thus been used for error model computation.

As can be seen on figure 1, GNSS errors can be divided into two main categories:

- Global Errors and
- Local Errors

GUIDE		RISK ASSESSMENT MATRIX		
		PROBABILITY OF ERRORS (Likelihood)		
		Unlikely	Moderately	Highly likely
IMPACT OF ERRORS (Consequence)	Major (>10m)	• Jamming	• Interference	• Obturation • Multipath
	Significant (<10m)	• Clock		• Diffraction • Unfitted Algorithm (Hybridizations, PPP,...)
	Moderate (<3m)	• Orbit	• Ionosphere	• Attenuation • Sensitivity • Antenna
	Minor (<1m)		• Geometry (DOP)	• Thermal noise

LOCAL
GLOBAL

Terminal

© GUIDE – All rights reserved

Figure 1: GNSS errors classification

The first category, identified as global errors, lists the classical sources known as contributing to degraded performance of the localisation. Some of these sources are caused by the system (wrong or inaccurate satellite orbit information for example), some others are perturbations observed on every satellite of the constellation such as ionosphere and troposphere delays. The second category, instead, includes all errors related to the local environment. The main source of errors is related to the environment surrounding the receiver and the impact that this last has on the signal reception.

These global effects are well-known and characterized in the literature. They can be shared by the different users of GNSS receivers, from aeronautics as well as land users. Effects like atmospheric impact (both from the ionospheric and the troposphere) are already implemented in some simulators and are not specific to railways.

Thus, in GATE4Rail WP3, we decided to focus on the second category of errors that are the local ones. By local, we define all the errors that will be generated by the close surrounding of the antenna such as obstacles or other radio sources that could disturb the GNSS signal reception.

Indeed, the error in the position is partly the consequence of the corrupted pseudorange estimation in the presence of the local effects. These local effects are widely investigated in the literature as they remain one of the most challenging issue to solve for accurate and safe localization in ground transport applications [ReD-14]. These effects keep varying as the train moves due to continuous change in the surrounding environment of the GNSS antenna. It becomes very challenging to detect and completely remove these local effects due to their characteristics that are given as:

- Local effects are both space and time dependent [ReD-15]. It means that, as the satellites are moving, the satellite signals do not reflect on the same position due to satellite azimuth and elevation.
- The geometry of the visible satellites is not always the same in time (DOP varies) because of the satellites movement. It can also bring a large uncertainty to the position.
- The train moves with an irregular speed due to accelerations and decelerations (when brakes are applied). The variable speed can also have an impact on the effects.

The local effects are typically multipath, NLOS effects and interferences.

3.1. Multipath and NLOS effects

Multipath refers to the phenomenon where a signal reaches the receiver after reflections and/or diffractions that are combined with the Line-Of-Sight (LOS) signal. Each Non Line-Of-Sight (NLOS) signal received is called an echo. Its delay and phase depends on the surface characteristics and geometry of the scene. Multipath distorts the shape of the correlation used by the receiver to extract the pseudo-range information; therefore it also has an impact on the position estimation. The NLOS label is used when the direct signal cannot be received because of the presence of a masking obstacle. Then, the correlation output is always characterized by a delay on the pseudo-range estimation caused by the absence of the direct path [ReD-9].

Multipath and NLOS effects differ with in the train run and from one run to another (trajectory but also time schedule of the train). It means that an error measured or simulated for one point at instant “t” is no more valid for a different point or a different time at same point. The following approaches have been investigated while considering these characteristics. In the following, we concentrate on local errors. We have split them into two sub-categories:

1. Typical Environment; frequently GNSS performances are associated to typical environment in which it is supposed to perform. We can distinguish typical environment as:
 - a. open-sky,
 - b. urban,
 - c. peri-urban,
 - d. forest.

causing different errors on GNSS measurements.

2. Local obstacles; over the errors introduced by the environment GNSS signals are also distorted by punctual obstacles (like tunnels, bridges and/or others) causing partial or total loss of GNSS signals and consequently different positions errors.

3.2. Interferences

GNSS signals are transmitted with very low power. That makes them sensitive to accidental as well as intentional interference. Intentional interference can be jamming or spoofing. A jammer deliberately transmits powerful signals at the GNSS frequencies and thus can disrupt GNSS-based services in wide geographical areas [ReD-10]. A spoofer can trick the receiver into outputting false estimate of its position. Since a GNSS receiver is embedded alone or inside a multi-sensor system, it is important to ensure that it will offer a service resilient to any kind of interference.

Interferences degrade the availability of GNSS measurements and their reliability [ReD-11]. They lead to disruption of the service or can even lead to a complete denial of the positioning capabilities of the receiver [ReD-12] [ReD-13].

The presence of interference depends of the presence of a transmitter in the surrounding of the train. However, their presence and positions are not referenced in any map or database. Even worse with jammers, by definition.

Yet, interferences will nevertheless be considered in the context of the project, and **most of the signals that can be used as interference to GNSS will be available in the Module 3** of the GATE4Rail simulation platform. This includes existing signals that are allowed in the GNSS bands (such DME and TACAN that are used in the aviation domain, or radar signals), but also signal usually used as specific GNSS jammer (such as Gaussian Noise, chirp signals, or spread spectrum signals).

The complete list of signals available is provided in the table here after.

Table 4: Available signals for interference simulation

Continuous wave (CW)	Wind profiler
Gaussian white noise (GWN)	JTIDS/MIDS
Radio amateur	Spread Spectrum GNSS like signals
VHF	Frequency hopping signals (slow and fast)
Primary radar	Chirp signals (with linear and non-linear frequency modulation)
Secondary radar	Tone/Multitone interference
DME/TACAN	Time hopping signals including radar like burst signals with typical modulations including phase-coded CW, FMCW

The use of advanced solutions based on the use of Antenna array and digital beamforming algorithms for jamming detection and exclusion or mitigation has been deeply analysed for rail application for the first time at the knowledge's writer in the context of ESA GSTP 6.2 Digital Beamforming for Rail [ReD-5, ReD-6].

In the context of the GATE4Rail project, the Module #3 of the chain is responsible for the generation of the interference/jamming synthetic waveforms and their impact is then evaluated through Modules #4 and #5. The occurrence of the feared events will depend on the characteristics of the environment.

Section 5 reminds the architecture chosen for the project and the main modules.

4. Simulation framework and tools definition

4.1. Architecture description

The main goal of the GATE4Rail test-bed is to simulate the performance of GNSS in the rail context in a virtualized and geo-distributed environment by minimizing the impact of the GNSS on the ETCS/ERTMS on-board and trackside subsystems as defined in the SUBSET-026-2 [ReD-8].

As shown in figure 7, five main modules have been defined, each one related to a specific area (GNSS, VBR, ERTMS/ETCS) and to a different ERTMS/ETCS and GNSS laboratory:

1. **Module #1: Emulated Moving Train.** It is a Railway Traffic Generator, implemented through the CEDEX ETCS/ERTMS laboratory that generates the railway traffic. Moreover, this block interacts with the Balise detected trigger that states if the train is passed over a VB and delivers the Balise reference Identification (ID).
2. **Module #2: Mission Control System and Train Dynamics.** It processes the information received from the traffic generator and translates them into a set of information usable by GNSS infrastructure. It is implemented through the CEDEX ETCS/ERTMS laboratory. In particular, the Mission Control System takes the input from the Global Track info and from the traffic generator and defines the route of the train. The Train Dynamics generates the movement trajectory of the train. Particularly, the output consists of Ground truth info that is a list of train positions and speed epoch-by-epoch.
3. **Module #3: GNSS Signal Generation (M3SB).** It is responsible for the generation of synthetic GNSS signals and observables both in nominal and fault conditions. It takes as input a train trajectory and a set of GNSS conditions parameters (such as the GNSS constellation(s) to be simulated, the atmospheric impact models to be used, the type of interferences to be account for – if any, ...). The GNSS signals generated is then sent to a GNSS receiver so that ‘real’ raw data (pseudoranges, carrier phase, Doppler shift, Carrier to Noise Ratio (C/N0)) can be provided as output of the module.
4. **Module #4: GNSS Augmentation and On-Board Unit (RDL).** This module represents the GNSS-based train positioning unit and it is implemented through the VIRGILIO simulator. It includes a GNSS-based OBU and an AIMN (based on local Reference Stations (RSs) network or EGNOS). Its inputs are represented by the raw data (pseudoranges, carrier phase, Doppler, C/N0), the AIMN information (navigation data and differential corrections), the ephemeris, Local Track info, Ground Truth info and the IMU observables.
5. **Module#5: Virtual Balise Trigger (RDL).** This module, implemented through the RDL lab, is responsible for triggering the event related to the VBD (Virtual Balise Detection). In particular, its output is the passage trigger with the balise reference ID. The inputs are the VB positions and the PVT estimates.

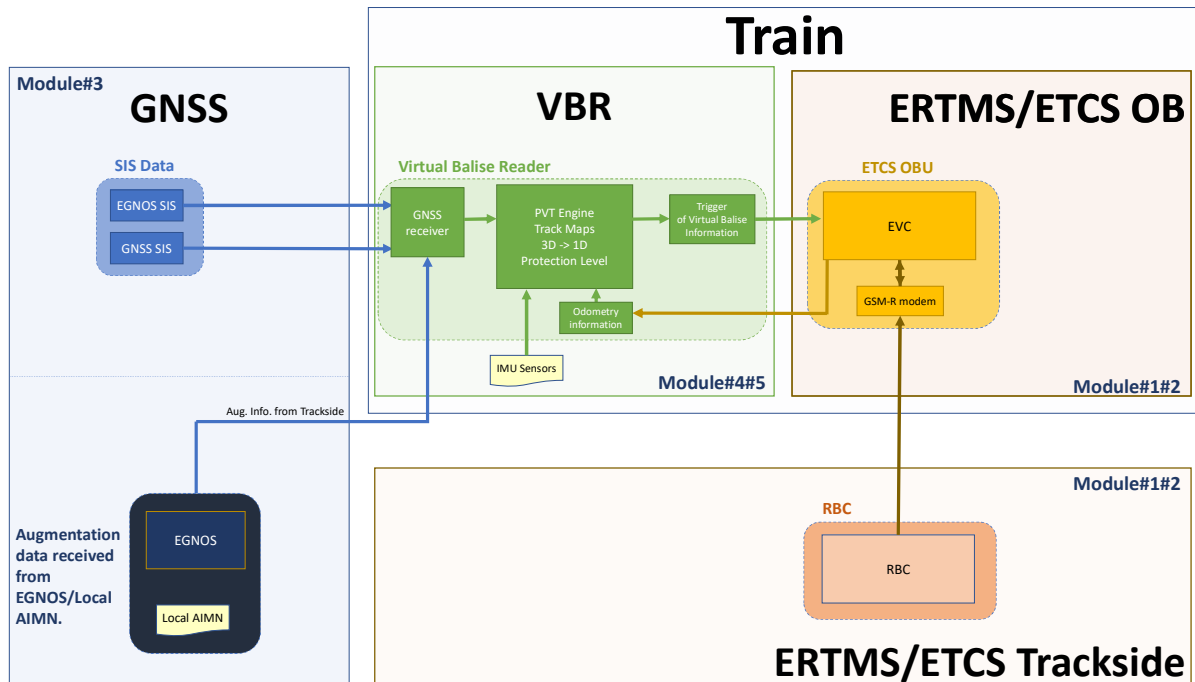


Figure 2: GATE4Rail architecture

The design and development of the main modules of the GATE4Rail architecture has been based on the tools already available in the GNSS labs (GUIDE, M3SB, IFSTTAR, RDL and INECO) and ETCS/ERTMS ones (CEDEX). Each tool generates the information required by the other modules for the execution of the simulations based on ERTMS/ETCS and GNSS Scenarios. In the following, the main tools required for implementing the virtualized test-bed to support the characterization of GNSS in the railway are described. More details about the tools related to the GNSS characterization in the railways domains are reported in D3.1 [ReD-1].

VIRGILIO (Virtual InstRuments for GNSS Augmentation and LocalizatIOn) simulator, developed by Radiolabs, is a complete multi-constellation software simulator able to characterize the performance of a LDS GNSS-based for rail. VIRGILIO simulator is used for simulating the AIMN (Augmentation and Integrity Monitoring Network), GNSS-OBU, and then the VBR (Virtual Balise Reader) trigger behaviours.

M3SB has developed a complete GNSS dedicated testbed product named STELLA®. STELLA enables both the simulation of GNSS signals and the record and playback of real signals collected on the field. The simulation function provides support to account for various errors that can affect GNSS performances (such as the impact of global errors as the propagation of the signal through ionosphere and troposphere, or clock errors, but also local errors such as multi-path). GNSS vulnerability to interferences (both intentional and non-intentional) can also be tested using STELLA.

STELLA is a multi-layer platform which can generate signals/data at different level for testing the various steps of signal processing within a GNSS receiver from the RF signals down to the PVT (Position, Velocity and Time) solution.

GUIDE implements tools of simulations reproducing the issues of propagation of the environment (obstructions, multipath, diffractions and fading). Typical 3D virtual scenes can be set to simulate the receiving conditions of GNSS signals. These scenarios can be then supplemented with other "feared events" outside local hazards, i.e. the global hazards such as satellite and ionospheric errors.

With the tools developed by IFSTTAR, GNSS performances are associated to typical environment in which it is supposed to perform. It can distinguish typical environment as open-sky, urban, semi-urban causing different errors on GNSS measurements. The role of the tool is to generate for any new trajectory, random errors along latitude and longitude axes in order to estimate a trajectory taken into account the corrupted reception of the signals.

CEDEX lab is composed by two test benches, the first one for the On-Board Unit (OBU or EVC) and the second one for the ETCS Trackside (Radio Block Center (RBC)), and by two main additional modules as Track Simulation Tool (TST) and the GSM-R network simulator. OBU test Bench simulates the train movement and also the interfaces between the OBU and the simulated train. Trackside test Bench simulates the interlocking (IxL) as well as the IxL GUI (Graphic User Interface), which is in fact the local operation post to set the routes. GSM-R Network module simulates the GSM-R network and is the way of inter-changing L2 messages between the RBC and the EVC. TST tool simulates the real trackside and provides to the lab the functionality of trains moving over the real track.

GNSS4RAIL simulator, developed by INECO, is able to provide the calculation, via a simulation, of satellite visibility parameters considering railways environment in terms of natural and artificial obstacles. Both constellations (Galileo and GPS) can be assessed for analysing geometry and number of visible satellites at each point of the track.

By taken into account the tools available in the consortium laboratories, the general architecture for GATE4Rail test-bed has been defined in sections 7 and 8 of the Deliverable 4.1 [ReD-2] while the main constituents and the interfaces has been described in details in Deliverable 4.2 [ReD-7].

4.2. Introduction of GNSS errors and their models into the GATE4Rail simulation chain

In the framework described in 5.1, the Module #3 is responsible of GNSS signals. This module will require knowledge of the different types of errors reminded in section 4. While considering existing error models described in the literature, and models already available in the GATE4Rail simulation modules (in particular Module #3), we have focused our study on the errors that originate due to the local effects and typically the ones that are associated to the railway environment.

In the GATE4Rail simulation chain that can simulate different levels of data, errors can be represented at different levels or with different sources of data. We have identified:

- Pseudo-range level: the delay caused by the local effects creates a delay in the signal reception,
- Positioning level: the local environment creates positioning error in 3D.

Considering the GATE4Rail simulator structure and the different available modules, the consortium agreed that the error to be modelled will be represented in the position domain. In concrete terms, based on the path ground truth, local error models will generate trajectory errors that will then be added on the ground truth path of the train in order to be used as input by the GNSS simulator (see illustration in Figure 3).

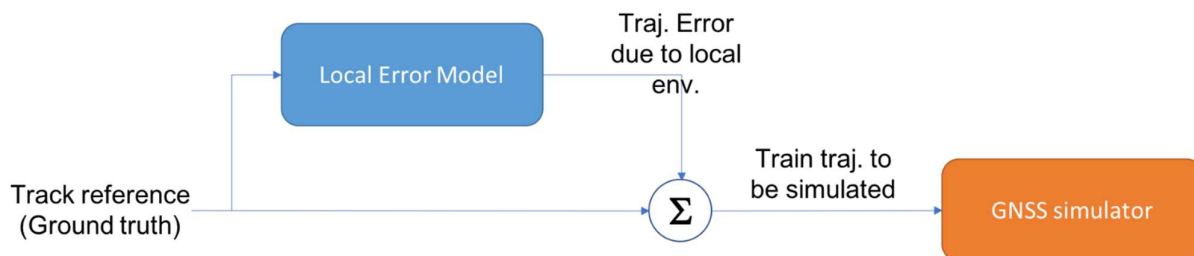


Figure 3: How GNSS error on track reference will be used by module#3

The choice of the typical error models and faulty events have been made considering: the environments defined in D3.1 [ReD-1] and the state of the art that also classically consider some typical environments.

One can notice that these typical environments are always characterized qualitatively. As an example, one use the term “urban” or “forest” in general without quantifying the height of the obstacles, the width of the street or the density of the foliage. Urban area is characterized the same way whatever the city can be as dense and high as Tokyo or a small European city. In this deliverable, we have kept this qualitative classification, keeping in mind that the databases are recorded in European environments, mostly rural or low density areas. At the time of this deliverable, bridges and tunnels are descriptive, but the other areas (urban, forest and open sky) are classified considering the user expertise or evaluation based on on-site knowledge or satellite-based images.

An on-going research task consists however to classify these areas in a more automatic manner with the definition of quantified criteria [ReD-18].

As an example, the process is illustrated in Figure 4 and in Figure 5 below representing open sky environment. Based on the train ground truth (in red), local error models generated position errors that are added on top of the ground truth to obtain the trajectory to be used as input of the Module#3 (in blue).

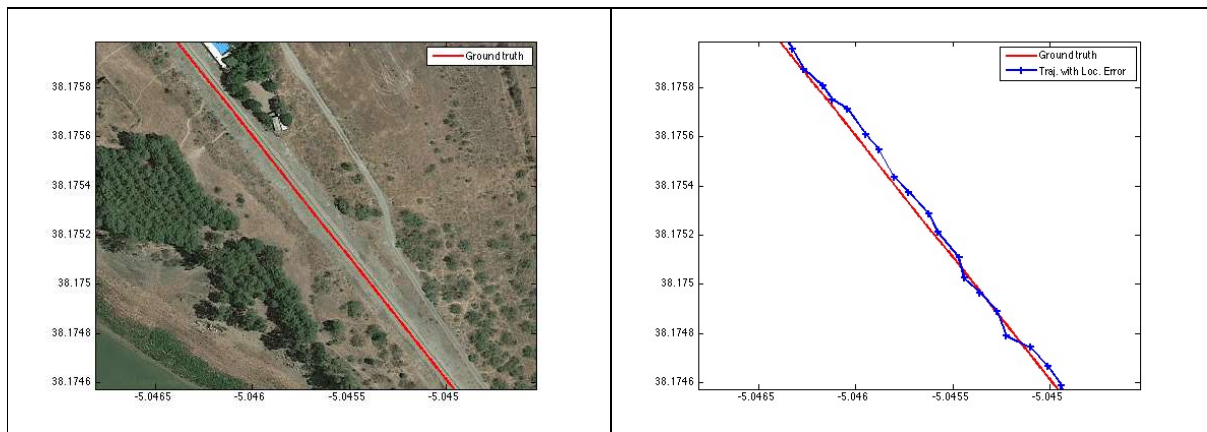


Figure 4: Module#3 input trajectory computation based on Local error models and ground truth.

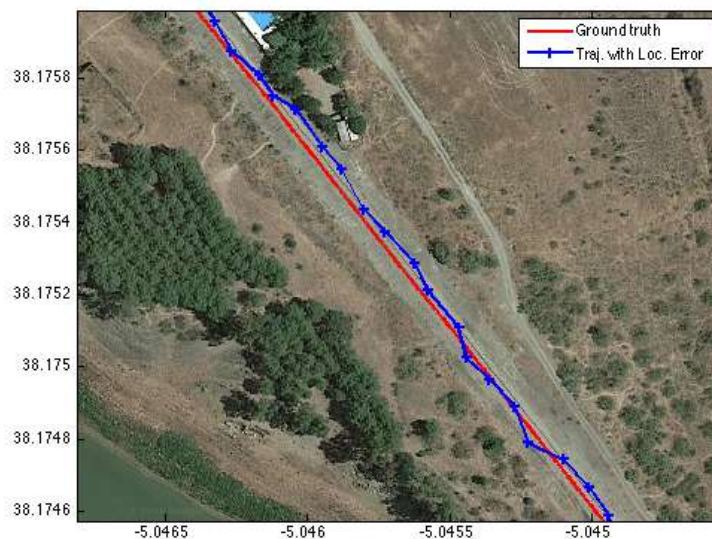


Figure 5: Module#3 input trajectory based on Local error models and ground truth.

5. Errors, faulty events and vulnerabilities modelling for Fail-Safe Positioning Components

5.1. Error model representation and methodology

5.1.1. Taking into account the estimator

The selection of the position estimator is an important aspect in error modelling as it is closely related to the shape of the error model. There are mainly two categories of estimation algorithms commonly used to compute position i.e. snapshot and recursive estimators. A Least Square is considered as a snapshot method as it estimates the position without any influence of the previously computed position. At each instant, the position is estimated from the available observations thus it has a large error variance. Conversely, Kalman filter is a recursive estimator as it requires the recent estimate of the state vector for the prediction. The predicted state is used together with the measurements to estimate position. As a result, it brings a smoothed effect in the resultant position.

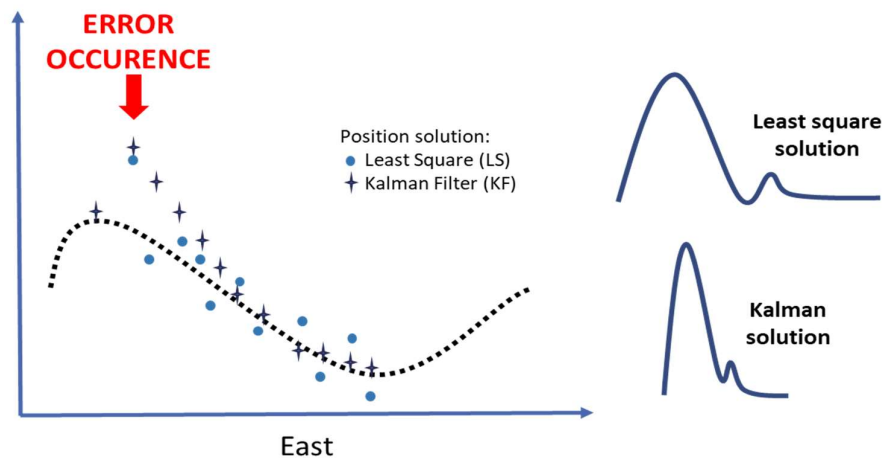


Figure 6: An illustration to differentiate LS and KF estimator.

Figure 6 summarizes the conceptual difference between the Least Square and Kalman filter in the context of error modelling. It illustrates particularly the behaviour of both estimation algorithms proceeding to the point of error occurrence. At each time instant, the least square solution independently estimates the position from the current measurements. Thus at any instant the error does not have any effect on the succeeding position. It tends to have large error variance as it does not filter/smooth the data samples. Kalman filter on the other hand carries a history of the previous position at each instant to do the prediction. Therefore, once the error has occurred its effects diminishes with time until the position is converged. Due to this smoothing/filtering effects KF has a small error variance.

It is expected that the two estimators will bring different shape of error distributions. With LS, the queues (distribution of large error point) would very well distant from the normal distribution while in KF these queues may not be very much evident (nearly merged with the main/normal error distribution) due to smoothing effects. The distribution curves of the two algorithms are illustrated in figure 6.

5.1.2. Position error computation

The position error is computed as the difference of the measured and the estimated positions given as,

$$E_{pos} = X_{measured} - X_{ref} \quad (1)$$

We have identified two possibilities to model the position errors according to the requirement they possessed. It may include 1) track-error identification and classification, and 2) environment identification and error classification, as presented in the following.

5.1.3. Different approaches investigated

5.1.3.1. Typical Environment

The first tool developed for GATE4Rail by UGE relies on the use of existing dataset of GNSS measurements along railway lines for error model extraction. Two approaches have been investigated first, that are presented here.

Typical track-error identification and their classification

This methodology requires error computation along a complete track without any a priori knowledge of the surrounding environment. These errors are then grouped into different classes w.r.t to their distribution curves. A simple clustering algorithm can do such a classification but in this manner the errors are not necessarily linked to the environmental surroundings of the track.

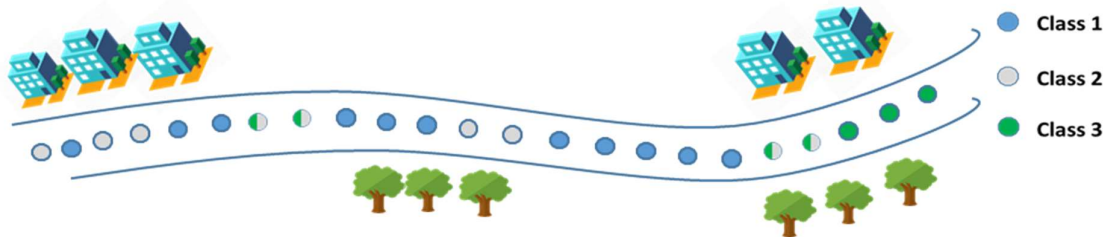


Figure 7: An example of Track error classification without a priori knowledge of the environment of the track.

Figure 7 illustrates an example of track error classification. The errors associated to the track position are grouped into 3 classes that are represented by distinct colours. Here the multi-coloured (e.g. green and grey) points symbolizes the error samples of a Gaussian mixture that falls in the shared region of the two distribution curves. With the clustering algorithm, these samples are classified in the group where they are most expected, due to larger probability.

There are several clustering algorithms that can easily find different groups of clusters within the mixed samples. The most common algorithms are k-mean, mean shift and Expectation maximization (EM). This is a very simple technique but its effectiveness is limited due to the characteristics of the errors in GNSS. It may require the division of a complete track into smaller regions to test for the repeatability with a large amount of data collected along several runs of the same track. Figure 8 summarizes the process.

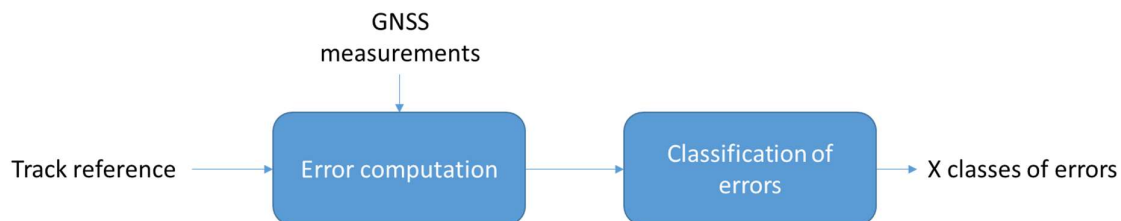


Figure 8: Schematic process of first approach.

Environment identification and track-error classification

This second approach requires pre-labelling of samples considering the surrounding environment of the track. As a first step, it requires a manual identification of the track environment. This can be done with google map. The complete route is divided into different regions representing the major proportion of environment around the track. The typical environment may include open-sky, foliage and buildings as depicted in Figure 9.

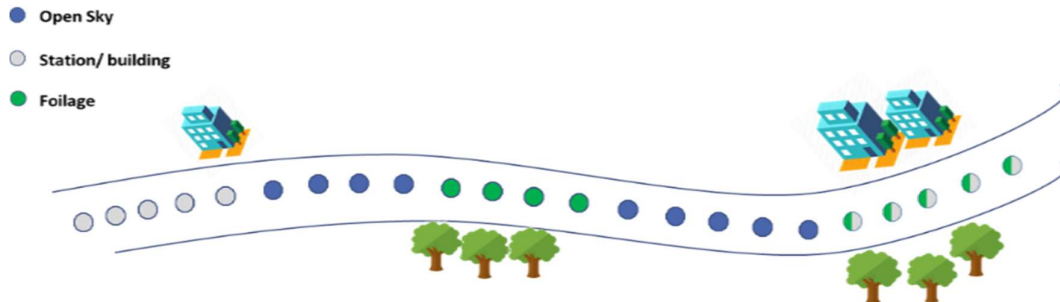


Figure 9: An example of a priori classification of the environment before error computation.

Figure 9 illustrates an example of track error classification with environment labels. The errors associated to the track position are then grouped into three classes, represented by different colours. The multi-colour point represent associated position error that shares two or more types of environments. For simplicity, such samples could fall into a class that has a major representation.

After the classification step, several different distributions are considered separately for each class to get the best modal representation. Besides general limitation presented above this method has a pre-labelling requirement; by inspection it could be laborious in case it involves several runs. Secondly, the local effects vary with composition of the local environment due to the dimensions, density, nature of the obstacles and others, therefore, the severity of error can differ largely even with in the same environment class for example spars tree vs. forests, buildings in urban vs. suburban region.

Figure 10 summarizes the process.

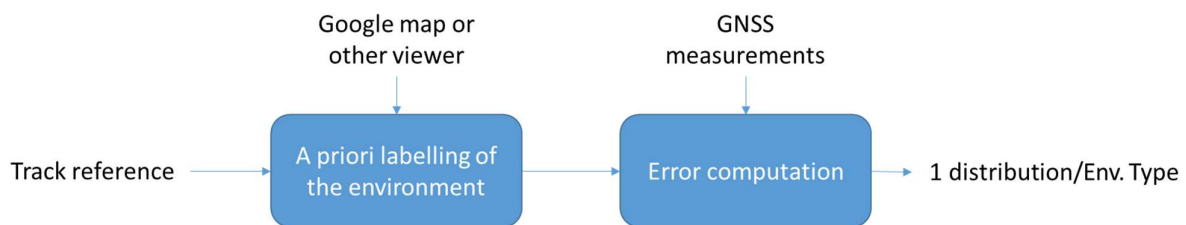


Figure 10: Schematic process of second approach.

5.1.3.2. Local Obstacles approach

To model the impact of errors generated by local obstacles GUIDE has used a “Record&Replay” (R&R) methodology. It consists in replaying several times the same set of digitized GNSS signals, recorded during a previous field test campaign, on a GNSS receiver with a defined configuration (single frequency or dual frequency, SBAS corrections active or not).



Figure 11: Single PVT track (cyan/blue) after a bridge crossing

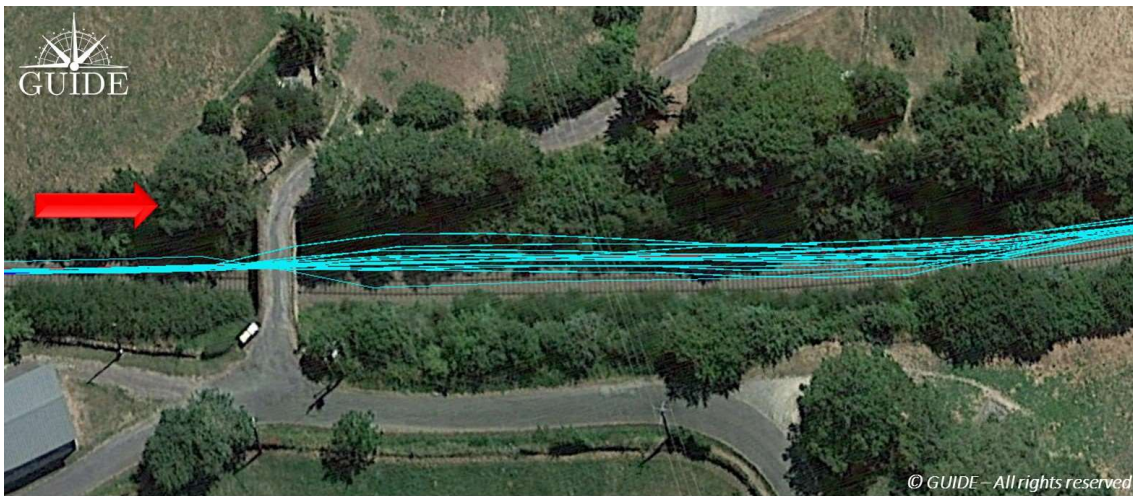


Figure 12: Multiple PVT tracks (cyan/blue) after bridge crossing

This methodology allows to better understand the effects of local obstacles on receivers position solutions. As visible in figure 11 and figure 12 using a single track it is not possible to derive any model for a bridge crossing. The use of the R&R methodology, instead, permits to better understand that a sort of random process causing the spread of the positions solution occurs after the obstacle crossing.

5.2. Use of the error models for the estimation of positioning errors on tracks

The routine developed to generate errors relies on the distributions built as described. As described in Figure 13 and presented before in section 5, an error is randomly applied to each point of the reference trajectory, depending on the local environment label associated.

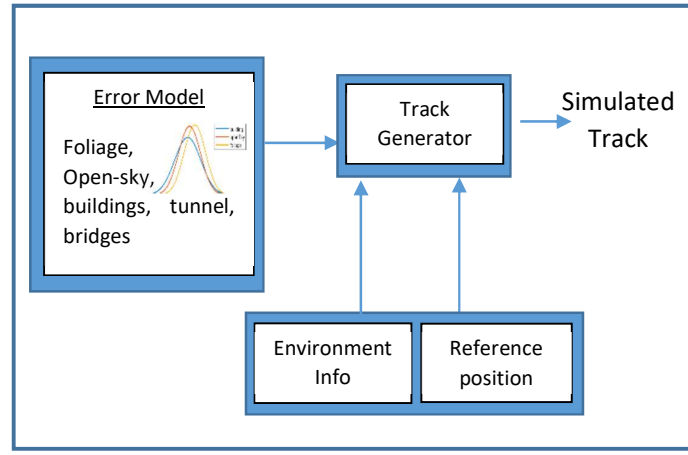


Figure 13: Use of error models to simulate errors on track reference.

The models will provide errors in meters for the along and cross-track errors. The used convention is to provide positive cross-track errors for positions on the left of the rail one and positive along-track errors for position forwarding the true one.

In order to make it compatible with a GNSS constellation simulator, these errors shall be converted at first step into north-east errors and then converted in degrees.

In order to perform the first step the heading of the train must be known, if this is not the case is must be estimated as follow:

$$\theta(t) = \tan^{-1} \left(\frac{\text{long}(t) - \text{long}(t-1)}{\text{lat}(t) - \text{lat}(t-1)} \right)$$

$$Err_{North}(t) = Err_{Cross}(t) \cos(\theta(t)) + Err_{Along}(t) \sin(\theta(t))$$

$$Err_{East}(t) = Err_{Cross}(t) \sin(\theta(t)) - Err_{Along}(t) \cos(\theta(t))$$

Err_{north} and Err_{East} are provided in meters.

For the conversion of meters in degrees this last shall keep into account the earth radius, 6371000 meters, and the latitude of the train for the point to convert. The formula are:

$$Err_{lat}(t) = \sin^{-1} \left(\frac{Err_{nort}(t)}{R_t} \right)$$

$$Err_{long}(t) = \sin^{-1} \left(\frac{Err_{nort}(t)}{R_t \cos(\text{lat}_{ref}(t))} \right)$$

For this model the vertical errors has not been modelled since the impact of a brined on this component is not relevant.

5.3. Error models: Results and analysis with real data

5.3.1. For typical environments

5.3.1.1. Computation of error distributions

In previous sections, we have investigated two approaches. For the GATE4Rail application, we have kept the second ones presented in section 6.2.3 due to its practical applicability in the chain while considering all practical limitations and in order to link environment and error.

Here below is the application of the process presented on a real dataset. For this purpose, we used a data campaign previously performed in Switzerland in the framework of the H2020 STARS project by Siemens. The complete route is 163 km long with nearly 3 hours of measurements. It has used odometry and Eurobalise system to estimate the ground truth [ReD-3]. The complete route incorporates several environments such as sparse trees, forests, tunnels, trenches, bridges and buildings.

In order to keep confidentiality of the data and with respect to the rules of the STARS consortium agreement, the data are not shared by the GATE4Rail consortium and are used only by the STARS ex-members for modelling.

As a first step, the complete route is divided into different regions according to the proportion of environment it represents around the track. As mentioned earlier the classification of the track is done visually/manually with google maps. The ground truth positions (points) are labelled with the environment tags and are represented in different colours in Figure 14. They have been defined, as a first set, with the following classes: open sky, forest or trees along the track and urban area.



Figure 14: Environment identification with Google maps.

Table 5 presents the samples from the run presented above and manually labelled samples belonging to each environment class. In the later study we will use these samples to estimate error model for each class.

Table 5. No of samples for each class

	Open Sky	Foliage	Buildings
No of samples	1854	1242	5465

We have used the RTKLib open source tool for the processing of the GNSS raw measurements. The least square algorithm is selected to estimate positions from the GPS and Galileo observations. Finally, error is

computed for each reference position using eq. 1 and these errors are projected to the track (body) frame. At each reference position, the error is distributed in two components lateral (cross-track) and forward (along-track) error. The forward and the lateral errors belonging to each class are shown in Figure 15. The contribution of the different environments is observed in the position estimation. It is expected that the position estimation should be optimal (i.e. close to the true position) in the open-sky region due to the no or minimal presence of obstacles. In urban regions, as the train passes through cities the positioning accuracy is badly affected due the presence of buildings along the track. The error is higher mainly due the signal reflections coming from the nearby obstacles. It can also be seen that the errors are more spread in the ‘building’ class though the significant numbers of samples are concentrated in the same area/region as for other environment classes. It could be due to two reasons i.e. the signals may not have interacted at all with the obstacles or the disturbance(s) due to the obstacles are absorbed in the position estimation.

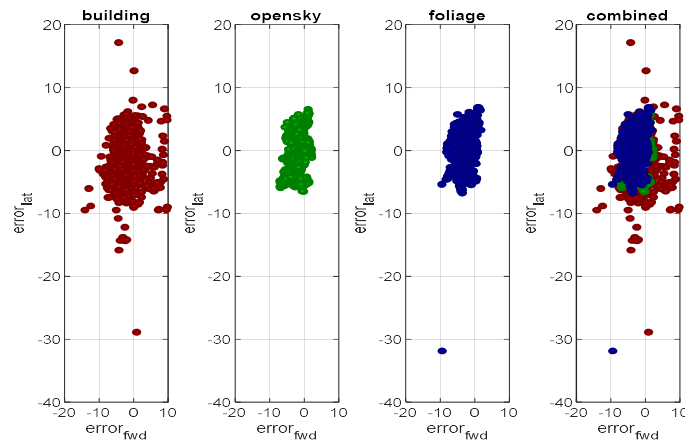


Figure 15: Error classified in typical environment.

After the error computation, the Matlab function ‘fitdist’ is used to approximate the error distribution. As an input, this function requires data samples and the distribution type to estimate the distribution parameters i.e. mean and the standard deviation. Figure 16 shows the error model estimated from the ‘normal’ distribution. The distributions intervals are highlighted with green, yellow and red colours that represent the standard deviation 1, 2 and 3 sigmas respectively. It appears that the model nearly fits the ‘building’ class in contrary to ‘open-sky’ and foliage classes. Meanwhile, the presence of Gaussian(s) mixture is also noticed with in other classes. It may be due to the smaller number of samples (missing samples) or wrong classification of samples or even the samples belonging to mixed environment.

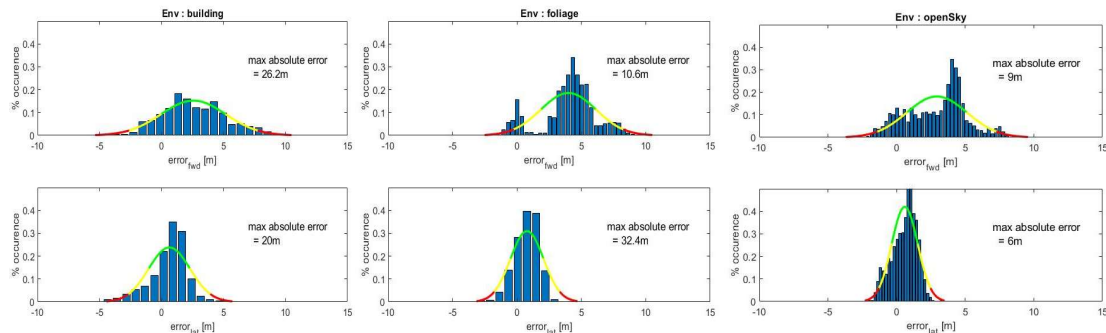


Figure 16: Histogram and error distribution.

With these Gaussian distributions, each environment can be represented by a mean μ and a variance σ^2 .

Based on STARS data, the distributions summarized in Table 6 have been proposed, that will require to be refined in the future with more extensive availability of data.

Table 6. Parameters of error distribution

	Latitude error (deg)		Longitude error (deg)		Height error (m)	
	mean	Variance	mean	Variance	mean	Variance
Open sky	-7.367e-7	5.9979e-10	-2.7472e-5	5.2232e-10	-1.4119	2.2552
Building/urban area	-1.285e-5	5.0427e-10	-2.7224e-5	7.4206e-10	-2.6868	10.287
Foliage/Forest	2.3099e-6	8.2942e-10	-3.7620e-5	7.9032e-10	-1.8310	3.1607

A main drawback of using statistical Gaussian distributions here is that larger effects, which do not often occur are largely ignored due to very minimal probability. In context to regeneration of the trajectory, these errors are either largely ignored due to very minimal occurrence probability or are otherwise associated to the wrong position/region of the track. In order to make them more visible, we discussed different possibilities:

- add punctually a supplementary large fault only to some point of interest positions/regions. That is to say, force the system to simulate such effects.
- These errors in the queues must be treated as a point of interest (POI) similar to the bridge model presented by GUIDE. Therefore, they must be modelled separately and must be linked with the time and the position/region on the track.
- Or, as a perspective, to define other distributions (as can be seen in the literature with Pareto distributions for example [ReD-4]).

5.3.1.2. Model Validation

A validation process is carried out to observe the effects of random errors on the track positions. For this purpose, a routine is developed to associate random errors to the reference position to create error induced trajectories. Consequently, 10 different trajectories are created and each time a 'random' Matlab function is called to generate random errors. These error are generated with pre-defined parameters (mean and variance) estimated in error model creation process for each environment. The complete process is illustrated in Figure 13. The validation process is based on the following assumptions.

1. The error components along the track and perpendicular to the track are independent (which in reality is not true); thus for each reference position, random error is generated separately for each component cross and along-track.
2. The error models rightly estimated the true error for each defined environment and can be applied to other trajectories with different location and time of acquisition.
3. Error models are estimated taking into consideration all the measurements acquired in a particular environment with varying density, type of material and the placement (to some extent) of the obstacle from the track position. So it probable that the severity of error may differ due to the random error association.

Figures 22, 23 and 24 compare different trajectories. For each, the Injected trajectory (red colour) represents the track positions that are determined from the real raw observations and using a least square estimator (explained in sec 6.3.1.1). These positions are used to estimate the error models for a defined environment. The generated trajectory (green colour) represents the track positions that are created by inducing the random error on the reference position. As expected the errors are diffused incoherently in both cross and along-track as a result of the assumptions presented above. The receiver output (green colour) trajectory represents the track positions that are estimated by the receiver from the real observation acquired during the acquisition campaign.

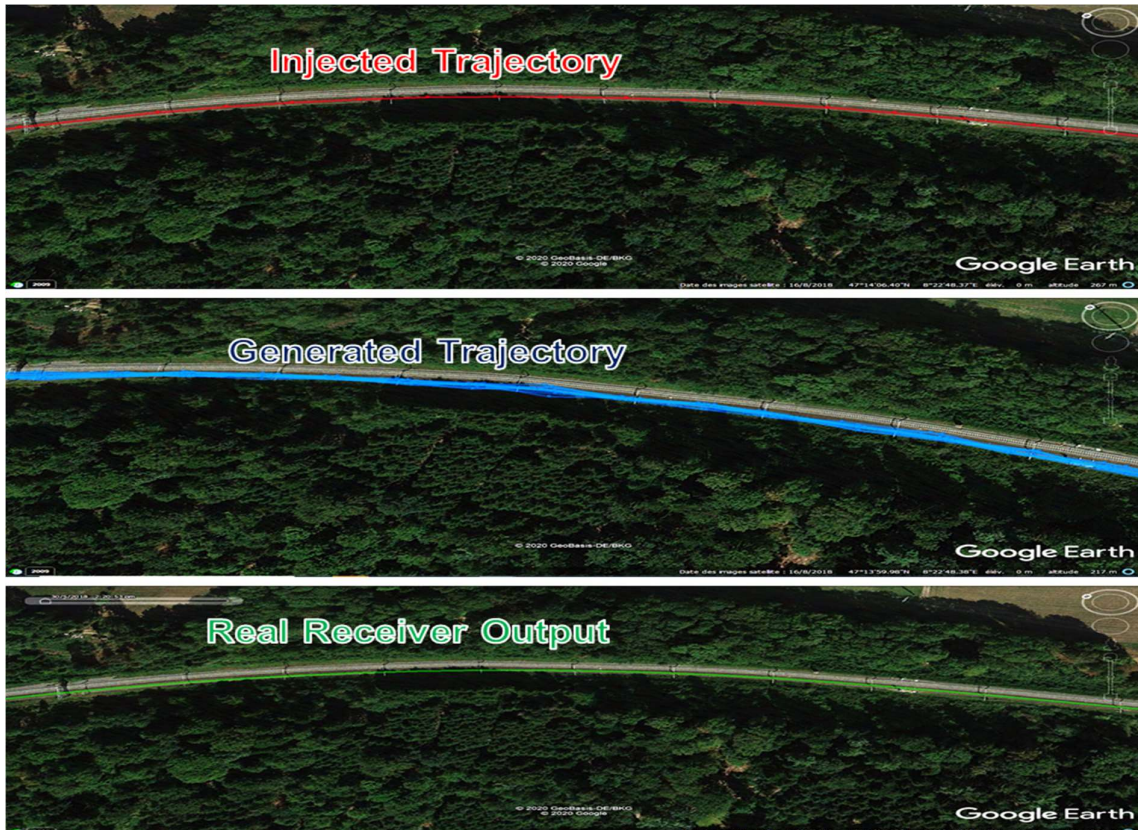


Figure 17: Comparison between simulated and real trajectories in Forest/Foliage

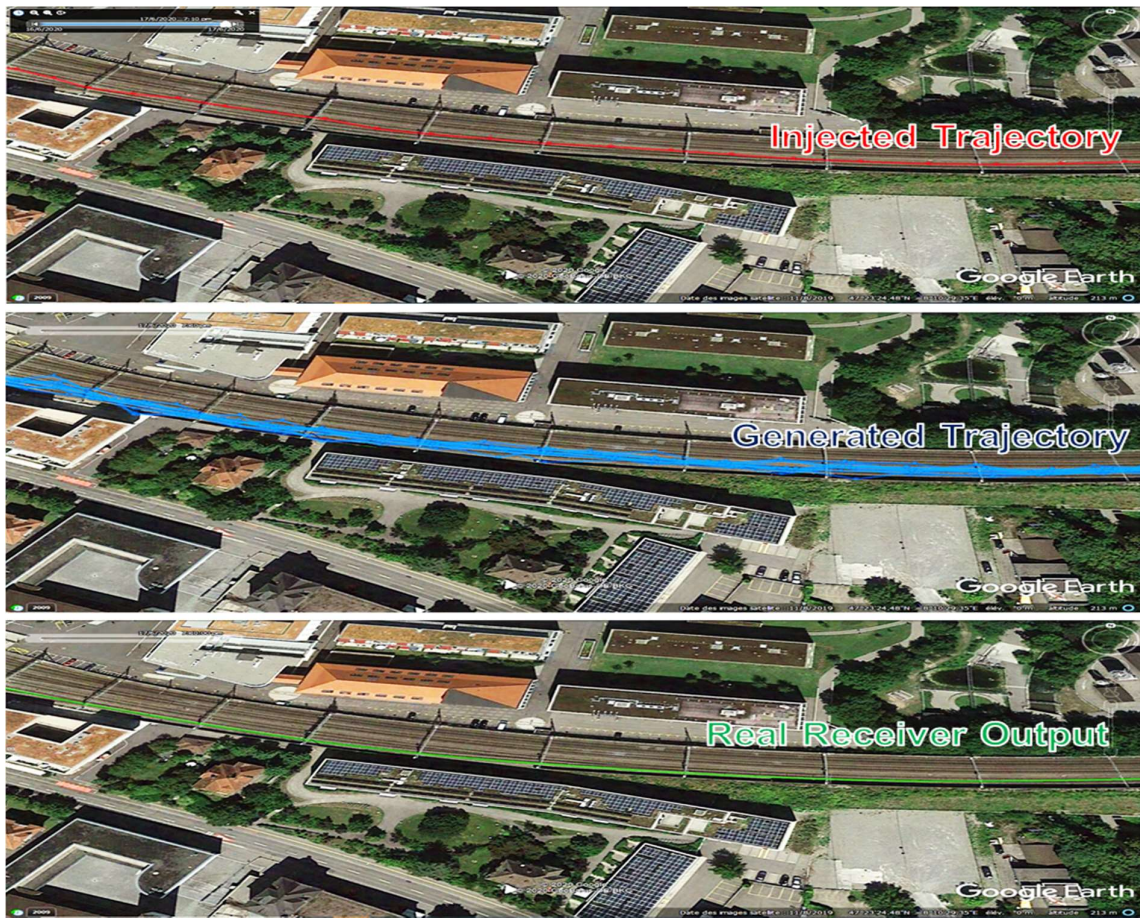


Figure 18: Comparison between simulated and real trajectories in Urban Environment

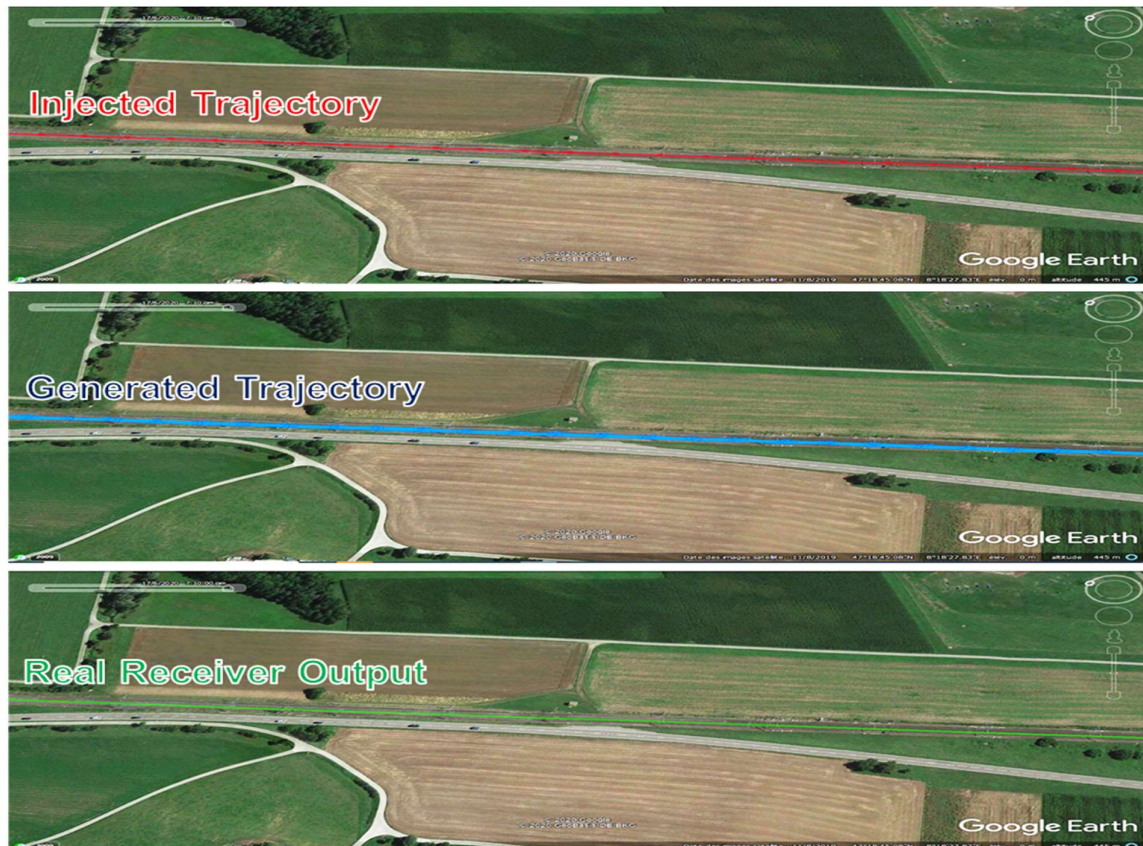


Figure 19: Comparison between simulated and real trajectories in Open sky environment

5.3.2. Specific obstacles (GUIDE)

In order to characterize more specifically some of the obstacles commonly encountered by the receiver along a railway run, position errors models have been generated for two of the most common obstacles presents on railway lines:

- Bridge
- Tunnel

5.3.2.1. Regional Line Bridge

The first obstacle analysed is a bridge typical of railway regional lines. The illustration here relies on the use of a Ublox M8T GNSS receiver. It is a single frequency and multi-constellation GNSS receiver using SBAS corrections. It belongs to the category of the mass-market receivers since its cost is below the 100€. Data have been collected by GUIDE in the frame of a test campaign on the railway line between Rodez and Albi (south of France) on April 2017.



Figure 20: Railway bridge present on regional lines

Thanks to the use of the “R&R” methodology the errors profile for the along and cross track has been analysed for 10 replays.

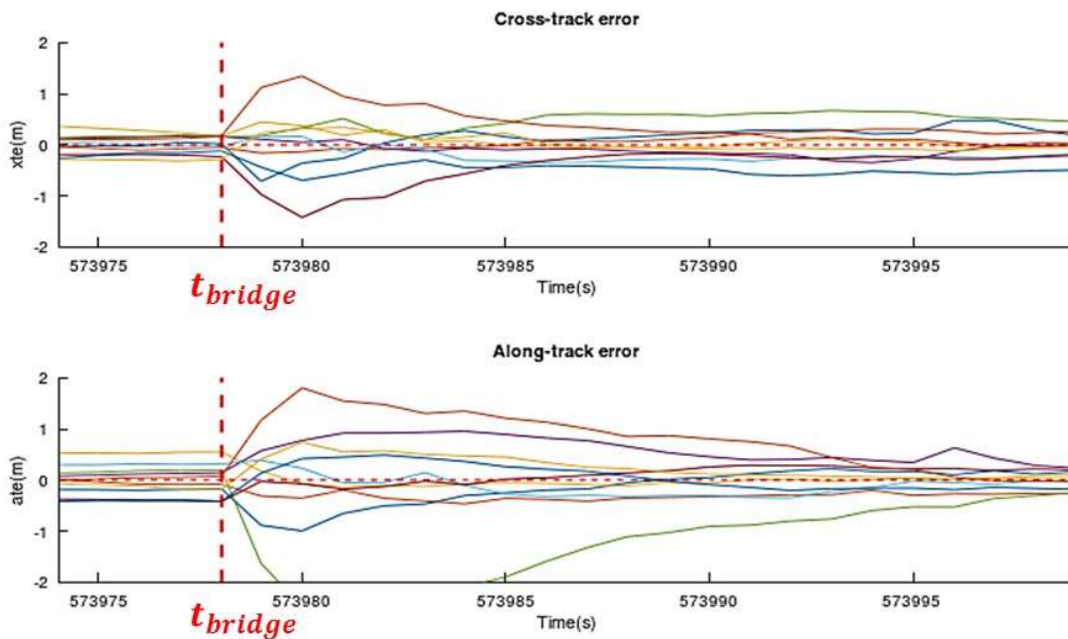


Figure 21: Bridge errors profile for along and cross track

The errors analysis shows a divergence of cross-track and along-track errors after the bridge crossing. This divergence can be represented by a peak occurring 1 second after the obstacle crossing. Representing the errors behaviour as a constant s , it can be treated as:

$$s(t) = \begin{cases} p, & t = t_{bridge} + 1 \\ \frac{p}{3}, & t = t_{bridge} + 2 ; \\ 0 & elsewhere \end{cases} \quad p = 7$$

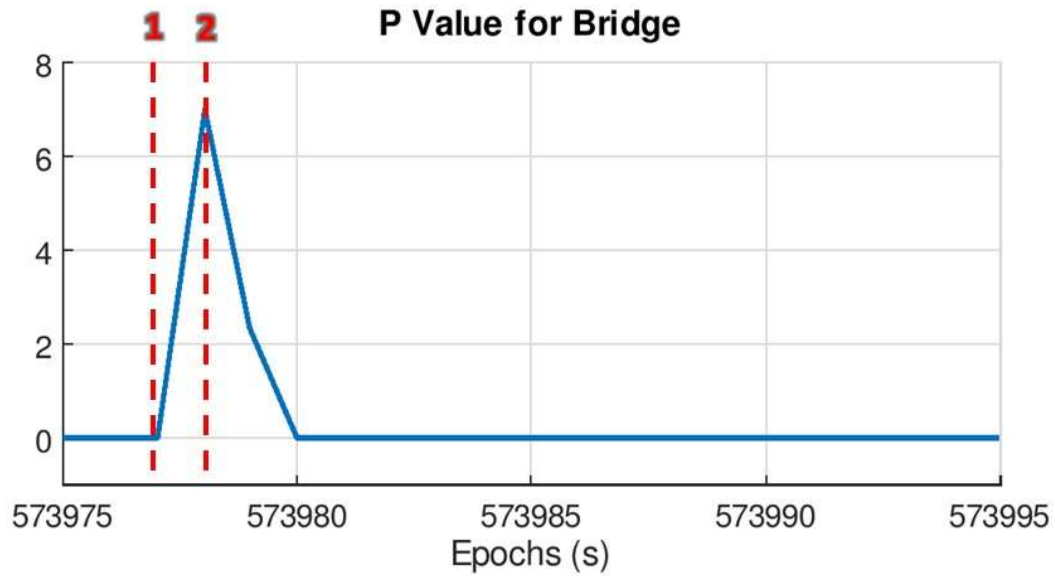


Figure 22: bridge error peak model

In Figure 22 “1” represents the bridge entry and “2” the exit.

Previous analysis, however, does not consider the fact that a longer bridge or a lower speed can cause the persistence of the divergence. In this case let $T_{bridge} = \left\lfloor \frac{D_{bridge}}{v_{bridge}} \right\rfloor$, with D_{bridge} the width of the bridge and v_{bridge} the speed of the rover under the bridge.

$$s(t) = \begin{cases} p, & t \in [t_{bridge} + 1; t_{bridge} + T_{bridge} + 1] \\ \frac{p}{3}, & t = t_{bridge} + T_{bridge} + 2 \\ 0 & elsewhere \end{cases} ; for T_{bridge} > 1$$

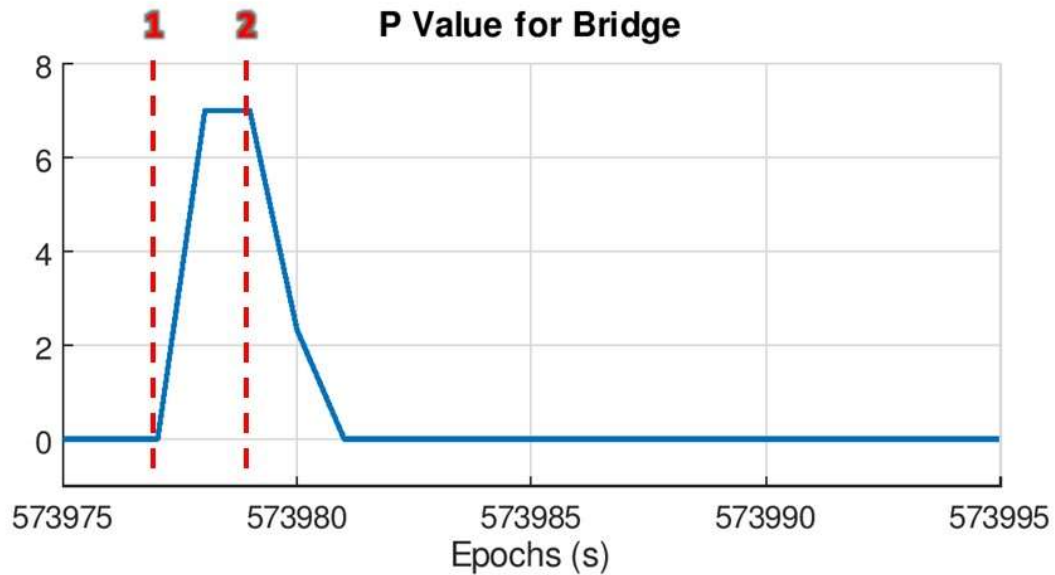


Figure 23: bridge error peak model with 2 seconds under bridge example

As for Figure 22, also in this case “1” represents the bridge entry and “2” the exit.

To match this model with errors analysed in the real case we can apply a moving average of previous signal $s(t)$, with a weight $\alpha = 0.2$. This value is valid, for instance, just for the M8T receiver. It can be used to represent GNSS receivers not using correction services (like NRTK or PPP). Model for NRTK, PPP, as well as for hybridized GNSS receivers shall be estimated.

$$\bar{s}(t) = \begin{cases} s(t_{bridge}), & t \leq t_{bridge} \\ \alpha \cdot s(t) + (1 - \alpha) \cdot \bar{s}(t - 1), & t > t_{bridge} \end{cases}$$

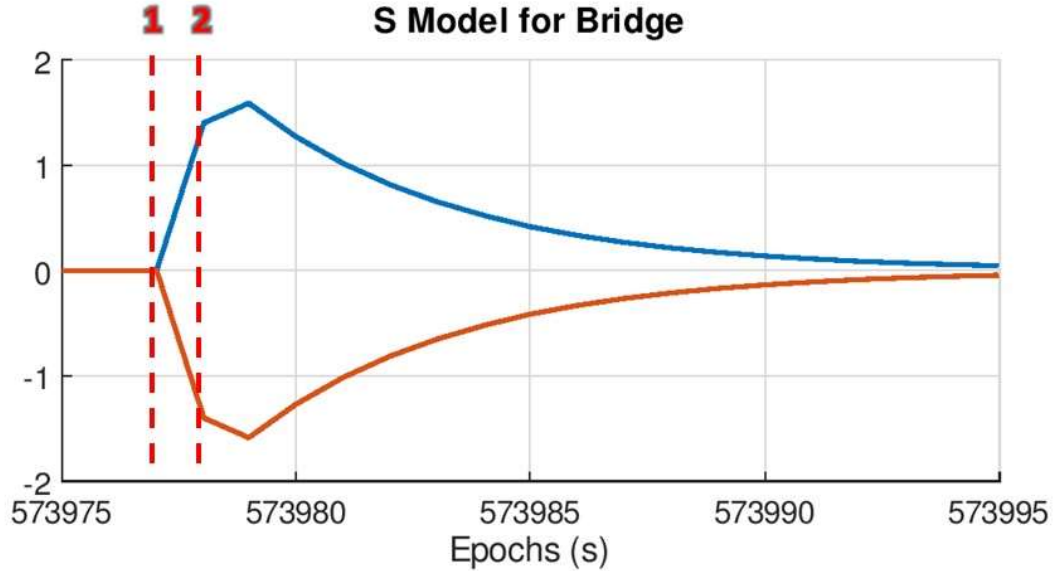


Figure 24: Bridge error model for positive/negative errors

Up to now we just considered the positive and negative extremes of the error model, by using a positive or negative value of p .

In order to represent all possible status inside the modelled diagram we can use a normal distribution using the following parameters.

$$\mathcal{N}\left(\mu = 0, \sigma = \frac{1}{2}\right)$$

The use of 0.5 as standard deviation allow to obtain values bigger than 1, or lower than -1, in order to simulate rare events giving bigger errors than ones seen in real data.

We can at each iteration estimate a value to apply to $\bar{s}(t)$ to obtain all possible curves inside the lobe.

$$\beta = \text{Random}(\mu; \sigma)$$

The new value of $\bar{s}(t)$ is :

$$\bar{s}_2(t) = \bar{s}(t) * \beta$$

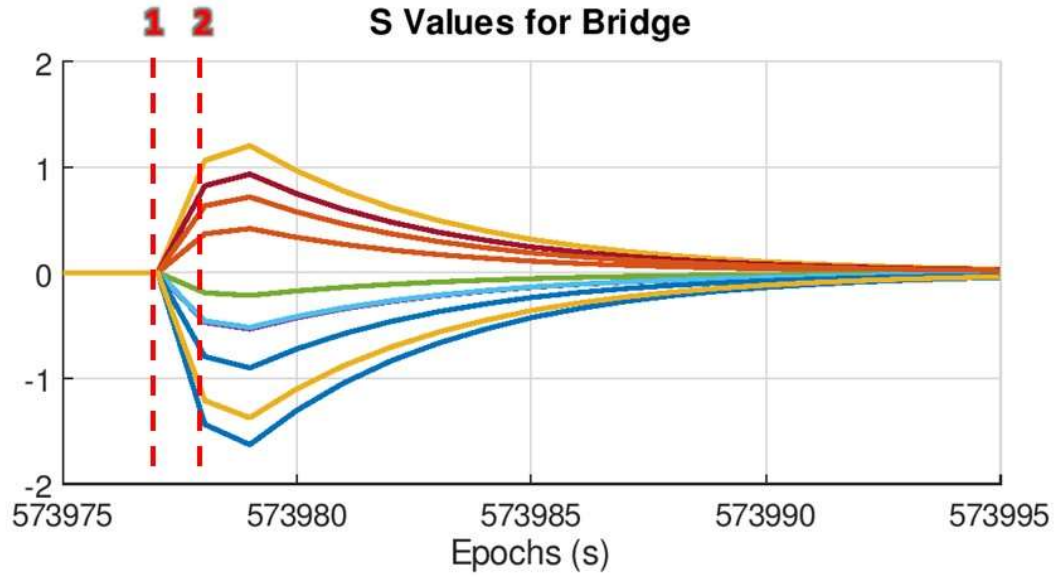


Figure 25: Bridge error model for all possible cases

Next figure shows a comparison between the analysed real errors and the ones generated using previous model. We add also a bit of dispersion to cope with effects introduced by global errors.

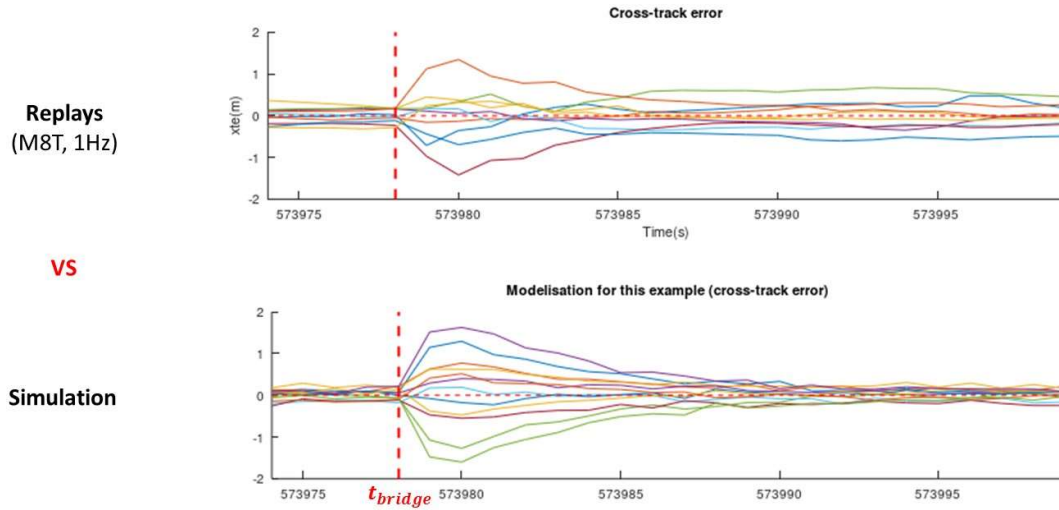


Figure 26: Comparison between real errors and modelled ones on cross-track

Next figure shows the behaviour of vertical errors for a bridge crossing.

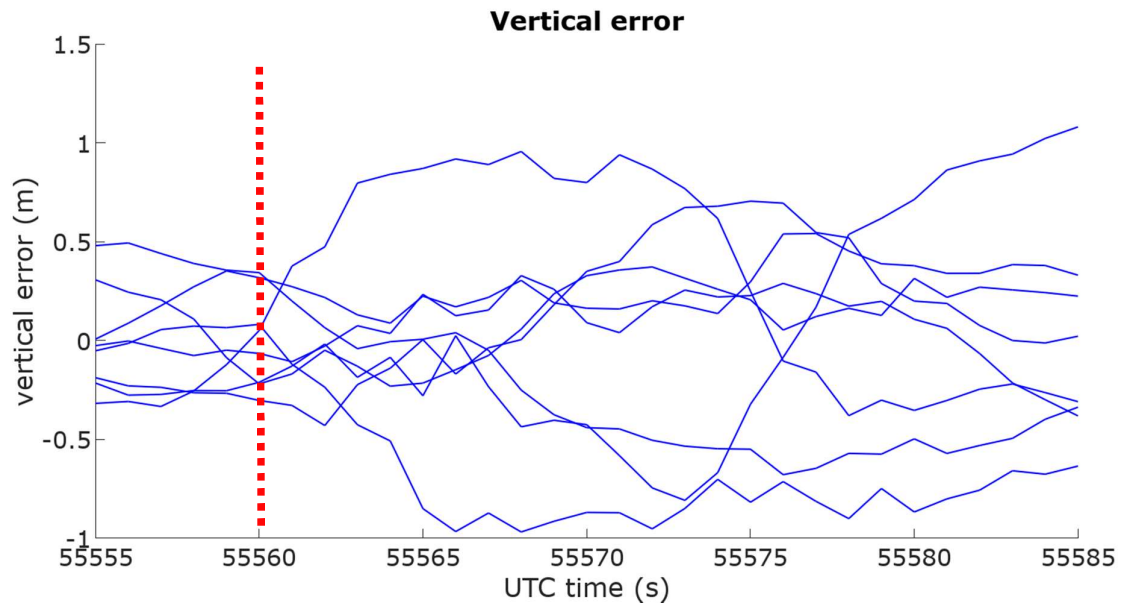


Figure 27: Vertical error analysis for bridge crossing.

As visible in Figure 27, the vertical errors are not impacted by the bridge presence. For this reason, and considering the impact of vertical errors for balise detection, they will not be modelled for bridges.

5.3.2.2. Single Track Tunnel

The second analysed obstacle is a single track tunnel typical of railway regional lines. In this case vertical errors are also considered since at tunnel exit the magnitude of these errors is not negligible. Data have been collected in the same data campaign, performed on April 2017, on the railway line between Rodez and Albi in the south of France.

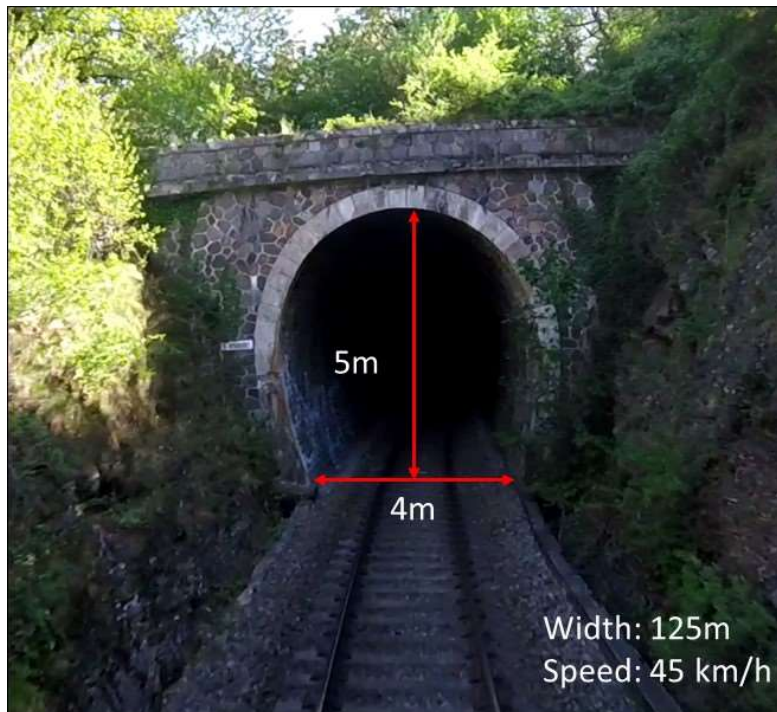


Figure 28: Railway tunnel present on regional lines

The analysis of behaviour of GNSS receiver in presence of a tunnel can be split in three sub-categories:

- The receiver loses the signal 1s after entering the tunnel.
- Without any signal, the uBlox receiver extrapolates future positions relying on the ones at $t_{i_{tunnel}}$ and $t_{i_{tunnel}} + 1s$.
- At tunnel exit positions calculated by the receiver converge rapidly to small errors.

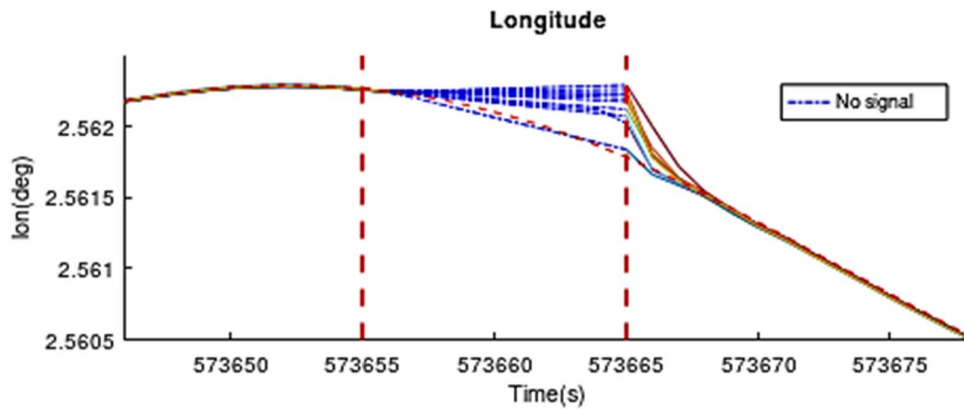


Figure 29: Longitude behaviour in presence of a tunnel

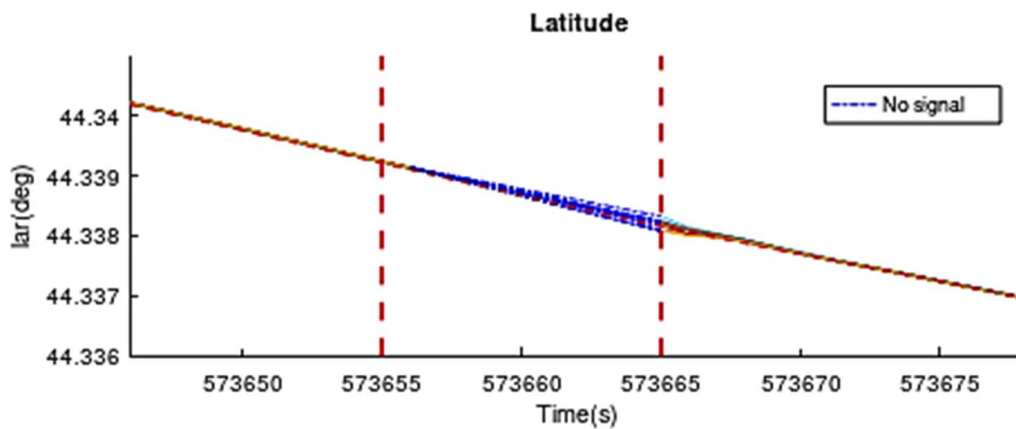


Figure 30: Latitude behaviour in presence of a tunnel

In order to have a better analysis the errors plot for the latitude component is given hereafter.

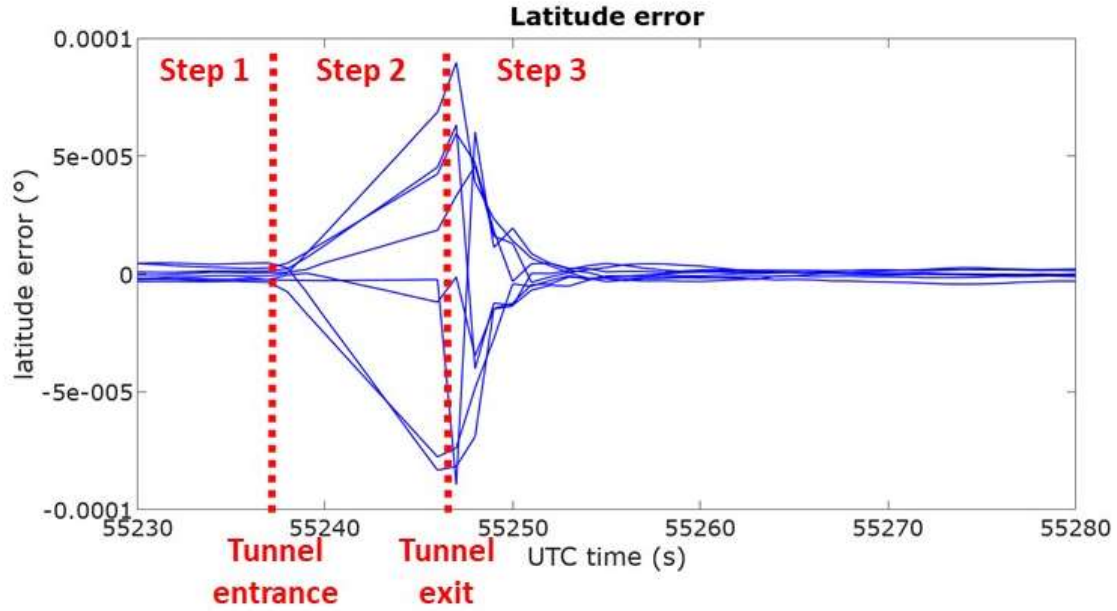


Figure 31: Latitude errors for tunnel pass

In **step 1** the GNSS receiver just behave as in the case of the bridge. Errors on positions increases for 1 seconds before GNSS signals are loss and PVT solutions are no longer provided.

The modes is the same seen in 5.3.2.1.

$$s(t) = \begin{cases} 0 & \text{for } t \leq t_{i_tunnel} \\ p & \text{for } t = t_{i_tunnel} + 1 \end{cases} ; \text{with } p = 7$$

Considering the use of a moving average model we obtain

$$\bar{s}(t) = \begin{cases} s(t_{i_tunnel}), & t \leq t_{i_tunnel}, \\ \alpha \cdot s(t) + (1 - \alpha) \cdot \bar{s}(t - 1), & t > t_{i_tunnel}, \end{cases} \quad \alpha = 0.2$$

For the vertical error a similar approach has been used. Nevertheless, value of p and α has been changed.

$$\begin{aligned} \bar{s}_v(t) &= s_v(t) & t \leq t_tunnel \\ \bar{s}_v(t_{tunnel} + 1) &= p_v + \bar{s}_v(t_tunnel) \\ p_v &= 0.15 \end{aligned}$$

In order to reproduce all possible cases inside the errors envelop a random value is applied to $\bar{s}(t)$

$$\begin{aligned} \beta &= \text{random}(\mu; \sigma); \text{with } \sigma = \frac{1}{3} \\ \bar{s}_2(t) &= \bar{s}(t) * \beta ; \bar{s}_{v_2}(t) = \bar{s}_v(t) * \beta \end{aligned}$$

For the **step 2** positions shall be extrapolated considering the last two valid PVT solutions estimated. This mean to use t_{tunnel} and $t_{tunnel} + 1$.

$$\begin{cases} \Delta_{lon} = lon(t_{itunnel} + 1) - lon(t_{itunnel}) \\ \Delta_{lat} = lat(t_{itunnel} + 1) - lat(t_{itunnel}) \\ \Delta_{alt} = alt(t_{itunnel} + 1) - alt(t_{itunnel}) \end{cases}$$

Noting the exit of the tunnel as $t_{f\ tunnel}$ positions at this point will be:

$$\begin{cases} long(t_{f\ tunnel}) = lon(t_{itunnel} + 1) + \Delta_{lon} * (t_{f\ tunnel} - t_{itunnel}) \\ lat(t_{f\ tunnel}) = lat(t_{itunnel} + 1) + \Delta_{lat} * (t_{f\ tunnel} - t_{itunnel}) \\ alt(t_{f\ tunnel}) = alt(t_{itunnel} + 1) + \Delta_{alt} * (t_{f\ tunnel} - t_{itunnel}) \end{cases}$$

The same behaviour can be reproduced considering errors introduced in step 1. In this case:

$$\begin{aligned} \Delta s &= \bar{s}_2(t_{itunnel} + 1) - \bar{s}_2(t_{itunnel}) \\ \Delta s_v &= \bar{s}_{v2}(t_{itunnel} + 1) - \bar{s}_{v2}(t_{itunnel}) \\ \bar{s}_2(t_{f\ tunnel}) &= \bar{s}_2(t_{itunnel} + 1) + \Delta s * (t_{f\ tunnel} - t_{itunnel}) \\ \bar{s}_{v2}(t_{f\ tunnel}) &= \bar{s}_{v2}(t_{itunnel} + 1) + \Delta s_v * (t_{f\ tunnel} - t_{itunnel}) \end{aligned}$$

In **step 3** a fast convergence of errors is observed. This last can be modelled as:

$$\bar{s}_2(t) = \gamma \cdot \bar{s}_2(t - 1) ; \forall t \geq t_{f\ tunnel} + 1$$

With $\gamma = 0.5$ or $\gamma_v = 0.8$

Step 3 is iterative and last until the estimated \bar{s}_2 is lower than 0.01 meters.

Once errors are estimated they can be converted in north and east components and then converted in degrees, to be integrated into the basic reference trajectory.

Present methodology provides the following errors.

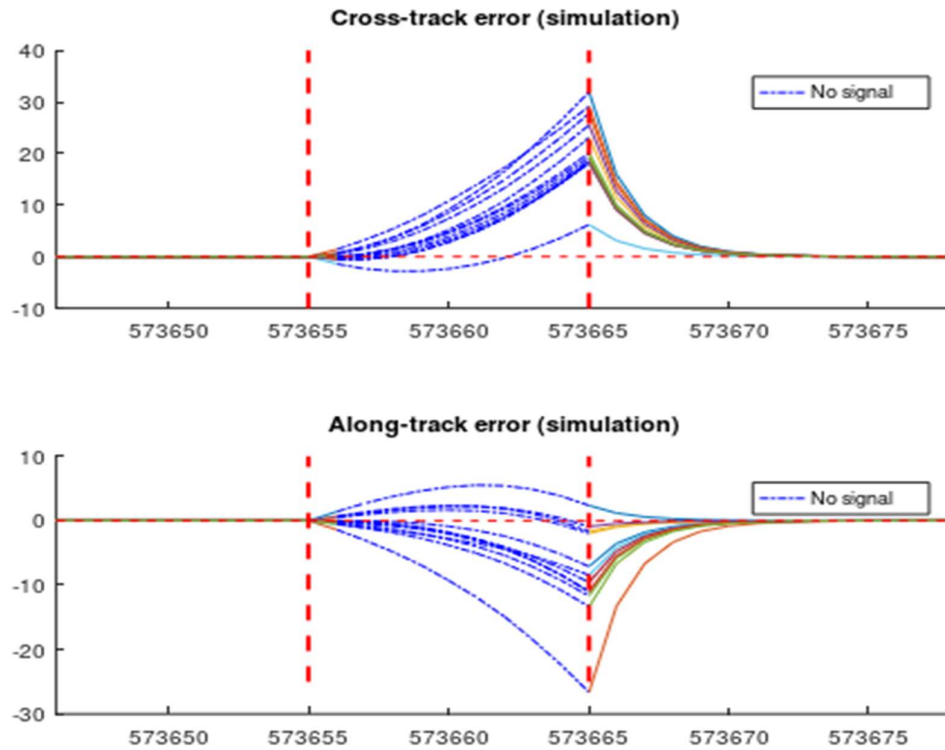


Figure 32: Simulated cross and along track errors in presence of a tunnel

5.3.2.3. Model Validation

Two previous models have been validated following the current methodology:

- Generation of errors on a rail line 10 times; two bridges and one tunnel are simulated and their impact added to ground truth
- Injection of 10 trajectories on a GNSS constellation simulator
- Comparison between injected trajectories and the ones obtained on GNSS receiver.
- Comparison with data obtained using digitized GNSS signals (R&R)

Next figures show the results obtained on a GNSS receiver.

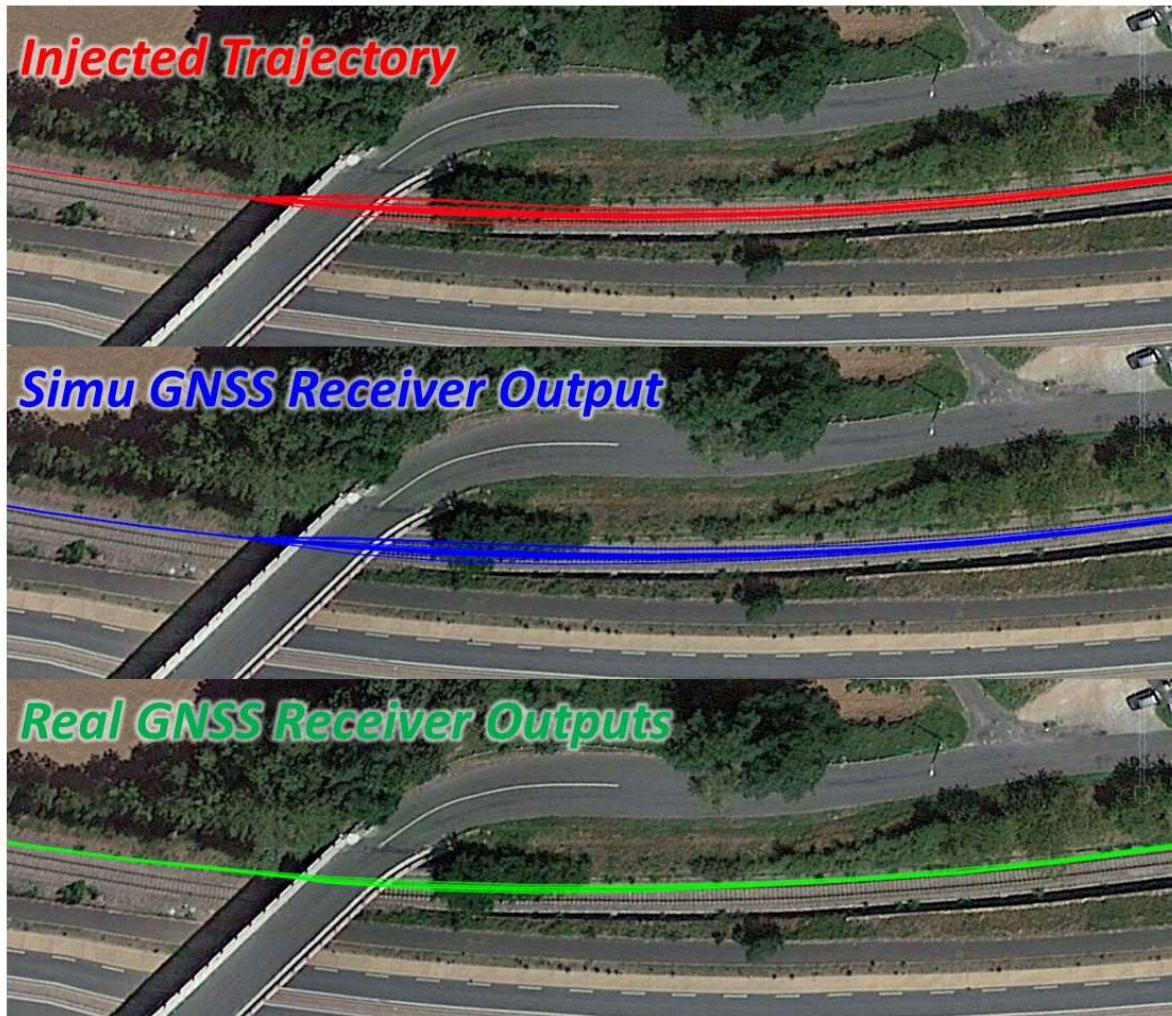


Figure 33: Comparison between simulated and real trajectories for a bridge crossing

Figure 33 shows how the error model developed for a bridge well represents the behaviour of a GNSS receiver used with real signals.

- The injected trajectories are well reproduced by the GNSS constellation simulator. They are well visible comparing red and blue lines.
- GNSS behaviour using real data is coherent with the developed model. The dispersion at bridge cross is less visible than on simulation since also before the bridge a small dispersion among trajectories is present (Phenomenon treated by UGE).

The model, also overestimates some of the error that it is possible to obtain.

In the case of the tunnel, the model also match very well the behaviour of the GNSS receiver observed on real data.

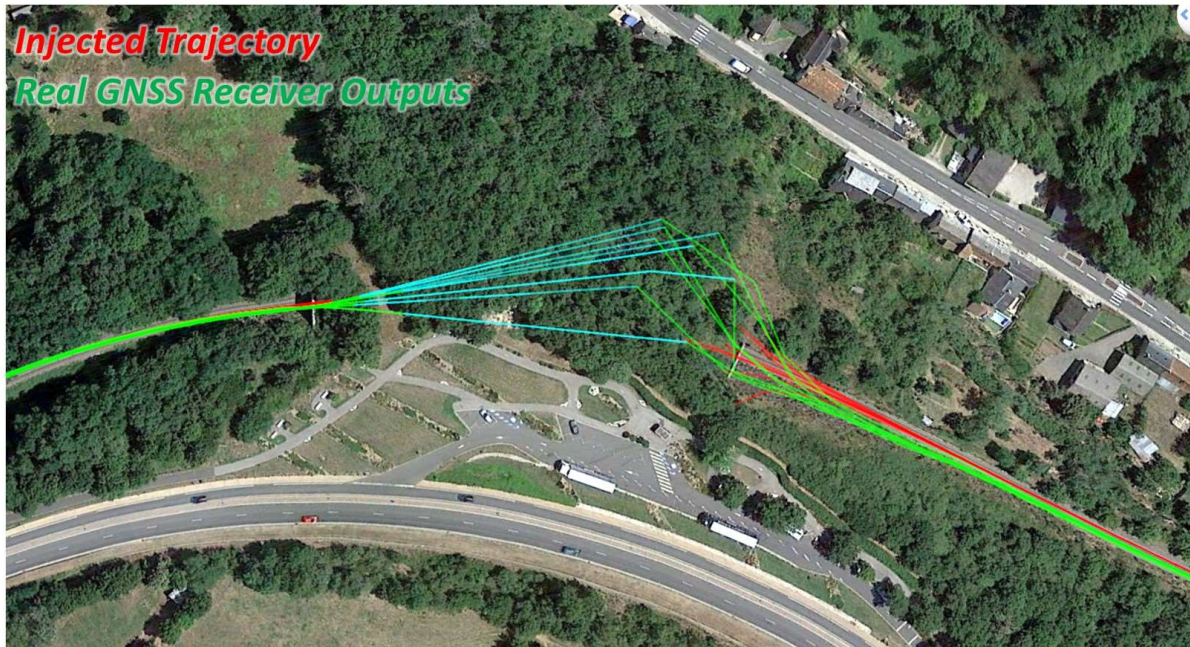


Figure 34: Comparison between simulated and real trajectories for a tunnel crossing

Also for the tunnel case the model developed matches quite well the behaviour of the used GNSS receiver. At the entrance and at the exit of the tunnel some discrepancies exist due to the different sensitivity that each GNSS receiver has in tracking weak GNSS signals.

The plot of simulation results have been not inserted since, for instance, GUIDE owned GNSS constellation simulator does not accept empty coordinates as the ones simulated inside the tunnel.

6. Conclusions and perspectives

This deliverable is the second report delivered by GATE4Rail WP3. The general objective of WP3 was to define a comprehensive methodology and the associated tools able to characterise the GNSS performance into the railways application scenarios defined in WP2 in the framework of a VB detection in ETCS/ERTMS context.

In particular, the developed methodology must allow to evaluate the effects of global and local GNSS faults on the properties of some Fail-Safe Train Positioning components.

Based on the state of the art, this report defines which effects of the railway environment have to be taken into account (local and global) and simulated in order to cover the main events that can occur during an operational scenario and how error models will be introduced in the simulation framework defined by the project

In absence of existing error models for local effects and in particular of the effects caused by masking or reflecting obstacles, two routines have been developed by the partners that aim to simulate them. This report details the process used for model generation. Two families of local errors have been selected and models: common errors observed in typical environments (forest, urban area, open sky...) and errors generated by local obstacles such as tunnels or bridges, causing partial or total loss of GNSS signals.

These two routines are complementary as they will apply at different points of the trajectory. Errors will be added along a reference track in order to generate local errors all along the track used as input for the chain.

Local and global errors will be simulated at different levels. At the end, the simulator will benefit of local effects generator, interference signals and global effects simulators played according to the tests scenarios defined in WP2.

The interfaces and the different modules will be applied in WP4 that deals with the development of the tool.

The models provided in the project rely on the existing databases, owned or accessible by the partners of the consortium. These databases are today not complete. They do not cover, in a statistical way, all the errors or environment types a train can crossed. However, they allow a proof of concept of the global GATE4Rail simulation framework.

The experience of this project allows us to highlight the following comments or recommendations for future work:

- The models studied in the current project are mostly based on the use of GNSS receivers providing 1 Hz output, without the use of corrections services (PPP or NRTK) and without any hybridization. Future works shall model at least these other two categories of GNSS receivers, hybridized and/or high precision terminals (PPP/NRTK) if investigated for future use.
- The queues of the errors for typical environments are small. So with the random process, the large errors that can occur and that are real feared events will rarely be seen in the simulation. In order to evaluate the behaviour of the system in harsh conditions, we can recommend to increase the probability of these events or injecting these large faults as a volunteer process.

Considering the GATE4Rail architecture and the available tools to be integrated in its framework, we have decided to model the errors in the position domain. This decision has limits: for example we cannot distinguish between the measurements which one to be impacted by an event, and thus we cannot assess the capacity of the solution to detect and exclude specifically such a measurement. For the future evolutions of the simulator, the introduction of error models at the measurement levels can be investigated.