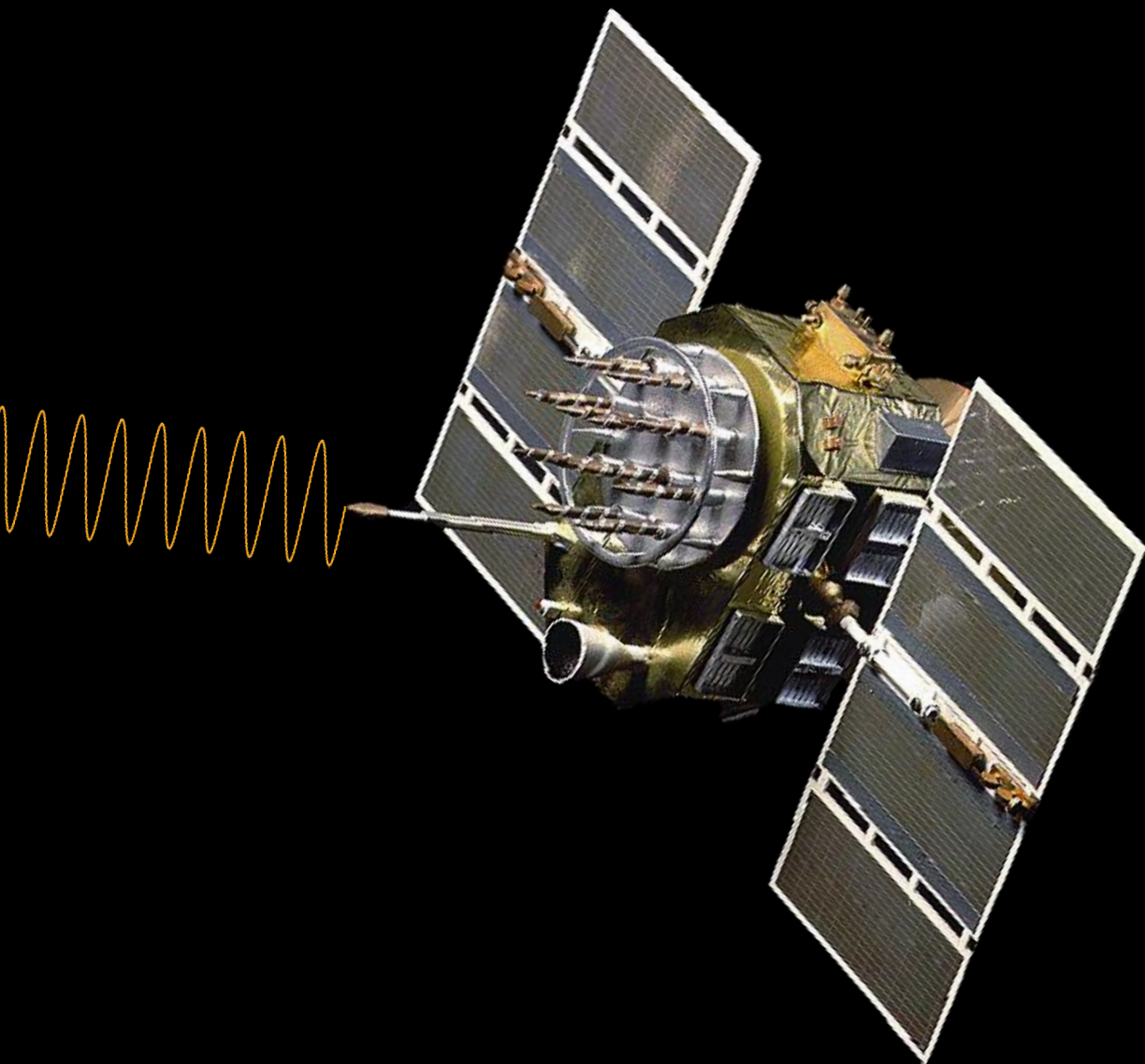


Ph.D. dissertation

**Contributions to multi-purpose GNSS
positioning to support multi-frequency
and multi-constellation with high
accuracy and integrity**

Deimos Ibáñez Segura



Earth image cover source: [iStock.com/Malekas85](https://www.iStock.com/Malekas85)
GPS Block IIA image cover source: [GPS.gov](https://www.gps.gov)

Ph.D. Thesis
Doctoral Program in Aerospace Science and Technology

**Contributions to multi-purpose GNSS positioning to support
multi-frequency and multi-constellation with high accuracy and
integrity**

Author:

Deimos Ibáñez Segura

Department of Mathematics and Physics

Universitat Politècnica de Catalunya

23rd of October 2023

Advisors:

Dr. Adrià Rovira Garcia

Prof. Dr. Jaume Sanz Subirana



Abstract

UNESCO codes: 1203.11, 2501.18, 2504.03, 2504.07

Global Navigation Satellite System (GNSS) have revolutionized location and timing technologies due to their low cost and wide availability. Initially, GNSS was limited to the American Global Positioning System (GPS) and the Russian GLObal NAVigation Satellite System (GLONASS). However, in recent years, new GNSS systems such as European Global Navigation Satellite System (Galileo), the Chinese BeiDou Navigation Satellite System (BDS), the Japanese Quasi-Zenith Satellite System (QZSS), and Indian Regional Navigation Satellite System (IRNSS) have been deployed. These systems offer multiple frequencies and signals, which have rendered the current processing of dual-frequency single-constellation GNSS data obsolete.

To address this issue, the Research group of Astronomy and Geomatics (gAGE) in Technical University of Catalonia (UPC) developed the GNSS-Lab Tool suite (gLAB) reference tool. Initially, gLAB was designed to support GPS only under a contract with the European Space Agency (ESA) and was later upgraded to become a reference tool for scientific studies. gLAB has demonstrated its research capabilities under ESA and European Union Agency for the Space Programme (EUSPA) contracts.

The gLAB tool has had a great impact on the GNSS community across the globe, with 222,526 downloads from 147 different countries, which demonstrates the demand for a tool with these capabilities. The positioning algorithms originally coded in gLAB, Standard Point Positioning (SPP) and Precise Point Positioning (PPP), are upgraded to work with all new signals. In addition, advanced algorithms have been implemented, including uncombined PPP, Fast Precise Point Positioning (Fast-PPP), and Satellite-Based Augmentation System (SBAS) Dual Frequency Multi Constellation (DFMC). These additional capabilities required optimizing the code and implementing multi-threading to reduce processing times.

The research carried out in this PhD has made several significant contributions.

IV

Firstly, an evaluation of European Geostationary Navigation Overlay System (EGNOS), initially developed for civil aviation, was carried out for its application for maritime purposes, proving that EGNOS is valid for this use, with even improved continuity figures. In addition, a methodology for the daily evaluation of the EGNOS Signal In Space (SIS) has been developed and implemented in a Global Monitoring System (GMS) based on gLAB.

Secondly, the impact of ionospheric activity and its correct modelling on navigation, especially in SBAS systems, has been analysed. In this area, the importance of having a global index, the Along Arc TEC Rate (AATR), to measure ionospheric activity has been shown and its correlation with the degradation of the quality of service of EGNOS has been analysed.

Thirdly, the effect of positioning anomalies (both in the SIS and in the receiver) has been studied, proposing methodologies for their detection based on pseudorange residuals or by combining several signals. In any of these methods, a high degree of data pre-processing is required, for which gLAB is a very suitable tool.

Fourthly, a method for measurement management in gLAB has been developed. It comprises complex algorithmics that allows any combination of observables (single, dual, triple, or quadruple combinations) on a per-satellite basis, as well as to provide either manual or automatic measurement selection (and extend it for cycle-slip detectors).

In conclusion, the present PhD has paved the way to modern GNSS data processing beyond the state of the art.

Resumen

Códigos UNESCO: 1203.11, 2501.18, 2504.03, 2504.07

El Sistema Global de Navegación por Satélite (GNSS) ha revolucionado las tecnologías de localización gracias a su bajo coste y amplia disponibilidad. Inicialmente, los GNSS se limitaban al Sistema de Posicionamiento Global (GPS) estadounidense y al Sistema Global de Navegación por Satélite (GLONASS) ruso, pero en los últimos años se han desplegado nuevos GNSS como el Sistema Europeo de Navegación Global por Satélite (Galileo), el Sistema de Navegación por Satélite BeiDou (BDS) chino, el Sistema de Satélites Cuasi Centrales (QZSS) japonés y el Sistema Regional Indio de Navegación por Satélite (IRNSS). Estos sistemas ofrecen múltiples frecuencias y señales que han dejado obsoleto el procesamiento actual de datos GNSS de doble frecuencia y una constelación.

Para abordar este problema, el grupo de Investigación en Astronomía y Geomática (gAGE) de la Universidad Politécnica de Cataluña (UPC) desarrolló la herramienta GNSS-Lab Tool suite (gLAB). gLAB se diseñó inicialmente para soportar sólo GPS en el marco de un contrato con la Agencia Espacial Europea (ESA) y posteriormente se actualizó para convertirse en una herramienta de referencia para estudios científicos. gLAB ha demostrado sus capacidades de investigación en el marco de contratos de la ESA y de la Agencia de la Unión Europea para el Programa Espacial (EUSPA).

gLAB tiene un gran impacto en la comunidad GNSS de todo el mundo, con 222.526 descargas en 147 países diferentes, lo que demuestra la demanda de una herramienta con estas capacidades. En cuanto a los algoritmos de posicionamiento codificados originalmente en gLAB, Posicionamiento de Punto Estándar (SPP) y Posicionamiento de Punto Preciso (PPP), éstos se han actualizado para trabajar con todas las nuevas señales. Además, se han implementado algoritmos avanzados, como el PPP no combinado, el PPP Rápido (Fast-PPP) y el Sistema de Aumentación Basado en Satélites (SBAS) Multi Constelación de Frecuencia Dual (DFMC). Estas capacidades adicionales han requerido la optimización del código y la implementación de

multihilo para reducir el tiempo de procesado.

La investigación realizada en este doctorado ha aportado varias contribuciones significativas. En primer lugar, se ha realizado una evaluación del Sistema Europeo de Navegación por Complemento Geoestacionario (EGNOS), inicialmente desarrollado para aviación civil, para su aplicación con fines marítimos, demostrando que EGNOS es válido para este uso, donde se mejoran, incluso, las figuras de continuidad. Asimismo, se ha desarrollado una metodología para la evaluación diaria de la Señal En El Espacio (SIS) de EGNOS, la cual se ha implementado en un Sistema de Monitorización Global (GMS) basado en gLAB.

En segundo lugar, se ha analizado el impacto de la actividad ionosférica y su correcto modelado en la navegación, especialmente en sistemas SBAS. En este ámbito se ha mostrado la importancia de tener un índice global, la Variación del contenido electrónico total en un arco (AATR), para medir la actividad ionosférica y se ha analizado su correlación con la degradación de la calidad de servicio del sistema EGNOS.

En tercer lugar, se ha estudiado el efecto de las anomalías en el posicionamiento (tanto de la SIS como en el receptor), proponiendo metodologías para su detección basadas en los residuos de los pseudorángos o mediante combinación de varias señales. Cualquiera de estos métodos, requiere un alto grado de preprocesado de los datos, para lo que gLAB es una herramienta muy adecuada.

En cuarto lugar, se ha desarrollado un método para la gestión de medidas en gLAB, Se trata de una algorítmica compleja, que permite procesar cualquier combinación de observables (simples, dobles, triples o cuádruples) por satélite, así como proporcionar una selección de medidas manual o automática (también para los detectores de salto de ciclo).

En conclusión, la presente tesis doctoral ha allanado el camino hacia el procesamiento moderno de datos GNSS más allá del estado del arte.

Resum

Codis UNESCO: 1203.11, 2501.18, 2504.03, 2504.07

El Sistema Global de Navegació per Satèl·lit (GNSS) ha revolucionat les tecnologies de localització gràcies al seu baix cost i àmplia disponibilitat. Inicialment, els GNSS es limitaven al Sistema de Posicionament Global (GPS) americà i al Sistema Global de Navegació per Satèl·lit (GLONASS) rus, però en els últims anys s'han desplegat nous GNSS com el Sistema Europeu de Navegació Global per Satèl·lit (Galileo), el Sistema de Navegació per Satèl·lit BeiDou (BDS) xinès, el Sistema de Satèl·lits Quasi Zenital (QZSS) japonès i el Sistema Regional Indi de Navegació per Satèl·lit (IRNSS). Aquests sistemes ofereixen múltiples freqüències i senyals que han deixat obsolet el processament actual de dades GNSS de doble freqüència i una constel·lació.

Per a abordar aquest problema, el grup de Recerca en Astronomia i Geomàtica (gAGE) de la Universitat Politècnica de Catalunya (UPC) va desenvolupar l'eina GNSS-Lab Tool suite (gLAB). gLAB es va dissenyar inicialment per a suportar només GPS en el marc d'un contracte amb l'Agència Espacial Europea (ESA) i posteriorment es va actualitzar per a convertir-se en una eina de referència per a estudis científics. gLAB ha demostrat les seves capacitats de recerca en el marc de contractes de l'ESA i de l'Agència de la Unió Europea per al Programa Espacial (EUSPA).

gLAB té un gran impacte en la comunitat GNSS de tot el món, amb 222.526 descàrregues en 147 països diferents, el qual demostra la demanda d'una eina amb aquestes capacitats. En quant als algorismes de posicionament codificats originalment en gLAB, Posicionament de Punt Estàndard (SPP) i Posicionament de Punt Precís (PPP), aquests han estat actualitzats per a treballar amb totes les noves senyals. A més a més, es van implementar algorismes avançats, com el PPP no combinat, el PPP Ràpid (Fast-PPP) i el Sistema de Augmentació Basat en Satèl·lits (SBAS) Multi Constel·lació de Freqüència Dual (DFMC). Aquestes capacitats addicionals han requerit l'optimització del codi i la implementació de multifil per a reduir el temps de processament.

VIII

La recerca realitzada en aquest doctorat ha aportat diverses contribucions significatives. En primer lloc, s'ha realitzat una avaluació del Sistema Europeu de Navegació per Complement Geoestacionari (EGNOS), inicialment desenvolupat per a aviació civil, per a la seva aplicació amb finalitats marítimes, demostrant que EGNOS és vàlid per a aquest ús, on es milloren, fins i tot, les figures de continuïtat. Així mateix, s'ha desenvolupat una metodologia per a l'avaluació diària del Senyal a l'Espai (SIS) de EGNOS, la qual s'ha implementat en un Sistema de Monitoratge Global (GMS) basat en gLAB.

En segon lloc, s'ha analitzat l'impacte de l'activitat ionosfèrica i el seu correcte modelatge en la navegació, especialment en sistemes SBAS. En aquest àmbit s'ha mostrat la importància de tenir un índex global, la Variació del contingut electrònic total en un arc (AATR), per a mesurar l'activitat ionosfèrica i s'ha analitzat la seva correlació amb la degradació de la qualitat de servei del sistema EGNOS.

En tercer lloc, s'ha estudiat l'efecte de les anomalies en el posicionament (tant de la SIS com en el receptor), proposant metodologies per a la seva detecció basades en els residus dels pseudorangs o mitjançant combinacions de diversos senyals. Qualsevol d'aquests mètodes, requereix un alt grau de preprocessament de les dades, per al que gLAB és una eina molt adequada.

En quart lloc, s'ha desenvolupat un mètode per a la gestió de les mesures en gLAB. Es tracta d'una algorítmica complexa, que permet processar qualsevol combinació d'observables (simples, dobles, triples o quàdruples) per satèl·lit, així com proporcionar una selecció de mesures manual o automàtica (també per als detectors de salt de cicle).

En conclusió, la present tesi doctoral ha aplanat el camí cap al processament modern de dades GNSS més enllà de l'estat de l'art.

Acknowledgements

I would like first to thank my thesis directors, Prof. Dr. Jaume Sanz Subirana and Dr. Adrià Rovira-García for persuading me to carry out this PhD thesis, as well as for their constant support, even in out-of-office hours or during weekends. They always kept me pushing forward for excellence, even when my methodology seemed too complex or costly (in time) and results were uncertain (specially when developing the gLAB measurement management and source code optimization). Without them, the tool would not be as good as it now is.

I also wish to thank my colleagues from gAGE, Prof. Dr. José Miguel Juan Zornoza, Dr. Cristhian Timoté, Dr. María Teresa Alonso, Dr. María Caamaño Albuérne and Mowen Li for their support and compliments regarding my work.

I would also like to thank Carlos López de Echazarreta for believing from the start in the gLAB tool and its potential, and for believing in gAGE in how to best develop the tool.

As around 70% of the time spent in this thesis has been working full time in GNSS related jobs (but aside from my thesis), I would like to thank my work colleagues, in both my former job in Toulouse and in my current job in Madrid, for the pleasant working environment. It is a pleasure to work with you.

I would also like to thank the gLAB users, especially those who have provided positive feedback. You have proved that the hard work involved in developing and improving the gLAB tool made sense.

I want to thank my friends for being on my side and understanding the importance of the PhD thesis for me.

Last but not least, I would like to thank my family for supporting me throughout this journey. I could not have done it without them.

Contents

Abstract	III
Resumen	V
Resum	VII
Acknowledgements	IX
List of Figures	XVII
List of Tables	XXI
Acronyms	XXIII
1 Thesis introduction	1
1.1 Purpose of the thesis	1
1.2 Thesis context	3
1.3 Thesis outline	3
1.4 Thesis publications	4
1.5 Standard development	6
2 GNSS Introduction	7

3	gLAB description	13
3.1	Main gLAB features	14
3.2	Graphical User Interface	16
3.2.1	Positioning mode	19
3.2.2	Analysis mode	20
3.2.3	Converter mode	21
3.2.4	Compare Orbits & Clocks mode	22
3.2.5	Show Orbits & Clocks mode	23
3.2.6	Add User Error to RINEX	24
3.2.7	GUI limitations	25
3.3	Data Processing Core	26
3.4	gLAB positioning example	28
3.5	Contributions to gLAB code	33
4	gLAB state-of-the-art algorithms	35
4.1	Standard Point Positioning	35
4.2	Precise Point Positioning	40
4.3	Satellite-Based Augmentation System single frequency	46
4.4	Differential Global Navigation Satellite System	57
5	Advanced gLAB algorithms	59
5.1	Uncombined Precise Point Positioning	59
5.2	Fast Precise Point Positioning	64
5.3	Satellite-Based Augmentation System Dual Frequency Multi Constellation	67
6	Challenges in gLAB upgrade	71
6.1	Measurement selection	71

6.1.1	Selection methodology	73
6.2	Cycle-slip detection	78
6.3	Output prints	80
6.4	Multi day SBAS availability plots	82
6.5	Optimization	84
6.6	Multi-threading	86
6.7	Receiver Antenna Phase Centre correction	88
7	Scientific studies with gLAB	91
7.1	EGNOS for maritime	91
7.1.1	Maritime requirements	91
7.1.2	EGNOS for maritime performance study	94
7.2	Global Monitoring System	100
7.3	AATR and SBAS	103
7.3.1	AATR definition	103
7.3.2	AATR application	104
7.3.3	Planetary index AATR	105
7.4	Ionosphere correction assessment	108
7.4.1	Introduction	108
7.4.2	Assessing ionosphere corrections	108
7.5	NeQuick-G integration into gLAB	112
7.5.1	Introduction	112
7.5.2	Integration in gLAB	112
7.6	“Prefit” outlier detection	116
7.7	GPS signal in space anomalies	118
7.7.1	Introduction	118

7.7.2	Methodology and data set	118
7.7.3	Results	119
8	Impact of gLAB tool	123
9	Conclusion	129
10	Future Work	133
A	gLAB source code optimizations	135
A.1	String generation for printing	135
A.2	Avoiding multiple recomputations of the same elements	136
A.3	Precise orbit and clocks management	137
A.4	Convert <i>if</i> chains to <i>switch</i> statements	138
A.5	Join multiple <i>if</i> conditions in a single <i>switch</i> statement	139
A.6	Memory allocation	141
A.7	Compiler flags	142
A.8	Line reading optimization	143
A.9	Micro-optimizations	148
A.9.1	Inlining functions	149
A.9.2	<i>Register</i> keyword	149
A.9.3	<i>Restrict</i> keyword	150
A.9.4	<i>Const</i> keyword	151
A.9.5	Variable declaration in <i>for</i> loop definition	152
A.9.6	Variable declaration inside a loop	153
A.10	Other possible optimizations	153
B	gLAB downloads per country	155

CONTENTS	XV
C Multi-day SBAS availability maps computation	161
Bibliography	189
Index	203

List of Figures

2.1	GPS, GLONASS, Galileo and BDS satellites	8
2.2	GNSS architecture.	9
2.3	GNSS frequency bands	10
3.1	gLAB modules	14
3.2	gLAB GUI initial screen	16
3.3	gLAB GUI help menu	19
3.4	gLAB GUI initial Positioning tab	20
3.5	gLAB GUI Analysis mode in SBAS mode	21
3.6	gLAB GUI Converter mode for SBAS files	22
3.7	gLAB GUI Compare Orbits & Clocks mode	23
3.8	gLAB GUI Show Orbits & Clocks mode	24
3.9	gLAB GUI Add User Error to RINEX mode	25
3.10	gLAB DPC “Show Input Mode” example	27
3.11	gLAB Input tab configured to perform SPP	28
3.12	gLAB Preprocess tab configured to perform SPP	29
3.13	gLAB Modelling tab configured to perform SPP	30
3.14	gLAB Filter tab configured to perform SPP	30
3.15	gLAB Output tab configured to perform SPP	31

3.16	NEU positioning error	32
4.1	Pseudorange measurement contents	36
4.2	SPP example GPS only and with all constellations	38
4.3	Layout of ARP, APC and MM	41
4.4	PPP example GPS only and with all constellations	44
4.5	SBAS service areas	49
4.6	EGNOS architecture	51
4.7	NSE and confidence bounds	52
4.8	Stanford plot example	53
4.9	1 Hz GPS only SPP and EGNOS example	54
4.10	Continuity risk with sliding window computation example	55
4.11	WAAS and EGNOS availability map	56
4.12	WAAS and EGNOS continuity risk map	56
4.13	GPS only SPP and DGNSS positioning	58
5.1	PPP and uncombined PPP example with all constellations	62
5.2	Kinematic PPP and Fast-PPP horizontal and vertical components	65
5.3	1 Hz GPS and Galileo SPP and SBAS DFMC example	68
5.4	Australia SBAS DFMC Availability map with GEO footprint on 02/07/2019 at 6 a.m.	69
5.5	Australia SBAS DFMC Availability map without GEO footprint on 02/07/2019 at 6 a.m.	70
6.1	Minimum noise IGF example	79
6.2	gLAB internal filter configuration print for multi-constellation processing in PPP	80
6.3	gLAB EPOCHSAT message PPP example	81
6.4	gLAB MEAS message example	81

6.5	gLAB processing times comparison with versions 5.5.1 and 6.0.0	85
6.6	Static PPP with 5 constellations with and without manual receiver APC correction	89
6.7	Static PPP with GPS and Galileo (left) or BDS (right) without manual receiver APC correction	90
7.1	Continuity risk with fixed window computation example	93
7.2	EGNOS maritime availability maps	95
7.3	EGNOS maritime continuity maps	96
7.4	gLAB and ESSP continuity maps without a Dilution Of Precision (DOP) mask and fixed window	98
7.5	Station map used in EGNOS maritime study	98
7.6	GMS website initial screen	101
7.7	Horizontal and vertical 95 th percentile PE maps on 18/03/2017	101
7.8	Horizontal and vertical WIR maps on 18/03/2017	102
7.9	Daily maximum AATR value vs. EGNOS APV-I availability . . .	105
7.10	AATR and SBAS availability	107
7.11	3D positioning error frequency percentage using code and CCL measurements	111
7.12	NeQuick-G and Klobuchar STEC and 3D positioning comparison CDF	115
7.13	Prefits and 3D positioning error on 4 th November 2018	117
7.14	Clock oscillation and clock pulse detections	120
7.15	IGS precise clock jump and cycle-slips	121
8.1	gLAB number of downloads per version	124
8.2	gLAB number of downloads per continent	125
8.3	Top ten gLAB number of downloads per country	126

List of Tables

3.1	Number of lines and documentation pages in gLAB	33
4.1	SPP 95 percentile comparison of errors and DOPs between GPS only and all constellations	39
4.2	Convergence time comparison in static and kinematic PPP	45
4.3	PPP 95 percentile comparison of errors and DOPs between GPS only and five constellations	46
4.4	Performance requirements	51
5.1	Convergence time comparison for kinematic PPP and Fast-PPP	66
6.1	RINEX version 3.05 GPS Observation Codes	72
6.2	gLAB default frequency priority list	74
6.3	gLAB default measurement attribute priority list	75
6.4	gLAB argument breakdown for measurement selection	76
6.5	gLAB single and multi-thread SBAS availability maps processing times	87
7.1	IMO A.1046(27) requirements	92
7.2	Maritime static positioning aggregation table	99
8.1	gAGE/UPC GNSS master courses with gLAB	127
8.2	gAGE/UPC professional training courses with gLAB	127

B.1 Full list of gLAB number of downloads per country 155

Acronyms list

AATR	Along Arc TEC Rate
ABAS	Aircraft-Based Augmentation System
AL	Alarm Limit
ANGA	Augmented NaviGation for Africa
ANTEX	ANTenna EXchange format
APC	Antenna Phase Centre
API	Application Programming Interface
APV	APproach operations with Vertical guidance
ARAIM	Advanced Receiver Autonomous Integrity Monitoring
ARM	Advanced RISC Machine
ARNS	Aeronautical Radio Navigation Service
ARP	Antenna Reference Point
AS	AntiSpoofing
AWGN	Additive White Gaussian Noise
Bash	Bourne Again SHell
BDGIM	BeiDou Global Ionospheric delay correction Model
BDS	BeiDou Navigation Satellite System
BDSK	BDS Klobuchar model
BDSBAS	BeiDou Satellite-Based Augmentation System
BIGF	British Isles continuous GNSS Facility
BM	BenchMark
C/A	Coarse/Acquisition
CCIR	Consultative Committee on International Radio
CCL	Carrier phase to Code Levelling
CCN	Contract Change Notification
CDDIS	Crustal Dynamics Data Information System

CDMA	Code Division Multiple Access
CDF	Cumulative Distribution Function
CLK	Clock
CNAV1	Civil NAVigation message 1
CNES	Centre National d'Études Spatiales
CODE	Centre for Orbit Determination in Europe
CoM	Centre of Mass
CONUS	CONTinental United States
CPF	Central Processing Facility
CPU	Central Processing Unit
CSNO	China Satellite Navigation Office
CTI	Continuity Time Interval
DAT	Data Analysis Tool
DCB	Differential Code Bias
DF	Dual Frequency
DF	Divergence Free
DFMC	Dual Frequency Multi Constellation
DGNSS	Differential Global Navigation Satellite System
DOP	Dilution Of Precision
DoY	Day of Year
DPC++	Data Parallel C++
DPC	Data Processing Core
EC	European Commission
ECAC	European Civil Aviation Conference
EDAS	EGNOS Data Access Service
EGNOS	European Geostationary Navigation Overlay System
EMS	EGNOS Message Service
ENAC	National School of Civil Aviation
EO	Earth Observation
EOF	End Of File
EP	European Patent
EPN	EUREF Permanent Network
EPO	EGNOS Project Office
ESA	European Space Agency
ESSP	European Satellite Services Provider

ESTEC	European Space Research and Technology Centre
EU	European Union
EUPL	European Union Public License
EUREF	European Reference Organisation for Quality Assured Breast Screening and Diagnostic Services
EUROCAE	European Organisation for Civil Aviation Equipment
EUROCONTROL	European Organisation for the Safety of Air Navigation
EUSPA	European Union Agency for the Space Programme
FAA	Federal Aviation Administration
FAQ	Frequently Asked Questions
FCC	Federal Communications Commission
FDMA	Frequency Division Multiple Access
FDSA	Flexible Statistics and Data Analysis
FIR	Flight Information Region
F/NAV	Freely accessible NAVigation message
Fast-PPP	Fast Precise Point Positioning
FOC	Full Operational Capability
FORTTRAN	FORmula TRANslator
FS	Forward Search
FSDA	Flexible Statistics and Data Analysis
GAGAN	GPS and GEO Augmented Navigation
gAGE	Research group of Astronomy and Geomatics
Galileo	European Global Navigation Satellite System
GAWK	GNU AWK
GB	GigaByte
GBAS	Ground-Based Augmentation System
GCC	GNU Compiler Collection
GDOP	Geometric Dilution Of Precision
GEO	Geosynchronous Equatorial Orbit
GF	Geometry Free
GIPSY	GPS Inferred Positioning SYstem
GIPSY-OASIS	GPS Inferred Positioning SYstem and Orbit Analysis and Simulation Software
gLAB	GNSS-Lab Tool suite
GLONASS	GLObal NAVigation Satellite System
GMS	Global Monitoring System

GNSS	Global Navigation Satellite System
GNU	GNU's Not Unix
GPL	GNU General Public License
GPS	Global Positioning System
GSA	Galileo Supervisory Authority
GUI	Graphical User Interface
HA	High Availability
HAL	Horizontal Alarm Limit
HAS	High Accuracy Service
HDOP	Horizontal Dilution Of Precision
HEO	Highly Elliptical Orbit
HF	High Frequency
HMI	Hazardous Misleading Information
HPE	Horizontal Positioning Error
HPL	Horizontal Protection Level
IAC	Information and Analysis Centre
IALA	International Association of Lighthouse Authorities
IAR	Integer Ambiguity Resolution
ICAO	International Civil Aviation Organization
ICD	Interface Control Document
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IERS	International Earth Rotation and Reference Systems Service
IF	Ionosphere Free
IGF	Ionosphere Geometry Free
IGN	Spanish National Geographic Institute
IGS	International GNSS Service
IGSO	Inclined Geosynchronous Satellite Orbit
IMO	International Maritime Organization
I/NAV	Integrity NAVigation message
I/O	Input / Output
IOD	Issue Of Data
ION	Institute Of Navigation
IONEX	IONosphere map EXchange format
IOV	In-Orbit Validation

IP	Internet Protocol
IPN	National Polytechnic Institute
IRNSS	Indian Regional Navigation Satellite System
IS	Interface Specification
ISCB	Inter System Clock Bias
ISO	International Organisation for Standardisation
ISRO	Indian Space Research Organisation
ISTF	Ionospheric Studies Task Force
JRC	Joint Reseach Centre
KASS	Korea Augmentation Satellite System
KML	Keyhole Markup Language
LAMBDA	Least-squares AMbiguity Decorrelation Adjustment
LEO	Low Earth Orbit
LLI	Loss of Lock Indicator
LNAV	Legacy NAVigation message
LMS	Least Median of Squares
LoS	Line of Sight
LT	Local Time
LTO	Link Time Optimization
LTS	Least Trimmed Squares
MATLAB	MATrix LABoratory
MB	MegaByte
MEO	Medium Earth Orbit
MGEX	Multi-GNSS EXperiment
MI	Misleading Information
MM	Monument Marker
MODIP	MOdified DIP latitude
MOPS	Minimum Operational Performance Standards
MSAS	Multi-functional Satellite Augmentation System
MSC	Maritime Safety Committee
MT	Message Type
MTBF	Mean Time Between Failures
MW	Melbourne–Wübbena
NASA	National Aeronautics and Space Administration
NAVITEC	NAVigation TEChnologies and European Workshop on GNSS Signals and Signal Processing

NEAD	Number of Epochs Affected by Discontinuities
NEU	North East Up
NSE	Navigation System Error
NPA	Non Precision Approach
OASIS	Orbit Analysis and Simulation Software
OP	Open Procedure
OpenMP	Open Multi-Processing
OS	Open Service
OS	Operative System
OSNMA	Open Service Navigation Message Authentication
PA	Precision Approach
PCT	Patent Cooperation Treaty
PDOP	Precision Dilution Of Precision
PE	Positioning Error
PEGASUS	Prototype EGNOS and GBAS Analysis System Using SAPPHIRE
PhD	Doctor of Philosophy
PL	Protection Level
PNT	Position Navigation Timing
PPP	Precise Point Positioning
PRN	Pseudo-Random Number
QZO	Quasi-Zenith Orbit
QZSS	Quasi-Zenith Satellite System
RAM	Random Access Memory
RAIM	Receiver Autonomous Integrity Monitoring
RAP	Andalusian Positioning Network
RIMS	EGNOS Ranging and Integrity Monitoring Stations
RINEX	Receiver INdependent EXchange format
RINEX-B	Receiver INdependent EXchange format SBAS binary broadcast messages
RISDE	Russian Institute of Space Device Engineering
RISC	Reduced Instruction Set Computer
RMS	Root Mean Square
RNSS	Radio Navigation Satellite Service
ROTI	Rate Of change of Total electron content Index
RTCA	Radio Technical Commission for Aeronautics

RTCM	Radio Technical Commission for Maritime
RTK	Real-Time Kinematics
SAO	Smithsonian Astrophysical Observatory
SAPPHIRE	Satellite and Aircraft Data Base for System Integrity Research
SBAS	Satellite-Based Augmentation System
SC	Special Committee
SDCM	System for Differential Corrections and Monitoring
SDD	Service Definition Document
SF	Single Frequency
SIF	Second Ionospheric Free
SINEX	Solution (Software/technique) INdependent EXchange format
SINEX BIAS	Solution (Software/technique) INdependent EXchange format for GNSS BIASes
SIS	Signal In Space
SISRE	Signal In Space Range Error
SNR	Signal to Noise Ratio
SoL	Safety-of-Life
SOLAS	Safety of Life at Sea
SouthPAN	Southern Positioning Augmentation Network
SP3	Standard Product #3
SPP	Standard Point Positioning
STEC	Slant Total Electron Content
SVN	Space Vehicle Number
TDOP	Time Dilution Of Precision
TEC	Total Electron Content
TECU	Total Electron Content Unit
TGD	Total Group Delay
TTA	Time-To-Alert
UHF	Ultra High Frequency
UN	United Nations
UNESCO	United Nations Educational, Scientific and Cultural Organization
UPC	Technical University of Catalonia
URL	Universal Resource Locator

USA	United States of America
VAL	Vertical Alarm Limit
VDOP	Vertical Dilution Of Precision
VPE	Vertical Positioning Error
VPL	Vertical Protection Level
WAAS	Wide Area Augmentation System
WIR	Worst Integrity Ratio
WLS	Weighted Least Squares
WWRNS	World-Wide Radio Navigation System
XML	eXtensible Markup Language

Thesis introduction

1.1 Purpose of the thesis

The aim of every scientist is not only the research itself, but to be able to effectively transmit its knowledge to other scientists, students, etc. The most common way is in writing (e.g. articles and conference proceedings), which has proven to be effective, but still have some limitations:

- Articles implementing algorithms describe the main equations, but do not provide an accessible implementation (e.g. source code or an executable file). If readers wish to replicate the results, they need to develop their own program.
- Dataset used in an article may not be easily accessible (even if all data used is public).
- Articles may be behind a paywall for the readers.

In the current era of digital data and worldwide internet access, the limitations mentioned above can be addressed. For instance, “Zenodo” ([European Organization For Nuclear Research and OpenAIRE \[2013\]](#)) is an open access portal for publishing articles (and its dataset), or even source code.

In the field of Global Navigation Satellite System (GNSS), there are several software tools available, but they have one or more limitations:

- No state-of-the-art algorithms.
- Limited focus.
- Limited compatibility (they usually only work on Windows).

- Limited processing speed (especially for programs coded in interpreted languages such as Python or Matlab).
- Limited documentation.
- Limited or no access to internal computations.
- Limited configuration options.
- Not suitable for education purposes.
- Licensing fees.
- No access to source code.

In the context of this Doctor of Philosophy (PhD) dissertation, the capability to process all constellations, algorithms and new processing methods requires not only implementing the algorithms itself, but solving other side problems. For this reason, the present PhD dissertation provides its research through the GNSS-Lab Tool suite (gLAB) tool, as on the one hand, it is used to compute the results in this thesis, and on the other hand, it allows one to show the challenges solved throughout its development and at the same time, to convert it into a tool to be used by scientists for research, by professionals in their work and by teachers or students for educational purposes. Moreover, gLAB has been developed in order to address all the limitations stated above, i.e.:

- Has state-of-the-art algorithms (e.g. Satellite-Based Augmentation System (SBAS) Dual Frequency Multi Constellation (DFMC), Fast Precise Point Positioning (Fast-PPP)).
- Processes in Standard Point Positioning (SPP), Precise Point Positioning (PPP), uncombined PPP, Fast-PPP, SBAS 1F, SBAS DFMC and Differential Global Navigation Satellite System (DGNSS).
- Works on Windows, Linux (also in Advanced RISC Machine (ARM) v7 and v8 architectures) and Mac platforms.
- Fast data processing.
- Software manual, extensive tooltips in the Graphical User Interface (GUI) and tutorial courses.
- Ability to print internal configuration and internal computations.
- Many configuration options.
- GUI designed for education, with a dedicated tutorial book (Sanz et al [2013b]).

- Free of charge
- Open source and with open licences: “Apache” licence for Data Processing Core (DPC) and Data Analysis Tool (DAT), GNU General Public License (GPL) v3 for the GUI.

1.2 Thesis context

This thesis has been mainly developed in the context of three projects (in order of occurrence):

1. ESA/ESTEC project “gEGNOS: gLAB Upgrade for EGNOS Data Processing” (with one Contract Change Notification (CCN)).
2. ESA/EPO project “Upgrade of gLAB Tool for Double Frequency Multi Constellation (DFMC)” (with one CCN).
3. EUSPA (former GSA) Specific Contract 1 under framework contract “Integration of Fundamental Elements”, number GSA/OP/12/16/Lot1.

The first project was aimed at upgrading gLAB for SBAS and developing a Global Monitoring System (GMS) for European Geostationary Navigation Overlay System (EGNOS) using gLAB, the second at the upgrade for multi-constellation and multi-frequency and SBAS DFMC for gLAB and the third one focused on performing a study on the usage of EGNOS for maritime purposes, which required adding a maritime mode in gLAB and developing a method for multi day SBAS availability maps.

Aside from these projects, uncombined PPP and Fast-PPP were added to gLAB in order to fulfil the tool’s capabilities.

In parallel to these tasks, the thesis author also made multi-platform installable files for all Research group of Astronomy and Geomatics (gAGE) GNSS courses involving gLAB, provided support for gLAB (including minor additions) to other gAGE members for their research work and also provided support to users by mail.

1.3 Thesis outline

This thesis is divided into ten chapters:

- **Chapter 1** (the current chapter) provides the thesis purpose, thesis outline, publications by the thesis author or where he collaborated in the context of this thesis and the standard co-authoring.
- **Chapter 2** provides an introduction to GNSS, including the different systems, its signals and current status.
- **Chapter 3** is a brief introduction to gLAB tool, its modules and capabilities, with a usage example and the list of all gLAB contributors to the source code.
- **Chapter 4** describes the state-of-the-art algorithms implemented in gLAB (SPP, PPP, SBAS 1F and DGNSS).
- **Chapter 5** describes the advanced algorithms implemented in gLAB (uncombined PPP, Fast-PPP and SBAS DFMC).
- **Chapter 6** lists the main challenges faced during the gLAB upgrades (measurement selection, cycle-slip detection, output prints, multi-day SBAS plots, code optimization and multi-threading).
- **Chapter 7** describes the main research studies in the context of the thesis.
- **Chapter 8** provides an assessment of the impact of gLAB.
- **Chapter 9** contains the conclusions of this thesis.
- **Chapter 10** describes future work that might be performed in gLAB.

1.4 Thesis publications

The following articles in peer-reviewed journals and proceedings were published as a result of the work in the thesis:

- **Articles:**
 1. Alonso MT, Ferigato C, Ibáñez D, Perrotta D, Rovira-Garcia A, Sordini E (2021) Analysis of 'Pre-Fit' Datasets of gLAB by Robust Statistical Techniques. *Stats* 4(2):400–418, doi: 10.3390/stats4020026, URL <https://doi.org/10.3390/stats4020026>
 2. Aragon-Angel A, Rovira-Garcia A, Arcediano-Garrido E, Ibáñez Segura D (2021) Galileo Ionospheric Correction Algorithm Integration into the Open-Source GNSS Laboratory Tool Suite (gLAB). *Remote Sensing* 13(2):191, doi: 10.3390/rs13020191, URL <https://doi.org/10.3390/rs13020191>

3. Ibáñez D, Rovira Garcia A, Alonso MT, Sanz J, Juan JM, González Casado G, López Martínez M (2020) EGNOS 1046 Maritime Service Assessment. *Sensors* 20(1):276, doi: 10.3390/s20010276, URL <https://doi.org/10.3390/s20010276>
4. Rovira-Garcia A, Ibáñez D, Orús R, Juan J, Sanz J, González-Casado G (2019) Assessing the quality of ionospheric models through GNSS positioning error: methodology and results. *GPS solutions* 24:4:1–4:12, doi: 10.1007/s10291-019-0918-z, URL <https://doi.org/10.1007/s10291-019-0918-z>
5. Juan J, Sanz J, Rovira-Garcia A, González-Casado G, Ibáñez D, Orús R (2018) AATR an ionospheric activity indicator specifically based on GNSS measurements. *Journal of Space Weather and Space Climate* 8(A14):1–11, doi: 10.1051/swsc/2017044, URL <https://doi.org/10.1051/swsc/2017044>
6. Rovira-Garcia A, Juan J, Sanz J, González-Casado G, Ibáñez D (2016b) Accuracy of ionospheric models used in GNSS and SBAS: methodology and analysis. *Journal of geodesy* 90:229–240, doi: 10.1007/s00190-015-0868-3, URL <http://hdl.handle.net/2117/84674>

- **Proceedings:**

1. Rovira García A, Ibáñez Segura D, Li M, Alonso Alonso MT, Sanz Subirana J, Juan Zornoza JM, González Casado G (2022) gLAB hands-on education on satellite navigation, Universitat Politècnica de Catalunya. doi: 10.5821/conference-9788419184405.010, URL <http://hdl.handle.net/2117/368624>
2. Ibáñez D, Rovira-Garcia A, Sanz J, Juan J, González-Casado G, Alonso M, López-Salcedo JA, Jia H, Pancorbo FJ, García C, Martin I, Rodrigo A, López M (2019) A Kinematic campaign to evaluate EGNOS 1046 maritime service, *ION Publications (The Institute of navigation)*, pp 840–854. doi: 10.33012/2019.16941, URL <https://doi.org/10.33012/2019.16941>
3. Sanz J, Juan J, López C, González-Casado G, Rovira-García A, Alonso M, Ibáñez D (2018) Automated GPS Signal-in-Space anomalies monitoring over more than 11 years, *ION Publications (The Institute of navigation)*, pp 2172–2185. doi: 10.33012/2018.15853, URL <http://hdl.handle.net/2117/128470>
4. Ibáñez D, Rovira-García A, Sanz J, Juan J, Gonzalez-Casado G, Jimenez-Baños D, López-Echazarreta C, Lapin I (2018) The GNSS

Laboratory Tool Suite (gLAB) updates: SBAS, DGNSS and Global Monitoring System. In: Proceedings of the 9th ESA Workshop on Satellite Navigation Technologies (NAVITEC 2018), ESA/ESTEC, Noordwijk, The Netherlands, doi: 10.1109/NAVITEC.2018.8642707, URL <https://ieeexplore.ieee.org/document/8642707>

1.5 Standard development

As a result of the knowledge developed during the realisation of the thesis, the thesis author is co-author of the Multi-Band EGNOS File standard, which is the new version of EGNOS Message Service (EMS) file standard for both SBAS 1F and SBAS DFMC messages:

- Buehler S, Ibáñez D (2020) Multi-band EGNOS file format description document. URL http://www.egnos-pro.esa.int/ems/EMS_UID_Multi-band.pdf, accessed: 2023-07-06

GNSS Introduction

Modern societies depend on several constellations of satellites for their correct functioning. Some of these constellations are GNSS, each of which is a constellation of satellites orbiting Earth, continuously transmitting signals which enable a user to determine their position in three dimensions, referenced from the centre of the earth.

The first GNSS to be developed was Global Positioning System (GPS), initially developed for military use, but in 1983 opened for civilian use (with limitations). Since that date, the civil community (scientific, commercial, etc.) has kept creating new use cases, which has meant that nowadays an important share of the global economy is dependent on GNSS (see [EUSPA \[2022b\]](#)). This fact has meant that several countries decided to develop their own systems.

There are currently four global and two regional GNSS:

- **Global Positioning System (GPS):** The United States of America (USA) system. Satellite deployment started at the end of the 1970s and reached Full Operational Capability (FOC) in July 1995 (from [Sanz et al \[2013a\]](#)). Constellation status can be consulted in [USA Navigation Center \[2023\]](#).
- **GLOBAL NAVIGATION SATELLITE SYSTEM (GLONASS):** The Russian system. Satellite launches started in 1982, achieving FOC in December 1995, but for only a few years due to lack of funds. In December 2011, it was again in FOC (from [Sanz et al \[2013a\]](#)). Constellation status can be consulted in [IAC \[2023\]](#).
- **European Global Navigation Satellite System (Galileo):** The European system. It currently has only 26 fully operational orbiting satellites. It was scheduled to reach FOC by 2023 (from [EUSPA \[2023b\]](#)), but due to the Russian invasion of Ukraine, the last two

launches were postponed. Constellation status can be consulted in [EUSPA \[2023a\]](#).

- **BeiDou Navigation Satellite System (BDS):** The Chinese system. It has been built in two phases: BDS-2, consisting of 5 Geosynchronous Equatorial Orbit (GEO), 5 Inclined Geosynchronous Satellite Orbit (IGSO) and 4 Medium Earth Orbit (MEO); and BDS-3, consisting of 3 GEO, 3 IGSO and 24 MEO (from [CSNO \[2023b\]](#)). On June 2020 the last BDS-3 was launched (from [Lu et al \[2020\]](#)) and reached FOC on July 2020 ([Zhang et al \[2022\]](#)). Constellation status can be consulted in [CSNO \[2023a\]](#).
- **Quasi-Zenith Satellite System (QZSS):** The Japanese system. It is a regional system focused on enhancing GPS in Asia-Oceania regions (especially in Japan). It consists of 3 IGSO and 1 GEO satellites (from [Cabinet Office, Government Of Japan \[2023c\]](#)). FOC was declared on November 1st 2018 (from [GPS World Staff \[2018\]](#)). Constellation status can be consulted in [Cabinet Office, Government Of Japan \[2023a\]](#).
- **Indian Regional Navigation Satellite System (IRNSS):** The Indian system. It is a regional system focused on India. It consists of 4 IGSO and 3 GEO satellites (from [U R Rao Satellite Centre \[2019\]](#)). The constellation was fully deployed in 2018 (from [ISRO \[2018\]](#)). Constellation status can be consulted in [ISRO \[2023\]](#).

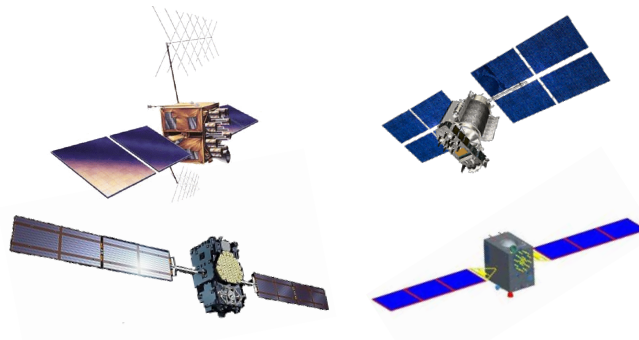


Figure 2.1: GNSS satellites: GPS IIR-M (top left), GLONASS-M (top right), Galileo IOV (bottom left) and BDS-M (bottom right). Image from [Sanz et al \[2013a\]](#).

Each GNSS is divided into three segments, as shown in Fig. 2.2:

- **Space Segment:** The constellation of satellites orbiting Earth.
- **Control Segment:** It is composed of one or more control centres for operating the system, several monitoring stations distributed around the world for data collection and ground antennas for uplinking data to the satellites.
- **User Segment:** Any receiver capable of receiving GNSS signals and computing the navigation solution. For example, any modern smartphone.

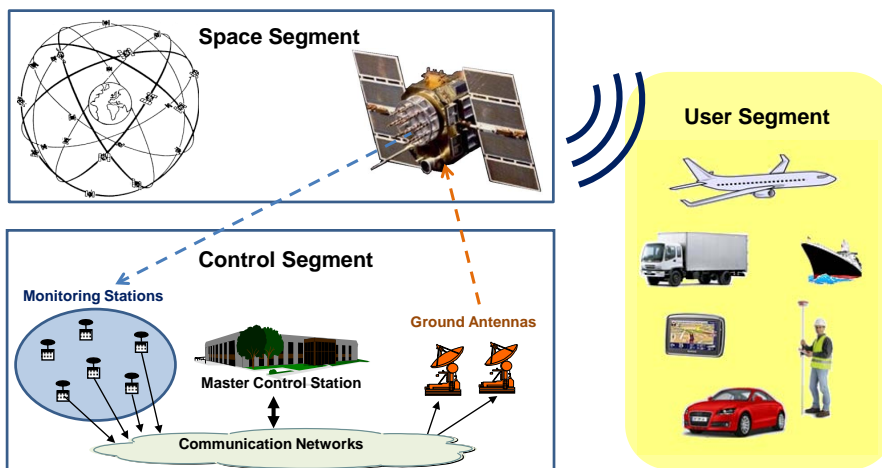


Figure 2.2: GNSS architecture (from Sanz et al [2013a]).

Every GNSS transmits signals on two or more different frequencies. There are some frequencies which are exclusive to GNSS, others are shared with other services (from [Federal Communications Commission \[2022\]](#)). The signals transmitted by satellites on these frequencies can be classified in three types:

- **Carrier:** Radio frequency sinusoidal signal at a given frequency, carrying the payload.
- **Ranging code:** Sequences of zeros and ones which allow the receiver, on the one hand, to separate the signal from different satellites, and on the other hand, to determine the travel time of the radio signal from the satellite to the receiver. They are called Pseudo-Random Number (PRN) sequences or PRN codes.

- **Navigation data:** A binary-coded message providing information on the satellite ephemeris (pseudo-Keplerian elements or satellite position and velocity), clock bias parameters, satellite health status and other complementary information. Satellites sending several GNSS signals may send different navigation messages in each signal.

In Fig. 2.3 is shown the frequency allocation for the four global constellations (GPS, GLONASS, Galileo and BDS). Shared frequencies allow for simultaneously receipt of signals from multiple constellations and improve accuracy.

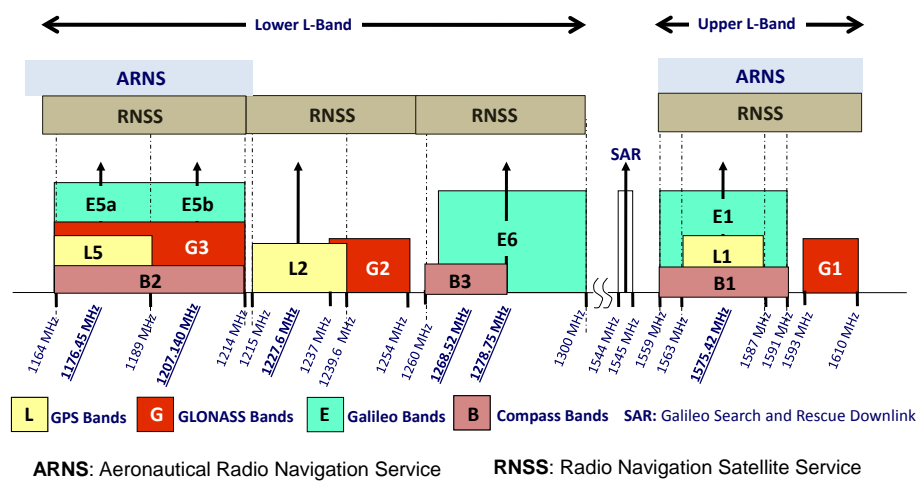


Figure 2.3: GNSS frequency bands (from Sanz et al [2013a]).

In Fig. 2.3 there are frequency bands marked as Aeronautical Radio Navigation Service (ARNS), which are exclusive for GNSS signals (not all ARNS have exclusive use). The rest of the Radio Navigation Satellite Service (RNSS) bands are shared with other services, such as radio location services (frequency allocation is described in [Federal Communications Commission \[2022\]](#)). ARNS bands are suitable for critical operations, such as Safety-of-Life (SoL) services.

As shown in Fig. 2.3, each constellation uses multiple frequencies. The lists of signals for each constellations are (from [Romero \[2021\]](#)):

- GPS has 3 frequencies, L1, L2 and L5. L1 has two civil and two military signals, L2 has one civil and two military signals, and L5 has only a civil signal. Additional info on GPS signals can be found at [GPS.gov \[2023b\]](#).
- GLONASS has 3 frequencies, G1, G2 and G3. G1 and G2 have both a civil and military signal, and use Frequency Division Multiple

Access (FDMA). G3 only has a civil signal (in deployment phase) with Code Division Multiple Access (CDMA). It is planned to add additional frequencies, G1a and G2a, which will both use CDMA. Additional info on GLONASS signals can be found at [PennState College of Earth and Mineral Sciences \[2016\]](#).

- Galileo has 3 frequencies E1, E6 and E5 (E5a and E5b signals are modulated in the same frequency E5). Galileo will include novel services, such as Open Service Navigation Message Authentication (OSNMA) or High Accuracy Service (HAS) (see [Fernández et al \[2023\]](#)). The list of services can be found at [EUSPA \[2023c\]](#).
- BDS-2 (BeiDou second generation) has 3 frequencies, B1, B2 and B3, whose signals were named B1I, B2I and B3 respectively. In BDS-3 (BeiDou third generation) additional frequencies and signals were added (B1C, B1A, B2a, B2b, B2a+b and B3A). A detailed characterization of BeiDou signals can be found at [Montenbruck et al \[2022\]](#).
- QZSS has 4 frequencies, L1, L2, L5 and L6, with a total of 9 civil signals and no military signals. The list of signals can be found at [Cabinet Office, Government Of Japan \[2023b\]](#).
- IRNSS has 2 frequencies, L5 and S, which both contain a civil and restricted signals (for authorized users only). Additional info on IRNSS signals can be found at [ISRO \[2022\]](#).

The current era of new constellations and signals opens new opportunities and challenges. Part of the work presented in this PhD is the gLAB upgrade for processing all constellations, frequencies and signals and the challenges encountered when working with so many different frequencies and signals.

Chapter 3

gLAB description

The GNSS-Lab Tool suite (gLAB) is an advanced interactive educational multi-purpose software package for processing and analysing GNSS data (Sanz et al [2013b]), developed by gAGE group. The first release was capable of performing precise modelling of the GPS observables (code and phase) at the centimetre level, allowing GPS SPP (USA Navigation Center [2022]) and PPP (Zumberge et al [1997]).

gLAB is designed to be multi-platform and easy to automate. It is divided into three independent software modules:

- **The Data Processing Core (DPC):** Implements all the data processing algorithms. It is written in C and can be executed in command line.
- **The Data Analysis Tool (DAT):** The plotting tool. Initially written in Python 2, it was later updated to work in both Python 2 and 3. It can be executed in command line.
- **The Graphical User Interface (GUI):** The graphical tool for interfacing with the user. Initially written in Python 2, it was later rewritten from scratch in C++/Qt (The Qt Company [2023]) since version 5.00.

Figure 3.1 shows the three gLAB modules. The GUI module main purpose is to interface with the user in a friendly way for both beginners and professionals, but implements additional useful features (such as extended tooltips and a list of Universal Resource Locators (URLs) list for downloading files). The DPC does all the computation, while the DAT generates the plots from the DPC output files. Both the DPC and DAT can be run in command line independently of each other.

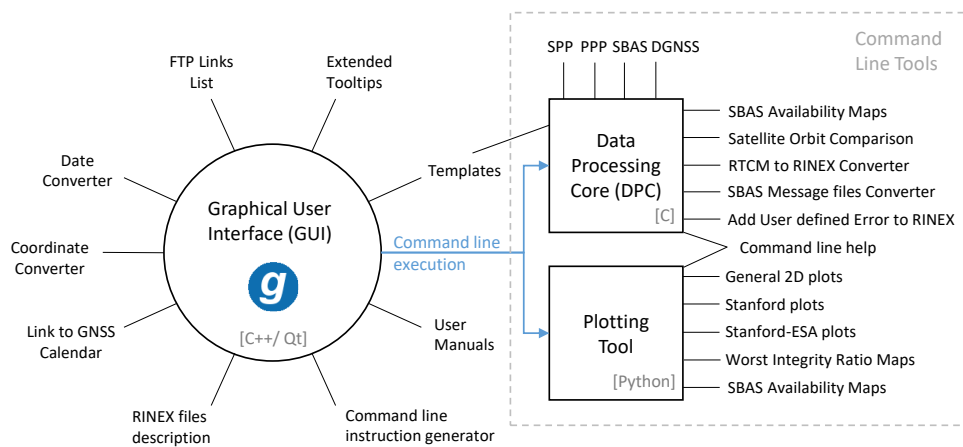


Figure 3.1: gLAB is divided into three independent executables: the GUI, the DPC and the DAT. The grey text inside square brackets shows the programming language. The callouts show the different features (in the GUI) or processing modes (in the DPC or the DAT). The blue lines show that the GUI generates the command line sentence to run the DPC or the DAT (from [Ibáñez et al \[2018\]](#)).

In the following sections and chapters, all gLAB capabilities or functionalities refer to the latest version (i.e. with all upgrades performed, including the ones made for the present PhD dissertation).

3.1 Main gLAB features

The main gLAB features are:

- **High Accuracy Positioning capability:** Precise modelling of the GNSS measurements (code and carrier phase) at the centimetre level, allowing both full multi-constellation SPP, PPP and Fast-PPP ([Rovira-Garcia et al \[2016a\]](#)).
- **Fully configurable:** gLAB has many configuration options (more than 500 in the DPC) for all steps in the GNSS data processing.
- **Easy to use:** The GUI has templates for each processing mode (based on the DPC defaults), therefore by default one only needs to provide the input files to get standard processing in each mode. The extended tooltips (text boxes with extended information shown when hovering over an element) in each option allows the user to understand not only what the option is for, but also its theory (e.g. the cycle-slips tooltips).

explain the algorithm used). Moreover, the DPC is designed to be used with the minimum parameters (the input files), as it applies the defaults in each mode automatically. Moreover, the DPC can benefit from the autocomplete feature when used in a Bourne Again SHell (Bash) shell (by installing the provided autocompletion script).

- **Access to internal computations:** There are several well documented output messages for each step in the GNSS data processing, including the intermediate computations.
- **Open source:** gLAB source code is distributed under two licences, the DPC and the plotting tool use the “Apache” licence version 2.0, while the GUI has the GPL 3.0, as it uses the Qt library. This allows the user to develop both free and commercial GNSS data processing tools using gLAB as a library.
- **Automate:** The DPC and the DAT can be run from the command line, and the configuration file or command line can be easily generated with the GUI, if needed.
- **Multi-platform:** gLAB can be run in Windows, Linux and Mac in x86_64 architectures. In Linux, it also supports ARM v7 and v8 architectures ([Wikipedia \[2023\]](#)). Moreover, the DPC can run in any platform with a C compiler, as it does not use any non-standard library (except for Open Multi-Processing (OpenMP) for multi-threading, but it is not used in single-thread compilation).

3.2 Graphical User Interface

The GUI is the link between the other two gLAB components (DPC and DAT), as its only purpose is to run the DPC or the plotting tool with the selected configuration. It is designed to be accessible for beginners (as the templates set the default values for each processing mode; therefore it is only necessary to provide the input files to run the tool) as well as complete for more advanced users, as it has most of the options available in the DPC.

The initial gLAB GUI screen can be seen in Fig. 3.2.

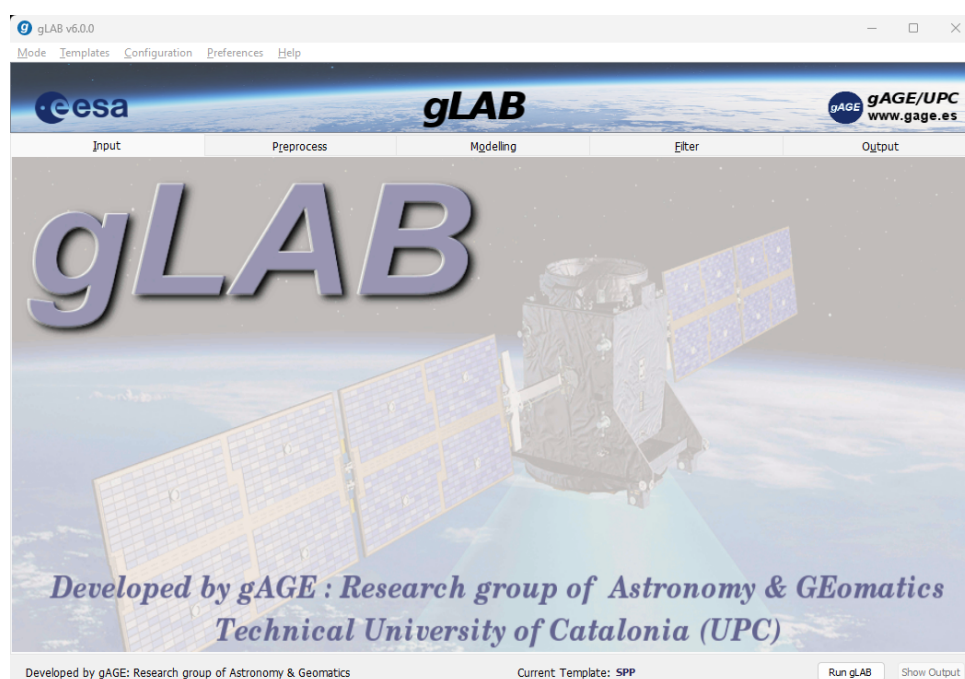


Figure 3.2: gLAB GUI initial screen.

At the top of gLAB GUI there is a menu bar with the following menus:

- **Mode:** Contains the various processing modes, as follows:
 - **Positioning:** To compute the navigation solution. This mode executes the DPC.
 - **Analysis:** To create the plots. This mode executes the DAT.
 - **Converter:** To convert between SBAS file formats, and from Radio Technical Commission for Maritime (RTCM) to Receiver

INdependent EXchange format (RINEX). This mode executes the DPC.

- **Compare Orbits & Clocks:** To compare orbits and clocks. Orbits and clocks to be compared may be from different files (e.g. a RINEX navigation file and an Standard Product #3 (SP3) file) or between different navigation message types (e.g. between Galileo Freely accessible NAVigation message (F/NAV) and Integrity NAVigation message (I/NAV)). This mode executes the DPC.
 - **Show Orbits & Clocks:** To print in columnar text format the satellite position (in XYZ and geodetic coordinates), the clock correction, Issue Of Data (IOD) and health status. This mode executes the DPC.
 - **Add User Error to RINEX:** To modify a RINEX observation, adding a user-defined error to the selected satellites and measurements, which are all set in a text file. The file format description is in the “Help” menu (subsection “User manuals”, item “User Added Error File Manual”). This mode executes the DPC.
- **Templates:** Templates to set the default settings in SPP, PPP, SBAS 1F, SBAS DFMC and DGNSS processing modes.
 - **Configuration:** To control general options provided to the DPC, as follows:
 - **Save configuration file:** Saves the current configuration file. The configuration is checked for errors before allowing a file to be saved.
 - **Load configuration file:** Loads a configuration file. The configuration file is checked for errors before it is actually applied to configure the GUI.
 - **Show configuration file:** Opens a text editor with the last saved configuration file.
 - **Show command line instruction:** Shows the command line instruction the GUI generates to run the DPC or the DAT. The instruction includes the full path to the executable and files; therefore it is suitable to be copied and executed from a terminal (independently of the terminal’s current directory).
 - **Generate minimum configuration file:** By default, the GUI generates a command line with the input and output files, and only the parameters to account for the user configuration differences with respect to the default settings (in the selected processing mode), in order to make the command line instruction as short as possible. If this option is disabled, the GUI sets all gLAB options,

even though they are the same as the defaults (this is useful for printing all parameters in the configuration file).

- **Set Multi-threading options:** Opens a menu to enable or disable multi-threading and set the number of cores to use in multi-thread processing.
- **Print messages in gLAB v5 format:** The multi-constellation version of gLAB (version 6) required to change the format of output messages. This option is for printing in the same format as previous version, in order to preserve backwards compatibility.
- **Preferences:** GUI options to disable tooltips or hide the gLAB top image in order to save screen size, tailored for small screens.
- **Help:** Menu with additional information or tools useful in GNSS processing (as shown in Fig. 3.3). Some of its elements are:
 - Menus related to the tool itself (credits, licence, release notes, etc.) and its manuals (embedded inside the executable).
 - A list of URLs for downloading files (RINEX, SP3, ANTenna EXchange format (ANTEX), etc.).
 - Built-in date and coordinate converter (also available in the command line).
 - Links to gAGE teaching material associated with gLAB.
 - Function to check for a gLAB update (the tool only checks for updates when the user clicks on this menu).

The GUI is designed to be as complete as possible for the user (in the sense that it does not need to look for information outside of the GUI). The templates set the defaults for each processing modes. The options can be configured as required by the user. Each option has a tooltip which explains what the option is for, including a link to the “Help” menu (e.g. a link to download any input file), and the “Help” menu contains all other useful information not directly related to processing (e.g. date converter).

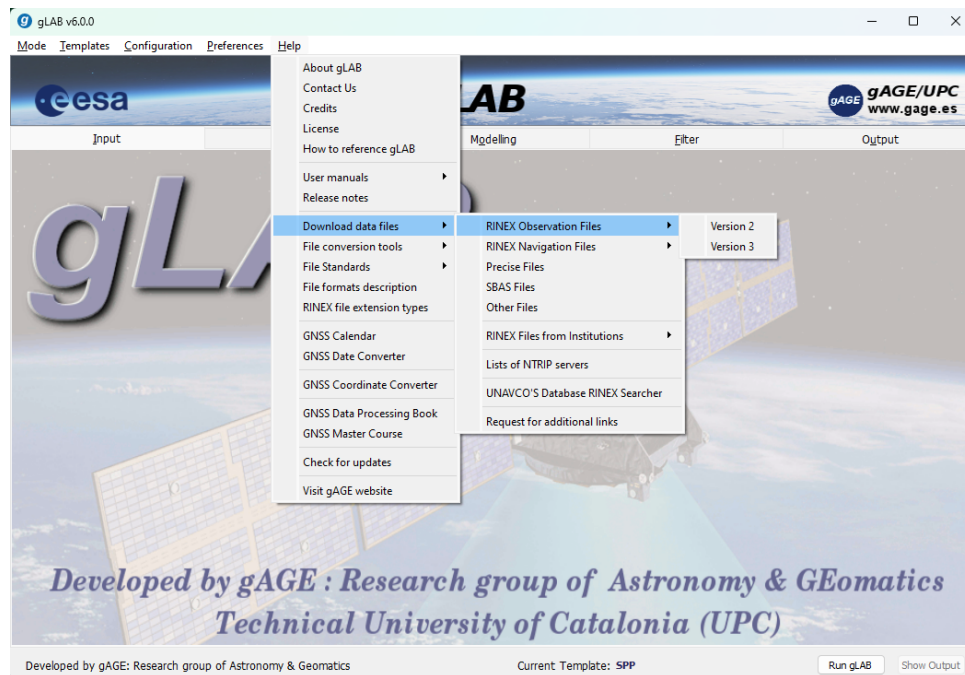


Figure 3.3: gLAB GUI help menu. The submenu “Download Data files” shown in the image contains the links for downloading all necessary GNSS files.

3.2.1 Positioning mode

The positioning mode computes the navigation solution in the various processing modes (SPP, PPP, etc.). As there are many options, it is divided into 5 different tabs, each tab representing a step in the data processing:

- **Input:** Includes all options regarding the data provided by the user to the tool, which are the files to be read and the receiver a priori position (as it is shown in Fig. 3.4).
- **Preprocess:** Includes the options regarding the data preprocessing (data decimation and satellite selection) and quality check of the data (such as cycle-slips, inconsistencies between code and carrier phase measurements, Signal to Noise Ratio (SNR), etc.).
- **Model:** Includes all options for modelling the receiver measurements in any processing mode (SPP, PPP, etc.).
- **Filter:** Includes the selection of measurements to be used in the navigation equations, the options for the Kalman filter (Kalman [1960]) implemented in gLAB, and other options regarding the navigation solution quality (e.g Dilution Of Precision (DOP)).

- **Output:** Includes the options for setting the paths for the output files (gLAB main output file, Keyhole Markup Language (KML) file, etc.), summary options and the selection of messages to be printed.

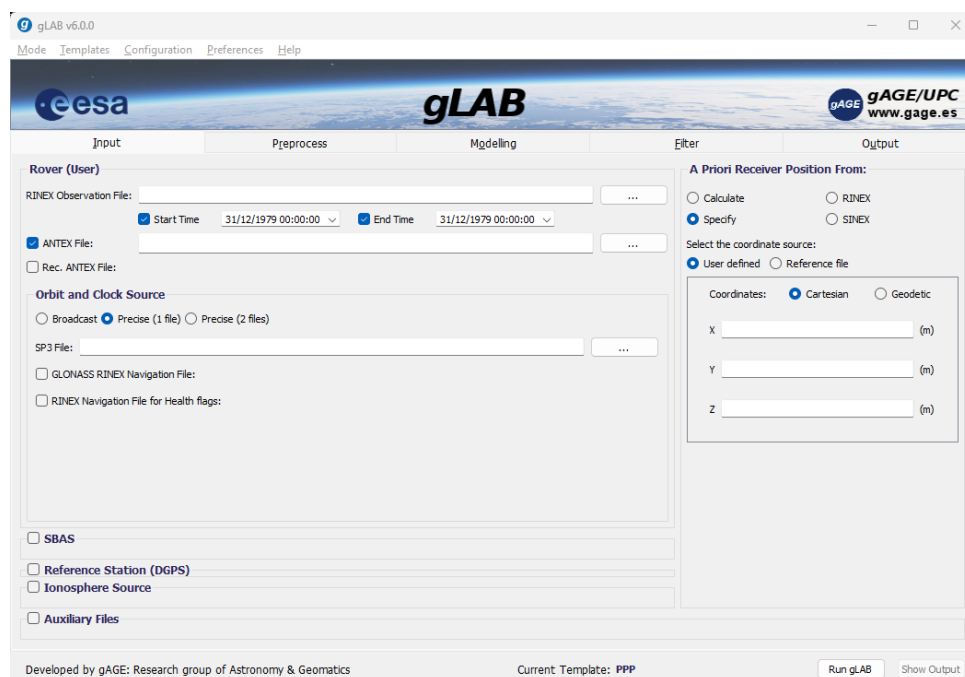


Figure 3.4: gLAB GUI initial Positioning tab in PPP mode.

3.2.2 Analysis mode

The analysis mode is devoted to the creation of plots from the DPC output files with the DAT. As can be seen in Fig. 3.5, the upper part consists of a series of buttons that set the default configuration for each type of plot. Some of these buttons change according to the processing mode set in the “Templates” menu (e.g. in SBAS processing mode there are buttons for Stanford plots). The lowest part are the configuration options for each plot. When the top buttons are used, there is no need to change these options, but the user can modify them as needed (even for creating plots with any other columnar text file not created by gLAB DPC).

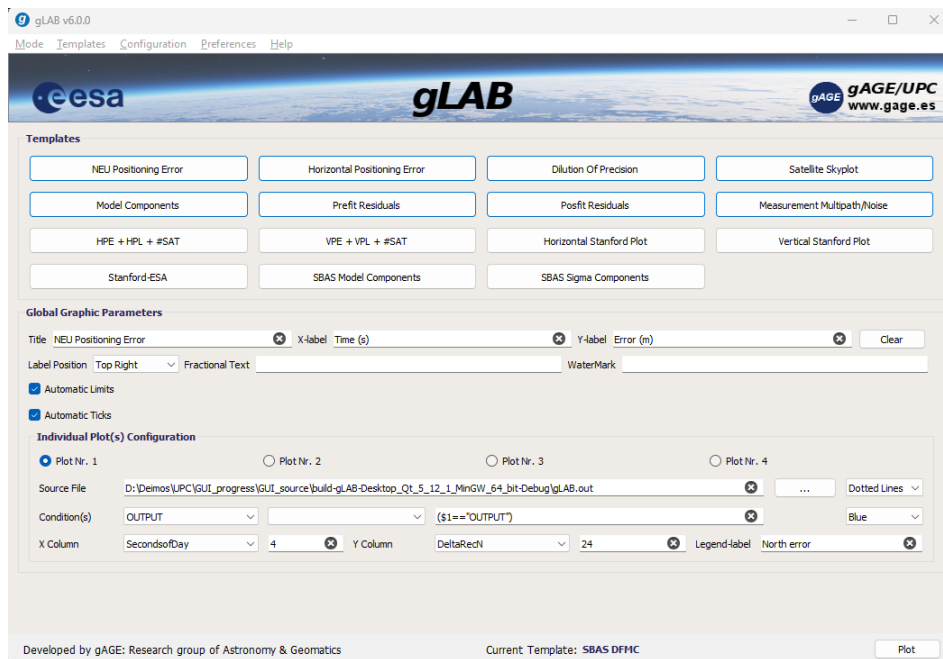


Figure 3.5: gLAB GUI Analysis.

As in the Positioning mode, the GUI can generate the corresponding command line, but it cannot save or load configuration files, because, unlike the DPC, the DAT cannot read configuration files.

3.2.3 Converter mode

This mode converts between file formats. The following conversion are possible:

- **SBAS files** (shown in Fig. 3.6):
 - Convert from Receiver INdependent EXchange format SBAS binary broadcast messages (RINEX-B) (Suard et al [2003]) to EMS (Buehler and Ibáñez [2020]) (and vice versa).
 - Convert from RINEX-B or EMS to Prototype EGNOS and GBAS Analysis System Using SAPPHIRE (PEGASUS) (EUROCONTROL [2001]).

- **DGNSS files:**

- Convert from RTCM version 2 (RTCM SC-104 [2010]) (GPS data only) to columnar text file (GPS data only).
- Convert from RTCM version 3 (RTCM SC-104 [2022]) (GPS data only) to RINEX observation file, either version 2.11 (Werner and Estey [2007a]) or 3.00 (Werner and Estey [2007b]).

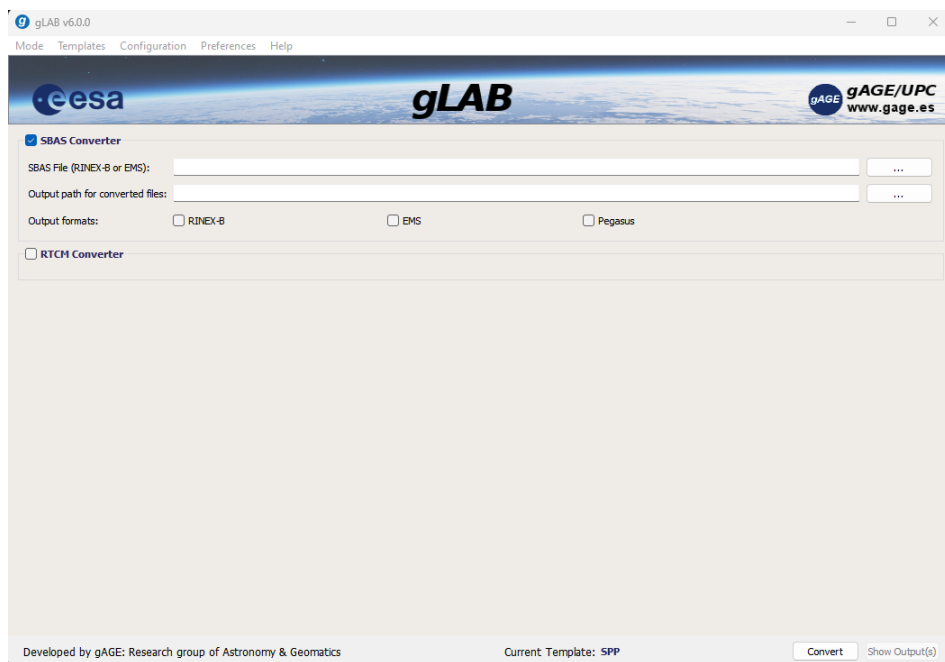


Figure 3.6: gLAB GUI Converter mode for SBAS files.

3.2.4 Compare Orbits & Clocks mode

This mode compares orbits and clocks from two files (which may be two RINEX navigation files, two SP3 files or a RINEX navigation file to an SP3 file). Moreover, when two RINEX navigation files are used, it is possible to compare orbits and clocks from two navigation message types. It currently supports comparing GPS Legacy NAVigation message (LNAV) to Civil NAVigation message 1 (CNAV1) and Galileo I/NAV to F/NAV, but the tool is prepared for additional comparison when it supports RINEX version 4.00.

The output messages contain the Signal In Space Range Error (SISRE) comparison and the difference of radial, along-track, cross-track and XYZ components. When using a RINEX navigation file, the IOD and health status are also provided.

The initial screen for comparing orbits & clocks mode can be seen in Fig. 3.7.

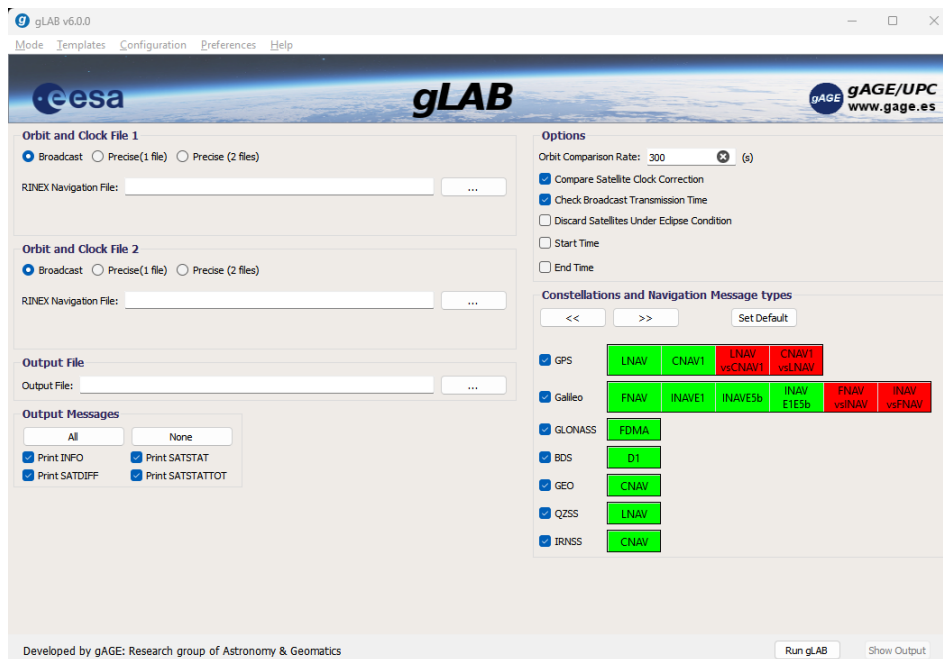


Figure 3.7: gLAB GUI Compare Orbits & Clocks mode.

3.2.5 Show Orbits & Clocks mode

This mode computes satellite orbits and clocks from a RINEX navigation file or an SP3 file. The output messages contain the satellite position in Cartesian and geodetic systems, as well as the IOD and health status (when a RINEX navigation file is used). Moreover, the satellite elevation and azimuth are also computed if the user provides a receiver position.

The initial screen for showing orbits & clocks mode can be seen in Fig. 3.8.

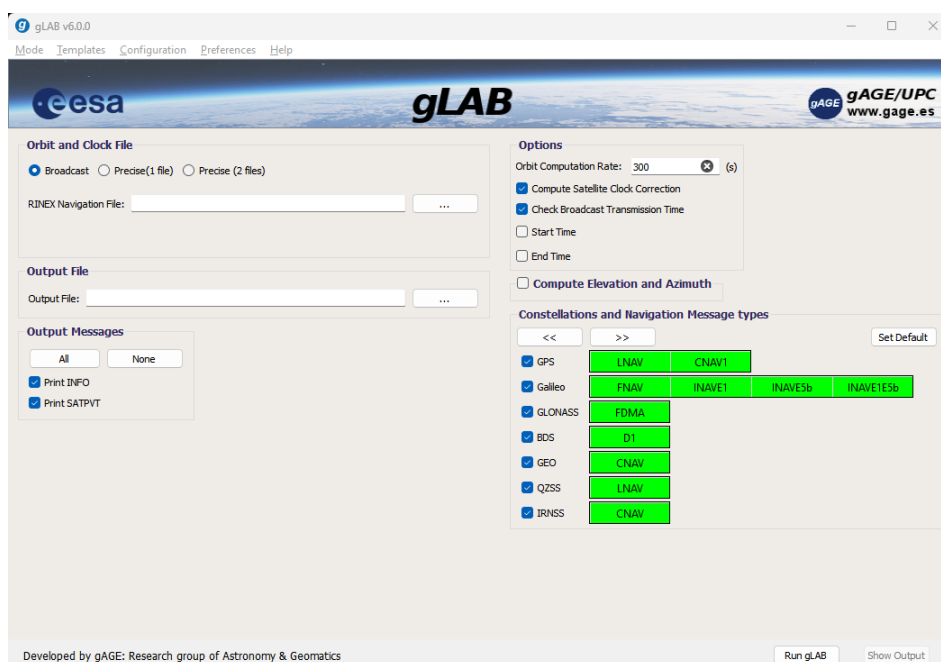


Figure 3.8: gLAB GUI Show Orbits & Clocks mode.

3.2.6 Add User Error to RINEX

This last mode adds user-defined error to measurements in RINEX observation files. The user-defined error type, duration and measurement to be added to is defined by a columnar text file (defined in the gLAB manual). With this mode, gLAB creates a new RINEX observation file with the user defined error (typically used to simulate satellite or receiver failures), which can be processed by gLAB or any other tool capable of reading a RINEX observation file.

The error types supported are the following:

- **Step:** Generated error is constant.
- **Ramp:** Generated error follows a ramp function.
- **Sinusoidal:** Generated error follows a sinusoidal function.
- **AWGN:** Generated error behaves as Additive White Gaussian Noise (AWGN).

The initial screen of add user error to RINEX mode can be seen in Fig. 3.9.

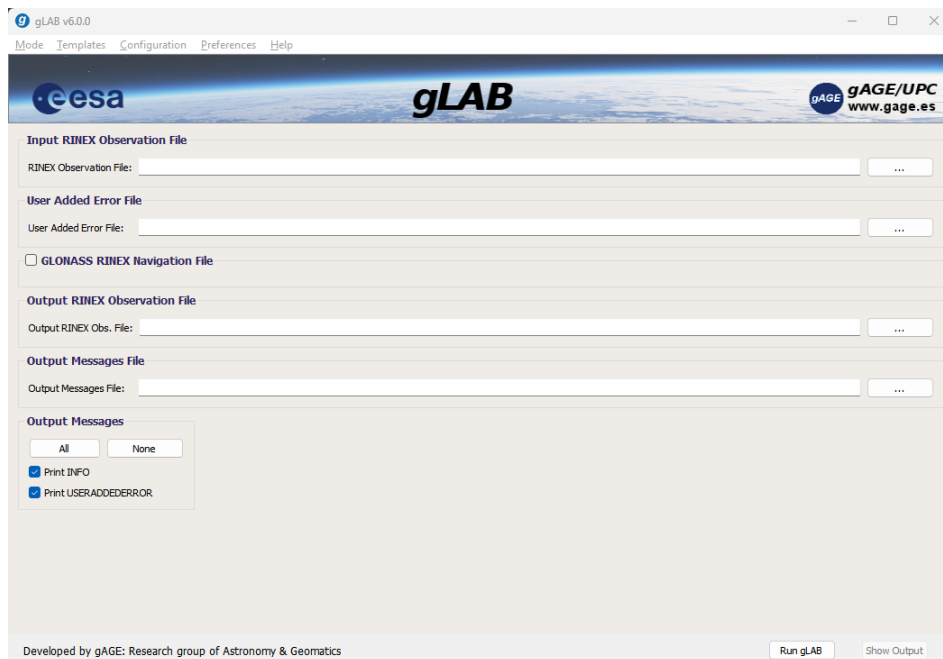


Figure 3.9: gLAB GUI Add User Error to RINEX mode.

3.2.7 GUI limitations

The GUI contains most of gLAB features, but some of them are too complex to implement in a GUI, or would clutter the interface or are too specific. The most important are:

- In “Positioning” mode, the user can only select cycle-slip detectors, SNR thresholds, measurement order and filter measurements on a per constellation basis, while in the DPC it is configurable on a per constellation and per satellite basis.
- In “Compare Orbits & Clocks” and “Show Orbits & Clocks” modes, the user can only select or unselect satellites on a per constellation basis, while in the DPC it is configurable on a per constellation and per satellite basis.
- The DPC mode “Show Input Mode” for only preprocessing RINEX observation files is not implemented in the GUI.
- The DPC mode “SBAS Plots” for computing SBAS availability and continuity maps is not implemented in the GUI.
- Fast-PPP processing mode in the DPC is not yet implemented in the GUI.

3.3 Data Processing Core

The DPC is the processing module of gLAB, and it was programmed in C, with no use of third-party libraries. Its main features are:

- Easy to use for an advanced user (as seen in section 3.2, minimum number of parameters are required when using default configuration).
- Contains a large amount of help (including input parameters, output messages format, Frequently Asked Questions (FAQ), SBAS maps computation, etc.).
- Supports parameter autocompletion (in a Bash shell only).
- Supports multi-threading.
- Very high processing speed.

The options of the DPC and GUI are basically the same with some exceptions that provide further flexibility. The DPC can be executed with the “-help” argument, which provides detailed information of the possible arguments. It is also worth mentioning that the DPC can also read the processing options from a configuration file, allowing easy repeatability of results and automatic batch processing.

The DPC can work in four different modes:

- **Positioning Mode:** This mode computes the navigation solution. It is automatically set when a RINEX observation file and orbits and clock are provided. Depending on the input files provided (broadcast orbits, precise orbits, SBAS files, etc.) it also automatically sets the default for each processing mode (SPP, PPP, etc.). Note that parameter names follow the structure in the GUI, as shown in section 3.2.1.
- **Converter:** This mode converts SBAS and RTCM files, as explained in section 3.2.3. It is triggered when only an SBAS or RTCM file is provided and the output formats are specified. Note that in the case of SBAS, it is possible to perform file format conversion also in positioning mode.
- **Product Comparison Mode:** This mode compares two orbit sources, as explained in section 3.2.4. It is triggered when only two products file (either broadcast or precise) are provided.
- **Show Product Mode:** This mode computes the satellite coordinates from a product file (broadcast or precise), as explained in section 3.2.5. It is triggered when only a single product file is provided.

- Show Input Mode:** This mode is designed to allow gLAB as a RINEX observation file preprocessor. In this mode, gLAB only prints INFO, MEAS and cycle-slip messages (cycle-slips by default are computed using the default PPP configuration). The MEAS message prints, for each satellite, the satellite block name and Space Vehicle Number (SVN) (if an ANTEX is provided), arc number, arc length, elevation, azimuth and finally a list of measurements. This list can be defined by the user (in both the order and which measurements are printed) on a per constellation and per satellite basis. As all measurements can be printed (pseudorange, carrier phase, SNR, Doppler and Loss of Lock Indicator (LLI) flag), the MEAS message effectively makes gLAB a RINEX to columnar text file converter, which can be easily read by any other tool. This mode is only available on command line, and is triggered when only RINEX observation file is provided. An example of the execution output of this mode is provided in Fig. 3.10.

```

MEAS 2021 006 300.00 00:05:00.00 GAL 2 GALILEO-2 211 1 1 0.000 0.000 2 C5Q:L5Q 28860411.3890 0.0000
MEAS 2021 006 300.00 00:05:00.00 GAL 7 GALILEO-2 207 1 301 0.000 0.000 2 C5Q:L5Q 25208383.0410 25208382.9288
MEAS 2021 006 300.00 00:05:00.00 GAL 12 GALILEO-1 102 1 301 0.000 0.000 2 C5Q:L5Q 25364195.4380 25364194.7614
MEAS 2021 006 300.00 00:05:00.00 GAL 14 GALILEO-2 202 1 301 0.000 0.000 2 C5Q:L5Q 24608135.1040 24608134.2153
MEAS 2021 006 300.00 00:05:00.00 GAL 19 GALILEO-1 103 1 301 0.000 0.000 2 C5Q:L5Q 26086933.7490 26086933.5469
MEAS 2021 006 300.00 00:05:00.00 GAL 20 GALILEO-1 104 1 1 0.000 0.000 2 C5Q:L5Q 0.0000 0.0000
MEAS 2021 006 300.00 00:05:00.00 GAL 21 GALILEO-2 215 1 301 0.000 0.000 2 C5Q:L5Q 28825766.1240 28825767.8356
MEAS 2021 006 300.00 00:05:00.00 GAL 26 GALILEO-2 203 1 301 0.000 0.000 2 C5Q:L5Q 24942217.4680 24942216.7522
MEAS 2021 006 300.00 00:05:00.00 GAL 27 GALILEO-2 217 1 301 0.000 0.000 2 C5Q:L5Q 26291684.9770 26291685.0838
MEAS 2021 006 300.00 00:05:00.00 GAL 30 GALILEO-2 206 1 301 0.000 0.000 2 C5Q:L5Q 25516066.9040 25516067.1577
MEAS 2021 006 300.00 00:05:00.00 GAL 33 GALILEO-2 222 1 301 0.000 0.000 2 C5Q:L5Q 24728183.4710 24728183.2859

```

Figure 3.10: gLAB DPC “Show Input Mode” example, with MEAS message set to print only Galileo satellites with C5Q and L5Q measurements. Parameters for this MEAS configuration would be ‘-pre:sat 0,+E0’ to select only Galileo satellites and ‘-print:meas:select E0-C5Q-L5Q’ to select C5Q and L5Q measurements on all Galileo satellites.

- SBAS plots:** This mode is for computing SBAS availability and continuity maps. This mode is only available on command line, and is triggered when an SBAS file and a RINEX navigation file are provided.

3.4 gLAB positioning example

In this section a full multi-constellation SPP example is described with gLAB. SPP algorithm is described in each constellation Interface Control Document (ICD) (USA Navigation Center [2022] for GPS, EU [2021] for Galileo, RISDE [2008] for GLONASS, CSNO [2019] for BDS, Cabinet Office, Government Of Japan [2021] for QZSS and ISRO [2017] for IRNSS). Furthermore, as the RINEX observation file includes a seventh constellation, the GEO constellation (which includes the navigation signals of all GEO satellites in view from all SBAS, but is not formally a constellation per se), gLAB can also use GEO in the SPP (with the same algorithm as GPS, but without a Total Group Delay (TGD) correction, as it is not needed for GEO signals RTCA-MOPS [2006]).

The first step is to set the “Positioning mode” (the initial mode on the GUI startup) in the “Mode” menu. Once in the correct mode, set the SPP template in the “Templates” menu (the initial positioning mode on the GUI startup).

On SPP, the only required fields to be input are the RINEX observation file (which includes the raw measurements) and the broadcast orbits. For the broadcast orbits can be provided through a single RINEX navigation file (which includes data on all constellations), or via multiple RINEX navigation files (up to one file per constellation). In Fig. 3.11 it is shown the GUI “Input” tab with the input files:

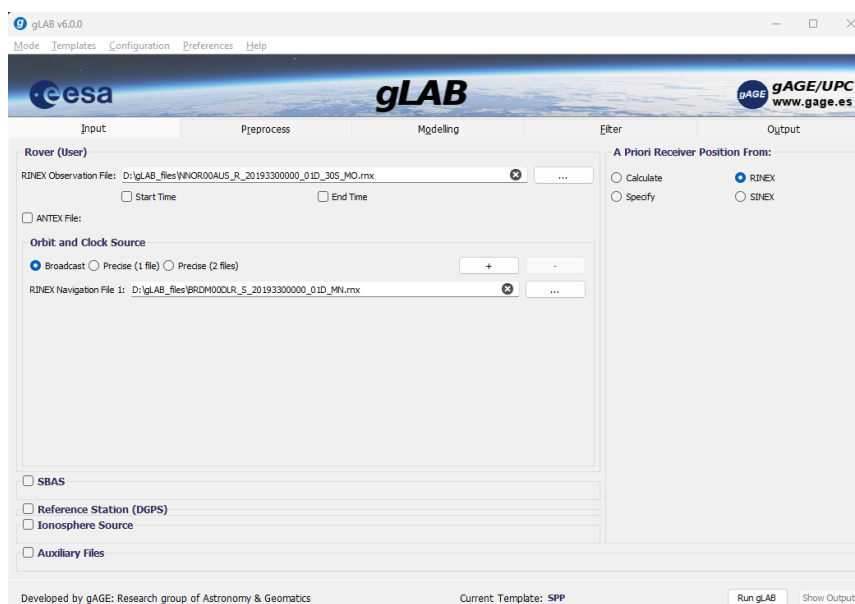


Figure 3.11: gLAB Input tab with the input files for multi-constellation SPP.

The station chosen for this example is “nnor”, located in New Norcia, Australia (with visibility of all GNSS constellations) on day 2nd of July of 2021 (Day of Year (DoY) 183). The station belongs to the International GNSS Service (IGS) network (Johnston et al [2017]), and their products can be downloaded from the Crustal Dynamics Data Information System (CDDIS) archive (Crustal Dynamics Data Information System [2023]). The RINEX observation filename is “NNOR00AUS_R_20211830000_01D_30S_MO.rnx” (which is a 30 second data rate file) and the RINEX navigation filename is “BRDM00DLR_S_20211830000_01D_MN.rnx”, which contains the broadcast navigation messages from all GNSS. These files follow the RINEX file naming convention (Romero [2021]).

The next step is to check the “Preprocess” tab. The only option to be modified is to enable GEO satellites, in order to use all constellations possible. Fig. 3.12 shows the configured GUI “Preprocess” tab:

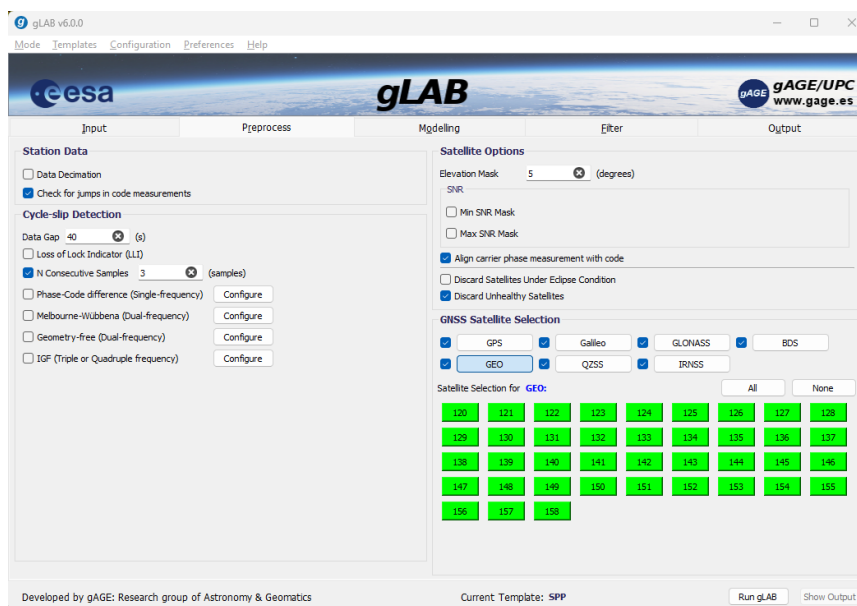


Figure 3.12: gLAB Preprocess tab configured to perform SPP with GEO satellites.

In the “Modelling” tab, the various models can be selected to compensate for any errors produced by propagation and electronic effects in the signal, such as atmospheric refraction. Also, the types of navigation messages to be used can be selected. The default options can be used, as show in Fig. 3.13.

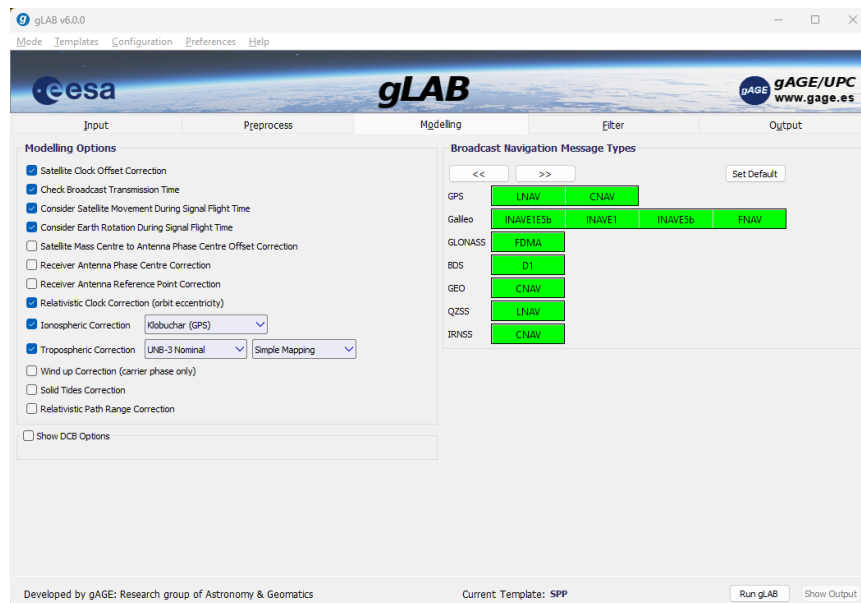


Figure 3.13: gLAB Modelling tab configured to perform SPP.

The “Filter” tab is where the satellite measurements are selected and the options for the Kalman filter (Kalman [1960]) are located. The default options are used to let gLAB decide automatically which measurements to use in the navigation solution, as show in Fig. 3.14.

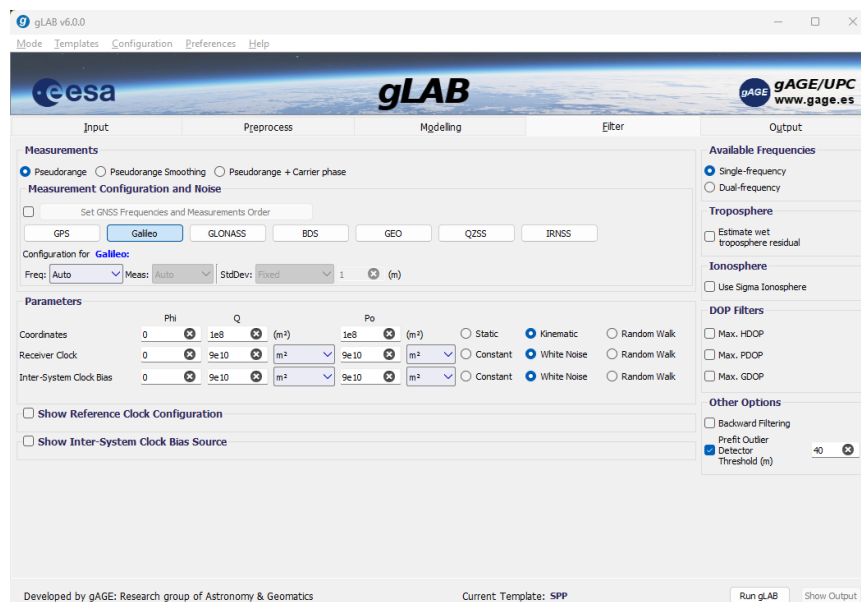


Figure 3.14: gLAB Filter tab configured to perform SPP.

Finally, the “Output” tab, where the gLAB output file can be set (by default to “gLAB.out” in the same directory as the input RINEX observation file). The default configuration for the output messages is to print only the INFO message (containing the internal gLAB configuration), OUTPUT message (containing the navigation solution) and the statistical summary. The interface should look like Fig. 3.15.

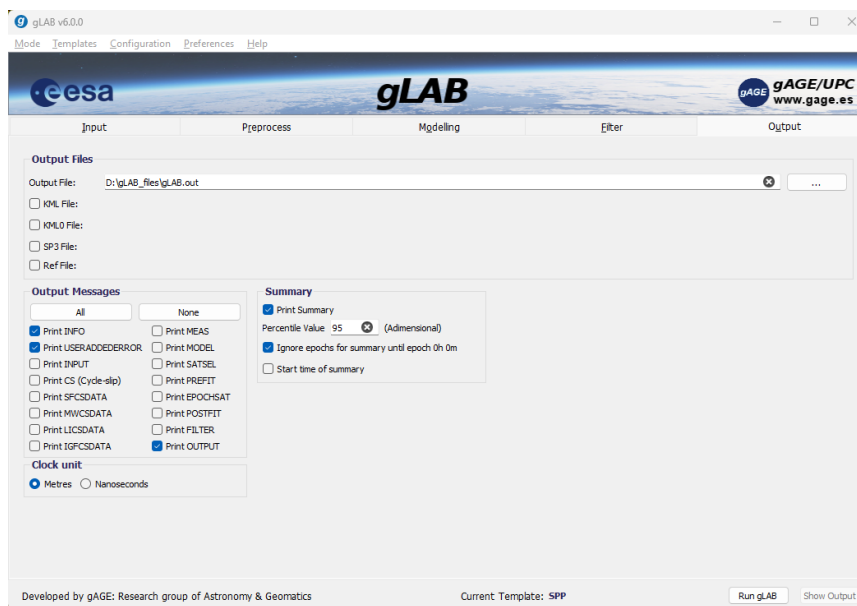


Figure 3.15: gLAB Output tab configured to perform SPP.

With the configuration set, the user has to click on “Run gLAB” button on the bottom right-hand side of the screen to run the DPC. While the DPC runs, the bottom bar of the GUI switches to a progress bar until the process is finished.

Once the run finishes, clicking on “Show Output” opens the gLAB output file in a text editor. At the end of the file there is a statistical summary, with information regarding epochs processed, satellites used, positioning errors, DOP, etc.).

To plot the North East Up (NEU) positioning error (see Appendix B of [Sanz et al \[2013a\]](#) for coordinate systems description), the user has to change to “Analysis” mode in the “Mode” menu, then click on “NEU positioning error” to set the default options for this plot (note that the source file changes automatically to the gLAB output file set in the “Output” tab) and finally click on “Plot” on the bottom right-hand side of the screen to run the DAT. The resulting plot is shown in Fig. 3.16.

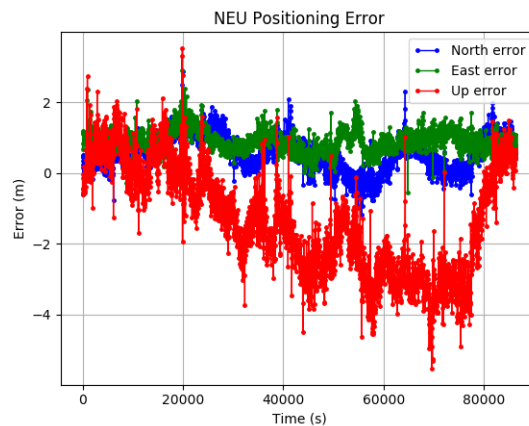


Figure 3.16: NEU positioning error for 24 hours on the 2nd of July 2021 for station “nnor” with 6 constellations and GEO satellites. The processing involves up to 62 satellites in view.

In Fig. 3.16, the North and East error (corresponding to a 2D positioning) is most of day below ± 2 metres, while the Up component is noticeably higher (up to 5 metres). The higher error in the latter component is mainly due to the inaccuracy of the ionospheric modelling (e.g. Klobuchar and NeQuick models) performed in single-frequency processing. Moreover, owing to the satellites in view providing signals in many directions except in the Up vertical, because the Earth blocks signals from satellite on the other side of the planet. Indeed, the user sees many satellites around its position in North and East axes (compensating each other’s errors on these axes), but not in the Up axis, for the aforementioned reason.

Peak errors that appear in the plot are linked to several causes, such as code multipath (see [USA Navigation Center \[2022\]](#)). Note also that error in all components is slightly biased. This is due to the default options use as an a priori receiver position (used to compute the error) the coordinates in the RINEX observation file header, which have an error around the metre level. To remove these biases, IGS also provides daily files with the precise station coordinates in the Solution (Software/technique) INdependent EXchange format (SINEX) file format (for this example, the corresponding file is “igs19P20812_all.snx” and the result can be seen in Fig. 4.2).

Finally, from the statistical summary at the end of the gLAB output file some metrics can be checked. For instance, the maximum number of satellites used (in one epoch) is 62 (11 GPS, 9 Galileo, 9 GLONASS, 26 BDS, 3 QZSS and 4 IRNSS), 6 or more constellations were used in 99.2% of epochs, Precision Dilution Of Precision (PDOP) 95 percentile of 0.8 and a 95 percentile

positioning error of 1.9 metres, with a total process runtime of 1.6 seconds (on an Intel i9-12900KS Central Processing Unit (CPU)).

3.5 Contributions to gLAB code

gLAB is currently a large program. In order to provide a rough idea of gLAB size, on table 3.1a the number of lines code in gLAB modules is provided, while table 3.1b provides the number of pages in gLAB documentation:

	Number of lines
DPC	140,000
GUI	105,000
DAT	10,000
Total	255,000

(a) Number of lines in gLAB modules.

	Number of pages
gLAB Manual	540
GUI Tooltips	200
Total	740

(b) Number of pages in gLAB documentation.

Table 3.1: Approximate number of lines and documentation pages in gLAB.

As for most large programs, gLAB also has had several contributors to its code:

- **Prof. Dr. Jaume Sanz Subirana:** Provided most of the technical GNSS knowledge, as well as provided multiple FORMula TRANslator (FORTRAN) prototypes for testing against gLAB.
- **Dr. Pere Ramos Bosch:** Developed the first version of gLAB and the following versions up to version 2.0.0.
- **Dr Adrià Rovira Garcia:** Developed the first version of the GUI (in Python 2), and has been the main gLAB user and tester since then.
- **Dr. Yixie Shao:** Developed the prototypes for RTCM decoder, the SBAS plots and Worst Integrity Ratio (WIR) maps (see Ibáñez et al [2018]) in the DAT.
- **Mr. Jesús Romero Sánchez:** Developed the DGNSS module, updated cycle-slips algorithm, integrated Yixie's RTCM decoder in gLAB and developed the first half of the new Qt GUI in version 5.0.0.
- **Dr. Ivan Lapin:** Developed the procedure to compile gLAB and to create the installable ".dmg" file in Mac Operative System (OS).

- **Dr. Àngela Aragon Angel and Mr. Enrique Arcediano Garrido:** Developed the NeQuick-G code integrated into gLAB ([Aragon-Angel et al \[2021\]](#)).
- **Mr. Mowen Li:** Upgraded the Qt GUI to multi-constellation (except for the tooltips).

The PhD author has been in charge of gLAB since version 2.0.0, maintaining and updating all the code (including all the code by the contributors), providing support to users and creating all the installable files for the gAGE courses using gLAB.

gLAB state-of-the-art algorithms

This chapter is devoted to providing an overview of the state-of-the-art GNSS algorithms implemented in gLAB. Namely SPP, PPP, SBAS 1F and DGNSS (algorithms implemented in gLAB are detailed in chapter 5).

4.1 Standard Point Positioning

Standard Point Positioning (SPP) algorithm is the standard strategy defined in the ICDs of all constellations ([USA Navigation Center \[2022\]](#) for GPS, [EU \[2021\]](#) for Galileo, [RISDE \[2008\]](#) for GLONASS, [CSNO \[2019\]](#) for BDS, [Cabinet Office, Government Of Japan \[2021\]](#) for QZSS and [ISRO \[2017\]](#) for IRNSS). The advantages of this algorithm are that it only requires the data provided in the broadcast navigation message and that it requires low computational power.

SPP uses a single pseudorange measurement in one frequency (L1 for GPS, QZSS, E1 for Galileo, G1 for GLONASS, B1I for BDS and L5 for IRNSS) in each satellite. The pseudorange needs to be corrected with the following terms (as shown in Fig. 4.1):

- **Geometric distance:** The distance between the receiver position and the satellite position (computed using the broadcast navigation message).
- **Satellite clock:** The offset between the satellite clock and the constellation master clock, provided in the broadcast navigation message.
- **Satellite instrumental delay:** The delay produced by the satellite hardware itself in the signal generation, provided in the broadcast

navigation message (except for GLONASS and GEO constellations).

- **Antenna Reference Point (ARP):** If the receiver antenna is fixed to the ground, a reference point is used to compute the receiver coordinates, known as the Monument Marker (MM). If the receiver needs to be slightly moved (for instance to elevate the antenna in order to reduce multipath), the ARP corrections accounts for the difference between the MM and its new position. This correction is provided in the header of the RINEX observation file. In Fig. 4.3 a layout of a receiver antenna is shown.
- **Relativistic clock offset:** The delay produced due to the difference in the gravitational potential between receiver and satellite locations (general relativity), as described in Sanz et al [2013a]. The only exception is GLONASS, which already incorporates this correction in their broadcast clock correction (Gunning et al [2017]).

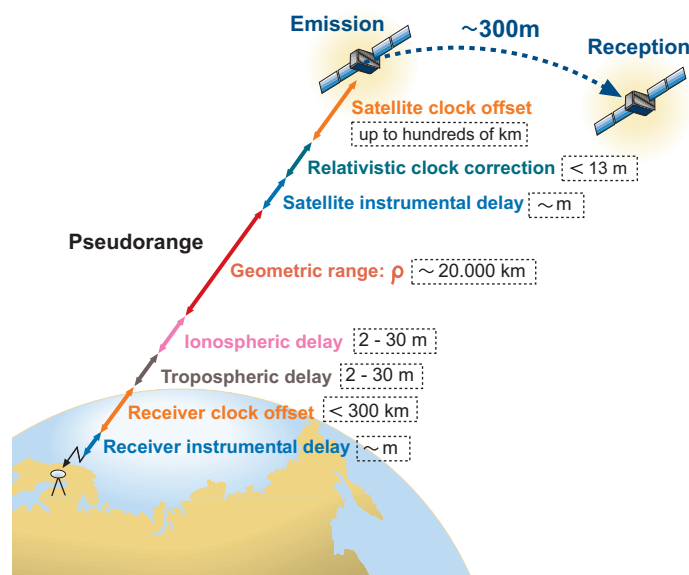


Figure 4.1: Pseudorange measurement contents (from Sanz et al [2013a]).

- **Ionosphere:** The delay produced by the refraction in the upper part of the atmosphere located from 60 km to more than 1,000 km (from Sanz et al [2013a]). Each constellation provides its own model (except GLONASS). GPS, QZSS, BDS-2 and IRNSS use ionospheric models based on Klobuchar model (Klobuchar [1987]), Galileo uses the NeQuick-G model (European Union [2016]) and BDS-3 uses the BeiDou Global Ionospheric delay correction Model (BDGIM) (CSNO [2017]). As any ionosphere model can be applied to any satellite (independently of the constellation), gLAB applies the selected model to all constellations (for

simplicity for the user). Broadcast ionosphere models from GPS, Galileo and BDS-2 are implemented in gLAB. The IONosphere map EXchange format (IONEX) files (Schaer et al [1998]) and SBAS 1F can also be used as ionosphere models.

- **Troposphere:** The delay produced by atmospheric layer between Earth's surface up to an altitude of 60 km (from Sanz et al [2013a]). For SPP, gLAB has implemented the model from SBAS (RTCA-MOPS [2006]) (as no model is broadcast in the navigation message of any constellation). This model is applied to all satellites of all constellations (for simplicity for the user).

In SPP, the receiver coordinates (dx, dy, dz) together with the receiver clock offset (δt) are estimated using Weighted Least Squares (WLS) or a Kalman filter in kinematic mode (method implemented in gLAB), as shown in Sanz et al [2013a]. When multiple constellations are used, one of the constellations is selected as the reference clock (which will be the receiver clock offset parameter), and the rest of the constellations will each have an additional parameter to estimate (common to each satellite in a constellation), named Inter System Clock Bias (ISCB) (δt_{const}), which is the difference between the current constellation master clock and the reference constellation master clock. In SPP, these ISCB have to be estimated as *white noise* (Sanz et al [2013a]), as each master clock constellation will not vary in a predictable way with respect to the reference constellation master clock. The parameters to be estimated are shown in vectorial form in equation 4.1:

$$\mathbf{x} = \left[dx, dy, dz, c \delta t, \underbrace{c \delta t_{const}^2, \dots, c \delta t_{const}^s}_{ISCB} \right]^T \quad (4.1)$$

Where:

- c : Speed of light (299792458.0 m/s) for converting seconds to metres.
- s : Number of systems.
- dx, dy, dz : Receiver coordinates.
- $c \delta t$: Receiver reference clock (e.g. GPS clock).
- $c \delta t_{const}^i$: ISCB for system i (except for the reference one, set as the first system in the list).

The advantages of SPP are its low complexity, it only requires the broadcast navigation message (allowing worldwide usage) and it has no convergence time, as it works in single epoch. Two examples of SPP positioning are shown in Figs. 4.2a and 4.2b, with the same dataset¹ as in Fig. 3.16 but using precise receiver coordinates from a SINEX file (IERS [2006]).

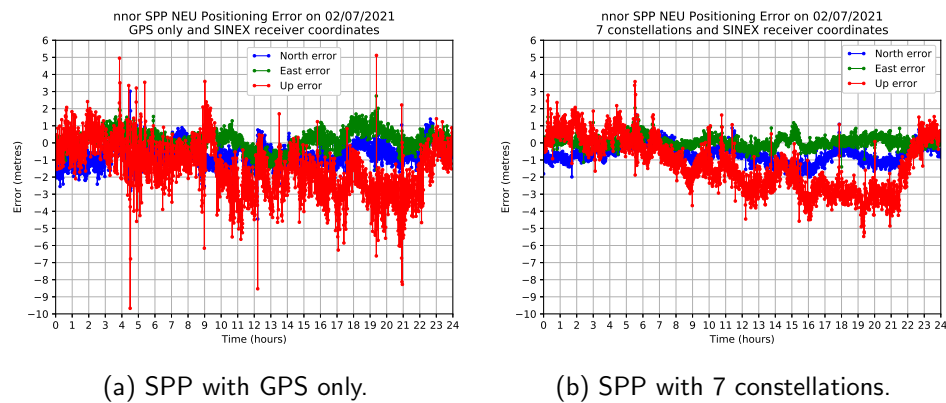


Figure 4.2: SPP example in “nnor” station on 02/07/2021, on the left-hand with GPS only and on the right-hand with 7 constellations (GPS, GLONASS, Galileo, GEO, BDS, QZSS and IRNSS). SINEX receiver coordinates are used in both cases as a ground truth.

The solution with full multi-constellation in Fig. 4.2b has lower positioning error and variance than the GPS only solution in Fig. 4.2a due to the improved geometry (the median satellites used is 9 for GPS only and 52 for multi-constellation), as it is shown in table 4.1 (with the values extracted from the gLAB statistical summary). Processing times for GPS only is 0.5 seconds and 1.6 seconds for multi-constellation (on an Intel i9-12900KS CPU). Note that PDOP, Geometric Dilution Of Precision (GDOP), Horizontal Dilution Of Precision (HDOP) and Vertical Dilution Of Precision (VDOP) components improve very significantly, but the Time Dilution Of Precision (TDOP) worsens due to the additional ISCB parameters in the filter estimation. This improvement in the DOP does not translate in a similar improve in the navigation solution error, as the error is driven by other factors (primarily the ionosphere and satellite clocks).

¹The corresponding files are available at the following URL: https://server.gage.upc.edu/PhD/Ibanez_Deimos/Ibanez_Deimos_PhD_files.7z

	95 percentile	
	GPS only	All constellations
PDOP	2.62	0.82
GDOP	3.09	1.98
TDOP	1.63	1.81
HDOP	1.27	0.47
VDOP	2.31	0.68
Horizontal Error (m)	1.79	1.51
Vertical Error (m)	3.92	3.56
3D Error (m)	4.08	3.74

Table 4.1: SPP 95 percentile comparison of errors and DOPs between GPS only and all constellations.

4.2 Precise Point Positioning

PPP (Zumberge et al [1997]) is a well established algorithm to achieve centimetre level in static positioning and decimetre level in kinematic positioning (Sanz et al [2013a]). In order to achieve this accuracy, PPP has several changes with respect to SPP:

- Two pseudoranges and two carrier phases measurements are used from two different frequencies. The usual frequencies are (indicated in their respective ICD): L1 and L2 in GPS , QZSS, G1 and G2 in GLONASS, E1 and E5a in Galileo, B1I and B2 for BDS-2, B1C and B2a in BDS-3, L5 and S for IRNSS.
- Both pseudoranges and carrier phases are combined using the Ionosphere Free (IF) combination, which removes about 99.9% of the ionosphere delay (Sanz et al [2013a]), therefore it is not needed to apply any ionospheric model.
- As carrier phases are used, it is necessary to estimate the carrier phase ambiguities (Sanz et al [2013a]) as constant parameters along continuous carrier phase arcs, but their value will change when a cycle-slip occurs. Therefore, it is necessary to implement cycle-slip detectors. There are several cycle-slip detectors depending on the number of frequencies used:
 - **Single Frequency (SF):** Only one frequency is used. It must be run independently in each carrier phase used to compute the navigation solution. gLAB implements the SF cycle-slip detector described in Sanz et al [2013a].
 - **Dual Frequency (DF):** Two frequencies are used. gLAB implements Melbourne–Wübbena (MW) and the Geometry Free (GF) cycle-slip detectors (described in Sanz et al [2013a]), which are the default cycle-slip detectors in PPP.
 - **Triple or quadruple frequency:** Three or four frequencies are used. The cycle-slip detectors implemented in gLAB are described in chapter 6.2.
- A simpler nominal tropospheric blind model is used, and a residual term is later estimated in the filter as an additional parameter (the zenith wet tropospheric delay), which is common to all satellites. The model used in gLAB for PPP is from GPS Inferred Positioning SYstem and Orbit Analysis and Simulation Software (GIPSY-OASIS) II software (Webb and Zumberge [1993]).

- The ISCB is typically estimated as constant parameter, as the clock correction for all constellations provided in SP3 or Clock (CLK) files refer to the common clock used to estimate the clock correction.

Moreover, there are additional differences in the modelling terms with respect to SPP:

- **Geometric range:** Satellite position are computed using precise orbits (in SP3 format, defined in [Hilla \[2016\]](#)), provided by an entity external to the GNSS service provider (e.g. Centre for Orbit Determination in Europe (CODE), as shown in [Prange et al \[2020\]](#)). Moreover, as SP3 file provides the satellite coordinates referred to the Centre of Mass (CoM) of the satellite, it is necessary to convert them to the satellite Antenna Phase Centre (APC). This correction is provided in the ANTEX file ([Rothacher and Schmid \[2010\]](#)), which is also provided by an external entity (e.g. IGS).
- **Receiver APC:** The receiver antenna also needs to be corrected to account for the elevation, azimuth and frequency with respect to the apparent source of radiation. This correction is also provided in the ANTEX file. In Fig. 4.3 a layout of a receiver antenna is shown.

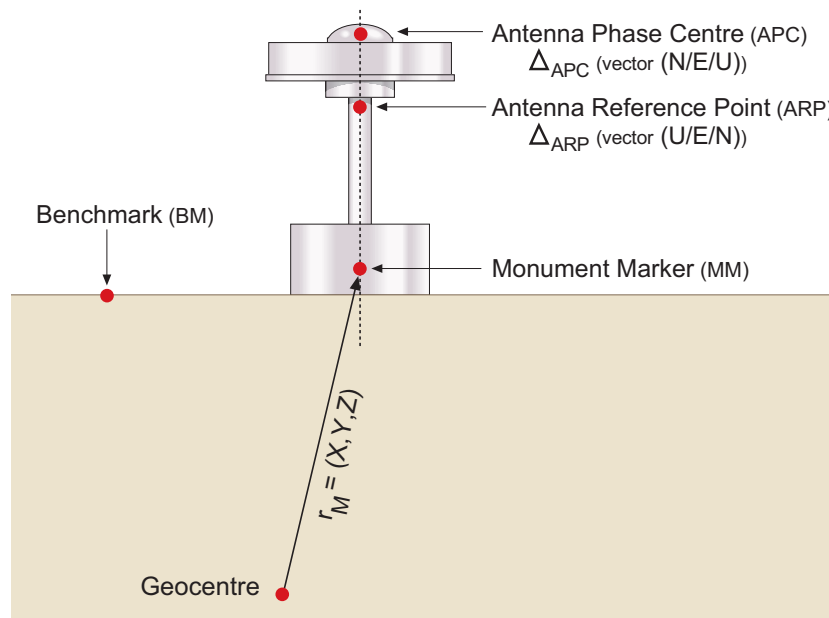


Figure 4.3: Layout of a fixed antenna showing the ARP, APC and MM. The BenchMark (BM) is an additional external reference point (from [Sanz et al \[2013a\]](#)).

- **Satellite APC:** The APC is defined as the *apparent source of radiation* (Sanz et al [2013a]), which depends on the elevation, azimuth and frequency of the signal. When converting the satellite coordinates from CoM to APC, it is first applied a correction to convert to the apparent source of radiation, and then an additional correction is performed (also provided in the ANTEX file) to account for the APC differences in elevation, azimuth and frequency.
- **Satellite clock:** Satellite clock correction is obtained from SP3 files or from CLK files (Ray and Gurtner [2010]), which have a higher data rate and are provided together with the SP3 files.
- **Satellite instrumental delay:** If the IF combination is the same as the one used to estimate orbits and clocks in SP3 and CLK file, there is no need to apply this correction. Otherwise, a Differential Code Bias (DCB) correction has to be applied.
- **Wind-up:** Delay only on carrier phase due to the rotation of the Line of Sight (LoS) vector in relation to the antenna. It is corrected following Wu et al [1993].
- **Solid tides:** To account for the elasticity of the Earth mantle due to Moon and Sun attractions. It is corrected following McCarthy and Petit [2004].
- **Relativistic path range correction:** An additional small effect (around 4 cm) due to the difference of the gravitational potential (special relativity). It is corrected following Ashby, N. [2003].

In PPP it is necessary to implement a Kalman filter, as there are several parameters to be estimated with different stochastic behaviour. Carrier phase ambiguities (B) and ISCB (δt_{const}) are constant parameters, the zenith wet tropospheric delay ($\Delta T r_{z,wet}$) is estimated as random walk (slow varying term) and the receiver coordinates (dx , dy , dz) can be estimated as static, random walk or kinematic depending on the receiver behaviour. The parameters to be estimated are shown in vectorial form in equation 4.2:

$$\mathbf{x} = \left[dx, dy, dz, c \delta t, \underbrace{c \delta t_{const}^2, \dots, c \delta t_{const}^s}_{ISCB}, \Delta T r_{z,wet}, \underbrace{B^1, \dots, B^{n[1]}}_{\substack{\text{Ambiguities} \\ \text{const 1}}}, \dots, \underbrace{B^j, \dots, B^{n[s]}}_{\substack{\text{Ambiguities} \\ \text{const s}}} \right]^T \quad (4.2)$$

Where:

- c : Speed of light (299792458.0 m/s) for converting seconds to metres.
- s : Number of systems.
- $n[i]$: Number of satellites in system i ($i = 1, \dots, s$).
- dx, dy, dz : Receiver coordinates.
- $c\delta t$: Receiver reference clock (e.g. GPS clock).
- $c\delta t_{const}^i$: ISCB for system i (except for the reference one, set as the first system in the list).
- $\Delta Tr_{z,wet}$: Zenith wet tropospheric delay.
- B^j : Carrier phase ambiguity, associated to ionosphere free combination of satellite j in system i .

The advantages of PPP are its high accuracy (centimetre level in static positioning or decimetre level in kinematic positioning) and the fact that it has global coverage. The drawback is that it requires external products (e.g. SP3) therefore requiring access to a service provider (e.g. internet access to IGS). Moreover, PPP presents a greater computational complexity and it has a long convergence time (around 1 hour in GPS, from Rovira-Garcia et al [2015]) due to the fact that estimation of the carrier phase ambiguities is linked to the satellite geometry, which varies slowly with respect to the user.

In Fig. 4.4 there are PPP examples² with a fixed receiver in static and kinematic modes with both GPS only and with 5 constellations (GPS, Galileo, GLONASS, BDS and QZSS). QZSS and GEO are not included in the processing examples due to unavailability of precise products for these constellations (but gLAB is capable to handle them). The GNSS measurements are the same as in Fig. 3.16, but with precise orbits and clocks (files “COD0MGXFIN_20211830000_01D_05M_ORB.SP3” and “COD0MGXFIN_20211830000_01D_30S_CLK.CLK”, also downloaded from IGS).

²The corresponding files are available at the following URL: https://server.gage.upc.edu/PhD/Ibanez_Deimos/Ibanez_Deimos_PhD_files.7z

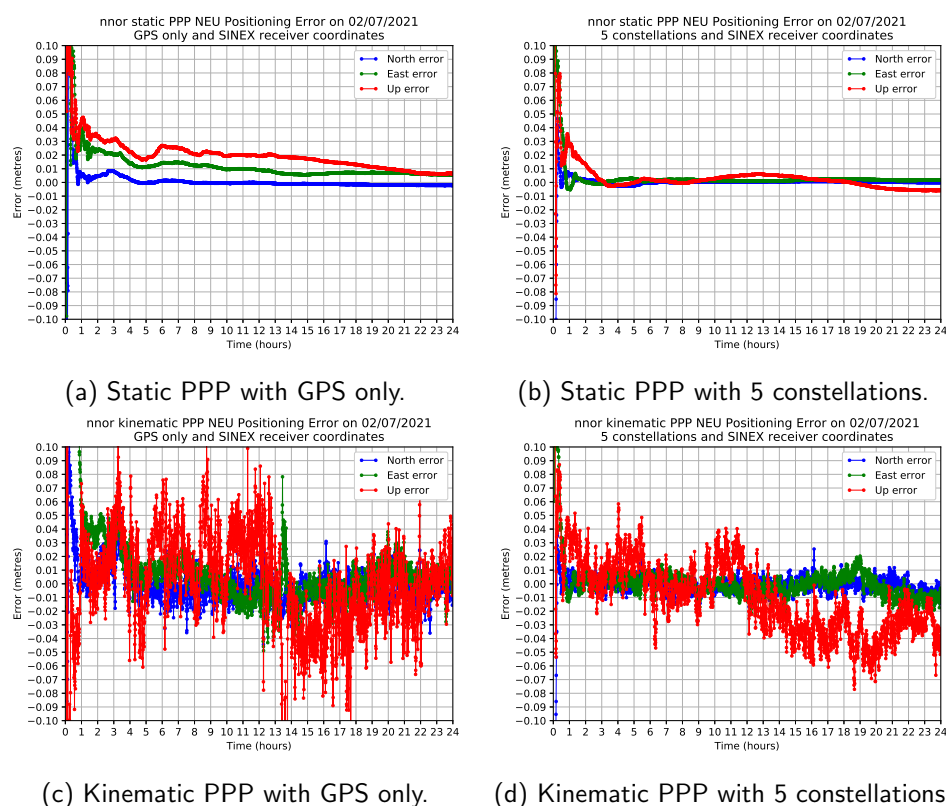


Figure 4.4: PPP example in “nnor” station on 02/07/2021. Top row is in static mode, while bottom is in kinematic mode. Left-hand side is with GPS only and right-hand side with 5 constellations. SINEX receiver coordinates are used in all cases.

The processing time for GPS only is 1.2 seconds and 2.2 seconds for multi-constellation (on an Intel i9-12900KS CPU). Note that for the multi-constellation case, the APC correction has been manually set (see section 6.7 for more details). The median of satellites used in the navigation solution is nine for GPS only and 39 for multi-constellation.

To compare the performances of GPS single and multi-constellation navigation solutions, the convergence time metric is used (provided in the statistical summary of the gLAB output file). In gLAB, the convergence time is computed in four different ways (independently for the horizontal, vertical and 3D components):

1. **Formal error:** The first epoch that the square root of the trace of the position covariance matrix is below a configurable threshold. This value does not necessarily match the actual error, but it has the advantage that

no receiver position is needed.

2. **Formal error maintained for a given time period:** The first epoch that the square root of the trace of the position covariance matrix is below a configurable threshold and it is maintained during a user defined time period. This time period is to ensure that the measured time is not affected by the value fluctuation during the convergence time.
3. **Positioning error:** The first epoch that the positioning error is below a configurable threshold. Precise receiver coordinates are needed for this method, as they are used for computing the positioning error.
4. **Positioning error maintained for a given time period:** The first epoch that the positioning error is below a configurable threshold and it is maintained during a user defined time period.

In this example the last method is used. The positioning error threshold is set to 1 cm (for each component) in static mode (the maximum accuracy in static PPP), 20 cm (horizontal) and 40 cm (vertical and 3D) in kinematic mode (same as Galileo HAS, from EUSPA [2023d]). The time period to keep the positioning error below the threshold is the gLAB default, 300 seconds (i.e. five minutes).

	Static positioning		
	GPS only (9 satellites)	5 constellations (39 satellites)	Difference
Horizontal (1 cm)	9h 41m 30s	0h 41m 0s	9h 0m 30s (-92.9%)
Vertical (1 cm)	19h 53m 30s	2h 8m 0s	17h 45m 30s (-89.3%)
3D (1 cm)	21h 32m 30s	2h 8m 30s	19h 24m 0s (-90.1%)

(a) Convergence time comparison in static PPP.

	Kinematic positioning		
	GPS only (9 satellites)	5 constellations (39 satellites)	Difference
Horizontal (20 cm)	32m 0s	4m 30s	27m 30s (-85.9%)
Vertical (40 cm)	9m 0s	2m 0s	7m 0s (-77.8%)
3D (40 cm)	14m 30s	3m 0s	11m 30s (-79.3%)

(b) Convergence time comparison in kinematic PPP.

Table 4.2: Convergence time comparison in static and kinematic PPP. The median number of satellites used is provided between parenthesis in the second row, and the positioning error thresholds between parenthesis in the first column.

From table 4.2, the convergence in multi-constellation is reduced around 90% in static and 80% in kinematic with respect to GPS only, which shows the high correlation between geometric variation and convergence time. The improvement in geometry is shown in table 4.3:

	95 percentile			
	Static		Kinematic	
	GPS only	5 constellations	GPS only	5 constellations
PDOP	2.91	1.01	2.91	1.01
GDOP	3.44	1.72	3.44	1.72
TDOP	1.79	1.41	1.79	1.41
HDOP	1.44	0.56	1.44	0.56
VDOP	2.59	0.86	2.59	0.86
Horizontal Error (m)	0.03	0.01	0.05	0.02
Vertical Error (m)	0.04	0.02	0.07	0.05
3D Error (m)	0.05	0.02	0.10	0.05

Table 4.3: PPP 95 percentile comparison of errors and DOPs between GPS only and five constellations in both static and kinematic modes.

4.3 Satellite-Based Augmentation System single frequency

GNSS, in standalone positioning, cannot meet the necessary requirements for some specific use cases. One of these cases is civil aviation, which has stringent requirements because it is an SoL application, which has the following requirements:

- Integrity:** The definition from ICAO [2006] is: *Integrity is a measure of the trust that can be placed in the correctness of the information supplied by the total system. Integrity includes the ability of a system to provide timely and valid warnings to the user (alerts) when the system must not be used for the intended operation (or phase of flight).* Thus, integrity measures the probability of utilizing erroneous information. As the goal is to ensure that the information provided can always be trusted by the navigation system, there has to be a quality check of the input data, and in the case there is unsafe data, the system must provide a protection

against or warn the user (time-to-alert) as soon as possible (e.g. 10 seconds for civil aviation in precision approach using GNSS). In the case of GNSS, satellite malfunction can occur, but the only way to warn the user is through the navigation message, which can take up to 2 hours until the navigation message gets updated with the health flag set to “don't use”. It is measured as the probability of receiving a false alarm (see table 4.4).

- **Availability:** The definition from ICAO [2006] is: *The availability of GNSS is characterized by the portion of time the system is to be used for navigation during which reliable navigation information is presented to the crew, autopilot, or other system managing the flight of the aircraft.* High Availability (HA) systems are designed to be online at least 99.999% of time within a year (TechTarget [2021]), or in the words, the acceptable downtime is 5 minutes and 15 seconds per year. From the worldwide GNSS in FOC in the past years (GPS and GLONASS), only GPS has met this requirement (see GPS performance reports on GPS.gov [2023a]), as GLONASS experienced a constellation fault in 2014 (Beutler et al [2014]).
- **Accuracy:** The definition from ICAO [2006] is: *GNSS position error is the difference between the estimated position and the actual position. For an estimated position at a specific location, the probability should be at least 95 per cent that the position error is within the accuracy requirement.* It measures the Navigation System Error (NSE). The level of accuracy required varies in function of the specific use case.
- **Continuity:** The definition from ICAO [2006] is: *Continuity of service of a system is the capability of the system to perform its function without unscheduled interruptions during the intended operation.* Measures the probability that the system will keep its performance during the duration of a phase of operation, as any transition from nominal to not nominal will result in an abort of the current operation (with the risk implied). GNSSs were not designed to provide a specific continuity goal, therefore no continuity can be assured without augmentation systems.

During the development of SoL systems for GNSS in the early 2000s, the only constellations available were GPS and GLONASS, both of which were emitting only in L1 in L2 frequencies, with a civil signal only on L1. As SoL requires the use of frequencies in the ARNS bands (recall Fig. 2.3, so the frequency is protected by law against interferences) and the use of a civil signal (as military signals may be modified according to the military needs), SoL systems were limited to use L1 frequency with C1C signal. This limits the accuracy to around 3 to 10 metres (see Fig. 4.2a) for a standalone receiver. Moreover, even though ICAO [2006] standard supports both GPS and GLONASS, only GPS is used,

as it was the only constellation meeting the availability requirements. Thus, to meet SoL requirements, a receiver needs additional data containing the following information:

- **Differential corrections:** The corrections to improve navigation message orbits and clocks precision.
- **Confidence bounds:** The confidence bounds to provide information on the quality of the corrections (measure of trust), while also provides information on which satellites can be used in the navigation solution.
- **Ionosphere model:** Broadcast ionosphere models (like Klobuchar) have low accuracy, and as ionosphere is one of the main signal delay and error sources, it is necessary to use a better ionosphere model to improve the navigation solution accuracy.

This additional data needs to be computed by an external system (neither GPS or GLONASS were designed with this capability) and sent to the user in real time. These systems are called augmentation systems, and for SoL purposes, there are currently three types:

- **GBAS:** Ground-Based Augmentation System. This augmentation system broadcast corrections in a local area, as they are limited by the range of the Ultra High Frequency (UHF) radio link (used to broadcast the corrections), which is forty-three kilometres (from [SC-159 \[2005\]](#)). It is installed in airports, and each airport GBAS is independent from any other. Due to the small coverage area, this system does not need to provide an ionospheric model, as it can be assumed that the error produced in the aeroplane measurement is the same as the error measured in the base station at the airport (although it is evolving to estimate ionospheric gradients using external networks, as described in [Caamaño \[2022\]](#)).
- **SBAS:** Satellite-Based Augmentation System. This augmentation system broadcast corrections in a wide area (continents) through geostationary satellites.
- **ABAS:** Aircraft-Based Augmentation System. The GNSS augmentation is fully performed on the aircraft, which is entirely responsible for the integrity monitoring.

As gLAB only has SBAS algorithm implemented, the rest of the present section will focus only on SBAS.

The current SBAS single frequency (or SBAS 1F) systems in operation or in development are as follows (as shown in [Fig. 4.5](#)):

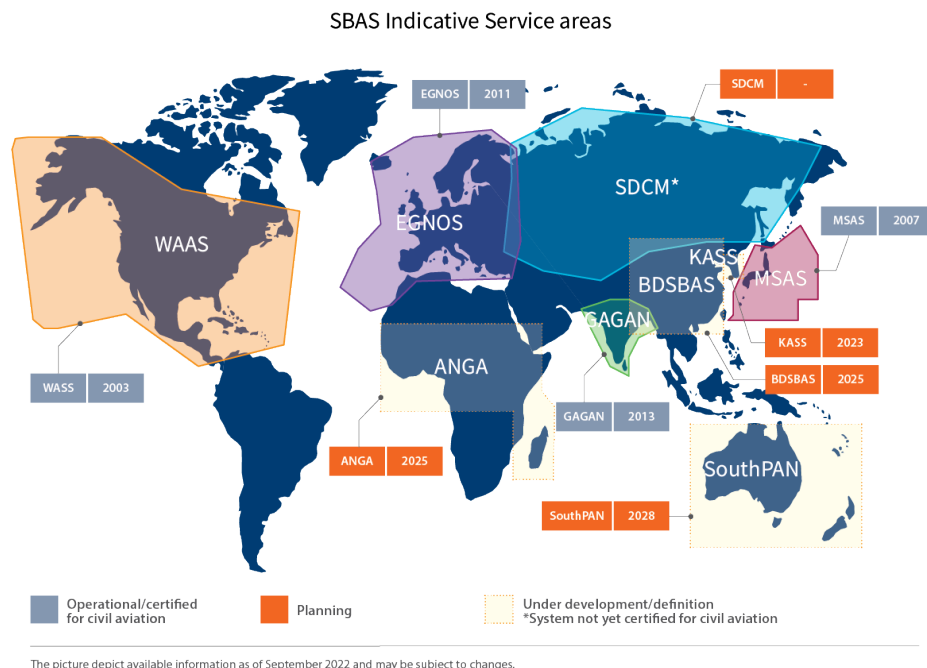


Figure 4.5: SBAS service areas (from EUSPA [2022c]).

- **WAAS:** Wide Area Augmentation System (WAAS) is the first SBAS developed and fully operational. It was developed by the Federal Aviation Administration (FAA). It became operational in 2003 (Navipedia [2011c]), initially providing service only to CONTinental United States (CONUS) area. On September 2007, coverage was extended to Canada and Mexico (FAA SatNav News [2007]).
- **EGNOS:** European Geostationary Navigation Overlay System (EGNOS) is the Europea SBAS. It provides service to the whole European Civil Aviation Conference (ECAC) area (see ECAC [2016]). It has been operational since 2011 (from Navipedia [2011a]).
- **MSAS:** Multi-functional Satellite Augmentation System (MSAS) is the Japanese SBAS. It provides service only to Japan. It has been operational since 2007 (from Navipedia [2011b]).
- **SDCM:** System for Differential Corrections and Monitoring (SDCM) is the Russia SBAS. It is in the development phase, designed to provide service only in Russia. It is the only system that will provide corrections for GPS and GLONASS constellations. The planned start of operations is currently unknown.

- **GAGAN:** GPS and GEO Augmented Navigation (GAGAN) is the India SBAS. It provides service only in India. It has been operational since 2013 (from JagranJosh [2015]).
- **BDSBAS:** BeiDou Satellite-Based Augmentation System (BDSBAS) is the Chinese SBAS. It is in the development phase, designed to provide service only in China. It is forecast to start providing service on 2025 (from EUSPA [2022c]).
- **KASS:** Korea Augmentation Satellite System (KASS) is the South Korea SBAS. It is in the development phase, designed to provide service only in South Korea. It is forecast to start providing service on 2023 (from EUSPA [2022c]).
- **SouthPAN:** Southern Positioning Augmentation Network (SouthPAN) is the Oceania SBAS. It is in the development phase, designed to provide service in Australia and New Zealand. It is forecast to start providing service on 2028 (from EUSPA [2022c]).
- **ANGA:** Augmented Navigation for Africa (ANGA) is the Africa SBAS. It is in the development phase, designed to provide service in Africa and Madagascar. It is forecast to start providing service on 2025 (from EUSPA [2022c]).

As stated in EUSPA [2022c], all SBAS have been designed to be compatible (not to interfere with each other) and interoperable (i.e. any standard receiver will be able to process any of the SBAS signals). Therefore, this will allow aeroplanes to benefit from SBAS on a worldwide scale without needing a receiver certification for each SBAS used.

Implementing an SBAS requires the following architecture (similar to GNSS). An example of SBAS architecture (EGNOS) is shown in Fig. 4.6:

- **Ground Segment:** A network of permanent GNSS stations distributed throughout the service area. The permanent stations send data to several master stations (for redundancy), which compute the SBAS corrections and send them to the uplink stations, which will upload the corrections to the GEO satellites. Note that each master station may produce slightly different SBAS corrections, in which case the GEOs assigned to each master station may send different message contents.
- **Space Segment:** Consists of a set of GEO satellites covering the desired area, which broadcast the SBAS corrections to the users. Each SBAS needs at least 2 GEO (for redundancy) or more for large area coverage (for instance SDCM, which covers the whole of Russia).

- **User segment:** The receiver capable of processing SBAS signals from the GEO. For SoL usage, the receiver must be certificated.

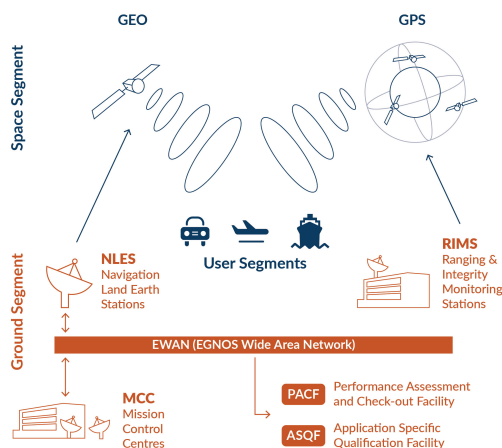


Figure 4.6: EGNOS architecture (from EUSPA [2022a]).

In civil aviation, the performance requirements vary according to the operation modes (as shown in table 4.4). The modes are as follows:

- **PA:** Precision Approach mode. This mode is used when an aircraft is approaching a terminal for landing.
- **NPA:** Non Precision Approach mode. Used in all other flight operations (i.e. en route over earth or ocean and departure from terminals).

Operation	Non Precision Approach (NPA)	Precision Approach (PA)
Horizontal Accuracy (95%)	220 m	16 m
Vertical Accuracy (95%)	N/A	20 m
Integrity	$1 \cdot 10^{-7}/h$	$2 \cdot 10^{-7}$ per approach
Time-To-Alert (TTA)	10 s	10 s
Continuity	$1 \cdot 10^{-4}/h$ to $1 \cdot 10^{-8}/h$	$8 \cdot 10^{-6}$ in any 15 s
Availability	0.99 to 0.99999	0.99 to 0.99999

Table 4.4: Performance requirements (from ICAO [2006]).

The integrity requirements in table 4.4 refer to the probability of having erroneous information (which is called as Misleading Information (MI)) and the probability of not fulfilling the operational performance requirements. In order to define when there is an integrity or continuity risk, the following terms are defined:

- **Protection Level (PL):** By combining all the confidence bounds of all satellites in the navigation solution provided, a global bound can be computed for both horizontal and vertical positioning errors. These bounds are called Protection Levels (Horizontal Protection Level (HPL) and Vertical Protection Level (VPL)), and they provide a measurement of confidence that the true position is within a cylinder around the computed position (see Fig. 4.7).
- **Alarm Limit (AL):** The threshold in protection levels for declaring a system as unavailable. Each component, horizontal and vertical, has a threshold, named Horizontal Alarm Limit (HAL) and Vertical Alarm Limit (VAL) respectively.

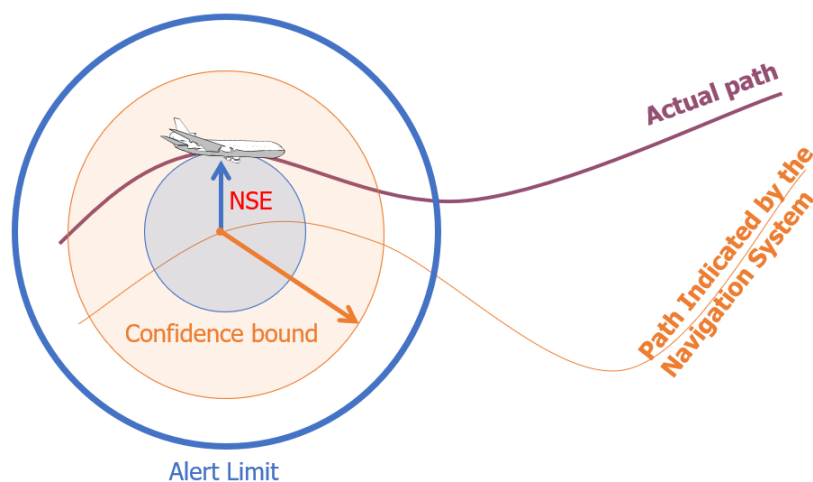


Figure 4.7: NSE, PL (shown as confidence bounds) and AL (from Sanz et al [2003]).

With the PL and AL, four modes of operation can be defined:

- **Normal operation:** When the navigation error is below the PL and the PL is below the AL.
- **Misleading Information (MI):** When the navigation error is above the PL.

- **Hazardous Misleading Information (HMI):** When the navigation error is above the PL and above the AL, but the PL is below the AL.
- **System unavailable:** When the PL is above the AL. If the error is above the PL, the SBAS positioning will also present MI.

These four modes of operation can be graphically represented using the Stanford plot, which uses the horizontal axis for positioning error and the vertical axis for protection levels (one plot is therefore needed for the horizontal component and another for the vertical component). Each position in the Stanford plot will have an associated colour, which represents the number of times a combination of positioning error and protection level occurred (in logarithmic scale). The Stanford plot values are quantized, with gLAB using a default grid of 0.5 per 0.5 metres. An example³ Stanford plot is shown in Fig. 4.8:

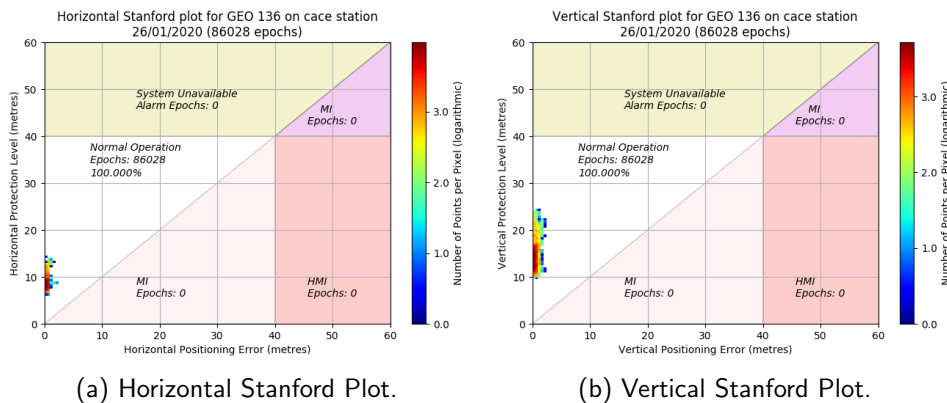


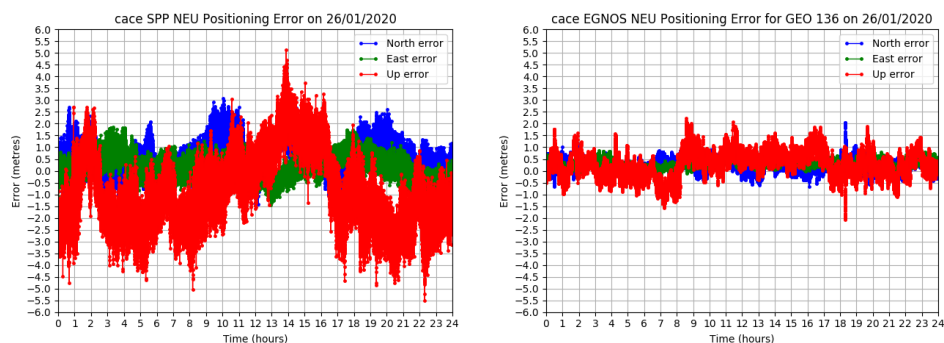
Figure 4.8: Stanford plot examples in “cace” station on 26/01/2020. On the left-hand with the horizontal component and on the right-hand with the vertical component. The AL for both components is set to 40 metres (the default value in gLAB).

SBAS broadcast the corrections through binary messages of one second of duration at a bit rate of 250 bits/s (as defined in [RTCA-MOPS \[2006\]](#)). The messages support only GPS and GLONASS constellations, and a maximum of 51 concurrently monitored satellites. A detailed explanation of how to apply these corrections can be found at [Todd Walter \[1999\]](#).

Once the SBAS corrections are applied, the computation of the navigation solution is the same as in SPP (WLS). An example³ of SBAS positioning with EGNOS corrections is shown in Fig. 4.9b, together with the processing in SPP

³The corresponding files are available at the following URL: https://server.gage.upc.edu/PhD/Ibanez_Deimos/Ibanez_Deimos_PhD_files.7z

with GPS for comparison only. In this case, a 1 Hz file has been used (as SBAS requires 1 Hz files) with station “cace” (Cáceres, Spain) on the 26th of January 2020.



(a) 1 Hz GPS only SPP processing.

(b) 1 Hz EGNOS processing.

Figure 4.9: 1 Hz processing with “cace” station on 26/01/2020, on the left-hand with GPS only SPP and on the right-hand using EGNOS corrections (GEO 136).

With EGNOS corrections, the 95 horizontal percentile error descends to 0.75 metres (from 1.89 metres in SPP), and the vertical to 1.15 metres (from 2.96 metres in SPP), as well as having much lower peak errors. The processing time for GPS only is 5.6 seconds and for SBAS is 8.4 seconds (on an Intel i9-12900KS CPU).

In order to measure availability and continuity in SBAS, there are availability and continuity maps. These maps are computed using broadcast navigation messages and SBAS corrections, that is, without using a RINEX observation file, as a fault-free receiver is assumed (no data gaps and no cycle-slips). To compute them, gLAB loops for each epoch during one day and for each position in a grid of points inside the SBAS coverage area (or user-defined area). At each point in the grid the number of satellites available is shown (by checking that satellite elevation is over 5°) and the PLs are computed (since for PL only sigmas and satellite geometry is needed).

The availability is computed as the percentage of the total epochs that were available (epochs with PLs below the ALs), as shown in equation 4.3:

$$\mathbf{Availability} (\%) = 100 \cdot \frac{\mathit{Avail\ epochs}}{\mathit{Total\ epochs}} \quad (4.3)$$

The continuity risk for aviation is computed using a sliding window according

to RTCA-MOPS [2006], which results in formula 4.4:

$$\text{Continuity Risk sliding window} = \frac{NEAD}{\text{Avail epochs}} \quad (4.4)$$

The Number of Epochs Affected by Discontinuities (NEAD) is the total sum of epochs with an available solution prior to a discontinuity (transition from an available epoch to an unavailable epoch) inside a Continuity Time Interval (CTI) of 15 seconds. If there are multiple discontinuities inside the CTI, only the available epochs prior to the current discontinuity, but after the previous discontinuity, are accounted for (to avoid double counting an epoch).

To illustrate the computation of continuity risk with the sliding window, Fig. 4.10 (taken from the gLAB manual) shows an example of computation of continuity risk. In this figure, 'X' are epochs with PA solution below alarm limits, while 'O' are epochs with no PA solution or with a PA solution but above any AL. The window size in this example is set to five epochs.

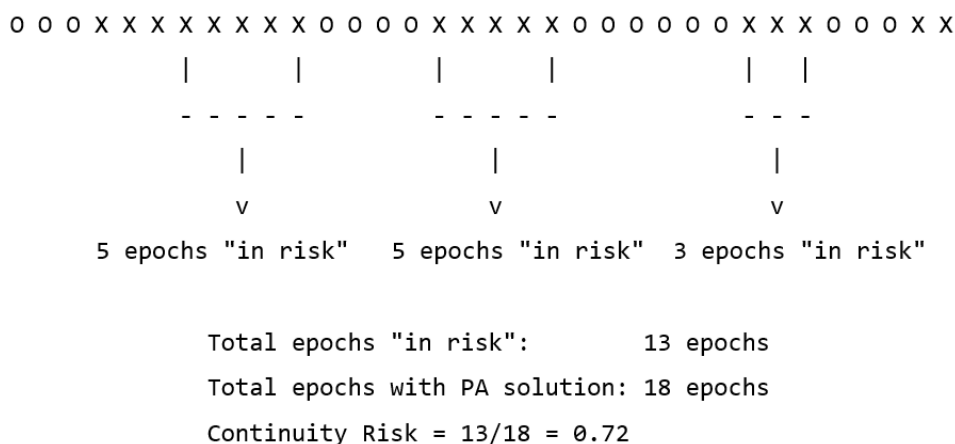


Figure 4.10: Continuity risk with sliding window computation example (from gLAB manual). 'X' are epochs with a PA solution below alarm limits, while 'O' are epochs with no PA solution or with a PA solution but above any AL. Epochs "in risk" are epochs affected by a discontinuity. The window size in this example is set to five epochs.

gLAB has an additional method for computing the continuity risk for maritime applications using SBAS, but will be explained in chapter 7.1.

An example availability map (computed in a single gLAB execution) is shown in Fig. 4.11. The availability map has several regions enclosed by white lines, named Flight Information Regions (FIRs), which delimit the SBAS service area (the European, CONUS, Canada and Mexico FIRs are hard-coded in the DAT,

but the user can set any other area). The table at the bottom provides the availability percentage for each region for several thresholds (all computed and plotted by the DAT), and all table properties can be tweaked by the user.

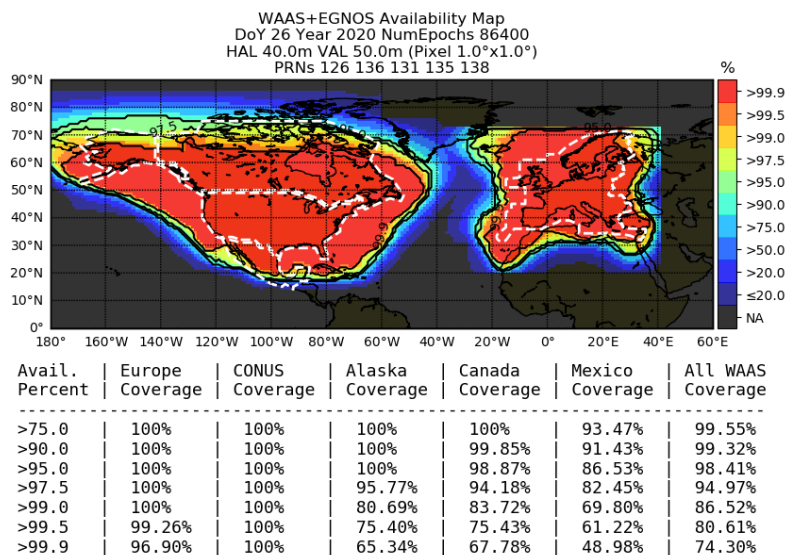


Figure 4.11: WAAS and EGNOS availability map for 26/01/2020.

Finally, an example continuity risk result is shown in Fig. 4.12.

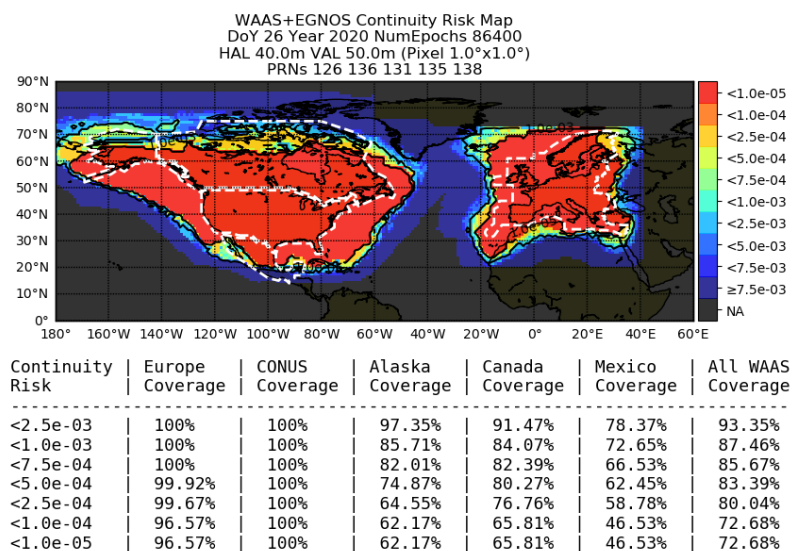


Figure 4.12: WAAS and EGNOS continuity risk map for 26/01/2020.

4.4 Differential Global Navigation Satellite System

DGNSS algorithm exploits the fact that two receivers with common satellites in view have correlated errors. The correlation is in time (a lower time difference increases correlation) and distance (a smaller distance increases correlation). These correlated errors are common range errors (e.g. orbit and clock errors, ionosphere and troposphere delays), are observed by a reference station (with precisely known coordinates) and are transmitted to the rover receiver, which uses them to mitigate or cancel these errors.

The reference station has two ways of providing the corrections to the rover:

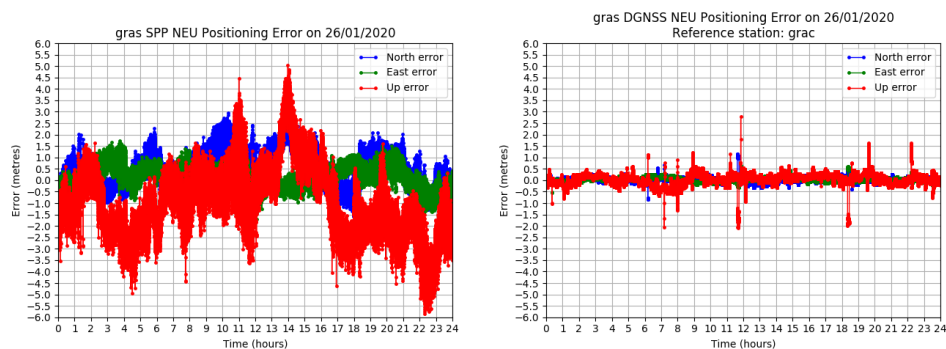
- Computing the corrections itself (each satellite will have a correction that includes all range errors) and transmitting them to the rover, as described in standard RTCM version 2 ([RTCM SC-104 \[2010\]](#)).
- By transmitting the pseudorange to the rover (and the rover computes the adequate correction), as described in standard RTCM version 3 ([RTCM SC-104 \[2022\]](#)).

gLAB supports both RTCM versions 2 and 3 for DGNSS, as well as using a RINEX observation file instead of RTCM version 3 (as both files provide the same data). In any of these formats, only GPS differential processing or RTCM decoding is supported (although all the input parameters for DGNSS and internal code have been prepared for a future multi-constellation DGNSS upgrade). Moreover, gLAB, by default, will apply pseudorange smoothing to reduce the pseudorange noise (which is not correlated with the reference station), which further improves the navigation solution.

Once the DGNSS corrections are applied, the computation of the navigation solution is the same as in SPP.

An example⁴ of DGNSS processing is shown in Fig. 4.13b together with the GPS only SPP counterpart in Fig. 4.13a. IGS station “gras” is used as reference station, while IGS station “grac” is the rover. The stations are about 32 metres apart (which is a short baseline scenario), and 1 Hz RINEX observation files were used for both stations.

⁴The corresponding files are available at the following URL: https://server.gage.upc.edu/PhD/Ibanez_Deimos/Ibanez_Deimos_PhD_files.7z



(a) GPS only SPP positioning with "gras" station. (b) DGNS positioning with rover "gras" and "grac" reference station.

Figure 4.13: GPS only SPP and DGNS positioning with rover "gras" and "grac" reference station on 26/01/2020, on the left-hand with GPS only and on the right-hand with DGNS.

With this scenario, the 95 positioning error percentile for the horizontal component is reduced to 0.23 metres (from 1.75 metres) and the vertical component to 0.46 metres (from 3.26 metres). The processing time in SPP mode is 5.5 seconds, while for DGNS mode is 9.4 seconds (on an Intel i9-12900KS CPU).

Advanced gLAB algorithms

In this chapter the advanced algorithms implemented in gLAB are explained. These are the uncombined PPP, Fast-PPP and SBAS DFMC.

5.1 Uncombined Precise Point Positioning

Uncombined PPP is similar to PPP but with the following differences:

- Pseudoranges and carrier phases are not combined in an IF combination (hence the name “uncombined”) but are provided to the filter as independent measurements.
- The following additional parameters must be estimated in the filter:
 - A slant ionosphere parameter per satellite (I). This parameter is shared by all measurements, as all satellite-receiver rays have the same satellite delay (in Total Electron Content Unit (TECU) units, which is independent of the frequency). It applies both to pseudoranges and carrier phases, but carrier phases have this parameter with a negative sign, due to the fact that ionosphere refraction delays the code but advances the carrier phase (Sanz et al [2013a]).
 - A satellite DCB (S) per satellite and per each pair of pseudorange and carrier phase (in TECUs). Following the same convention as with the ionosphere, carrier phases have this parameter with a negative sign (Sanz et al [2013a]).

- A receiver DCB (R), per each pair of pseudorange and carrier phase in a constellation (in TECUs). For instance, all GPS satellites with C1W and L1W measurements will share the same receiver DCB. Carrier phases will also have this parameter with a negative sign.
- On a per satellite basis, two reference measurements must be selected (from different frequencies). These two reference measurements will have their satellite DCBs set as equal (in TECUs) with an additional equation. This step emulates the behaviour in the IF combination, where it is assumed that in the IF combination the satellite DCBs are removed. In order to select the reference measurements, gLAB starts from the default measurements (which can be modified by the user). Should these measurements not be available, gLAB will select first another measurement from the default frequencies, if possible, or else it will loop through the available measurements until it can select one.
- On a per constellation basis, two reference measurements must be selected (from different frequencies). The same criterion is applied as for satellite DCB, but the receiver DCBs is shared by all satellites in a constellation with common measurements (e.g. all GPS satellites with C1C measurement will share the same receiver DCB). In order to select the reference measurements, gLAB will use the two most common measurements (from different frequencies) in a constellation from the list of selected measurements to be used in the navigation solution.

Typical uncombined PPP processing uses only two pairs of measurements (pseudorange and carrier phase) per satellite (the default in gLAB), therefore these two measurements will be the reference measurements for both satellite and receiver DCBs. gLAB allows more than two pairs of measurements per satellite, so in this case it was necessary to program a criterion to select the reference measurements (as explained above).

For ionosphere, satellite and receiver DCB parameters, a conversion factor has to be applied to convert them from TECU to metres (which will vary depending on the frequency of the signal). In the filter, ionosphere and DCBs parameters are estimated as white noise (but it can be configured as static or random walk), to let their value vary as needed.

The parameters to be estimated are shown in vectorial form in equation 5.1:

$$\begin{aligned}
\mathbf{x} = & \left[dx, dy, dz, c\delta t, \underbrace{c\delta t_{const}^2, \dots, c\delta t_{const}^s}_{ISCB}, \Delta Tr_{z,wet}, \right. \\
& \underbrace{R_1(1), \dots, R_1(k[1]), \dots, R_s(1), \dots, R_s(k[s])}_{Receiver\ DCBs} , \\
& \underbrace{B_1^1(1), \dots, B_1^1(k[1]), S_1^1(1), \dots, S_1^1(k[1]), I_1^1, \dots, B_1^{n[1]}(1), \dots, B_1^{n[1]}(k[1]), S_1^{n[1]}(1), \dots, S_1^{n[1]}(k[1]), I_1^{n[1]}}_{Satellite\ DCBs\ of\ system\ 1} , \\
& \underbrace{B_s^1(1), \dots, B_s^1(k[s]), S_s^1(1), \dots, S_s^1(k[s]), I_s^1, \dots, B_s^{n[s]}(1), \dots, B_s^{n[s]}(k[s]), S_s^{n[s]}(1), \dots, S_s^{n[s]}(k[s]), I_s^{n[s]}}_{Satellite\ DCBs\ of\ system\ s} \left. \right]^T \\
& \vdots \\
& \underbrace{B_s^1(1), \dots, B_s^1(k[s]), S_s^1(1), \dots, S_s^1(k[s]), I_s^1, \dots, B_s^{n[s]}(1), \dots, B_s^{n[s]}(k[s]), S_s^{n[s]}(1), \dots, S_s^{n[s]}(k[s]), I_s^{n[s]}}_{Satellite\ DCBs\ of\ system\ s} \left. \right]^T
\end{aligned} \tag{5.1}$$

Where:

- c : Speed of light (299792458.0 m/s) for converting seconds to metres.
- s : Number of systems.
- $n[i]$: Number of satellites in system i ($i = 1, \dots, s$).
- $k[i]$: Number of pairs of pseudorange and carrier phases in system i ($i = 1, \dots, s$).
- dx, dy, dz : Receiver coordinates.
- $c\delta t$: Receiver reference clock (e.g. GPS clock).
- $c\delta t_{const}^i$: ISCB for system i (except for the reference one, set as the first system in the list).
- $\Delta Tr_{z,wet}$: Zenith wet tropospheric delay.
- $B_i^j(k[i])$: Carrier phase ambiguity for each carrier phase of satellite j in system i . Note that there will be as many carrier phases as pairs of pseudorange and carrier phase (indicated by $k[i]$).
- $S_i^j(k[i])$: Satellite DCB for each pair of pseudorange and carrier phase (indicated by $k[i]$) of satellite j in system i .

- I_i^j : Ionosphere delay of satellite j in system i .
- $R_i(k[i])$: Receiver DCB for each pair of pseudorange and carrier phase (indicated by $k[i]$) in system i .

An example¹ of uncombined PPP is shown in Fig. 5.1, together with the PPP example in section 4.2 in order to compare the results. The same dataset as in section 4.2 is used, (“nnor” station on 02/07/2021 with a 30 second file):

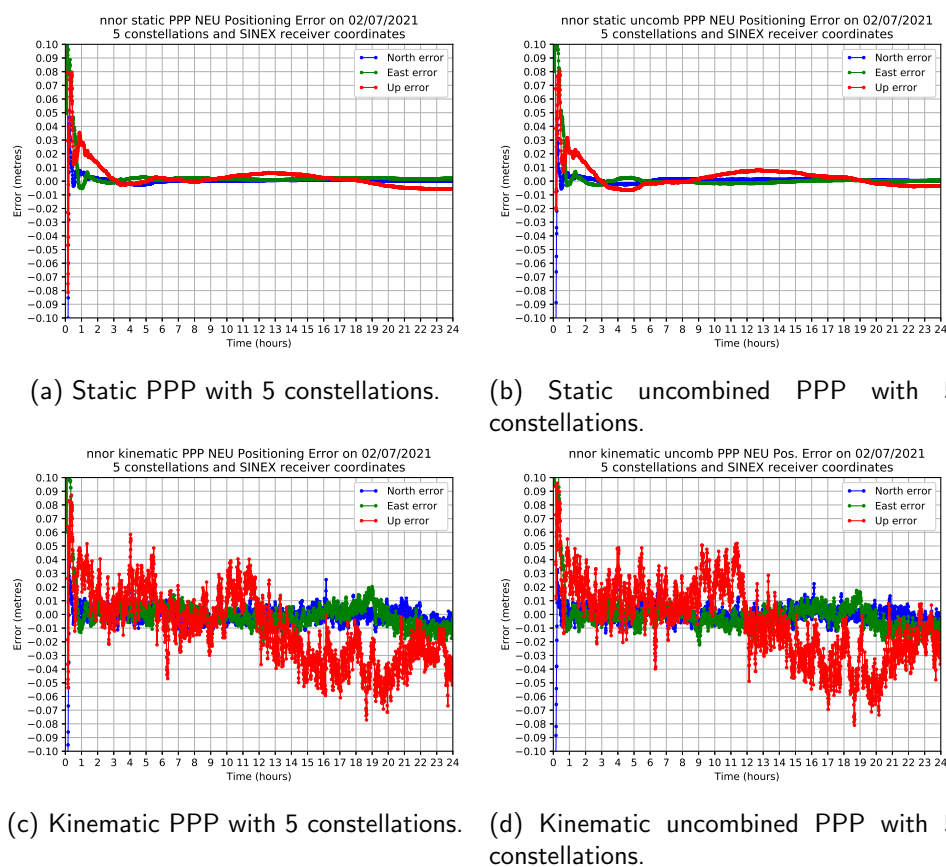


Figure 5.1: PPP and uncombined PPP example in “nnor” station on 02/07/2021. The top row is in static mode, while the bottom row is in kinematic mode. The left-hand side is with PPP and the right-hand side with uncombined PPP (both with 5 constellations). SINEX receiver coordinates are used in all cases as a ground truth.

¹The corresponding files are available at the following URL: https://server.gage.upc.edu/PhD/Ibanez_Deimos/Ibanez_Deimos_PhD_files.7z

The results in both PPP and uncombined PPP are nearly the same (with differences of ± 1 cm). This is the expected result, for the following reasons:

- The geometry of the satellites is the same in both cases.
- Although the number of equations has more than doubled (there is double the number of pseudoranges and carrier phases, plus the equations to link satellite and receiver DCBs), the number of parameters to estimate in the filter have also increased accordingly (the additional carrier phase ambiguities, ionosphere, satellite and receiver DCB parameters).
- The ionosphere, satellite and receiver DCB parameters are loosely coupled in the filter, as on the one hand, they are estimated as white noise, and in the other hand, each parameter appears only in one satellite (except for the receiver DCBs). Moreover, the estimation of these parameters will be rather unreliable, as the filter cannot separate the delay produced by each parameter (as they are all frequency dependent parameters).
- From the algebraic point of view, the uncombined PPP can be considered to be equivalent to the ionosphere-free combination when there is no external information about ionosphere and DCBs, or equivalently, with the use of external information but with a large variance. Indeed, these parameters can be isolated from the equation associated with one of the two frequencies and substituted in the other equation to obtain the ionosphere-free combination solution.

The processing time is 20.2 seconds (on an Intel i9-12900KS CPU). This is nine times more than in PPP (2.2 seconds). This increase is due exclusively to the size increase of the navigation equations (and all the code associated to fill the data), as the cycle-slip detection and measurement modelling did not change. This increase in computation time does not provide any advantage with respect to PPP, which means that uncombined PPP does not provide any benefit by itself, but it will be key for the Fast-PPP algorithm in which ionospheric corrections and satellite DCBs are exploited.

5.2 Fast Precise Point Positioning

Fast-PPP algorithm (Juan et al [2012], Rovira-Garcia et al [2016a]), developed by gAGE (patent reference PCT/EP2011/001512), is aimed to provide the same level of accuracy as in PPP, but with a convergence time of a few minutes. It is based on uncombined PPP, but with the following differences:

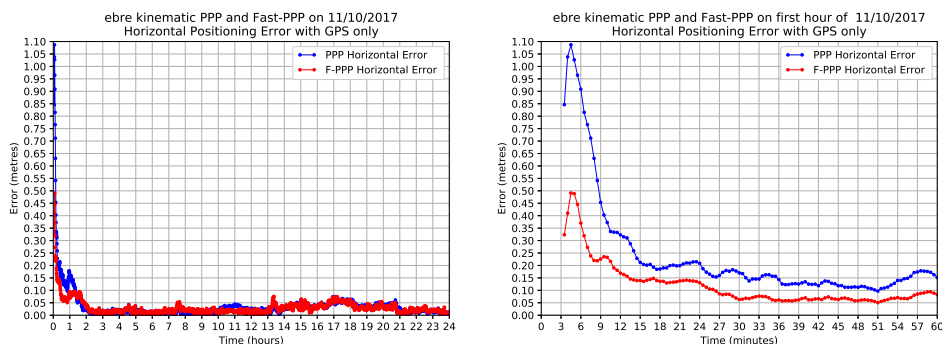
- Use of external values of ionosphere and satellite DCBs. These values and associated sigmas are estimated by an external Central Processing Facility (CPF) (for instance the gAGE CPF, described in Rovira-Garcia et al [2015]). The ionosphere and satellite DCB can be provided in two ways (which are equivalent):
 - Skip the ionosphere and satellite DCB during the measurement modelling and provide these values with the additional equations, weighted by the confidence value of the corrections.
 - Correct the ionosphere and satellite DCB during the measurement modelling and set the values in the additional equations (weighted by the confidence value of the corrections) to zero (this latter approach is the method implemented in gLAB, as it allows existing code from SPP processing mode for ionosphere and satellite DCB modelling to be reused).
- Ionosphere and satellite DCB are constrained in the navigation filter, using their confidence bounds as the weights. These parameters are now estimated as random walk, instead of white noise.

With these changes, Fast-PPP exploits the advantage of uncombined PPP, which is to introduce an ionospheric model and its corresponding satellite DCB estimates as constraints in the navigation filter. This additional information reduces the convergence time, provided that the ionospheric corrections and confidence bounds are realistic and are of better quality than the code pseudorange, because the code noise plays the main role in the first epochs after filter initialization in which all parameters (coordinates, clocks, ambiguities, etc.) are estimated.

In gLAB, Fast-PPP supports full multi-constellation (as it uses the same code as in uncombined PPP) but it cannot take advantage of the full capabilities of the algorithm, because gLAB lacks support for the Solution (Software/technique) INdependent EXchange format for GNSS BIASes (SINEX BIAS) file format (Schaer, S. [2018]). This file is necessary for reading DCB data which varies over time (at any time rate), as Fast-PPP provides ionosphere and satellite DCB data at the same data rate (usually at a five-minute rate). Without this

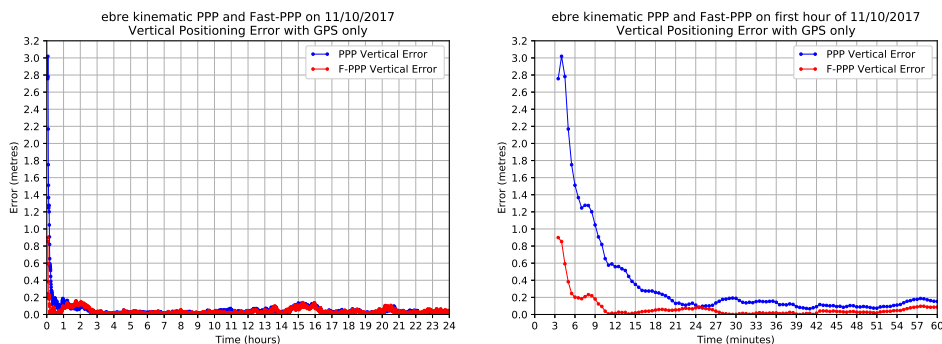
file, gLAB is limited to use only a single DCB per satellite and day, read from the IONEX header (although not officially supported in the standard, gLAB can read DCBs from any constellation from the IONEX header).

An example² of kinematic Fast-PPP processing is shown in Fig. 5.2, where the horizontal and vertical components are shown with the equivalent components in the PPP processing. The dataset is from 11th of October 2017 (around the last solar maximum), using station “EBRE” for 24 hours at a 30 second data rate and precise orbits and clocks from IGS.



(a) Kinematic PPP and Fast-PPP GPS only horizontal component.

(b) First hour of kinematic PPP and Fast-PPP GPS only horizontal component.



(c) Kinematic PPP and Fast-PPP GPS only vertical component.

(d) First hour of kinematic PPP and Fast-PPP GPS only vertical component.

Figure 5.2: Kinematic PPP and Fast-PPP horizontal and vertical components in “EBRE” station on 11/10/2017 (30 second data rate). The top row is in the horizontal component, while the bottom row is in the vertical one. The left-hand side is full day processing and the right-hand side is a zoom of the first hour. GPS only is used in all cases. Note that processing starts at 3 minutes and 30 seconds due to the time needed to fill the cycle-slip detector buffers.

²The corresponding files are available at the following URL: https://server.gage.upc.edu/PhD/Ibanez_Deimos/Ibanez_Deimos_PhD_files.7z

Figures 5.2a and 5.2d, showing the full day processing. Once both algorithms are converged, they perform mostly in the same way. This is expected from the uncombined PPP, provided that the ionospheric corrections do not introduce any bias errors. Regarding the convergence time, in figures 5.2b and 5.2d there is a zoom for the first hour of processing in the horizontal and vertical components respectively. In both cases, Fast-PPP starts with much lower error than the classical PPP (even though in this example PPP uses satellite GPS04, which is discarded in Fast-PPP due to lack of satellite DCB correction) and stays mostly below the classical PPP, with a final convergence to around five centimetres at nearly the same time. In table 5.1 the convergence times for PPP and Fast-PPP are provided, compared with the Galileo HAS requirements (20 cm for the horizontal component and 40 cm for the vertical component in 100 seconds, from EUSPA [2023d]):

	Kinematic convergence time		
	PPP (10 satellites)	Fast-PPP (9 satellites)	Difference
Horizontal (20 cm, 100 s)	1260 s (43 epochs)	450 s (16 epochs)	-64.29%
Vertical (40 cm, 100s)	660 s (23 epochs)	90 s (4 epochs)	-86.36%

Table 5.1: Convergence time comparison for kinematic PPP and Fast-PPP. The satellite used median is provided between parentheses in the second row, while the Galileo HAS accuracy and convergence time are provided between parentheses in column one. The criterion for computing the convergence time is to select the first epoch under the given threshold and it is maintained for 300 seconds.

From table 5.1, Fast-PPP manages to accomplish the vertical component convergence time requirement from Galileo HAS, even though it uses only a single constellation, uses one fewer satellite (as in this case satellite G04 has no corrections) and has the satellite DCB input data limitation commented above. The latter reason is not critical since GPS DCBs are very stable (as demonstrated in Sanz et al [2017]).

The processing time for GPS only in Fig. 5.2 is 0.8 seconds (on an Intel i9-12900KS CPU) with a median of 10 satellites used, while for Fast-PPP is 1.6 seconds with a median of nine satellites used.

5.3 Satellite-Based Augmentation System Dual Frequency Multi Constellation

The evolution of SBAS single frequency and single constellation is SBAS DFMC. The differences between SBAS DFMC and SBAS 1F are as follows (from EUROCAE [2018]):

- Use of dual frequency measurements (L1 and L5 frequencies) with an IF combination in both the pseudorange and carrier phase (used for smoothing the code).
- No ionospheric model is transmitted, as the IF is used, which saves bandwidth.
- Support for Galileo, BDS and QZSS constellations.
- Support for monitoring up to 92 satellites simultaneously.
- No “Fast” and “Slow” corrections (as in SBAS 1F), as all corrections of a satellite are provided in a single message (message type 32).
- Support for all types of orbits (IGSO, Highly Elliptical Orbit (HEO), MEO and GEO) for the SBAS satellite broadcasting the messages.

The processing of SBAS DFMC messages is much simpler than in SBAS 1F, due to the lack of ionosphere and the fact that each satellite correction is sent in a message (in SBAS 1F, “Fast” and “Slow” corrections were sent in different messages that had to be synchronized) which results in a reduction in the number of messages (from 16 in SBAS 1F to 10 in SBAS DFMC).

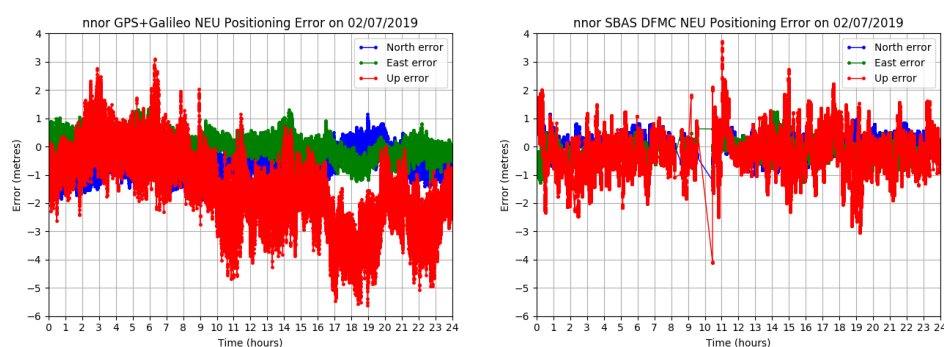
As in SBAS 1F, the computation of the navigation solution is the same as in SPP (WLS). In order to test with real data, SBAS DFMC messages from the Australia SBAS test-bed demonstration project (Mitchell et al [2019]) were used³. This test bed used DFMC Minimum Operational Performance Standards (MOPS) version 0.3.13 (EUROCAE [2017]) with some modifications (an option in gLAB allows these modifications to be taken into account):

- Messages type 34, 35 and 36 were sent inside Message Type (MT) 0. Message types are distinguished by using the two spare bits in positions 222 and 223. Value “00”, correspond to MT0, value “01” correspond to MT34, “10” to MT35 and “11” to MT36.

³Australian test-bed data and configuration were kindly provided by Eldar Rubinov and Julián Barrios. The corresponding files are available at the following URL: https://server.gage.upc.edu/PhD/Ibanez_Deimos/Ibanez_Deimos_PhD_files.7z

- The GPS TGD correction is already provided in the message corrections, therefore it does not need to be applied (contrary to what the standard states).

Station selected was “nnor” (in Australia) on the 2nd of July 2019. Figure 5.3b shows the SBAS DFMC positioning and the comparison with a GPS and Galileo SPP positioning:



(a) 1 Hz GPS and Galileo SPP processing with “nnor” station. (b) 1 Hz Australia SBAS DFMC test bed processing with “nnor” station.

Figure 5.3: 1 Hz processing with “nnor” station on 02/07/2019, on the left-hand with GPS and Galileo SPP and on the right-hand using Australia SBAS DFMC corrections. The gap around 10 a.m. is due to a data gap in the SBAS DFMC data file.

The 95 error positioning percentile in SBAS DFMC for the horizontal component is 0.83 metres (compared to 1.36 metres in SPP) and for the vertical component is 1.51 metres (compared to 3.64 metres in SPP). The processing time for both SPP and SBAS DFMC is 11.8 seconds (on an Intel i9-12900KS CPU). The median number of satellites used in the navigation solution is 15 for GPS and Galileo SPP and 9 for SBAS DFMC. This difference in number of satellites (a 40% reduction) is due to the fact that, on the one hand, in the case of GPS, not all operational satellites support L5 frequency (as shown in [GPS.gov \[2023b\]](#)), and on the other hand, that new satellites will be launched into orbit (for both GPS and Galileo).

These positioning results are very similar to those achieved with SBAS 1F, but they will improve, as SBAS DFMC standard is evolving (note that the version used is from 2017, and the contents of the message have slightly changed) and there will be additional usable GPS and Galileo satellites in orbit.

As there is no need to broadcast an ionosphere model (which was the most limiting factor for the coverage area), it is expected to have a better coverage

area, especially in unsounded areas such as the seas (i.e. areas without permanent stations for computing the ionospheric model). The availability map for this day for 6 a.m. is shown in Fig. 5.4 (an hourly map was used as the SBAS DFMC has data gaps, as seen in Fig. 5.3b).

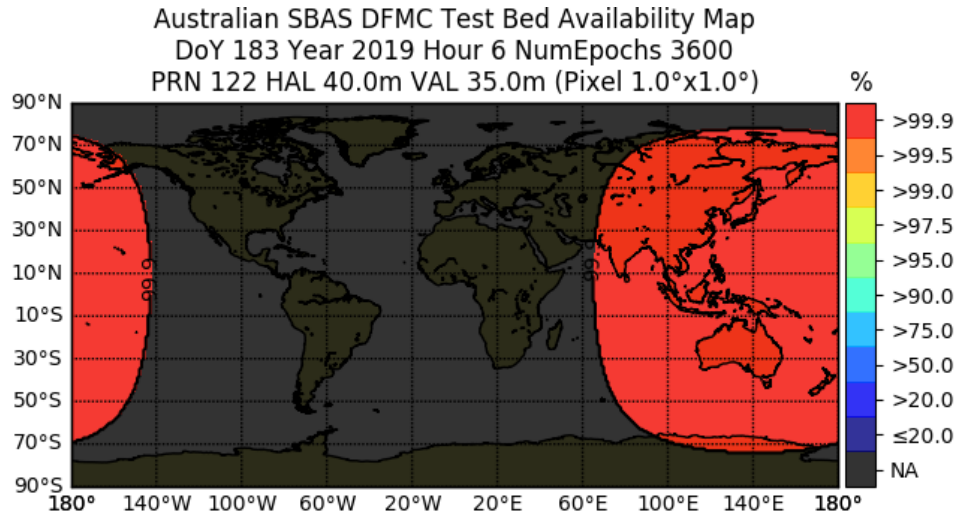


Figure 5.4: Australia SBAS DFMC Availability map with GEO footprint on 02/07/2019 at 6 a.m.

Surprisingly, Fig. 5.4 shows that there is coverage in the entire GEO footprint, which shows that stations outside Australia had to be used. The availability map can be recomputed by removing the GEO footprint limitation, as shown in Fig. 5.5. Note that transmitting the corrections via the internet is feasible even for aircrafts (for instance through the use of Low Earth Orbit (LEO) constellations, such as [Starlink \[2023\]](#), which would also make it possible to reduce the TTA due to its higher bandwidth and lower latency).

Without the GEO footprint, the availability shown in Fig. 5.5 is almost worldwide, which suggests that a worldwide set of stations was used to generate the corrections. It is also a proof a concept that a single SBAS DFMC provider would be capable of providing worldwide service. In practice, however, this option is not possible, as SoL systems have stringent requirements (in terms of availability and security) which a single provider cannot achieve on a worldwide scale.

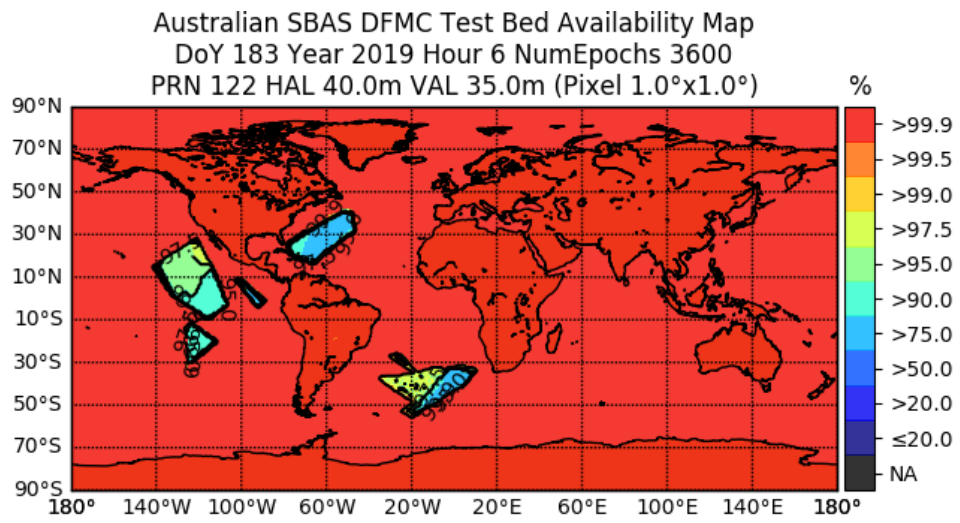


Figure 5.5: Australia SBAS DFMC Availability map without GEO footprint on 02/07/2019 at 6 a.m.

Challenges in gLAB upgrade

In this chapter the main challenges during the gLAB update are explained. The measurement selection, cycle-slip detection, multi-day SBAS availability plots and receiver APC correction challenges are necessary for the GNSS data processing, while the output prints, optimization and multi-threading are tied to the capabilities and performance of gLAB.

6.1 Measurement selection

When GPS and GLONASS reached FOC, GPS has one civil signal and two military signals, while GLONASS had two civil and two military signals. This fact made the selection process easy (due to the limited number of possibilities) and gLAB selection capability was designed only to handle these GPS signals (other signals were read but not used in the navigation filter).

With the new constellations (Galileo, BDS, QZSS, IRNSS) and the new signals in GPS and GLONASS (refer to chapter 2), it was necessary to develop a new method for measurement selection to support the new signals and all possible combinations (dual, triple or quadruple frequency combinations).

To illustrate the magnitude of the problem, according to [Romero \[2020\]](#), GPS has 3 frequencies with 22 measurements (shown in table 6.1); 5 frequencies for GLONASS with 13 measurements; 5 frequencies in Galileo with 19 measurements, 2 frequencies for GEO satellites with 4 measurements, 6 frequencies for BDS with 27 measurements; 4 frequencies for QZSS with 20 measurements; 2 frequencies for IRNSS with 8 measurements. Therefore, there are 14 different frequencies and 113 different measurements.

GNSS System	Freq. Band / Frequency	Channel or Code	Observation Codes			
			Pseudo Range	Carrier Phase	Doppler	Signal Strength
GPS	L1/1575.42	C/A	C1C	L1C	D1C	S1C
		L1C (D)	C1S	L1S	D1S	S1S
		L1C (P)	C1L	L1L	D1L	S1L
		L1C (D+P)	C1X	L1X	D1X	S1X
		P (AS off)	C1P	L1P	D1P	S1P
		Z-tracking and similar (AS on)	C1W	L1W	D1W	S1W
		Y	C1Y	L1Y	D1Y	S1Y
		M	C1M	L1M	D1M	S1M
		codeless		L1N	D1N	S1N
	L2/1227.60	C/A	C2C	L2C	D2C	S2C
		L1 (C/A) + (P2-P1) (semi-codeless)	C2D	L2D	D2D	S2D
		L2C (M)	C2S	L2S	D2S	S2S
		L2C (L)	C2L	L2L	D2L	S2L
		L2C (M+L)	C2X	L2X	D2X	S2X
		P (AS off)	C2P	L2P	D2P	S2P
		Z-tracking and similar (AS on)	C2W	L2W	D2W	S2W
		Y	C2Y	L2Y	D2Y	S2Y
		M	C2M	L2M	D2M	S2M
	codeless		L2N	D2N	S2N	
	L5/1176.45	I	C5I	L5I	D5I	S5I
		Q	C5Q	L5Q	D5Q	S5Q
		I+Q	C5X	L5X	D5X	S5X

Table 6.1: RINEX version 3.05 GPS Observation Codes (from [Romero \[2020\]](#)).

Moreover, it is necessary to deal not only with such a large number of measurements and frequencies, but also with all the possible dual or triple frequency combinations (IF, Divergence Free (DF), GF, MW, Ionosphere Geometry Free (IGF) and Second Iono Free (SIF), described in [Sanz et al](#)

[2013a]) in each constellation. Finally, some GNSS receivers add another layer of complexity, as they may track different signals on different frequencies (e.g. new generation of GPS satellites are tracked with a new civil signal in L2 frequency, while the former satellites are tracked with the legacy encrypted signal). This fact made it necessary gLAB to select and process measurements based on a satellite basis and not on a constellation basis.

To simplify this huge processing problem, most tools allow only a limited number of combinations (usually hard-coded), as they are designed for specific use cases and receivers. On the other hand, gLAB is developed as a generic tool, focused on both research and professional use. As a research tool, it must be able to cope with all possible combinations, even though some of them may currently have no practical use (since a use may appear in the future).

6.1.1 Selection methodology

The internal management of this huge number of combinations inside gLAB code is relatively simple. Internally, gLAB has a hard-coded list of all possible dual, triple and quadruple frequency combinations (without considering the different measurements) and all single possible measurements. At processing level, it has on a per satellite basis, several variables with the type of measurement to use (single or combination, frequencies, signals, etc.). This requires more memory usage, but the computational cost is very low, as despite the large number of combinations, they are internally selected using *switch* statements. This design choice keeps runtime computational cost constant despite the number of possibilities.

The challenge resides in the interaction with the user and the selection of the measurements. That is, the user must be able to select which measurements and combinations to use (on a per satellite basis) or let the tool decide automatically. Therefore, gLAB has three modes of measurement selection:

- **Automatic mode:** gLAB performs an autonomous selection of the combinations, frequencies and measurements to be used according to the input RINEX observation file and the processing mode (SPP, PPP, uncombined PPP, Fast-PPP, SBAS 1F, SBAS DFMC, DGNSS).
- **Semi-automatic mode:** The user sets which combinations or single measurements together with its frequencies (e.g. L1), but not the measurement attribute (e.g. C1C). In this case, gLAB selects the measurement to use according to the input RINEX observation file.
- **Manual mode:** The user specifies the combinations, frequencies and measurements to be used in the GNSS processing.

Note that each satellite may have multiple measurements selected for the filter, and each measurement can be set by the user in either semi-automatic or manual mode.

In automatic and semi-automatic mode, in order to select the frequencies and measurements, gLAB has several internal priority lists for frequencies and measurements for each constellation and satellite. It starts with a default hard-coded list (see table 6.2 and table 6.3), which is then modified according to input parameters (if provided).

GNSS System	gLAB default frequency priority list					
	#1	#2	#3	#4	#5	#6
GPS	L1 (F1)	L2 (F2)	L5 (F5)			
Galileo	E1 (F1)	E5a (F5)	E6 (F6)	E5b (F7)	E5a+b (F8)	
GLONASS	G1 (F1)	G2 (F2)	G3 (F3)	G1a (F4)	G2a (F6)	
GEO	L1 (F1)	L5 (F5)				
BDS	B1 (F1)	B2a (F5)	B1-2 (F2)	B2b (F7)	B2a+b (F8)	B3 (F6)
QZSS	L1 (F1)	L2 (F2)	L5 (F5)	L6 (F6)		
IRNSS	L5 (F5)	S (F9)				

Table 6.2: gLAB default frequency priority list. Frequencies are provided with their given names in their respective ICDs, and between parenthesis the frequency number assigned in RINEX standard.

Furthermore, if an ANTEX file is provided, gLAB assigns each satellite to a block type (according to its generation) and removes all frequencies and measurements remaining in the list that are not available for a given generation block (performed on a satellite per satellite basis). Later, these lists are filtered again according to the available measurements in the header of the RINEX observation file.

GNSS System	Frequency	gLAB default measurement attribute priority list									
		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
GPS	L1 (F1)	1P	1W	1C	1Y	1L	1S	1X	1M	1N	
	L2 (F2)	2P	2W	2C	2Y	2L	2S	2X	2D	2M	2N
	L5 (F5)	5Q	5I	5X							
Galileo	E1 (F1)	1C	1X	1Z	1B	1A					
	E5a (F5)	5Q	5I	5X							
	E6 (F6)	6C	6X	6Z	6B	6A					
	E5b (F7)	7Q	7I	7X							
	E5a+b (F8)	8Q	8I	8X							
GLONASS	G1 (F1)	1P	1C								
	G2 (F2)	2P	2C								
	G3 (F3)	3Q	3I	3X							
	G1a (F4)	4A	4B	4X							
	G2a (F6)	6A	6B	6X							
GEO	L1 (F1)	1C									
	L5 (F5)	5Q	5I	5X							
BDS	B1 (F1)	1P	1D	1L	1S	1X	1Z	1A	1N		
	B1-2 (F2)	2Q	2I	2X							
	B2a (F5)	5P	5D	5X							
	B3 (F6)	6Q	6I	6P	6D	6X	6Z	6A			
	B2b (F7)	7Q	7I	7P	7D	7X	7Z				
	B2a+b (F8)	8P	8D	8X							
QZSS	L1 (F1)	1C	1L	1S	1X	1Z	1B				
	L2 (F2)	2L	2S	2X							
	L5 (F5)	5Q	5I	5P	5D	5X	5Z				
	L6 (F6)	6L	6S	6X	6E	6Z					
IRNSS	L5 (F5)	5A	5B	5C	5X						
	S (F9)	9A	9C	9B	9X						

Table 6.3: gLAB default measurement attribute priority list. The default list assumes it is the same for pseudorange, carrier phase and Doppler, although each type can have its list modified independently from each other.

As a practical example of this versatility, the user can set gLAB to process in PPP mode with dual frequency IF combination with frequencies L1 and L5 for those satellites with L5 available, and use L1 and L2 for the ones without L5 by simply modifying the frequency priority to make L2 the lowest.

Once the generation of the priority lists are completed (i.e. they have been modified according to user inputs and RINEX observation file header), the selection process is performed when an epoch is read from the RINEX observation file. For each epoch and each satellite, gLAB will try to select the first measurement in the priority list. If there is no data available, gLAB will try with the next one in the list. If there are no measurements with data in the current epoch, gLAB will repeat the process for each epoch until one of the measurements has valid data, except in the event that only one measurement is in the priority list, in which case it will be selected regardless of the available data in the current epoch. In the case of concatenated RINEX observation files, every time a new RINEX observation header is found, gLAB checks whether any of the selected measurements has been removed from the RINEX observation file header. If any measurement is removed, the whole selection process is restarted (including the priority list adjustment).

In order to allow the user to select the measurements to use in a compact and versatile manner, a new argument format was developed. This new format is defined in command line as in equation 6.1 with its breakdown in table 6.4:

$$[+][g][g]#[-][#].[#[-][#]]-<filtermeas>[<meas1>-<meas2>[-<meas3>-<meas4>]] [\dots] \quad (6.1)$$

Table 6.4: gLAB argument breakdown for measurement selection with usage examples.

Section format	Definition
[+-]	Optional symbols for selecting (with a plus '+') or unselecting (with a minus '-'). If no symbol is provided, the default behaviour is to select measurements.
g	Character determining GNSS constellation following RINEX convention (e.g. 'G' → GPS). If g=0, apply it to all constellations.
[g]	More than one character for GNSS constellation may be provided, so multiple GNSS can be selected at once.
#	PRN number. If #=0, it will be applied to all satellites of the selected GNSS constellation.
[-#]	Optionally, a minus sign ('-') and a second PRN number can be provided in order to set a range of satellites (of the same constellation).
[.#[-#]]	Optionally, more than one satellite PRN or satellite range can be given, by separating each value or range with a dot ('.').

Continued on next page

Table 6.4: gLAB argument breakdown for measurement selection with usage examples. (Continued)

<filtermeas>	<p>Set the measurement or combination to be used in the filter. <i>For single measurement and user selected measurement, the measurement name must be provided (in RINEX 2 or 3 format).</i> <i>For single measurements and measurements to be selected by gLAB:</i> Set value 'Pseudorange' or 'Code' (case insensitive), a hyphen ('-') and the frequency number to use, OR Set value 'Phase' or 'Carrierphase' (case insensitive), a hyphen ('-') and the frequency number to use. <i>For dual/triple/quadruple combinations, the combination name with its frequencies must be given.</i></p>
[<meas1>-<meas2>[-<meas3>-<meas4>]]	<p>If measurement is a combination, optionally set the measurements to be used. <i>For dual frequency combinations, two measurements are needed. For triple or quadruple frequency combinations, four measurements are needed.</i> <i>If measurements are not set, they will be automatically selected using the measurement list.</i> <i>In the case of dual, triple or quadruple measurements, if the user does not set the measurements, the type of measurements (pseudorange or carrier phase) to be used is undefined. The user can set it by appending '-code' or '-pseudorange' (for pseudorange) or '-phase' or '-carrierphase' (for carrier phase). For setting pseudorange and carrierphase at once (in two different measurements), use the word '-codephase' or '-phasecode'.</i> <i>All arguments are case insensitive. If not set, gLAB will assume that the type to be used is a carrier phase.</i></p>
[...]	<p>User can optionally set multiple filter measurements in the same parameter, inste.</p>
Examples:	<p><i>G0-L1P</i> → Select L1P measurement for all GPS satellites.</p> <p><i>G0-Code-1 GE0-Phase-5</i> → Select a pseudorange in L1 (which will be automatically selected by gLAB) for all GPS satellites and a carrier phase (also automatically selected by gLAB) in L5 for all GPS and Galileo satellites.</p> <p><i>GR0-IF12-CodePhase</i> → Select ionosphere-free combinations of pseudorange and carrier phases with frequencies L1 and L2 for all GPS and GLONASS satellites. All measurements in the selected frequencies will be automatically selected by gLAB.</p> <p><i>G0-L1P -G1-L1P +G1-L1C</i> → Select L1P measurement for all GPS satellites, then remove L1P measurement from GPS PRN 1 and then add L1C measurement to GPS PRN 1.</p>

Continued on next page

Table 6.4: gLAB argument breakdown for measurement selection with usage examples. (Continued)

	<p><i>GR0-IF12-code EJ0-IF15-code EC1-10.15-IF27-code</i> → Select the pseudorange ionosphere-free combination with frequencies L1 and L2 for all GPS and GLONASS satellites, then select the pseudorange iono-free combination with frequencies L1 and L5 for all Galileo and QZSS satellites, then select the pseudorange ionosphere-free combination with frequencies E2 and E7 for satellites 1 to 10 (both included) and satellite 15 from Galileo and BDS. All measurements in the selected frequencies will be automatically selected by gLAB.</p>
	<p><i>0-IF15-C1X-C5X 0-IF15-L1X-L5X</i> → Select all satellites from all constellations the ionosphere-free combination with C1X and C5X measurements and then select all satellites from all constellations the ionosphere-free combination with L1X and L5X measurements.</p>

This argument format is also applied for other options concerning measurement selections, for instance, for filter weight selection or cycle-slip detection.

Finally, gLAB keeps backwards compatibility for all parameters. When the user inputs the legacy format for any parameter, gLAB will automatically detect it and apply the parameter options to all GPS satellites, as this was the behaviour in previous versions.

6.2 Cycle-slip detection

In legacy gLAB versions, all the cycle-slip detectors had their measurements hard-coded due to the limited number of measurements. In order to be able to cope with any measurement, the following upgrades were needed:

- Allow the selection of any measurement (either manually or automatically) in the cycle-slip detectors. This selection is performed in the same way as for the filter measurements (but with different parameters and internal variables).
- Allow multiple iterations of each cycle-slip detector (e.g. when using dual frequency measurements, use multiple iterations of the single frequency cycle-slip detector for each carrier phase used).
- Check that all carrier phases used in the filter are processed in each of the enabled cycle-slip detectors. If any carrier phase is missing, automatically add a new iteration of the cycle-slip detector. Unless explicitly disabled,

this check is performed independently of whether the cycle-slip detectors were set automatically or manually.

Furthermore, a new cycle-slip detector has been added using triple or quadruple frequency combinations, called the Ionosphere Geometry Free (IGF). The IGF can be computed in two modes:

- **Dual ionosphere-free:** Two ionospheric-free combinations are computed (with either three or four different frequencies), and then these two IF combinations are subtracted.
- **Minimum noise combination:** Compute the IGF combination using the formulas explained in [Montenbruck et al \[2010\]](#).

In both cases, the remaining value contains only the ambiguities, residual wind-up, residual APC correction and carrier-phase noise, which is compared with the predicted value computed by gLAB (in the same manner as in the GF cycle-slip detector defined in [Sanz et al \[2013a\]](#)).

In Fig. 6.1 it is shown an example¹ of a cycle-slip visible with the minimum noise IGF combination. The figure also contains a zoom to the last hour of the plot, where it can be seen that the noise (low elevation) is around three centimetres.

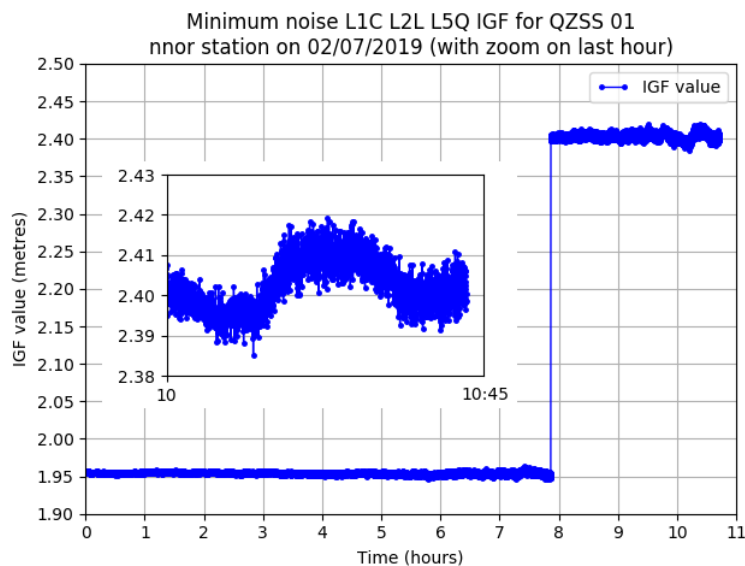


Figure 6.1: Minimum noise IGF with zoom of the last hour (the epochs with lowest elevation). The jump around hour eight is due to a cycle-slip.

¹The corresponding files are available at the following URL: https://server.gage.upc.edu/PhD/Ibanez_Deimos/Ibanez_Deimos_PhD_files.7z

6.3 Output prints

One gLAB feature is its ability to print its internal configuration in the INFO messages. Given that the user is allowed to set the measurement selection on a per satellite basis (as shown in section 6.1), and that different constellation do not share all the frequencies or measurements, it is necessary to compact the output prints as much as possible, as one line per satellite would bloat the output file. Therefore, gLAB groups satellites by PRN ranges and constellations which share the same configuration, meaning that the number of output lines for printing a configuration will depend on the number of different configurations selected, as shown in Fig. 6.2:

```
INFO FILTER Meas: G01 G03-04 G06 G08-11 G14 G18 G23-27 G30 G32 E01-36 C15 C17 C19-63 IF15-Auto-Pseudorange StdDev: 3 (metres)
INFO FILTER Meas: G01 G03-04 G06 G08-11 G14 G18 G23-27 G30 G32 E01-36 C15 C17 C19-63 IF15-Auto-Carrier phase StdDev: 0.03 (metres)
INFO FILTER Meas: G02 G05 G07 G12-13 G15-17 G19-22 G28-29 G31 R01-32 J01-10 IF12-Auto-Pseudorange StdDev: 3 (metres)
INFO FILTER Meas: G02 G05 G07 G12-13 G15-17 G19-22 G28-29 G31 R01-32 J01-10 IF12-Auto-Carrier phase StdDev: 0.03 (metres)
INFO FILTER Meas: C01-14 C16 C18 IF27-Auto-Pseudorange StdDev: 3 (metres)
INFO FILTER Meas: C01-14 C16 C18 IF27-Auto-Carrier phase StdDev: 0.03 (metres)
```

Figure 6.2: gLAB internal filter configuration print for multi-constellation processing in PPP processing with automatic measurement selection and selected to use frequency L5 first for GPS and use first frequencies B1 (F1) and B2a (F5) for BDS (if the satellites have them).

All other gLAB output messages had to be redefined in order to accommodate the new fields for multi-constellation. The main changes in the messages are as follows:

- Messages with measurements; if these measurements are combinations, the frequencies and measurements to build them will also be printed (previously only the combination name was printed).
- Smoothing measurements and the computed smoothed value are now printed in PREFIT and POSTFIT messages.
- Fields for SVN, satellite block type, arc number and arc length were added in most of the messages which are printed in per satellite basis (it facilitates message parsing).
- Messages printed on a per satellite basis are sorted by constellation (with the fixed order GPS, Galileo, GLONASS, GEO, BDS, QZSS and IRNSS). If these message have separate entries for pseudoranges and carrier phase measurements (one line for each one), pseudoranges are printed first.
- New messages have been created for each cycle-slip detector in order to print their internal computations.

- The message with the selected satellites (EPOCHSAT) has been completely redesigned. This message now prints one line per each measurement used in the filter, together with the number of satellites using this measurement that were selected or unselected (in total and separated by constellation), and also the list of satellites involved (selected first with a '+' sign, unselected after with a '-' sign). An example of this message is show in Fig. 6.3.

```

EPOCHSAT 2019 183 19.00 00:00:19.00 IF27-C2I-C7I - 5 +5 G 0 E 0 R 0 S 0 C 5 J 0 I 0 -0 G 0 E 0 R 0 S 0 C 0 J 0 I 0 +C06 +C08 +C09 +C13 +C16
EPOCHSAT 2019 183 19.00 00:00:19.00 IF15-C1C-C5Q - 1 +1 G 0 E 1 R 0 S 0 C 0 J 0 I 0 -0 G 0 E 0 R 0 S 0 C 0 J 0 I 0 +E04
EPOCHSAT 2019 183 19.00 00:00:19.00 IF15-C1W-C5Q - 4 +4 G 4 E 0 R 0 S 0 C 0 J 0 I 0 -0 G 0 E 0 R 0 S 0 C 0 J 0 I 0 +G01 +G03 +G08 +G32
EPOCHSAT 2019 183 19.00 00:00:19.00 IF12-C1W-C2W - 5 +5 G 5 E 0 R 5 S 0 C 0 J 0 I 0 -0 G 0 E 0 R 0 S 0 C 0 J 0 I 0 +G11 +G14 +G18 +G22 +G23
EPOCHSAT 2019 183 19.00 00:00:19.00 IF12-C1W-C2L - 1 +1 G 1 E 0 R 0 S 0 C 0 J 0 I 0 -0 G 0 E 0 R 0 S 0 C 0 J 0 I 0 +G31
EPOCHSAT 2019 183 19.00 00:00:19.00 IF12-C1C-C2L - 4 +3 G 0 E 0 R 0 S 0 C 0 J 3 I 0 -1 G 0 E 0 R 0 S 0 C 0 J 1 I 0 +J01 +J02 +J03 -J07
EPOCHSAT 2019 183 19.00 00:00:19.00 IF12-C1C-C2P - 5 +5 G 0 E 0 R 5 S 0 C 0 J 0 I 0 -0 G 0 E 0 R 0 S 0 C 0 J 0 I 0 +R05 +R07 +R15 +R20 +R21
EPOCHSAT 2019 183 19.00 00:00:19.00 IF12-C1C-C2C - 1 +0 G 0 E 0 R 0 S 0 C 0 J 0 I 0 -1 G 0 E 0 R 1 S 0 C 0 J 0 I 0 -R19
EPOCHSAT 2019 183 19.00 00:00:19.00 IF27-L2I-L7I - 5 +5 G 0 E 0 R 0 S 0 C 5 J 0 I 0 -0 G 0 E 0 R 0 S 0 C 0 J 0 I 0 +C06 +C08 +C09 +C13 +C16
EPOCHSAT 2019 183 19.00 00:00:19.00 IF15-L1C-L5Q - 5 +5 G 4 E 1 R 0 S 0 C 0 J 0 I 0 -0 G 0 E 0 R 0 S 0 C 0 J 0 I 0 +G01 +G03 +G08 +G22 +E04
EPOCHSAT 2019 183 19.00 00:00:19.00 IF12-L1C-L2W - 5 +5 G 5 E 0 R 0 S 0 C 0 J 0 I 0 -0 G 0 E 0 R 0 S 0 C 0 J 0 I 0 +G11 +G14 +G18 +G22 +G23
EPOCHSAT 2019 183 19.00 00:00:19.00 IF12-L1C-L2L - 5 +4 G 1 E 0 R 0 S 0 C 0 J 3 I 0 -1 G 0 E 0 R 0 S 0 C 0 J 1 I 0 +G31 +J01 +J02 +J03 -J07
EPOCHSAT 2019 183 19.00 00:00:19.00 IF12-L1C-L2P - 5 +5 G 0 E 0 R 5 S 0 C 0 J 0 I 0 -0 G 0 E 0 R 0 S 0 C 0 J 0 I 0 +R05 +R15 +R20 +R21
EPOCHSAT 2019 183 19.00 00:00:19.00 IF12-L1C-L2C - 1 +0 G 0 E 0 R 0 S 0 C 0 J 0 I 0 -1 G 0 E 0 R 1 S 0 C 0 J 0 I 0 -R19
EPOCHSAT 2019 183 19.00 00:00:19.00 ALL - 1 +0 G 0 E 0 R 0 S 0 C 5 J 3 I 0 -23 G 1 E 6 R 4 S 6 C 5 J 1 I 0 +G01 +G03 +G08 +G11 +G14
+G18 +G22 +G23 +E04 +R05 +R07 +R15 +R20 +R21 +C06 +C08 +C09 +C13 +G16 +G18 +G22 +G23 +E12 +E18 +E19 +E20 +E26 +E33 +R06 +R16 +R19 +R26 +S22 +S26 +S28 +S29
-530 -532 -C19 -C20 -C29 -C30 -C32 -J07

```

Figure 6.3: gLAB EPOCHSAT message PPP processing with automatic measurement selection and selected to use frequency L5 and signal L2C first for GPS and use first frequencies B1 (F1) and B2a (F5) for BDS (if the satellites have them).

- As explained in section 3.3, the MEAS message has been redefined to act as a converter from RINEX observation file to columnar text file, by printing the following fields: satellite block name and SVN (if an ANTEX is provided), arc number, arc length, elevation, azimuth and a list of measurements (automatically selected or set by the user). An example MEAS message is shown in Fig. 6.4.

```

MEAS 2022 100 3570.00 00:59:30.00 BDS 1 BEIDOU-2G 3 1 120 43.124 46.984 2 C2I:L2I 37538883.0140 37538879.9159
MEAS 2022 100 3570.00 00:59:30.00 BDS 2 BEIDOU-2G 16 1 120 40.494 -51.795 2 C2I:L2I 37689016.5160 37689013.5958
MEAS 2022 100 3570.00 00:59:30.00 BDS 3 BEIDOU-2G 18 1 120 55.252 -11.880 2 C2I:L2I 36944971.3880 36944969.9242
MEAS 2022 100 3570.00 00:59:30.00 BDS 4 BEIDOU-2G 6 1 120 30.969 62.079 2 C2I:L2I 38511309.1040 38511306.9408
MEAS 2022 100 3570.00 00:59:30.00 BDS 5 BEIDOU-2G 11 13 5 19.419 -72.510 2 C2I:L2I 39596259.4850 39596259.7086
MEAS 2022 100 3570.00 00:59:30.00 BDS 6 BEIDOU-2I 5 1 120 22.449 4.913 2 C2I:L2I 39286278.3100 39286279.5230
MEAS 2022 100 3570.00 00:59:30.00 BDS 7 BEIDOU-2I 7 1 120 25.090 -19.393 2 C2I:L2I 39112658.7690 39112652.6313
MEAS 2022 100 3570.00 00:59:30.00 BDS 8 BEIDOU-2I 8 1 120 55.254 -170.000 2 C2I:L2I 36693184.2040 36693181.8902
MEAS 2022 100 3570.00 00:59:30.00 BDS 9 BEIDOU-2I 9 1 3 7.982 -7.434 2 C2I:L2I 41064554.4130 41064554.8731
MEAS 2022 100 3570.00 00:59:30.00 BDS 10 BEIDOU-2I 10 1 120 30.283 -33.618 2 C2I:L2I 38852102.2770 38852097.3695
MEAS 2022 100 3570.00 00:59:30.00 BDS 13 BEIDOU-2I 17 1 120 49.559 -141.936 2 C2I:L2I 36980490.1350 36980488.1514
MEAS 2022 100 3570.00 00:59:30.00 BDS 16 BEIDOU-2I 19 1 120 24.626 7.652 2 C2I:L2I 39012907.8070 39012910.3654
MEAS 2022 100 3570.00 00:59:30.00 BDS 27 BEIDOU-3M-SECM 203 1 120 75.351 -141.680 2 C2I:L2I 21494243.4040 21494241.9623
MEAS 2022 100 3570.00 00:59:30.00 BDS 28 BEIDOU-3M-SECM 204 1 120 48.192 43.384 2 C2I:L2I 22808935.2090 22808929.7683
MEAS 2022 100 3570.00 00:59:30.00 BDS 30 BEIDOU-3M-SECM 208 1 26 22.112 -138.413 2 C2I:L2I 24748465.5020 24748464.6432
MEAS 2022 100 3570.00 00:59:30.00 BDS 32 BEIDOU-3M-CAST 213 1 120 44.660 -57.057 2 C2I:L2I 23345451.1430 23345452.8598
MEAS 2022 100 3570.00 00:59:30.00 BDS 33 BEIDOU-3M-CAST 214 2 112 22.724 137.170 2 C2I:L2I 25110769.3520 25110762.3446
MEAS 2022 100 3570.00 00:59:30.00 BDS 38 BEIDOU-3I 220 1 120 60.223 166.295 2 C2I:L2I 36439039.5050 36439037.7796
MEAS 2022 100 3570.00 00:59:30.00 BDS 39 BEIDOU-3I 221 1 120 34.039 14.322 2 C2I:L2I 38160765.1250 38160765.2749
MEAS 2022 100 3570.00 00:59:30.00 BDS 40 BEIDOU-3I 224 1 120 20.759 -11.161 2 C2I:L2I 39451218.3430 39451211.0984
MEAS 2022 100 3570.00 00:59:30.00 BDS 41 BEIDOU-3M-CAST 227 1 120 74.083 164.581 2 C2I:L2I 21981343.7700 21981342.0729
MEAS 2022 100 3570.00 00:59:30.00 BDS 56 BEIDOU-3SI-CAST 104 1 1 0.000 0.000 2 C2I:L2I 46229950.2690 46229950.2778
MEAS 2022 100 3570.00 00:59:30.00 BDS 58 BEIDOU-3SM-CAST 103 1 1 0.000 0.000 2 C2I:L2I 37383033.7270 37383033.6057
MEAS 2022 100 3570.00 00:59:30.00 BDS 59 BEIDOU-3G-CAST 217 1 1 45.491 40.448 2 C2I:L2I 37931486.9710 37931486.8094
MEAS 2022 100 3570.00 00:59:30.00 BDS 60 BEIDOU-3G-CAST 229 1 1 38.176 -56.528 2 C2I:L2I

```

Figure 6.4: gLAB MEAS message example with BDS satellites using C2I and L2I measurements.

- The statistical summary has been expanded. It now prints the epochs with highest or lowest DOP, error, satellites used or satellites not used on a per constellation basis, as well as the number of epochs for each combination of constellations used and the number of epochs a constellation is used as the reference clock. Moreover, for PPP, uncombined PPP and Fast-PPP

processing modes, the convergence time is also computed (as explained in section 4.2).

Finally, backwards compatibility is achieved by adding an option to print in the same format as in version 5.5.1 (which affects all messages except for the INFO message).

6.4 Multi day SBAS availability plots

As explained in section 4.3, gLAB computes SBAS availability plots on a daily basis (one day for each execution), which is the most common usage. When computing multiple day maps, availability and continuity risk metrics need to be calculated as a continuous period of time, in the following way:

- **Availability:** As described in equation 4.3, the availability is the total number of epochs available divided by the total number of epochs processed. Adding additional days (more epochs) is straightforward, as it only requires the two counters to be incremented.
- **Continuity risk:** As described in equation 7.2, continuity risk requires the number of available epochs in the sliding window when a discontinuity occurs to be taken into account, which can be a problem when a discontinuity occurs during a day transition.

In order to correctly compute the continuity risk during day transitions, there are two strategies:

1. Modify the software to be able to compute multiple days (in chronological order). With the available epochs of the previous day's available epochs in memory, the discontinuities occurring during a day transition can be handled just like any other discontinuity.
2. The software computes independent daily SBAS availability maps, and the total availability and continuity risk for the full period are computed externally (taking into account the day transitions).

The first method is straightforward for the user, as one the period to be computed needs to be set. But as the time to compute SBAS availability maps in chronological order increases linearly with the number of days to be processed, hence, continuity calculations involving large time spans (which can

be weeks, months or years) will take a long time to complete. For instance, gLAB takes around two hours to process one day (with the default configuration, which uses the former coverage EGNOS area). Therefore, processing one year of data would take one month. Moreover, these kinds of processing usually need to be run multiple times (e.g. for testing several configurations, such as DOP filters), which multiplies the processing time.

The second method requires the user to manually add the values from all the processed days, but in turn, allows the execution of multiple days in parallel (as each day is computed independently), limited only by the number of CPUs available (SBAS availability maps require very low Input / Output (I/O) operations, so access to hard disk is not a bottleneck even with a high number of gLAB processes running in parallel). For instance, a server with 30 CPUs can process one year of data in slightly more than one day (with a large number of concurrent processes, a small slowdown is expected due to the concurrent access to system resources).

gLAB follows the second method, as the tool can only compute single day SBAS availability maps. To add the data for multiple days, gLAB prints a file with all the discontinuities found in the processing (for both daily and hourly maps), and a detailed explanation from the manual, which includes:

- An explanation of how the continuity risk is computed for aviation (shown in equation 7.2) and for maritime use (shown in the following chapter, in equation 7.1).
- A comprehensive list of all possible cases during a day transition, with an example of how to manage them.
- Command line instructions and a GNU AWK (GAWK) script ([GAWK \[2023\]](#)) to add up availability and continuity risk for any time span. These commands are meant to be copied and pasted into the user terminal, requiring only a change to the path to the input data files. The commands are designed to be executed in a Bash terminal (the default in Linux and Mac), which is also available in Windows through Cygwin ([Cygwin \[2023\]](#)), provided with the Windows installable of gLAB.

As the detailed explanation in the manual is very long, the gLAB manual section is provided in annex C. The only topic not covered by the manual is how to parallelize multiple executions of gLAB (or any other tool). This can be easily achieved by using tools such as “Parallel” ([Tange \[2020\]](#)).

6.5 Optimization

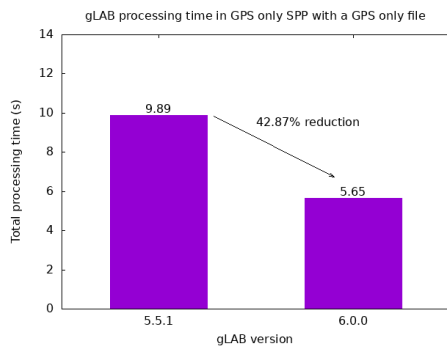
The transition in gLAB from GPS only to multi-constellation and multi-frequency implied that the amount of data to be read and processed has increased exponentially. On the one hand, 1 Hz GPS only RINEX observation files sizes occupied around 60 MegaByte (MB), while multi-constellation and multi-frequency files can occupy between 500 MB and 1.5 GigaByte (GB). On the other hand, the number of satellites with GPS only per epoch were 12, while with full multi-constellation it can be higher than 60 (as seen in section 3.4).

With this huge increase in the amount of data to be processed, any inefficiencies in the gLAB source code results in noticeably longer processing times. For this reason, it was necessary to review and optimize the code.

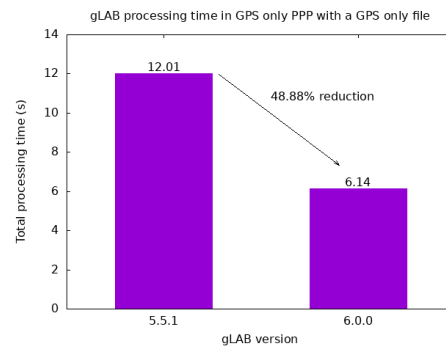
A gLAB processing time comparison of versions 5.5.1 and 6.0.0 is provided in Fig. 6.5 (measured on an Intel i9-12900KS CPU). The data set used for this comparison is on the 2nd of July of 2019, with the following setup²:

- **SPP GPS only file:** IGS “nnor” station (24 hours at 1 Hz data rate) with all measurements removed except for GPS. For orbits and clocks, IGS “brdc” RINEX navigation file was used (which only contains GPS broadcast navigation messages).
- **SPP multi-constellation file:** IGS “nnor” station (24 hours at 1 Hz data rate) with all constellations available (except IRNSS). For orbits and clocks, an IGS “brdm” RINEX navigation file was used (which contains broadcast navigation messages for all constellations).
- **PPP GPS only file:** IGS “nnor” station (24 hours at 1 Hz data rate) with all measurements removed except for GPS. For orbits and clocks, IGS final orbits and clocks were used (which only contain GPS products).
- **PPP GPS multi-constellation file:** IGS “nnor” station (24 hours at 1 Hz data rate) with all constellations available (except IRNSS). For orbits and clocks, CODE orbits and clocks were used (which contain products for all constellations except GEO and IRNSS).

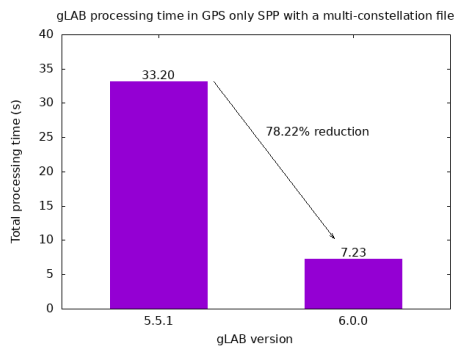
²The corresponding files are available at the following URL: https://server.gage.upc.edu/PhD/Ibanez_Deimos/Ibanez_Deimos_PhD_files.7z



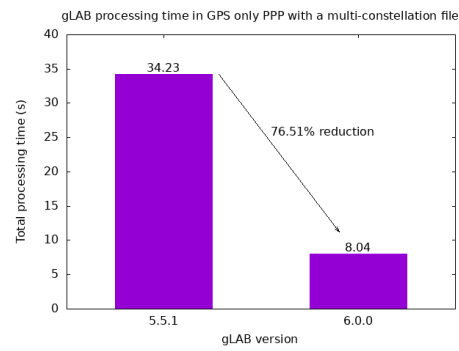
(a) GPS only SPP processing times with GPS only RINEX observation file.



(b) GPS only PPP processing times with GPS only RINEX observation file.



(c) GPS only SPP processing times with all constellations RINEX observation file.



(d) GPS only PPP processing times with all constellations RINEX observation file.

Figure 6.5: gLAB processing times comparison with versions 5.5.1 and 6.0.0 with a GPS only SPP (left-hand column) and PPP (right-hand column) processing modes with 1 Hz RINEX observation files containing GPS only data (top row) or all constellation data (bottom row).

The details of how the optimization was performed are highly technical, so they are included in annex A.

6.6 Multi-threading

In order to increase processing speed, aside from optimizing the code (shown in section 6.5), multi-threading was implemented to parallelize code execution. However, only the code sections that meet the following criteria can be parallelized:

- Code executed by each thread must be independent from each other (in terms of input data and memory accesses), as the exact order of execution is unknown.
- Code executed in parallel must be long, otherwise the additional time spent in creating and synchronizing the multiple threads will be greater than the time saved by the parallel execution of code.

In gLAB, the only sections that meet these criteria are the cycle-slip detection and the satellite modelling, which are the only sections executed in multi-threading.

The implementation of multi-threading was performed using the OpenMP library (OpenMP [2023]), which automatically adds all the multi-threading code in the section indicated by the user with preprocessing directives. However, the programmer needs to make sure that the sections to be executed in multi-threading do not have any race conditions (errors produced by concurrent access to shared variables). In this regard, several functions of gLAB code had to be modified or moved entirely to another section (for instance, the function that checks for alarm messages in SBAS GEO satellites).

As processing times in gLAB in positioning modes (SPP, PPP, etc.) are already low, the most benefited processing mode for multi-threading is SBAS availability maps (when only one day needs to be processed). In table 6.5 it is provided the comparison of execution for EGNOS availability computation (on the 26th January 2018) in single and multi-threading (4 and 20 threads), executed on an Intel i9-12900KS CPU (with the default configuration, which covers from -30° West to 40° East in longitude and from 25° to 70° North)³:

Table 6.5 shows the processing time separated by “user” and “kernel” time (the percentages are provided by gLAB). The “user time” is the actual time spent in code execution, while the “kernel time” measures the time spent in system calls (I/O, memory allocation and freeing, thread creation and synchronization).

³The corresponding files are available at the following URL: https://server.gage.upc.edu/PhD/Ibanez_Deimos/Ibanez_Deimos_PhD_files.7z

	Processing time			
	User time (all threads)	Kernel time (all threads)	Total processing time (all threads)	Wall time
Single thread	31m 0.48s (99.998%)	0.044s (0.002%)	31m 0.792s	31m 0.792s
Multi-thread (4 threads)	1h 5m 24.419s (99.388%)	24.159s (0.612%)	1h 5m 48.478s	16m 27.205s
Multi-thread (20 threads)	6h 34m 54.589s (99.832%)	39.763s (0.168%)	6h 35m 34.352s	19m 47.247s

Table 6.5: gLAB single and multi-thread (with four and 20 threads) EGNOS availability map processing times (on the 26th January 2018). Cells in green indicate the best processing times. User time, kernel time and processing time refer to the total amount of time spent by each thread running in parallel (on user space, kernel space and both respectively). Wall time refers to the actual time passed.

From the values in table 6.5, the following conclusions can be drawn:

- Multi-threading has a much larger total processing time than single thread when summing the processing time of all concurrent threads, but as they can run in parallel, the wall time (the user-perceived time) is smaller than single-thread execution.
- The “kernel time” spent in multi-threading is much higher than in single-thread mode. This is because multi-thread requires additional code for creating and synchronizing the threads (executed by the OS).
- Using more threads does not always translate into less wall time. Above a certain number of threads, the wall time will actually increase. This is because when a large number of threads is used, it is not always possible to distribute work in all threads at all times, therefore there will be small periods during which some threads are idle, but they still have to run through the synchronization code.

Moreover, when using multi-threading, the architecture of the CPU and cooling of the CPU directly affect performance. In this case, with the Intel i9-12900KS CPU, the following behaviour occurs (according to its specifications on [Intel \[2022\]](#)):

- Each core can handle two threads simultaneously.
- In single threaded applications, one core can boost its frequency up to 5.5 GHz (provided there is enough cooling).
- The CPU has two types of cores. There are eight E-cores (Efficiency cores) and eight P-cores (Performance cores). The former have a maximum frequency of 4.0 GHz, while the latter have a maximum frequency of 5.2 GHz. P-cores will be preferred for intensive processing tasks and will all be boosted to maximum frequency if there is enough cooling.

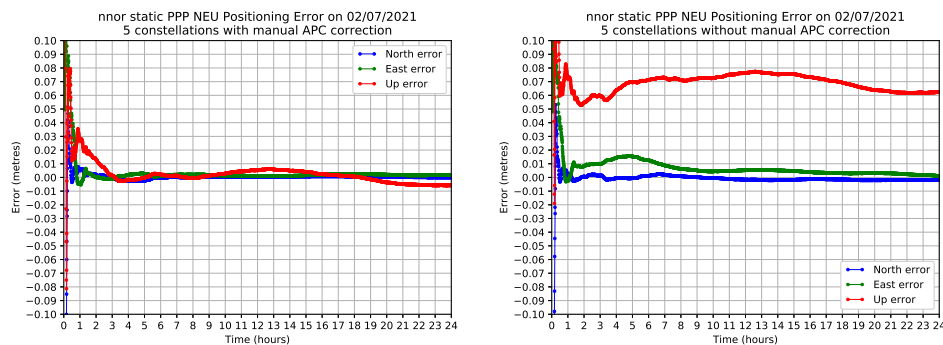
In the test results shown in table 6.5, the single-thread test was run with one core with a clock frequency of 5.5 GHz, the four-thread test was run using only P-cores at 5.2 GHz, while the 20 thread test was run with P-cores and E-cores at 5.2 GHz and 4.0 GHz respectively. The effect of the difference in clock frequency and core type is difficult to measure (and is out of the scope of this thesis), but good practice is to use only the cores that can boost to a higher frequency.

6.7 Receiver Antenna Phase Centre correction

As explained in section 4.2, PPP positioning requires the receiver APC correction, which is read from an ANTEX file, but IGS ANTEX files only have APC correction for GPS L1 and L2 frequencies and GLONASS G1 and G2 frequencies for currently installed receivers. Since GPS week 2148 (since 13/03/2021) the ANTEX file started to include APC for all constellations and frequencies, but only for new receivers.

Therefore, when computing a multi-constellation PPP with Galileo, BDS, QZSS or IRNSS, a bias will appear, as the receiver APC is not corrected (the APC mainly consists of a bias). An example is shown in Fig. 6.6, where the same processing as in Fig. 4.4b (in section 4.2, also provided below) is performed, but without manually correcting the receiver APC⁴:

⁴The corresponding files are available at the following URL: https://server.gage.upc.edu/PhD/Ibanez_Deimos/Ibanez_Deimos_PhD_files.7z



(a) Static PPP with 5 constellations with manual receiver APC correction. (b) Static PPP with 5 constellations without manual receiver APC correction.

Figure 6.6: Static PPP with 5 constellations with and without manual receiver APC correction.

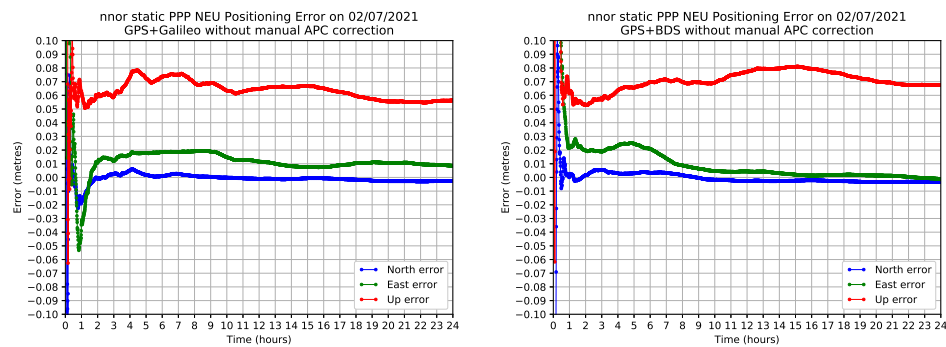
In Fig. 6.6b a bias is clearly visible in the up component of 0.6 cm, which is the combined bias of all uncorrected receiver APC. There is also a small oscillation of less than 1 cm (more visible in Fig. 6.6a), which can be explained to the receiver APC correction term dependent on the elevation of the satellite, which cannot be manually corrected.

To overcome this issue (which also makes gLAB validation more difficult), gLAB has two options for providing the receiver APC correction:

1. Provide the numerical correction (in North, East and Up components) on a per constellations and frequency basis.
2. Instruct gLAB to use the data from another constellation and frequency on a per constellations and frequency basis. This assignment can be arbitrary (e.g. the GPS receiver APC correction from L1 frequency can be used for IRNSS S frequency).

The second option is preferred, as it will copy both the constant term and the elevation-dependent term. Unfortunately, this is only possible for QZSS, as both frequencies used in the PPP are the same as for GPS.

For the two constellations remaining, Galileo and BDS, the only option is to manually provide the numerical correction to gLAB. To do so, a rough estimation must be made by performing a PPP with GPS and the constellation to which the receiver APC needs to be measured. As in this case two constellations need their receiver APC measured, two PPP must be performed. The results are shown in Fig. 6.7:



(a) Static PPP with GPS and Galileo without manual receiver APC correction. (b) Static PPP with GPS and BDS without manual receiver APC correction.

Figure 6.7: Static PPP with GPS and Galileo (left) or BDS (right) without manual receiver APC correction.

From Fig. 6.7a, the receiver APC correction value at the end of the processing is around 6 cm (measured as the vertical offset in the vertical component), while in Fig. 6.7b the correction is around 7 cm. These values correspond to the APC correction of the ionosphere-free combination, but gLAB only accepts the correction on a per frequency basis. Therefore, either the user must calculate the equivalent correction to be applied in only one of the frequencies, or by trial and error guess the right value, which usually is within ± 3 cm of the estimated ionosphere-free correction. For this specific case, the adequate correction for both constellations was 5 cm in the vertical component.

Scientific studies with gLAB

In this chapter the main scientific studies conducted by the thesis author are explained, where gLAB played a main role as a GNSS data processing tool. The first two studies are in the context of two projects: one with European Union Agency for the Space Programme (EUSPA) assessing the suitability of EGNOS for maritime use and another with European Space Agency (ESA) for developing a Global Monitoring System (GMS) for SBAS and DGNSS. The other contributions are four peer-reviewed scientific papers and one conference proceeding to the Institute Of Navigation (ION), where the author was strongly involved in the data processing.

7.1 EGNOS for maritime

7.1.1 Maritime requirements

EGNOS was designed for civil aviation (like all other SBAS, as shown in section 4.3). Due to its cost-effectiveness (it does not require additional frequencies in the GNSS receiver), the fulfilment of maritime requirements by EGNOS (and other SBAS as well) is being evaluated for use by ships, vessels, etc.

Maritime requirements are within the scope of the International Maritime Organization (IMO), which is the United Nations (UN) authority for the safety, security, and environmental performance of international shipping, as defined in the Safety of Life at Sea (SOLAS) convention of 1974 (IMO [1974]). This convention is updated and amended through resolutions by the IMO Assembly, the IMO Maritime Safety Committee (MSC). The current maritime requirements in force are established in its resolution A.1046(27) (IMO [2011]), shown in table 7.1:

Parameter	Navigation Phase	
	Ocean Waters	Harbor Entrances, Harbor Approaches, and Coastal Waters
Horizontal Accuracy 95%	100 m	10 m
Signal Availability	99.8%	99.8%
Service Continuity (over 15 min)	N/A	99.97%
Position update rate	2 s	2 s
Time-To-Alert (TTA) ^a	Maritime safety information as soon as practicable	10 s
System Coverage	Adequate ^b	Adequate ^b

Table 7.1: IMO A.1046(27) requirements (from IMO [2011]).

^a Generation of integrity warnings in cases of system malfunctions, non-availability, or discontinuities.

^b Making into account the radio frequency environment, the coverage of the system should be adequate to provide position fixing throughout this phase of navigation.

From the requirements in table 7.1, the position update rate of 2 s is compliant by design of GNSS, the TTA is also accomplished by the requirements for SBAS (in table 4.4). The horizontal accuracy and signal availability requirements are accomplished by EGNOS according to European Satellite Services Provider (ESSP) report ESSP [2019]. Therefore, the service continuity will be the requirement with higher focus, as it needs to be tested whether EGNOS achieves it.

In maritime use, service continuity is computed using the formula defined by International Association of Lighthouse Authorities (IALA) in its guidelines for monitoring DGNSS services (IALA [2015]), as follows:

$$\text{Continuity fixed window} = 1 - \frac{CTI}{MTBF} = 1 - n \cdot \frac{CTI}{\text{Total epochs}} \quad (7.1)$$

The Continuity Time Interval (CTI) is 15 min (due to the long time needed for vessel manoeuvres), and the Mean Time Between Failures (MTBF) is computed as the total number of epochs processed (independently whether or

not the epoch was available) divided by the total number of discontinuities (n). To illustrate the computation of continuity risk with the fixed window, Fig. 7.1 (extracted from the gLAB manual) shows an example of computation of continuity risk. In this figure, 'X' are epochs with PA solution below alarm limits, while 'O' are epochs with no PA solution or with a PA solution but above any AL. The window size in this example is set to five epochs.

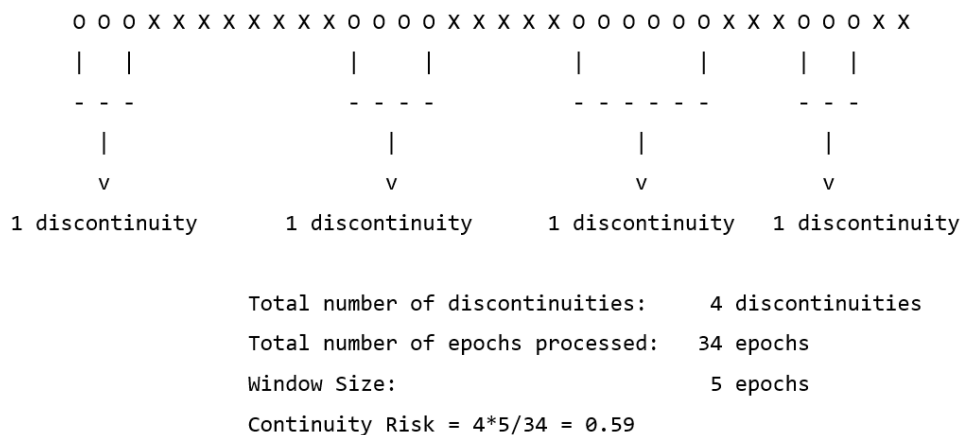


Figure 7.1: Continuity risk with fixed window computation example (from gLAB manual). 'X' are epochs with a PA solution below alarm limits, while 'O' are epochs with no PA solution or with a PA solution but above any AL. The window size in this example is set to five epochs.

In gLAB, the continuity with fixed window is computed together with the sliding window, shown in equation 7.2 and described in section 4.3:

$$\text{Continuity sliding window} = 1 - \frac{NEAD}{\text{Avail epochs}} \quad (7.2)$$

Both continuity methods are computed independently of the SBAS mode (aviation or maritime), as the code overhead is very small (as both share the code to detect discontinuities). Note that the formula 7.1 is for the continuity value. The continuity risk is equal to the difference between the continuity and one.

Moreover, in maritime mode, the degradation factors (factors that indicate the degradation of all SBAS corrections over time) transmitted in SBAS MT10 are not used. As stated in appendix S of [RTCA-MOPS \[2006\]](#), when MT10 is not used, instead of applying the degradation parameters, a conservative bound of eight metres is used.

7.1.2 EGNOS for maritime performance study

In EUSPA Specific Contract 1 under framework contract “Integration of Fundamental Elements”, developed by gAGE/Technical University of Catalonia (UPC), where the PhD author was responsible for performing an assessment of EGNOS performance with maritime requirements for a period of two years (May 2016 to April 2018). The study contained the following assessments:

1. Signal In Space (SIS) availability.
2. Service availability.
3. Service continuity.
4. Static positioning accuracy (using fixed stations on the coast).
5. Kinematic positioning accuracy (mounting a GNSS receiver in a vessel around the Canary Islands).

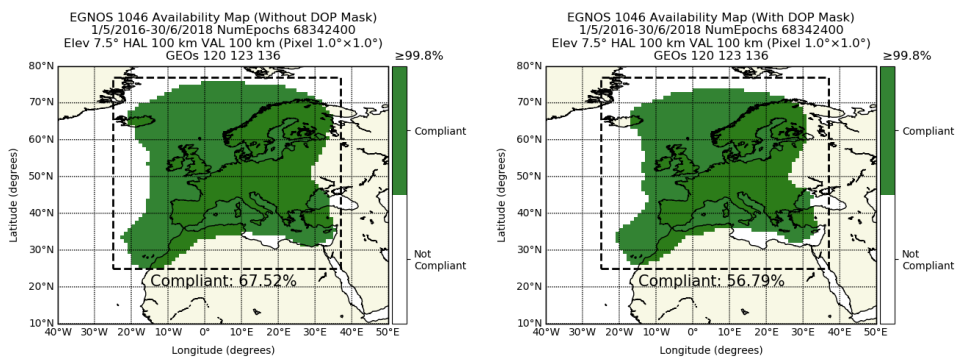
The first four assessments results were published in Ibáñez et al [2020] and the latter in Ibáñez et al [2019] with the two year study, but the results in this section will refer to the final report provided to EUSPA (EUSPA [2019]), which contains the assessments for an extended period of two years and two months and additional results not published in these papers. In this report, although not initially foreseen, the following comparisons were made:

- A comparison of the availability with and without a DOP mask (geometries with $PDOP \geq 6$ and $HDOP \geq 4$ are filtered, according to MSC [2015]).
- A comparison of the continuity with and without a DOP mask.
- A comparison of the continuity computed with a sliding and a fixed window.
- A comparison of the Positioning Error (PE) and PL obtained with and without the usage of SBAS MT10.

The results of these comparisons will be shown in the following sections, as they are the ones with the most scientific interest.

7.1.2.1 EGNOS for maritime availability

Fig. 7.2 shows the availability results from May 2016 to June 2018 (from EUSPA [2019]). Green pixels show where the availability requirement of 99.8% is met. To quantify the results, a fictitious FIR of coordinates (25°W, 25°N), (25°W, 77°N), (37°E, 77°N), (37°E, 25°N) was defined, where the availability percentage is computed as the number of pixels in green inside the FIR divided by the total number of pixels inside the FIR. The pixel resolution is 1° per 1° without interpolation, as this produces the most accurate results.



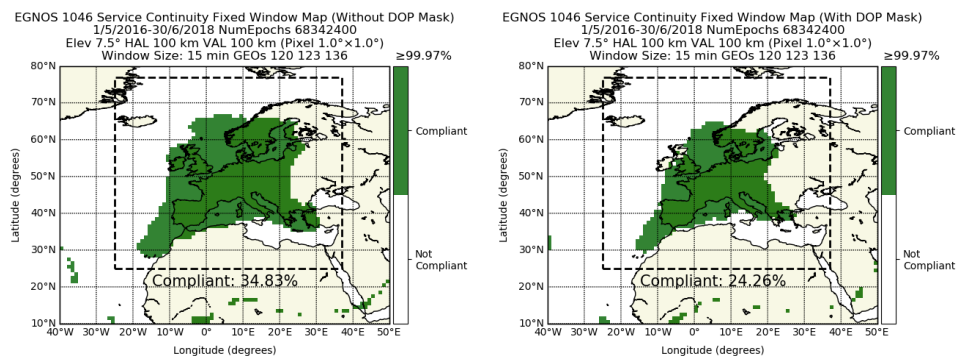
(a) EGNOS maritime availability without a DOP mask (from EUSPA [2019]). (b) EGNOS maritime availability with a DOP mask (from EUSPA [2019]).

Figure 7.2: EGNOS maritime availability maps (equal or greater than 99.8%) without a DOP mask (left) and with a DOP mask (right) from May 2016 to June 2018.

The difference in availability due to the DOP mask is 10.73%, with the most relevant difference being in the North (especially Iceland). This is because in the North there are fewer and more scattered EGNOS Ranging and Integrity Monitoring Stations (RIMS) due to the sea, while in the south there is a higher density of RIMS (RIMS map can be found in EU [2017]).

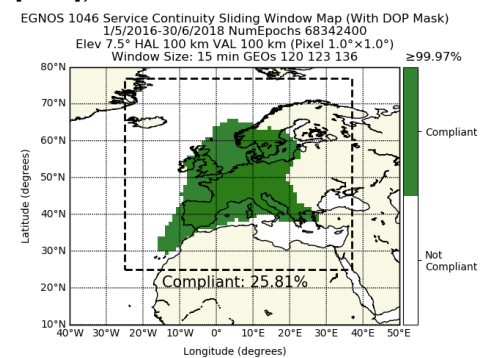
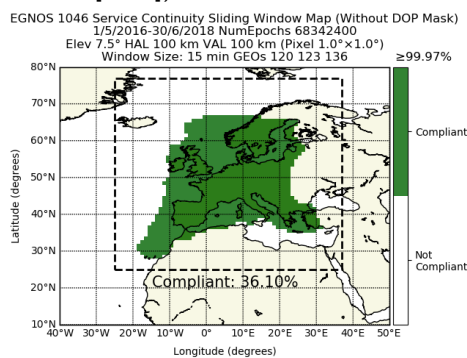
7.1.2.2 EGNOS for maritime continuity

Fig. 7.3 shows continuity maps without a DOP mask (left-hand column) and with a DOP mask (right-hand column) and using a fixed window (top row) or a sliding window (bottom row). The timespan, figure format, FIR area and computed percentage value are the same as for the availability maps, but using a threshold of 99.97%.



(a) EGNOS maritime continuity using fixed window without a DOP mask (from EUSPA [2019]).

(b) EGNOS maritime continuity using fixed window with a DOP mask (from EUSPA [2019]).



(c) EGNOS maritime continuity using a sliding window without a DOP mask (from EUSPA [2019]).

(d) EGNOS maritime continuity using a sliding window with a DOP mask (from EUSPA [2019]).

Figure 7.3: EGNOS maritime continuity maps (equal or greater than 99.97%) without a DOP mask (left-hand column) and with a DOP mask (right-hand column) and using a fixed window (top row) or a sliding window (bottom row) from May 2016 to June 2018.

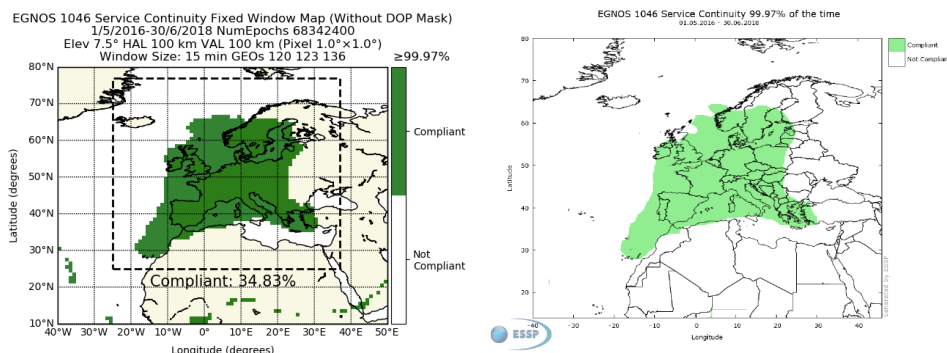
From figures in 7.3, the following conclusions can be made:

- With the same configuration, the two methods provide similar results (the coverage area has the same shape), proving the results are consistent.

- The DOP mask reduces the continuity by a similar amount for both methods (10.57% with the fixed window and a 10.29% with the sliding window).
- The usage of a sliding window provides slightly better results than the fixed window (1.27% without the DOP mask and 1.55% with the DOP mask). In the case without the DOP mask, the difference is not very relevant (it is mainly in the sea), but with the DOP mask, it affects Scotland and Ireland.
- The continuity in the fixed window has multiple compliant continuity pixels in the very south (10°S) well within the border of EGNOS coverage, which is not true. This is due to a limitation in the computation of continuity with a fixed window, as it only accounts for the number of discontinuities, but not for the actual available epochs. In these pixels, it so happens that there are very few available epochs, and thus very few discontinuities (transitions from available to not available), which makes the quotient in equation 7.1 small enough to meet the requirement. Moreover, the continuity with the fixed window could even become negative if there were too many discontinuities (e.g. a system continuously switching between available and unavailable), the quotient would be greater than 1 (as the product of the number of discontinuities and the window size would be greater than the total number of epochs processed), which in the author's opinion, makes no sense.

Moreover, in the comparison of the continuity computed with gLAB and by ESSP (using only the fixed window method and without the DOP mask), the results agreed in most part, except in Scotland and Ireland, as shown in Fig. 7.4. The root cause of this difference was solely due to the criteria used for switching GEO. gLAB keeps the same GEO while it provides a valid navigation solution. When the GEO does not provide a valid solution, it tries with the next one available until one of them provides a valid solution or until the list is exhausted (in this case the epoch is skipped). This process is performed for every epoch. ESSP, in order to switch GEO, has to wait four epochs without a valid solution to perform a GEO switch¹.

¹This information was kindly provided by ESSP.



(a) gLAB EGNOS maritime continuity using fixed window without a DOP mask (from [EUSPA \[2019\]](#)). (b) ESSP EGNOS maritime continuity using fixed window without a DOP mask (from [Magdaleno et al \[2019\]](#)).

Figure 7.4: gLAB and ESSP continuity maps without a DOP mask and fixed window.

7.1.2.3 EGNOS for maritime positioning

The 108 stations selected (from multiple networks) for static positioning were at most 20 km away from the coast, and with a data rate of 1 Hz. The station map is shown in Fig. 7.5 (the map was prepared using the gLAB DAT tool):

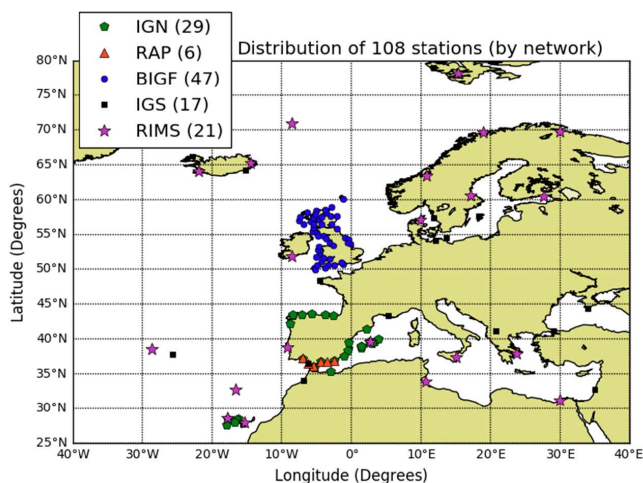


Figure 7.5: Station map used in EGNOS maritime study (from [EUSPA \[2019\]](#)). 6 stations are from the Andalusian Positioning Network (RAP); 29 stations are from the Spanish National Geographic Institute (IGN); 35 stations are from the British Isles continuous GNSS Facility (BIGF); 17 stations are from the IGS; and 21 are RIMS. Specifically, at each RIMS, the receiver designated as “A” from the two co-located receivers “A” and “B” was used.

Even though maritime requirements do not impose a vertical protection limit or a vertical accuracy level, these figures were computed to complete the positioning assessment. In table 7.2 it is shown the aggregated positioning results from all stations for the two years and two months period (the same as in section 7.1.2.1) with and without using the MT10.

Metric	MT10	95%	99%	99.9%	MAX	RMS	Mean
HPE	NO	0.91	1.29	2.20	100.58	0.51	0.44
	YES	0.86	1.20	2.07	100.39	0.48	0.41
VPE	NO	1.68	2.30	3.58	2124.02	0.95	0.68
	YES	1.57	2.15	3.47	1100.68	0.85	0.65
HPL	NO	70.32	107.48	193.58	812.656	49.00	46.40
	YES	14.97	28.21	74.38	764.16	11.11	9.66
VPL	NO	118.19	175.50	367.03	26725.38	85.22	73.51
	YES	24.76	44.23	131.14	16856.30	20.21	15.85

Table 7.2: Maritime static positioning aggregation table for all 108 stations for the Horizontal Positioning Error (HPE), Vertical Positioning Error (VPE), HPL and VPL with and without MT10. For each of these metrics, it is depicted the 95, 99, 99.9 percentiles, maximum value, Root Mean Square (RMS) and mean. All values are in metres.

The positioning results show that the horizontal accuracy is far below the 10 m requirement, even for the 99.9% percentile. This is also met in the vertical accuracy, although there is no requirement. The difference between using MT10 in the positioning error is small (between 0.05 to 0.15 metres depending on the percentile), but in the protection levels it is very high (an increase factor of 2.6 to 4.7 depending on the percentile), which makes them too high and not representative. In the author's opinion, MT10 should be used, as not using this message does not provide any advantage to the user.

Finally, it is worth mentioning that following this analysis, the possibility is under consideration (by the maritime authorities) of using the sliding window for continuity computation or the usage of the DOP mask in maritime mode, but at the time of writing this thesis the discussion is still on-going.

7.2 Global Monitoring System

In the CCN of ESA/ESTEC project “gEGNOS: gLAB Upgrade for EGNOS Data Processing”, one of the tasks was to develop from scratch a Global Monitoring System (GMS) for monitoring SBAS and DGNSS (which suits as a demonstrator of gLAB capabilities). The GMS computed the following metrics per day:

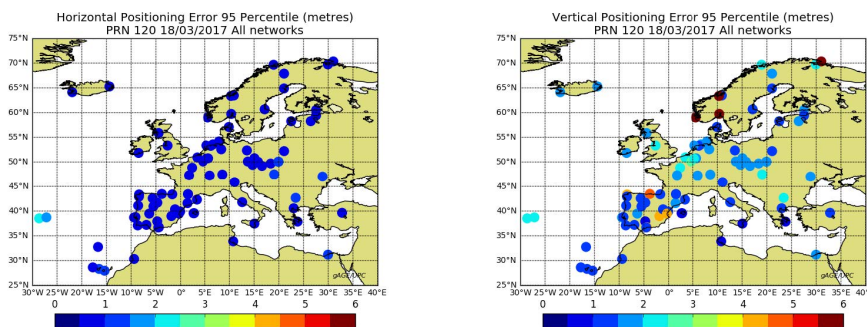
- Daily and hourly availability and continuity maps for EGNOS coverage area for each EGNOS GEO.
- Daily and hourly availability and continuity maps for WAAS and EGNOS coverage area using all GEO (figures 4.11 and 4.12 in section 4.3) are from the GMS.
- The positioning error plot with the selected stations and for each EGNOS GEO.
- The horizontal and vertical Stanford plot with the selected stations and for each EGNOS GEO.
- The horizontal and vertical Stanford-ESA plot (Tossaint et al [2007]) with the selected stations and for each EGNOS GEO.
- An horizontal and vertical 95 percentile positioning error map with the selected stations and for each EGNOS GEO.
- An horizontal and vertical 95 percentile protection level map with the selected stations and for each EGNOS GEO.
- An horizontal and vertical Worst Integrity Ratio (WIR) map with the selected stations and for each EGNOS GEO.
- For DGNSS, a list of manually selected pairs of stations (each station pair close to the other), and the DGNSS positioning is computed with each pair in turn (using the other as the reference station).

The GMS was run automatically on a daily basis and the results published through a web server (it was publicly accessible if access was requested to gAGE group, but is currently discontinued). The initial screen is shown in Fig. 7.6:



Figure 7.6: GMS website initial screen (from Ibáñez et al [2018]).

SBAS data was downloaded from Centre National d'Études Spatiales (CNES), while the station RINEX observation files were downloaded from IGS, EUREF Permanent Network (EPN) and EGNOS Data Access Service (EDAS). Nearly all of the stations available with 1 Hz data rate were used, which accounted for more than 100 stations. An example of positioning error map is shown in Fig. 7.7 (showing also the station distribution):



(a) Horizontal 95th percentile PE map on 18/03/2017 with GEO 120 (from Ibáñez et al [2018]).

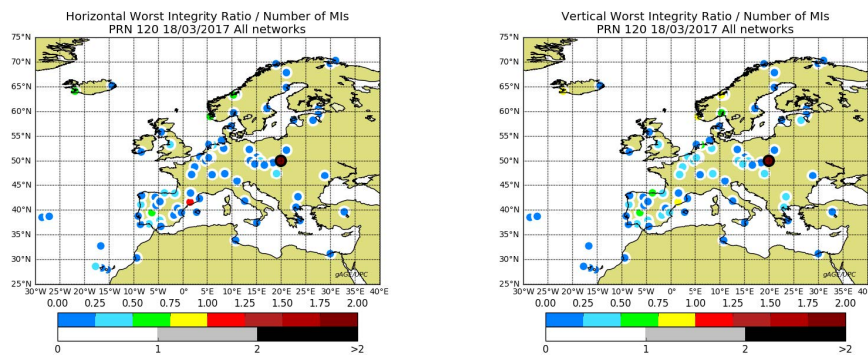
(b) Vertical 95th percentile PE map on 18/03/2017 with GEO 120 (from Ibáñez et al [2018]).

Figure 7.7: Horizontal and vertical 95th percentile PE map maps on 18/03/2017 with GEO 120.

In the processing, no protection against local errors was used (e.g. profit outlier detector), as the goal was to search for SIS or SBAS message errors, which

could potentially be filtered out by such protection.

The WIR is the largest ratio between the PE and the PL in the worst geometry computed. If this ratio is equal or greater than one, an MI has occurred. An example of a WIR map is shown in Fig. 7.8, which has, for each station, an inner circle with the WIR for all possible combinations from four to all-in-view satellites and an outer corona for the all-in-view combination.



(a) Horizontal WIR map on 18/03/2017 with GEO 120 (from Ibáñez et al [2018]). (b) Vertical WIR map on 18/03/2017 with GEO 120 (from Ibáñez et al [2018]).

Figure 7.8: Horizontal and vertical WIR maps on 18/03/2017 with GEO 120. The colour of the inner circle in each station indicates the maximum PE and PL ratio for all possible combinations from four to all-in-view satellites. The outer corona surrounding the WIR colour circle indicates the number of MIs when using all-in-view satellites.

From figures 7.8a and 7.8b, the following conclusion can be derived:

- The station at (50°N, 20°E) shows MIs for the all-in-view satellites set. Indeed, the outer corona in black indicates a number of all-in-view MIs of two or greater. Moreover, the inner corona in magenta indicates a WIR of two or larger.
- MIs are due to local problems (multipath, receiver errors, etc.) and not to the SIS, as the receivers surrounding this station did not experience any MI.
- The station in the north-east of Spain (about 41°N, 2°E) experiences MIs for specific satellite geometry combinations, but not for the all-in-view satellites, as shown by the inner corona in red, surrounded by a white corona. This example illustrates the usefulness of exploring all possible combinations of satellites to fully guarantee integrity (which is performed in the Stanford-ESA plot).

- As no MIs are experienced by the other stations around the previously mentioned, we can conclude that they are linked to local effects (multipath, receiver anomaly, etc.). Indeed, any SIS event or error in SBAS messages will create MIs in multiple stations, as satellite clocks and ephemeris are common to all stations and ionosphere has a strong spatial correlation.

Finally, during the time that the GMS was online (nearly three years), it did not detect any MI in the EGNOS SIS (as expected). During this period, there were no SIS anomalous events in GPS satellites.

7.3 AATR and SBAS

One of the open problems in GNSS positioning is to be able to detect anomalous conditions in the ionosphere (large gradients, ionospheric disturbances, scintillation, etc.), which is the upper part of the atmosphere where propagation of radio waves can be affected due to the quantity of ions and free electrons (Rawer [1963b]). In order to monitor the status of the ionosphere, several metrics have been developed. A customary choice is to use the time derivative of the ionosphere status, such as the Rate Of change of Total electron content Index (ROTI) (Pi et al [1997]), or the Along Arc TEC Rate (AATR) index, proposed in Juan et al [2018].

The AATR index is well correlated with the ionospheric activity and, unlike other global indicators linked to the geomagnetic activity (i.e. D_{ST} or A_P , defined in Rostoker [1972]), it is sensitive to the regional behaviour of the ionosphere and identifies specific effects on GNSS users. Moreover, from a devoted analysis of different SBAS performances in different ionospheric conditions, it follows that the AATR indicator is a very suitable mean to reveal whether SBAS service availability anomalies are linked to the ionosphere (see section 7.3.2).

7.3.1 AATR definition

To compute the AATR index using GPS measurements, the steps are (according to Juan et al [2018]):

1. Compute the Geometry Free (GF) combination (using frequencies L1 and L2) using equation 7.3, where $L_{1_i}^j$ and $L_{2_i}^j$ are the carrier phase measurements for station i and satellite j of frequencies L1 and L2 respectively:

$$L_{GF_i}^j = L_{1_i}^j - L_{2_i}^j \quad (7.3)$$

Once the GF is computed, cycle-slips have to be detected using this combination. gLAB dual frequency cycle-slip detectors and its output prints (as explained in sections 6.2 and 6.3) can be used, as gLAB has a cycle-slip detector based on the GF combination.

2. Compute the Slant Total Electron Content (STEC) variation ($\Delta STEC$) for station i and satellite j between two consecutive observations separated Δt (usually at a 30 second rate) with the formula 7.4:

$$\Delta STEC_i^j(t) = L_{GF_i}^j(t) - L_{GF_i}^j(t - \Delta t) \quad (7.4)$$

Note that formula 7.4 has to be computed only between epochs in the same arc (otherwise a jump would occur due to the difference in the carrier phase ambiguity values), therefore it is critical to correctly detect cycle-slips.

3. Compute the instantaneous AATR index for station i and satellite j with formula 7.5:

$$AATR_i^j(t) = \frac{1}{(M(\varepsilon))^2} \frac{\Delta STEC_i^j(t)}{\Delta t} \quad (7.5)$$

Where $M(\varepsilon)$ is the mapping function, defined as the secant of the zenith angle at the mean ionospheric height (350 km), which converts the vertical delay to slant delay.

4. Compute the Root Mean Square (RMS) of the instantaneous $AATR_i^j(t)$ for a pre-defined period for all j satellites in view from a station, resulting in the following AATR index per a given receiver i using equation 7.6:

$$RMS_{AATR,i}(t) = \sqrt{\frac{1}{N} \sum_{t=T}^{T+\Delta T} \sum_{j=1}^{n_{sat}(t)} (AATR_i^j(t))^2} \quad (7.6)$$

Where N is the total number of observations during the selected interval ΔT (usually one hour), after adding the AATR of all satellites in view ($n_{sat}(t)$) at each epoch t .

7.3.2 AATR application

As seen on section 7.3.1, the AATR is computed by using differences in time of the GF combination, which is measuring the variation in the STEC value in time. Therefore, it allows to measure the ionospheric perturbation using solely GNSS measurements. This fact is very important, as many of the applications using GNSS have their performance very correlated with the ionosphere activity. In particular, one of the services is SBAS, as they need to provide reliable ionospheric corrections. Indeed, the AATR index has been adopted as a standard tool by the International Civil Aviation Organization (ICAO) for joint ionospheric studies in SBAS (ICAO [2015]).

For instance, on Fig. 7.9 it is compared the daily maximum value of AATR in “mas1” station (located in the Canary Islands) with the EGNOS APV-I availability for a fault-free receiver.

The correlations between both terms can be clear seen (i.e. high ionosphere activity resulted in lower EGNOS availability) until the EGNOS update v.2.4.1M around July 2015:

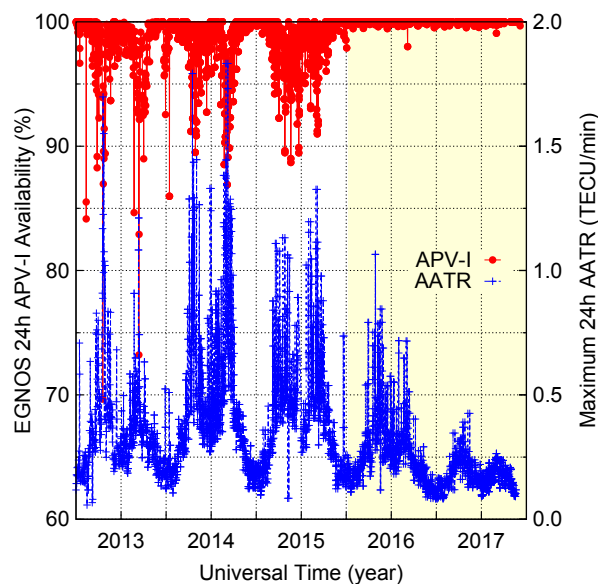


Figure 7.9: Daily maximum AATR value (blue pulses) for “mas1” station vs. EGNOS APV-I availability (red points) for a fault-free receiver, both located in the Canary Islands. The yellow shadow indicates the EGNOS update to v.2.4.1M performed around July 2015 (from Juan et al [2018]).

7.3.3 Planetary index AATR

The AATR index defined in 7.3.1 is a local index, but it can be extended to be a planetary index. According to Juan et al [2018], the proposed planetary index AATR can be computed following formula 7.7, where the AATR indexes situated between MODified DIP latitude (MODIP) latitudes (Rawer [1963a]) of 45 and 55° and a Local Time (LT) between 12 and 14 hours (at noon) are averaged:

$$\overline{\text{AATR}} = \frac{1}{N_{\text{AATR}}} \sum_{\text{LT}=12}^{14} \sum_{\text{MOD}=45}^{55} \text{AATR}_{\text{MOD}}(\text{LT}) \quad (7.7)$$

Where N_{AATR} is the number of AATR indexes used and AATR_{MOD} the local AATR index.

With a planetary AATR index, it is possible to study global events. For instance, in Fig. 7.10 it is depicted the correlation between the planetary AATR index and SBAS availability maps (using GEO 133 for WAAS and GEO 120 for EGNOS), which was the main author contribution to the paper. To compute these maps, it was necessary to add in gLAB the capability to compute hourly availability maps (the customary processing used in the aviation community analysis was to compute daily maps only). On the top part of the image the AATR index for the Saint Patrick storm on 17th March 2015 is shown (National Weather Service [2015]), with a green rectangle (at 2 a.m.) indicating a low activity hour, and two red rectangles (at 4 p.m. and 11 p.m.) with the top values of the AATR index. Below, combined APV-I availability maps of both WAAS and EGNOS at these three hours are provided. At 2 a.m., before the development of the geomagnetic storm, both SBAS have complete coverage of North America and Europe. At 4 p.m., WAAS has its availability reduced in Alaska and North East of Canada, while in Europe North and East regions are the most affected. At 11 p.m., WAAS has recovered the Alaska coverage, but it is being impacted in the East coast of USA, while EGNOS has lower availability than at 4 p.m. This difference in the overall availability due to ionospheric activity is regional. Furthermore, storm evolution is coupled with the diurnal variation (night/day).

Finally, it is worth noting that although the AATR indexes (local and planetary) in Juan et al [2018] were computed in post-processing, the AATR indexes can be computed in real time, as shown in Timoté [2023].

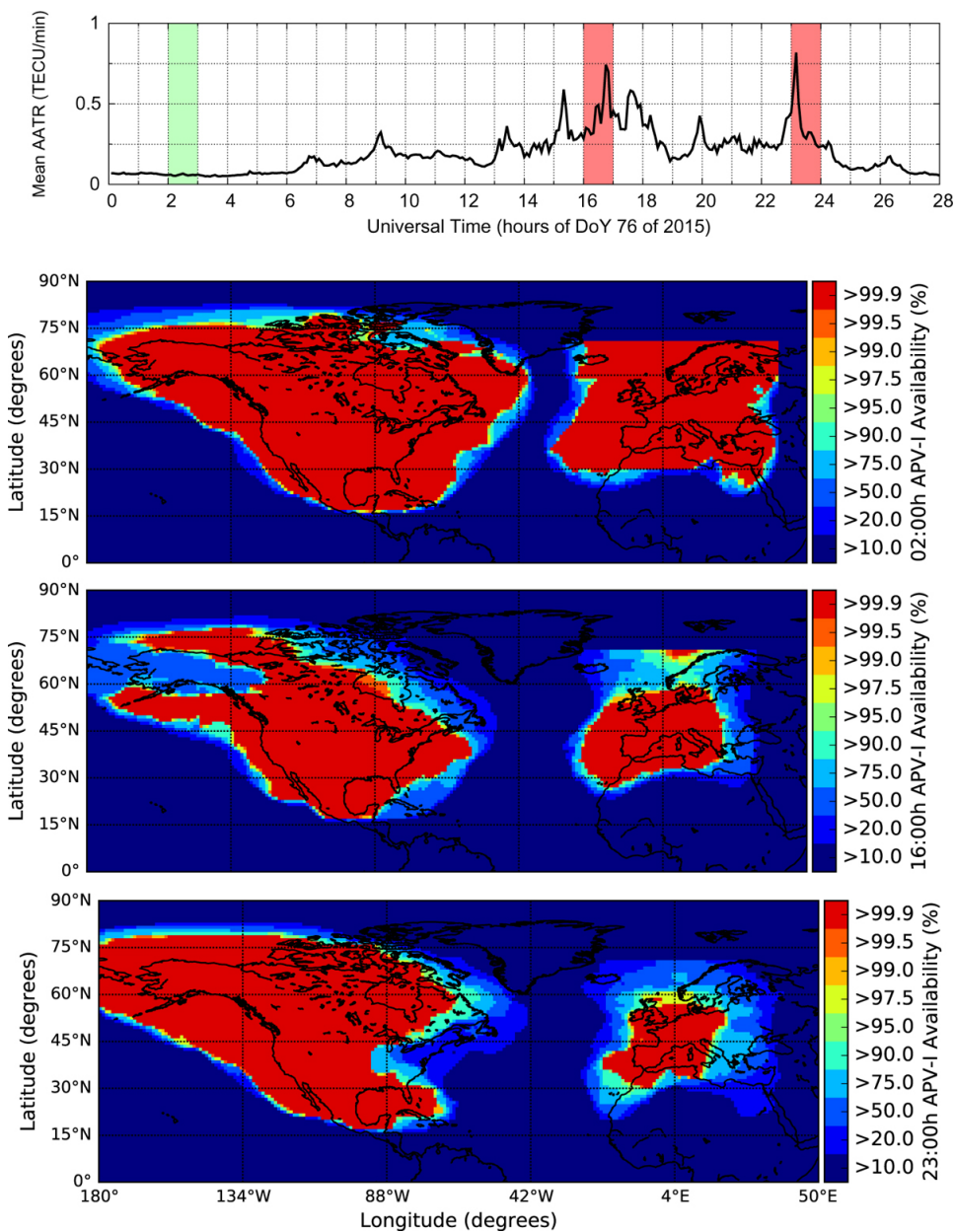


Figure 7.10: AATR index (top) and SBAS APV-I availability maps (using GEO 133 for WAAS and GEO 120 for EGNOS) on 17th March 2015 (Saint Patrick storm) at 2 a.m. (nominal ionosphere activity), 4 p.m. and 11 p.m. (both with high ionosphere activity). When the mean AATR planetary index is high (marked in red), the availability of both WAAS and EGNOS is significantly reduced. The pixel resolution is 1° per 1° per without interpolation. Image is from [Juan et al \[2018\]](#).

7.4 Ionosphere correction assessment

7.4.1 Introduction

The ionosphere is one of the challenging source of signal delay for GNSS signals, due to its difficulty to model and its effects on GNSS navigation, as shown in section 7.3 (see for instance Rovira-Garcia et al [2016b]; Rovira-Garcia et al [2019]; Caamaño [2022]; Timoté [2023]). Dual frequency users can remove 99.9% of the ionosphere delay using the Ionosphere Free (IF) combination, as shown in equation 7.8 (Sanz et al [2013a]):

$$\mathbf{L}_{IF} = \frac{f_1^2 \mathbf{L}_1 - f_2^2 \mathbf{L}_2}{f_1^2 - f_2^2} \quad (7.8)$$

Where L_{IF} is the carrier phase IF combination, f_1 and f_2 the two frequencies used for the IF combination (for GPS L1 and L2 is used), L_1 and L_2 the two carrier phase measurements in these frequencies. The IF combination for pseudoranges uses the same equation, but replacing the carrier phase terms for the counterpart pseudoranges.

Single frequency users need to correct this delay (as explained in section 4.1), while users with high accuracy and rapid convergence times use the ionospheric delay of faster convergence (as explained in section 5.2). In both cases, the accuracy of the ionospheric model is directly translated to the navigation solution of the user.

There are currently two on-going problems with the ionospheric correction. One is to generate an accurate model (ideally in real time), and the second one is to assess the quality of the ionospheric corrections. This section will be devoted to the latter issue.

7.4.2 Assessing ionosphere corrections

The most straightforward method would be to compare it with a reference, but there are currently no reference ionospheric values accurate enough (centimetre level) (Rovira-Garcia et al [2019]). This has been a long-standing problem, and there are two main methods for performing the assessment. One is in the SIS domain, performed in Rovira-Garcia et al [2016b] (where the thesis author was responsible to compute the comparison between models). In paper Rovira-Garcia et al [2019], the proposed method is to assess the ionospheric corrections through the positioning error.

The usual technique for assessing the error in the position domain is by computing the navigation solution using precise orbits and clocks and a single pseudorange measurement. As pseudorange measurements are absolute, the accuracy of the ionospheric model translates directly to the accuracy of the navigation solution up to a limit, which is the pseudorange noise (usually around one metre). However, due to the large code noise, the use of pseudorange measurements does not provide accurate results for assessing the ionospheric corrections.

To overcome this limitation, [Rovira-Garcia et al \[2019\]](#) uses the Carrier phase to Code Levelling (CCL) method ([Mannucci et al \[1998\]](#)), which consists of estimating the carrier phase ambiguities (in post-processing) by averaging the difference between the Geometry Free (GF) combinations of pseudorange and carrier phase during the whole duration of an arc, using equation 7.9. With the ambiguities estimated and added to the carrier phases, they can be used as precise pseudoranges.

$$\mathbf{B}_{\text{CCL}}^j = \frac{1}{N} \sum_{i=1}^N \left(\mathbf{P}_{\text{GF}}^j(i) - \mathbf{L}_{\text{GF}}^j(i) \right) \quad (7.9)$$

Where P_{GF}^j and L_{GF}^j are the GF combinations of pseudorange and carrier phase respectively for satellite j for the duration of one arc (of length N). L_{GF} is computed using equation 7.3, while P_{GF}^j is computed as in equation 7.3, but with the sign changed ([Sanz et al \[2013a\]](#)).

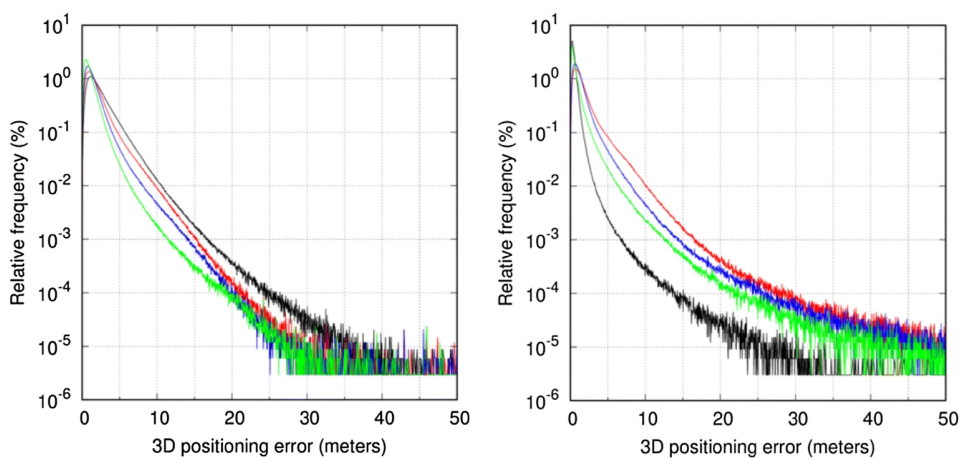
In [Rovira-Garcia et al \[2019\]](#), the dataset used 34 Multi-GNSS EXperiment (MGEX) stations distributed worldwide for the whole year of 2014. The ionosphere models assessed were “IGSG” IONEX, “UQRG” IONEX, Fast-PPP ionosphere model versus the Ionosphere Free (IF) combination (with frequencies L1 and L2) as a reference. The navigation position solution was computed with gLAB for each station, day and ionospheric model using C1C GPS measurement, except for the case with the IF combination, where no ionospheric model was needed because the IF already removes the ionosphere. This process was repeated, first with pseudorange measurements and later with CCL measurements. The Root Mean Square (RMS) error was then assessed by comparing the navigation solution with precise receiver coordinates.

The navigation solution was computed with gLAB with the two measurement types, pseudorange and CCL. The former is straightforward, as gLAB has always supported processing in single frequency with precise orbits and using a ionospheric model. The challenging one was the latter, since gLAB does not support using carrier phases as pseudoranges (note that the gLAB version was

from 2018). The workaround was to use the gLAB option to add user defined error to measurements, which allows an arbitrary value to be added to any measurement read from the RINEX observation file (before or after the cycle-slip detection). Using this option, the value added in each epoch subtracted the pseudorange value and adding the corresponding CCL measurement. In this way, gLAB processed the CCL measurement as a normal pseudorange measurement. The downside to this method is that the wind-up correction is not applied, but this correction is not critical for the level of accuracy required. Note also that it was not possible to add the estimated ambiguity to the carrier phase, as the gLAB version used at that time did not have the option to not estimate ambiguities when carrier phases are used (in the multi-constellation version presented in this thesis a new option has been added for this purpose).

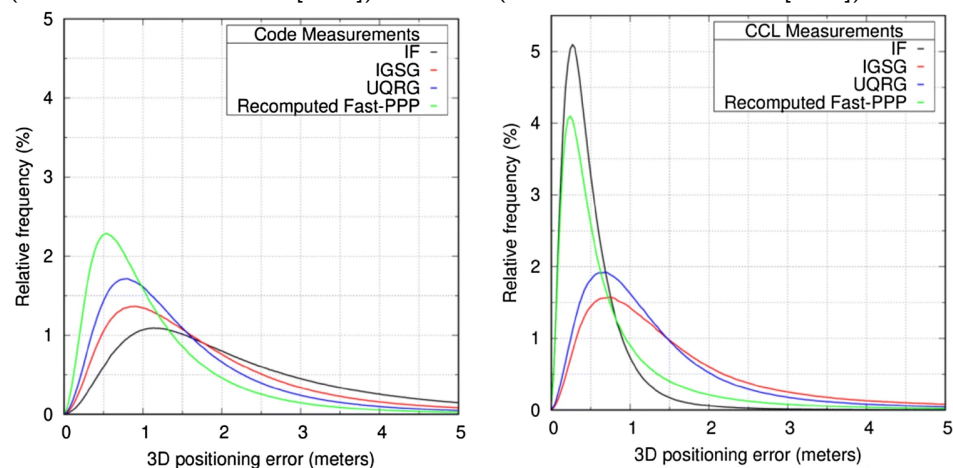
On Fig. 7.11 the frequency of the 3D positioning error for the 34 stations and the whole of 2014 with the two measurements types are shown. Fig. 7.11a and Fig. 7.11c depicts the results using code measurements, while Fig. 7.11b and Fig. 7.11d show the results with the CCL measurements. From these figures, the following can be concluded:

- All ionosphere models perform worse using code measurements than with CCL measurements, as the pseudorange noise is the main error source. The IF combination performs the worst as the noise has been increased by a factor of three.
- When using CCL measurements, the Fast-PPP ionosphere model and the IF combination perform much better, as the main error source is the error in the model itself (for Fast-PPP) or the estimation of the CCL (for the IF).
- “IGSG” IONEX and “UQRG” IONEX ionosphere models slightly improve their results, meaning that most of the error in the navigation solution was due to the error in the model itself.



(a) 3D positioning error frequency percentage using code measurements (from Rovira-Garcia et al [2019]).

(b) 3D positioning error frequency percentage using CCL measurements (from Rovira-Garcia et al [2019]).



(c) 3D positioning error frequency percentage using code measurements (zoom to first five metres) (from Rovira-Garcia et al [2019]).

(d) 3D positioning error frequency percentage using CCL measurements (zoom to first five metres) (from Rovira-Garcia et al [2019]).

Figure 7.11: 3D positioning error frequency percentage using 34 stations for the full year of 2014. The top row shows the positioning error up to 50 metres, while the bottom row is a zoom of the first five metres of error. The left-hand column shows the results using code measurements, while the right-hand column shows the results with CCL measurements.

7.5 NeQuick-G integration into gLAB

7.5.1 Introduction

When gLAB was first released, it only supported the GPS ionosphere correction model, which is the Klobuchar model (Klobuchar [1987]; USA Navigation Center [2022]). Later updates added support for IONEX files (Schaer et al [1998]) and for SBAS 1F ionosphere models. With the gLAB multi-constellation upgrade, it was necessary to add support for the Galileo broadcast ionosphere model, NeQuick-G (EC [2016]).

NeQuick-G ionospheric correction algorithm requires three parameters (a_0 , a_1 and a_2) as input data, which are transmitted in the Galileo navigation message (EU [2021]). To compute the ionospheric delay, the following steps are performed (according to Sanz et al [2013a]):

1. The *effective ionisation level parameter*, A_z , is computed using equation 7.10:

$$A_z = a_0 + a_1\mu + a_2\mu^2 \quad (7.10)$$

Where a_0 , a_1 and a_2 are the parameters sent in the navigation message and μ is the MODIP latitude.

2. Compute the electron density for a point in the satellite receiver path.
3. Repeat steps 1 and 2 for all the discrete points along the satellite receiver path.
4. Integrate all electron density values along the ray to obtain the STEC (in TECUs).
5. Convert the STEC from TECUs, to metres for correcting the pseudorange, using equation 7.11:

$$I_f = \frac{40.3 \cdot 10^6}{f^2} \text{STEC} \quad (7.11)$$

Where STEC is the value obtained in step 4 (in TECUs), and f is the frequency of the pseudorange to be corrected (in Hz).

7.5.2 Integration in gLAB

There are two independent implementations of the algorithm, one from ESA and one from Joint Research Centre (JRC), both implemented in C, provided as self-executing programs (Aragon-Angel et al [2021]). As the ionosphere model is one step in the satellite modelling for the navigation solution computation, the integration of the NeQuick-G model into gLAB was considered and realized in the context of the present PhD.

The ESA cannot be integrated into gLAB due to its licence (ESA [2023]), which restricted its usage to ESA member states. This restriction is incompatible with gLAB, as the software is distributed worldwide together with its source code. In contrast, the JRC implementation uses the European Union Public License (EUPL) (EC [2017]), which is compatible with gLAB as it has no geographic restriction.

In Aragon-Angel et al [2021], both NeQuick-G implementations were integrated into gLAB and compared. In particular, we assessed the correction computed by each independent version of the NeQuick-G model. The testing methodology (in terms of the correction computed by each implementation), checks the implementation in gLAB and assesses the algorithm against Klobuchar, in both the range domain and the position domain. To perform this comparison, the dataset comprised 12 days of 2019 (the first day of each month in 2019, so all sets of internal coefficients within NeQuick-G are used) with 597 permanent receivers distributed worldwide.

The comparison of the STEC between ESA and JRC output value was performed with more than 2125 million samples. As expected, the vast majority of samples (99.998%) were the same. Samples with different values were due to two reasons, the first being to rounding errors in the fifth (and last) decimal (which can be disregarded) and the second due to a difference in the implementation in satellite elevation below 0.5° , where ESA implementation returns a value of zero, but JRC implementation does not return any value (i.e. no correction available). As the minimum elevation mask is usually 5° , it is not critical for navigation purposes. Following the crossed validation of the models, it was decided to integrate the JRC version into gLAB in the version 5.5.0 released in 13th November 2020.

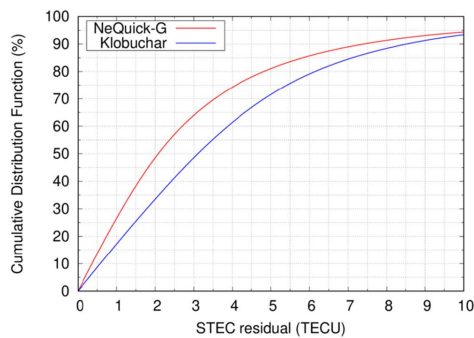
Regarding the implementation in gLAB, as NeQuick-G requires data from 12 Consultative Committee on International Radio (CCIR) files and one file for the MODIP (EC [2016]). The model was able to be implemented by having these files as external libraries (i.e. gLAB needs to read the files when the program starts) or by embedding the data in the source code. Both methods were tested, and the latter method was selected, as it reduced the processing time by 2.25%. Furthermore, when comparing the embedded implementation against the ESA

implementation, the JRC model was 13.87% faster (Aragon-Angel et al [2021]).

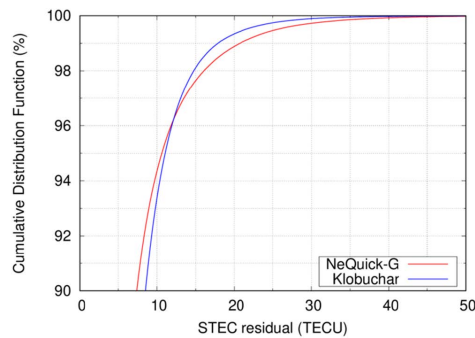
For the accuracy assessment in the range domain, the STEC were compared against reference values computed from the Geometry Free (GF) combination of carrier phase measurements after performing Integer Ambiguity Resolution (IAR) (Shi and Gao [2014]), while in the position domain the navigation solution was compared against precise station coordinates. The navigation solution was computed using GPS C1P and CCL measurements (Mannucci et al [1998]; refer also to section 7.4).

The results of these comparisons are shown in Fig. 7.12. Figures 7.12a and 7.12b show the Cumulative Distribution Function (CDF) (in %) of both NeQuick-G and Klobuchar of the STEC residual against the reference value (in TECU). Figures 7.12c and 7.12d show the same CDF (in %) but in the position domain (with the 3D positioning error). The following conclusions can be drawn:

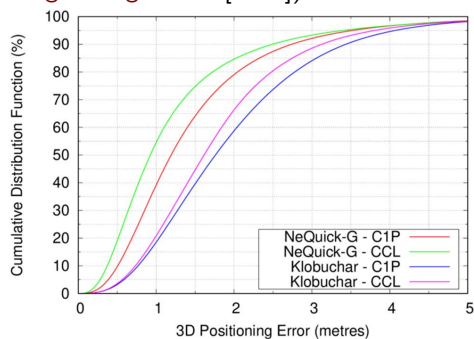
- In the range domain, NeQuick-G performs better than Klobuchar up to an error of 12.15 TECUs (percentile 96.23) or 1.97 metres in the L1 frequency. The RMS of the STEC error for NeQuick is 5.29 TECUs and 5.54 TECUs for Klobuchar.
- In the position domain, NeQuick-G performs better than Klobuchar up to an error of 5.76 metres (percentile 99.18) using C1P and up to an error of 4.98 metres (percentile 98.29). The RMS using CCL is 1.79 m for NeQuick-G and 2.21 for Klobuchar, while with C1P the RMS is 1.83 for NeQuick-G and 2.28 for Klobuchar.
- In the position domain, NeQuick-G performs better with CCL up to 4.18 metres (percentile 97.13), while Klobuchar performs better with CCL up to 5.23 metres (percentile 98.60). The worst performance of CCL method in larger percentiles is due to a smaller number of satellites, as it is necessary to perform cycle-slip detection in the CCL process.
- In the position domain, NeQuick-G CCL has a greater improvement than for Klobuchar, showing that in NeQuick-G the pseudorange noise had a greater impact in the navigation solution, or in other words, in Klobuchar the error in the model was at similar levels to the pseudorange noise.
- In both range and position domain, the CDFs of NeQuick-G and Klobuchar end up crossing, proving the consistency of the results.



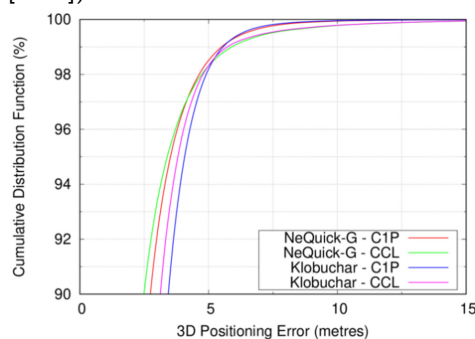
(a) NeQuick-G and Klobuchar STEC comparison CDF up to 10 TECU (from Aragon-Angel et al [2021]).



(b) NeQuick-G and Klobuchar STEC comparison CDF (from Aragon-Angel et al [2021]).



(c) NeQuick-G and Klobuchar 3D positioning comparison CDF up to 5 metres (from Aragon-Angel et al [2021]).



(d) NeQuick-G and Klobuchar 3D positioning comparison CDF (from Aragon-Angel et al [2021]).

Figure 7.12: NeQuick-G and Klobuchar model comparison CDFs (in %) for the first day of each month in 2019 and 597 permanent receivers. The top row shows the CDF comparison in the range domain against reference STEC, while the bottom row shows the CDF comparison is in the position domain against precise reference station coordinates (using GPS C1P and CCL measurements). The left-hand column shows a zoom of the CDFs, while the right-hand column shows the full CDF.

7.6 “Prefit” outlier detection

In order to perform accurate GNSS navigation, one step is to detect anomalous samples and discard them. These non-nominal code and carrier-phase measurements can be originated at both ends of the processing chain, at the satellite transmitter or at the user receiver. For the satellite errors, the control segment has the ability to make the satellite unusable by means of the healthy flag broadcast in the navigation message (updated in GPS every two hours). For the detection of receiver errors, one algorithm known for this purpose and used in SoL navigation is Receiver Autonomous Integrity Monitoring (RAIM) (Walter, T. and Enge, P. [1995]), which has a high degree of complexity and is currently not implemented in gLAB.

gLAB already has some algorithms that filter anomalous carrier-phase measurements, which are the cycle-slip detectors (note that most of the cycle-slip detectors use both pseudoranges and carrier-phase measurements). With these detectors, if there is a discontinuity in the measurements (for any reason), the sample will be flagged as a cycle-slip and the satellite will be skipped until the cycle-slip detector buffer is filled again (see example on Fig. 7.15 in section 7.7). However, anomalous samples which do not have jumps will not be detected (as in Fig. 7.14 in section 7.7); therefore cycle-slip detectors can only be considered as partial protection. In the case of SPP processing, no cycle-slip detector is used (as only pseudoranges are used); therefore it has no protection against anomalous samples unless a dedicated outlier detector is added.

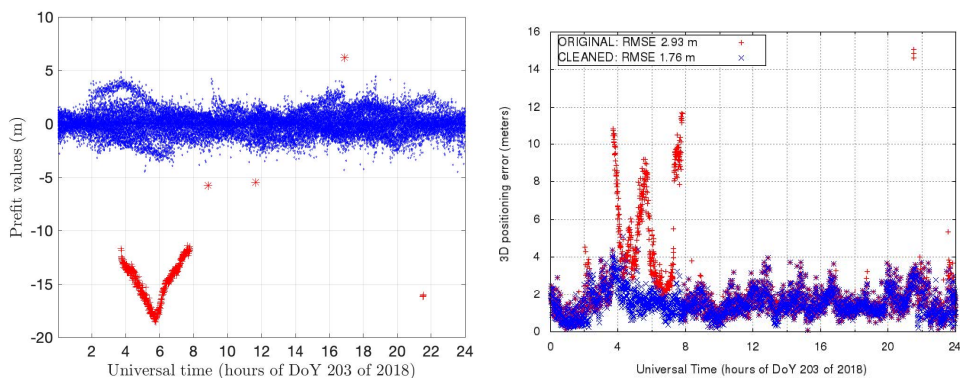
To detect these outliers, several methods use the “prefits” values (such as RAIM). According to Sanz et al [2013a], “prefits” are the values obtained after removing the geometric range and modelling terms from the pseudorange (or carrier phases), which only contain the receiver clock, noise and modelling errors (and also the ambiguity in the case of carrier phases). As a receiver clock is common to all satellites, the pseudorange “prefits” from all satellites will be centred on a common value (the receiver clock) and “prefits” from each satellite can differ up to the metre level (due to multipath and modelling errors). However, the similar values in the pseudorange “prefits” for all satellites will only be true if the a priori position used for modelling is accurate enough.

In gLAB, the method for detecting anomalous samples (hereinafter called outliers) is by comparing the “prefit” value of each satellite to the median of all satellite “prefits” (the algorithm performs such comparison epoch by epoch independently). If the absolute difference between the “prefit” and the “prefit” median is larger than a threshold (by default 40 metres), the satellite is discarded. This process is repeated for all satellites, and if more than half of

the satellites are discarded, then the algorithm considers that the detection is not reliable and does not discard any satellite, which occurs when the a priori position is far from the real position. This algorithm is simple to implement, but has the drawback that it cannot filter out small outliers (e.g. outliers less than 10 metres).

In this context, during the PhD and in collaboration with JRC, some statistical functions were developed and analysed to detect outliers. One dataset used is from 4th November 2018 using station “kir0” (from IGS network) with a 30s data rate and GPS C1C measurement. With this dataset, the Least Median of Squares (LMS) estimator (Rousseeuw [1984], Torti et al [2012]) is used to estimate “predit” outliers, using 24 hours of samples. This algorithm is included in the Flexible Statistics and Data Analysis (FDSA) toolbox (Riani et al [2012]) in MATrix LABoratory (MATLAB) program (MathWorks [2023]).

In Fig. 7.13a the prefit dataset and detected outliers are shown in red. gLAB and LMS both detect outliers greater than 10 metres, but LMS also detected outliers slightly higher than five metres. With this detection, the navigation solution improvement is shown in Fig. 7.13b.



(a) Prefits on 4th November 2018 (from Alonso et al [2021]).

(b) 3D positioning error on 4th November 2018 with and without prefit outlier detection (from Alonso et al [2021]).

Figure 7.13: Left-hand image contains the prefits on 4th November 2018 where outliers are shown in red (outliers with ‘+’ symbols were detected by gLAB algorithm and LMS, while outliers with ‘*’ were only detected by LMS), while the right-hand image contains the 3D positioning error with and without prefit outlier detection.

In the published versions of gLAB, the “predit” outlier detector is disabled by default on all processing modes, as for GPS processing the presence of large outliers is less common. However, based on the thesis author experience, the presence of outliers are common when using all constellations, but they are

usually filtered by the cycle-slip detectors. To characterize the frequency and typical constellations with outliers, a devoted follow-on study is required.

Due to this increased presence of outliers, the “prefit” outlier detector in the multi-constellation version of gLAB has been enabled by default on SPP processing mode to avoid large solution errors in SPP mode, whilst being an indication that robust outlier detectors (as in [Alonso et al \[2021\]](#)) will be more necessary for multi-constellation processing.

7.7 GPS signal in space anomalies

7.7.1 Introduction

Electronic hardware, during its expected lifetime, can potentially have malfunctions. GNSS satellites are no exception, and in the rare cases when a malfunction occurs, it is translated into unexpected behaviour in the SIS. Due to several systems rely on GPS (especially for SoL systems such as SBAS or GBAS), it is essential to be able to detect and characterize these anomalous events. For instance, in [Sanz et al \[2018\]](#), an study of GPS SIS for more than 11 years is performed where the thesis author has contributed (this study was later expanded in [Alonso \[2023\]](#)).

Following [Sanz et al \[2018\]](#), this proceeding is focused on a new detection methodology using code and carrier-phase measurements from a worldwide distributed network of permanent receivers (IGS). The details on the algorithm is out of scope of this thesis, but it involves the geodetic detrending ([Juan et al \[2022\]](#)) of the code and carrier phase measurements, which were performed with gLAB.

7.7.2 Methodology and data set

In [Sanz et al \[2018\]](#) it is proposed a new detection methodology targeted to detect very short duration events (with time scales of few seconds) using pseudorange and carrier phase measurements available from a set of IGS receivers worldwide distributed, which are otherwise undetectable using only IGS products (precise orbits and clocks). With this detection methodology, it is assessed 11 years of data since 2004, focusing on the following list of anomalies: pseudorange step, ramp and acceleration errors, code-carrier incoherency, hardware bias jump and drift.

Each of the above SIS anomalies are assessed by analysing code and carrier phase

measurements for a minimum set of receivers worldwide distributed. Combining these measurements allows to remove the contribution of ionosphere, geometry or receiver clocks, as these terms could lead to false event detections. Events that were detected on multiple receivers simultaneously were recorded, where as events on a single receiver were discarded as receiver artefacts.

The data set used in this study involved 11 years of dual frequency GPS RINEX observation files at 1Hz from IGS network ([International GNSS Service Data \[2023\]](#)), available in RINEX-2 format ([Werner and Estey \[2007a\]](#)). Receivers and its measurements were selected with the following criteria ([Sanz et al \[2018\]](#)):

- A satellite must be seen by at least three stations at any time.
- Elevation mask is 15° for carrier measurements and 20° for pseudorange measurements.
- Receiver's type should be as much homogeneous as possible (i.e. with similar figures of code and carrier phase noise) and the GPS receivers have to provide C1 (L1 civil signal), P1 (L1 military signal) and P2 (L2 military signal) codes. This condition reduces the number of possible sites, as not all receivers provide P1 code.

The data preprocessing consisted in detecting cycle-slips using the GF combination and MW combination (the same implemented in gLAB, as described in section 4.2), removing the geometry (as the receiver position is known) and satellite clocks. Then, satellite clock correction is obtained from IGS precise products, therefore, the anomaly detections cannot be performed if the IGS products were not available or there was a data gap equal or greater than ± 1 hour around the interpolation epoch.

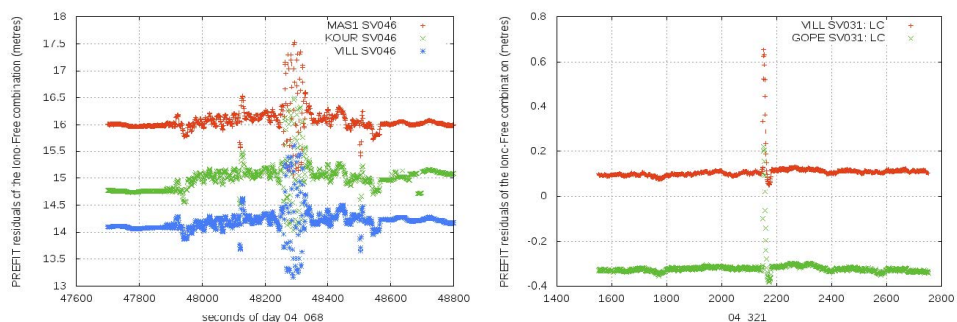
7.7.3 Results

From the 11 years analysed period, a total of 124 days having satellite clock events were detected, involving 23 different satellites. Then, 72 of the days with events were associated to SVN 063 (there were no multiple satellite events in a single day). The analysis of the detected events using this carrier based monitor (carrier jump, drift and acceleration) proved that of them are related with satellite clock anomalies in the GPS satellites. These anomalies can be classified in three different groups: clock oscillations, clock jumps and clock pulses, or combinations between them. Some relevant events are depicted next.

An example of three events were detected with these monitors are given next: a satellite clock oscillation, a satellite clock pulse and a satellite clock jump. The

first two events are very similar, they are both an oscillation of the clock, but the clock pulse (second event) is a very short event. Figure 7.14a depicts the clock oscillation detected on GPS SVN 046 on 8th March 2014 at 13:21:59 is shown, while in Fig. 7.14b the clock pulse detected on GPS SVN 031 on 16th November 2004 at 00:35:52 is shown. Both figures depict “prefit” residuals from the carrier phase IF combination. In particular, Fig. 7.14a, correspond to the permanent receivers “mas1” (located in the Canary Islands), “kour” (in the French Guiana) and “vill” (Madrid, Spain), while in Fig. 7.14b they correspond to receivers “vill” and “gope” (Ondrejov, Czechia) permanent receivers. In both events, even though the stations are far away from each other, they present the same pattern in the same satellite and epoch, confirming the anomaly in the satellite.

In Fig. 7.14b, the magnitude of the pulse is around 0.5 metres, which is below the code pseudorange noise level (metre level), while in Fig. 7.14a the oscillation is around 1.5 metres, which is slightly above the code pseudorange noise level, but it could be masked by other errors (e.g. troposphere residual). In both cases, if navigation was performed using only pseudoranges (like in SPP), these events would pass mostly unnoticed (due to the pseudorange noise) and would not be detected by the “prefit” outlier detectors shown in section 7.6, but these events could occur in a much higher magnitude. In any case, if the carrier phases were used for code smoothing or in precise navigation, these events would deteriorate the navigation solution position.



(a) Clock oscillation on GPS SVN 046 on 8th March 2014 at 13:21:59 (from Sanz et al [2018]). (b) Clock pulse on GPS SVN 031 on 16th November 2004 at 00:35:52 (from Sanz et al [2018]).

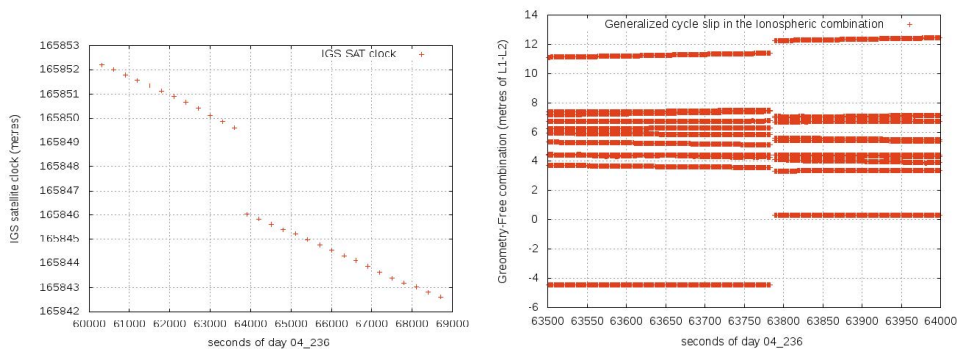
Figure 7.14: “Prefit” residuals from the carrier phase IF combination during the satellite clock events for different permanent receivers. The left-hand plot shows the aforementioned “prefits” from receivers “mas1”, “kour” and “vill”, while the right-hand plots are from “vill” and “gope” receivers.

The third event shown in Sanz et al [2018] is a clock jump, detected on GPS SVN 031 on 16th November 2004 at 00:35:52, lasting 11 seconds (the clock

jumps during this time period until it stabilizes again). Fig. 7.15a shows the IGS clock precise determination (accurate at the level of a few centimetres) at a 300 second rate, where the clock correction jumps around 3.5 metres (11.6 nanoseconds), well above the pseudorange noise level. The jump in the carrier-phase measurements is detected as a cycle-slip by the cycle-slip detectors in all stations, as shown in Fig. 7.15b.

In this case, if the user is utilizing cycle-slip detectors, they would discard the satellite when the clock jumps start until the clock stabilizes again (and the cycle-slip buffers are filled again). Once stabilized, if the clock correction is not updated, the jump will appear in the satellite “pfit”. In the case of using IGS clock correction, this correction is capable of following the jump, but as they are at a 300 second rate, the clock is linearly interpolated between samples. Therefore, until the interpolation with the next clock correction sample with the jump incorporated, there will be an error in the clock correction. If the user utilizes broadcast ephemerides, as they are estimated in advance, this jump will not be updated in the clock correction until a new navigation message is broadcast (typically every two hours) or until the satellite is flagged as unhealthy. The cases in which there is a satellite anomaly but the health flag is set as healthy are known as “feared events”.

In any of the cases presented above, a “pfit” outlier detector (as described in section 7.6) can serve as protection against these anomalies, provided that the magnitude of the anomaly is larger than the detection threshold.



(a) IGS precise clock jump on GPS SVN 030 on 23rd August 2004 at 17:43:04 (from Sanz et al [2018]). (b) Cycle-slip in all stations on GPS SVN 030 on 23rd August 2004 at 17:43:04 (from Sanz et al [2018]).

Figure 7.15: Clock jump for GPS SVN 030 on 23rd August 2004 at 17:43:04. Th. The left-hand figure shows the IGS precise clock determination (at 300s rate) jump, while the right-hand figure show the cycle-slip produced in the GF combination of L1 and L2 carriers (cancelling the non-frequency dependent terms, such as the clocks, geometry and troposphere delay) in IGS stations with the satellite in view during the event.

Impact of gLAB tool

As gLAB is a free and open source tool, with no requirement of any type of registration for downloading it, and it has no built in telemetry (and will never have any), it is impossible to make a complete assessment of the impact of the tool. Thus, the assessment is limited to the following metrics:

- **Number of downloads:** This metric is quite representative and also allows the user's country to be determined from its Internet Protocol (IP) address, but it has the following limitations:
 - Some users may download the tool multiple times.
 - A user may download the tool and then distribute it internally to other users (usual in public institutions or large companies).
 - Some downloads are performed by automatic web crawlers, which it is not always possible to identify by their user agent.
- **User emails:** Some of the users contact gAGE for troubleshooting or support. Only a limited number of the total user base may end up contacting gAGE, but from these contacts, the thesis authors knows that gLAB is being used in universities, public institutions, private companies (small and large) and private users. All of these users are scattered around the world in all five continents (which is the expected result, as GNSS is a worldwide constellation). No information of the users can be disclosed, as user privacy must be respected.
- **Citations:** Citations of gLAB papers is a very limited metric, as many of the gLAB users may not end up writing any article or paper, especially students and professionals working in private companies. The former are studying (so they do not publish anything), and the latter use the tool as an algorithm implementation reference or for cross-authentication of their internal tools.

- **GNSS courses:** gLAB is used for the laboratory sessions in the gAGE/UPC GNSS courses (the thesis author has been responsible for preparing the course installable files). These courses have had a significant impact, for many reasons; it is one of the few cases where a complete independent software solution compatible with all OSs and adjusted for the laboratory sessions is provided in a course.
- **Time of downloads:** The new versions of each gLAB tool were usually released on Fridays at the end of the evening. Even though gLAB has no automatic check for updates (i.e. the user needs to manually check for updates through the GUI option or in gAGE website), some users started downloading the tool on the Friday itself, or over the following weekend, which indicates that users are actively looking for new gLAB releases.

To quantify the impact of the tool, the number of downloads per gLAB version (from the 30th April 2019 to 12th of March 2023) is provided in Fig. 8.1:

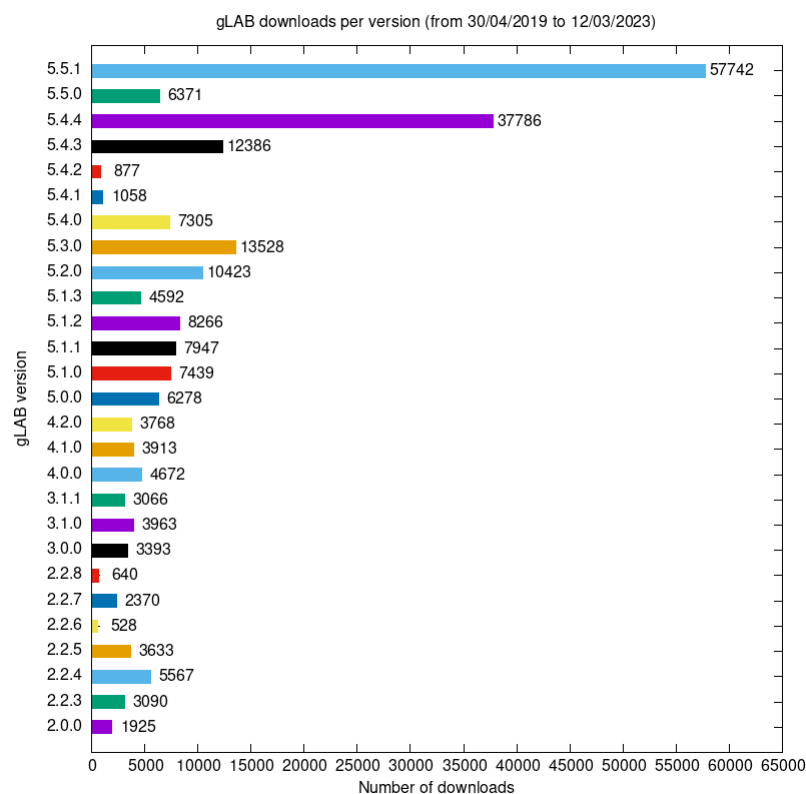


Figure 8.1: gLAB number of downloads per version (from 30th April 2019 to 12th of March 2023).

The period chosen in Fig. 8.1 is because this is the period with available web

server logs. The number of downloads were filtered in order to count local IPs, public IPs for UPC and automatic web crawlers (the filter for automatic web crawlers is based on [Klausmeier, E. \[2021\]](#)). With this filtering, the total number of downloads is 222,526, from 11,658 unique IPs. The number of downloads is smaller for older versions, mainly due to the unavailability of older logs or relatively short lifespan of the version. Version 5.4.4 and 5.5.1 are by far the most downloaded versions, as they are the two latest versions with long time-spans.

In order to dig more on the gLAB downloads, in [Fig. 8.2](#) a breakdown of the number of gLAB downloads per country is provided (from all gLAB versions) in the same period. The geographical information was obtained from [DB-IP \[2023\]](#).

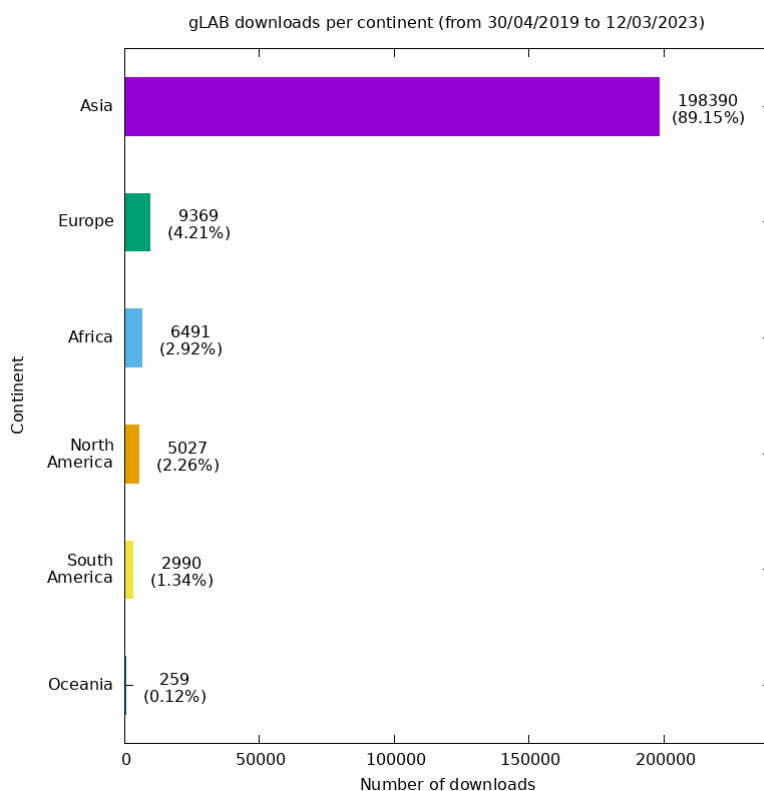


Figure 8.2: gLAB number of downloads per continent (from the 30th April 2019 to 12th of March 2023).

From [Fig. 8.2](#), the continent with the highest number of downloads by far is Asia, followed by Europe, Africa, America and Oceania. To follow the breakdown, the top ten countries with most downloads are shown in [Fig. 8.3](#):

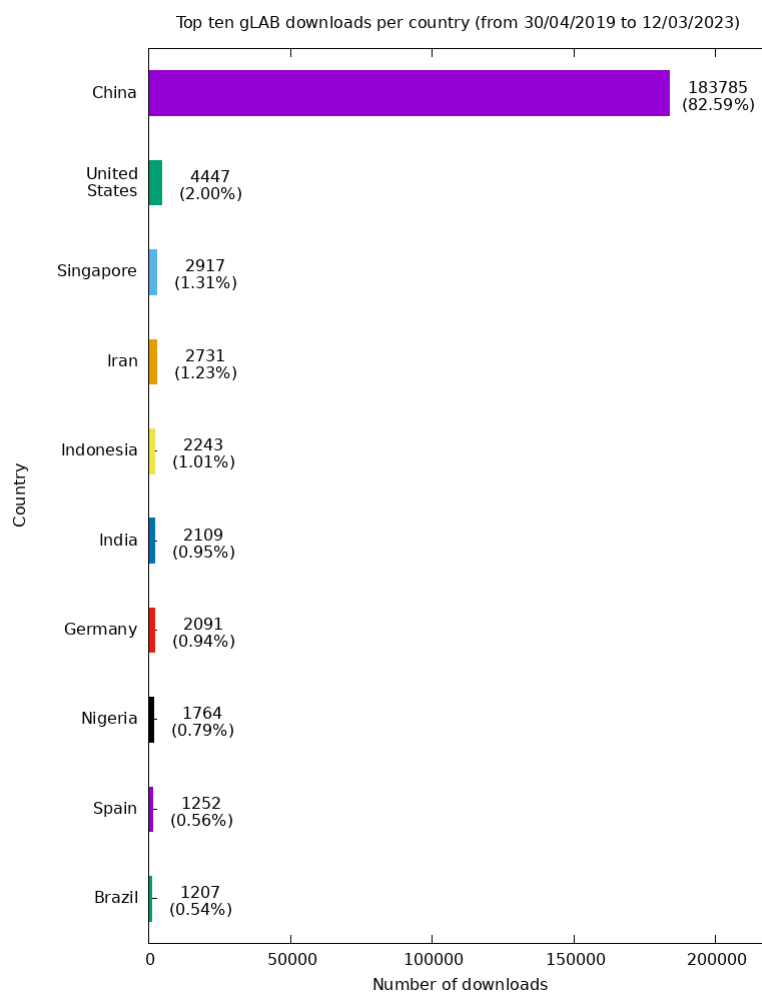


Figure 8.3: Top ten gLAB number of downloads per country (from the 30th April 2019 to 12th of March 2023).

From Fig. 8.3, the country with most downloads is China. From Asia are also Singapore (third position), Iran (fourth position), Indonesia (fifth position), India (sixth position). From North America there is only USA (in second position). From South America there is Brazil (tenth position). From Europe, there is Germany (seventh position) and Spain (ninth position). Finally, from Africa, there is Nigeria (eighth position). A full list of countries (147) where gLAB was downloaded is provided in annex B.

Finally, a list of the GNSS courses (that used gLAB) performed by gAGE/UPC during the duration of the thesis is provided in table 8.1. These courses demonstrate the capability of gLAB for educational uses (as it is the tool for the laboratory sessions).

Course	Institution	Location	Degree	Duration (Hours)
2021-2022	Technical University of Catalonia (UPC)	Terrassa, Spain	Master in Aerospace Engineering	72
2020-2021	Technical University of Catalonia (UPC)	Terrassa, Spain	Master in Aerospace Engineering	72
2017-2018	National Polytechnic Institute (IPN)	Mexico City, Mexico	Master in Telecommunications Engineering Science	54
	National School of Civil Aviation (ENAC)	Toulouse, France	Master in Aerospace Systems - Navigation and Telecommunications	35
	Technical University of Catalonia (UPC)	Castelldefels, Spain	Master in Aerospace Systems Engineering	87

Table 8.1: gAGE/UPC GNSS master courses with gLAB. The duration of the master courses account for both theory and laboratory sessions.

On table 8.2 below are provided the professional trainings where gLAB was used:

Year	Location
2022	Barcelona, Spain
2021	Fortaleza, Brazil
2019	Mollet del Vallès, Barcelona, Spain
2019	Bandung, Indonesia
2018	Barcelona, Spain
2017	Warsaw, Spain
2017	Barcelona, Spain

Table 8.2: gAGE/UPC professional training courses with gLAB. The second row in blue is the course that the thesis author attended and was one of the UPC lecturers.

Conclusion

This thesis has focused on the development of the gAGE/UPC GNSS Laboratory (gLAB), an advanced GNSS processing platform designed to handle new processing requirements and new signals and frequencies introduced by the deployment of new GNSS. The aim of gLAB is not only to meet the demands of the GNSS community but also to be accessible to scientists, professionals, and students.

The first contribution in this topic is the assessment of EGNOS signal. As explained in section 7.1, SBAS usage is being expanded for maritime purposes. The maritime community has different requirements with respect to aviation, the most relevant ones being the different method for continuity computation, the usage of a DOP mask and omitting the usage of SBAS MT10. Aside from assessing the performance of EGNOS for maritime use during a two-year period (requiring the development of a method for multi-day availability maps, detailed in annex C), these three requirements were challenged by comparing the results with the method used in aviation (in continuity computation and MT10 usage) and without a DOP mask. In the case of the continuity and DOP mask, the continuity was computed four times, with a sliding and a fixed window, with and without a DOP mask. Results show that the sliding window provides better results in all cases (between 1.27% and 1.55%), as well as always providing realistic results in the edges. Moreover, during this computation, it was proved that the gLAB method for switching GEOs (which is not to wait any epoch when there is a discontinuity) maximized the continuity. Last, the omission of MT10 in the SBAS corrections barely affects the positioning, but increases protection levels by a factor between 2.6 and 4.7, resulting in unrealistic protection levels. With these results, the maritime authorities are discussing these requirements.

Moreover, a demonstration of a method for detecting SIS errors for both satellite signals and EGNOS messages (as shown in chapter 7.2) has been developed. In this regard, a GMS was built with gLAB, which required computing availability

and continuity maps, positioning error plots, 95 percentile positioning error and protection level plots, Stanford plots, Stanford-ESA plots and WIR maps for many stations distributed around the EGNOS coverage area. It was shown that to be able to detect any error, no local error filtering can be performed, as a priori local errors are indistinguishable from SIS errors, therefore the way to distinguish them is by assessing whether the error occurred at a single station (a local error) or at multiple relatively nearby stations. As expected, no MI was found during the nearly three years duration of the GMS campaign.

The second contribution is the assessment of the ionospheric activity and its impact in navigation, specially in SBAS. On the one hand, in section 7.3, the AATR index is presented, which its key feature is that is computed solely using GNSS measurements (i.e. the same measurements as any GNSS user). The AATR index is a local index and it was correlated with the EGNOS availability. Moreover, the AATR definition was extended to convert the AATR into a global index, and it was demonstrated its correlation with the availability with EGNOS and WAAS. For this reason, the AATR index was adopted by ICAO for joint ionospheric studies in SBAS. On the other hand, in section 7.4 it was demonstrated the importance of correctly assessing the ionospheric effects on navigation. By using CCL measurements, which require a low level of preprocessing, the positioning results became representative of the quality of the ionospheric model used.

The third contribution is on the study of the effect of anomalies in GNSS positioning, both in the receiver and in the SIS. The receiver anomalies were assessed in section 7.6 using gLAB as the main tool to implement a set of detection algorithms. These SIS anomalies are analysed in 7.7, where several short events of different magnitude and types (clock oscillation, pulse and jump) are detected with this new methodology, that requires high level of preprocessing, for which gLAB has been proved to be a very appropriate tool.

The fourth contribution is the gLAB tool itself. As demonstrated in chapter 8, gLAB has a significant impact worldwide, which demonstrates the demand for this kind of tool. gLAB has had the SPP and PPP algorithms upgraded to full multi-constellation (as demonstrated in chapter 4), as well as adding the advanced algorithms uncombined PPP, Fast-PPP and SBAS DFMC (also with full multi-constellation support), as shown in chapter 5. The upgrade for multi-constellation has had several challenges (as shown in chapter 6), being the most important ones the capability to manage all the possible combinations of signals and frequencies in an automatic or manual way (to make it suitable for any kind of user) and the optimization of the source code (detailed in annex A). Furthermore, the capability of the tool for research purposes is demonstrated with the results in chapter 7.

With the distribution of the updated gLAB version, the GNSS community will

have a reference tool for algorithm implementation, comparison of results, education or research, as in this thesis.

Chapter 10

Future Work

During the work on this thesis and in the author's previous two final degree projects (Ibáñez [2014] and Ibáñez [2016]), gLAB has had many upgrades (as shown in this thesis and as in its release notes at gAGE/UPC [2020]), but still there are many more possible improvements or upgrades that might be performed. A list of the most relevant upgrades is provided below:

- Support for RINEX observation and RINEX navigation files version 4.00 (Romero [2021]).
- Support for SINEX BIAS files (Schaer, S. [2018]).
- Add uncombined PPP and Fast-PPP to the GUI.
- Processing of Galileo HAS data (EUSPA [2023d]), so gLAB can become an independent validation or troubleshooting tool.
- IAR to fix carrier phase measurement with Least-squares AMbiguity Decorrelation Adjustment (LAMBDA) method (de Jonge, Paul and Tiberius, C. [1998]).
- RAIM (Walter, T. and Enge, P. [1995]) and Advanced Receiver Autonomous Integrity Monitoring (ARAIM) (Blanch et al [2012]) processing.
- New output messages with the computed sigmas (in all steps of the processing).
- Multi-constellation decoding of RTCM messages.
- Multi-constellation DGNSS processing.
- Real-Time Kinematics (RTK) processing.

- Real-time processing.
- Support for modelling ocean loading.
- Support for modelling polar tides.

Aside from the updates to gLAB itself, the tutorials related to gLAB or which use gLAB will need to be updated with the new functionalities, especially for multi-constellations processing.

In summary, a lot of work has been done, but there is also much more work ahead.

Appendix A

gLAB source code optimizations

In this annex the gLAB source code optimizations are detailed. These optimizations are for the DPC, which is written in C99 (ISO/IEC [1999]). Some of these optimizations are specific to C language, others can be exported to other languages. Apart from the techniques shown below, programmers should always reassess their algorithms (in order to see if it is possible to improve them), functions (some functions may be possible to simply integrate in their parent function) and use code profilers (e.g. GNU's Not Unix (GNU) Profiler, described in GNU [1997]) to check for bottlenecks in the code execution.

A.1 String generation for printing

In gLAB, when an internal integer value was converted to text (for instance from internal constellation type to string), the function responsible for performing it copied the requested string to a buffer. As these strings will never change, the function can be changed to simply return a pointer to the position in a static array with the requested string, thus avoiding the if conditions and copying the string to another buffer. This approach also has the advantage that static variables are initialized at compile time, so no computation time is lost in initialization of variables. A minimal example is shown in code snippets A.1 (where the string is copied) and A.2 (where only the pointer to the array is returned):

```
1 char *gnsstype2gnssstr (enum GNSSystem GNSS) {
2     static char str[10];
3     switch(GNSS) {
4         case GPS:
5         strcpy(str, "GPS");
```

```
6     break;
7     case Galileo:
8         strcpy(str, "Galileo");
9         break;
10    default:
11        strcpy(str, "NAN");
12    }
13    return str;
14 }
```

Listing A.1: Internal value to string copying the string.

```
1 char *gnsstype2gnssstr (enum GNSSystem GNSS) {
2     static char str[2][10]={"GPS", "Galileo"};
3     return str[GNSS];
4 }
```

Listing A.2: Internal value to string using pointers.

A.2 Avoiding multiple recomputations of the same elements

In many cases, it is much easier and faster to program such that different functions in the code compute the same thing, as it allows the use of local variables for the computations and avoids the hassle of using pointers to pass the data from one function to another, but it has the downside of wasting computational time.

gLAB code had this issue, therefore the following improvements were made:

- All possible items which could only be computed once for the entire program were moved (e.g. the generation of the selected measurements text strings for printing only need to be generated once after the measurement selection ends).
- All possible items which have to be computed once each (such as the string with the current epoch) were moved at the start of the epoch.
- Functions “getstr” (which extracts a portion of text in a given string) and “trim” (which removes leading and trailing white spaces from a string), were computing the string length (passed by parameter) inside them, but this was not necessary, as the string length had been calculated when a line was read. Therefore, the string length was passed by parameter, avoiding the computation of the string length in each function call.

A.3 Precise orbit and clocks management

Precise orbits and clock products (SP3 and CLK files) from any provider (IGS, CODE) have small data gaps in most of their daily files. These data gaps are small (usually only affecting very few satellites in very few epochs).

To overcome these data gaps, gLAB saved the data read (for each satellite) in a continuous array, in the order in which the samples were read. If a data gap occurred, no data was written in the array until the next sample was read (so there were no blank slots in the array). This method used the minimum amount of memory, but in order to search for a particular sample, it was necessary to loop through the array until the desired sample was found (or not), as the samples were not always evenly spaced in time, due to the data gaps. This search method gets slower the larger the array is (which is very high when using high rate CLK files), and gLAB had to perform this search several times (for computing position, speed and clock correction) per each epoch and satellite.

The new algorithm saves the data in an array where each position in the array is equally separated in time from each other. If there are data gaps, these positions are filled with zeros. In this way, looking for a specific sample always takes the same amount of computation time, as the position index in the array is computed with equation A.1:

$$Position\ index = \frac{Current\ epoch - Initial\ Epoch}{Time\ between\ samples} \quad (A.1)$$

Where:

- **Current epoch:** The current epoch (in GPS time).
- **Initial Epoch:** The epoch of the first sample in the array (in GPS time).
- **Position index:** The index with the position in the array.

If the position in the array is where a data gap occurred (its value will be zero) and interpolation is disabled, no value is computed. When interpolation is enabled, gLAB will search for the rest of the samples to be used for interpolation by increasing or decreasing the index by 1. If any data gap is found, it will be skipped. As gLAB has a (configurable) maximum of data gaps allowed (by default four), and the number of additional samples to be searched is limited by this maximum. If the maximum data gap check is disabled, then the sample in the computed index (or the previous one) must exist. If no interpolation is performed, then the sample in the computed index must exist.

This optimization was implemented in gLAB version 5.4.0.

A.4 Convert if chains to switch statements

If chains have the advantage that they work with any type of condition and with multiple conditions in a single statement, but require all statements to be processed until the desired one is reached.

Switch statements only need to check one condition in order to jump to the desired element inside the *switch* and are also independent of the number of elements inside it, as it uses an internal look-up table. The downsides are that it only works with integers and that each case number must be set at compile time.

As *switch* statements are faster than *if* chains when there are more than two conditions, in gLAB, the *if* chains have been changed to *switch* statements whenever possible. A minimal example of this change is shown in code snippets A.3 and A.4:

```
1 if (GNSS==GPS) {
2     /*GPS instructions*/
3 } else if (GNSS==Galileo) {
4     /*Galileo instructions*/
5 } else if (GNSS==GLONASS) {
6     /*GLONASS instructions*/
7 }
```

Listing A.3: *If* chain example.

```
1 switch(GNSS) {
2     case GPS:
3         /*GPS instructions*/
4         break;
5     case Galileo:
6         /*Galileo instructions*/
7         break;
8     case GLONASS:
9         /*GLONASS instructions*/
10        break;
11    default:
12        break;
13 }
```

Listing A.4: *Switch* statement.

Another limitation of the *switch* statements is that they do not accept a range of values. GNU Compiler Collection (GCC) (GNU [2023b]), Clang (The LLVM Project [2023]) and Intel oneAPI DPC++/C++ compilers (Intel [2023]) support *switch* statements with ranges (as an unofficial C extension). Other compilers, such as Visual Studio C/C++ compiler (Microsoft [2023]), do not support this extension. Therefore, in gLAB, the code is written twice (with *if* chains and *switch* statements) but between an exclusive preprocessor directive, in order to select only one of them at compile time. By default, the *switch* statement version will only be selected if the flag "RANGECASE" is provided to the compiler and the GCC flag or Clang flag or Intel oneAPI DPC++/C++ flag is found. In code snippet A.5 it is shown the preprocessor commands to decide which code to use, and on code snippet A.6 a minimal example of code (with both methods) is shown:

```

1 #if (defined RANGECASE) && ((defined __GNUC__) || (defined
  __clang__) || (defined __INTEL_COMPILER) || (defined
  __INTEL_LLVM_COMPILER))
2     #define RANGECASEENABLED
3 #endif

```

Listing A.5: Preprocessor commands for detecting “RANGECASE” flag and compiler flags. If both are found, the flag “RANGECASEENABLED” is enabled.

```

1 #if defined RANGECASEENABLED
2 /*Code using switch range extension*/
3 switch(meas) {
4     case G1A ... S1N: case I1: case X1: case G1:
5         return 1;
6         break;
7     case G2A ... S2N: case I2: case X2: case G2:
8         return 2;
9         break;
10    default:
11        return -1;
12        break;
13 }
14 return -1;
15
16 #else
17 /*Code with if chain*/
18 if (meas <= S1N || meas==I1 || meas==X1 || meas==G1 ) return 1;
19 else if (meas <= S2N || meas==I2 || meas==X2 || meas==G2 )
20     return 2;
21 else return -1;
22 #endif

```

Listing A.6: Code in both *if* chains and *switch* statements. The flag “RANGECASEENABLED” will make the preprocessor select only one of the two versions.

A.5 Join multiple if conditions in a single switch statement

When there are multiple nested *if* conditions both of which check integer variables, it may be possible to join them in a single *switch* by multiplying one of the variables by the maximum order of magnitude plus one of the other variable, sum it to another variable and use the result to the *switch* statement.

For instance, when gLAB computes satellites orbits, it has to check if it is a BDS satellite in geostationary orbit (which are PRN 1, 2, 3, 4, 5, 59, 60, 61, 62 and 63). Using *if* conditions, it is necessary to check for the constellation, and if the PRN number is equal or less than five or equal or greater than 59. As PRN

is always smaller than 100 and the constellation variable is of type *enumerator* (which is an integer), the three conditions can be merged by multiplying the constellation by 100 and adding the PRN number. In gLAB, BDS constellation has value 4, so in the switch we would need to check for values 400, 401, 402, 403, 404, 405, 459, 460, 461, 462 and 463. The code with *if* conditions is shown in code snippet A.7 and the code with *switch* statement in code snippet A.8:

```

1 if (GNSS==BDS) {
2   if (PRN<=5 || PRN>=59) {
3     /*BeiDou GEO satellite*/
4   } else {
5     /*Other BeiDou
6      satellites. Shares the
7      same code as non BeiDou
8      satellites*/
9   }
10 } else {
11   /*Other non BeiDou
12    satellites*/
13 }

```

Listing A.7: Nested *if* conditions with integers (constellation and PRN). The constellation variable (“GNSS”) is an *enumerator*.

```

1 switch(GNSS*100+PRN) {
2   case BDS*100+1:
3   case BDS*100+2:
4   case BDS*100+3:
5   case BDS*100+4:
6   case BDS*100+5:
7   case BDS*100+59:
8   case BDS*100+60:
9   case BDS*100+61:
10  case BDS*100+62:
11  case BDS*100+63:
12    /*BeiDou GEO satellite*/
13    break;
14  default:
15    /*Other satellites*/
16    break;
17 }

```

Listing A.8: *Switch* statement with merged conditions.

Another case can be that there are multiple *if* conditions with an integer compared to another variable integer (in previous cases the values compared with were constants). It is possible to merge the conditions by creating an array which saves the expected type (code in an integer format). For instance, several *if* conditions check if a value is of “type A” by checking whether its value is greater than “minValueA” and smaller than “maxValueA” or if it is of “type A” by checking whether its value is greater than “minValueB” and smaller than “maxValueB”. The array to be created will contain from its position “minValueA” plus one to “maxValueA” minus one the value 0, and from “minValueB” plus one to “maxValueB” minus one the value 1. The *switch* statement will then check the position of the array indicated by the value. An example with *if* conditions is shown in code snippet A.9, with the *switch* statement version in A.10:

```

1 if (i>minValueA && i<
2   maxValueA) {
3   /*i has a value of type
4     A*/
5 } else if (i>minValueB && i<
6   maxValueB) {
7   /*i has a value of type B
8     */
9 }

```

Listing A.9: *If* conditions with variable integers.

```

1 for(j=0;j<maxSizeArray;j++)
2 {
3   /*Initialize the array to
4     an invalid value*/
5   array[j]=-1;
6 }
7 for(j=minValueA+1;j<
8   maxValueA;j++) {
9   /*Set the positions for
10    Value A*/
11  array[j]=0;
12 }
13 for(j=minValueB+1;j<
14   maxValueB;j++) {
15  /*Set the positions for
16    Value B*/
17  array[j]=1;
18 }
19 switch(array[i]) {
20   case 0:
21     /*i has a value of type
22     A*/
23     break;
24   case 1:
25     /*i has a value of type
26     B*/
27     break;
28   default:
29     break;
30 }

```

Listing A.10: Switch statement using array to merge conditions. The for loops may be integrated in other existing loops for further optimization.

A.6 Memory allocation

Dynamic memory allocation requires the memory to be first allocated and later freed. The easy and fast approach is to allocate and free the memory inside the function where it is used (in each function call), as it reduces the chances of memory related errors (which are the most difficult to track down and fix). This is not the optimal way, so three approaches were used to optimize the memory allocation:

- **Allocate memory by chunks:** In the case of the statistical summary,

each epoch the buffer was increased by one to make space for the new data. This was changed to allocate in chunks of 86400 slots (one day of data), and more memory was allocated if the buffer became full. This reduced the number of memory allocations from one every epoch to one per day processed (or even less if data rate was less than 1 Hz). An example of this optimization can be found in gLAB source code function "SummaryDataUpdate".

- **Allocate memory as needed:** In some cases, the size of allocated memory varies per epoch. For instance, when selecting the GEO with the best protection levels, it is necessary to save all the data computed with that GEO. As the number of GEO used will vary (since it may not be possible to compute the navigation solution) and the size of the data is large, gLAB will allocate memory required in the first epoch. In the following epochs, if more memory is required (not necessarily in all following epochs), memory will be allocated, but it will not be freed until gLAB terminates. Therefore, memory allocation only occurs when additional memory is required (which will be in few epochs) and freed only once. An example of this optimization can be found in gLAB source code function "SaveDataforGEOselection".
- **Usage of static variables:** If the allocated memory is only used in current function and the worst case of memory usage is known at compile time (and is not too large), the memory allocation can be completely substituted by a static variable with the size of the worst case. Static variables are saved in the "static" memory region, which has no limit, so large variables can be used) but its size is fixed at compile time. An example of this optimization can be found in gLAB source code function "cholinv'opt".

A.7 Compiler flags

Compilers have a lot of parameters to change the generation of compiled code. The best known for optimization is "-O3", which is a shortcut for enabling many optimization procedures in the compiler. Aside from this flag, gLAB uses these two flags at compile time:

- **-fno-stack-protector:** This flag is for disabling the addition of code to check for buffer overflows in the stack every time a function is exited (by default it is enabled). As gLAB does not interact with the user during its execution, this check is unnecessary, as gLAB has safeguards against this attack when parsing parameters or reading the input files. This compilation flag was added in version 5.4.4.

- **-flto**: This flag enables Link Time Optimization (LTO). When the compiler compiles multiple files, each file is considered a separate entity, named as a translation unit. Each translation unit is compiled and optimized separately, and then linked to produce the final executable. When LTO is enabled, each file is compiled separately with additional data, which is used later by the linker to optimize all the files as they were a single translation unit (as though all code had been provided in a single source file).

Furthermore, if the user only needs to compile the program for a specific CPU architecture, the flag “-march=native” makes the compiler compile the code for the architecture where the compilation is executed. This flag is not used by gLAB as it needs to run in a wide variety of CPUs.

Note that it is not meant for there to be a comprehensive list of safe flags to be used for the compiler. However, in other applications, if some accuracy can be spared and standard compliance is not a requirement, some other potentially dangerous flags can be enabled, such as “-ffast-math”.

A.8 Line reading optimization

With the usage of multi-constellation files, the amount of data to be read by gLAB is huge (even if it is not used). For instance, multi-constellation RINEX observation files at 1 Hz can easily be between 600 MB and 1 GB. Therefore, being able to read lines from files quickly is now critical.

In version 5.5.1 and older, the function “getL” to read line from a stream is shown in code snippet A.11:

```
1 int getL (char *lineptr, int *n, FILE *stream) {
2     int rd;
3
4     *n = 0;
5     while ((rd=fgetc(stream))!=EOF) {
6         if (rd==13 || rd=='\0' || rd=='\n') { /*Character 13 is
7             ^M (for Windows)*/
8             if(rd=='\r') {
9                 lineptr[*n] = '\n';
10                (*n)++;
11                lineptr[*n]='\0';
12            } else {
13                lineptr[*n]=rd;
14                if (rd!='\0') {
15                    (*n)++;
16                    lineptr[*n]='\0';
17                }
18            }
19        }
20    }
```

```

17     }
18     break;
19   } else {
20     lineptr[*n] = rd;
21     (*n)++;
22     if (*n >= MAX_INPUT_LINE - 3) return -1; /*Line too long
, quit reading to avoid buffer overflow*/
23   }
24 }
25
26 if (rd == 13) fgetc(stream);
27 if (rd == EOF && (*n) == 0) return -1;
28 else return (*n);
29 }

```

Listing A.11: gLAB 5.5.1 function for reading a line from a data stream.

The main problem in code snippet A.11 is that every time a byte is read, it is checked first to determine whether it is a End Of File (EOF) (to end the *while* loop), which will only occur at the end of the file. Then it is checked to determine whether the byte is a carriage return, a newline or a *NULL* in order to check whether the end of the line has been found. Taking into account that gLAB reads hundreds of millions of bytes, performing four checks per read byte is very slow. The code was optimized as shown in code snippet A.12:

```

1 int getL (char *lineptr, int *n, FILE *stream) {
2   register int rd;
3   /*The "register" keyword is to indicate the compiler to try
to save this variable in a CPU
4   register instead of in memory, so it has faster access times
*/
5   *n = 0;
6   do {
7     rd = fgetc_function(stream);
8     switch (rd) {
9       case '\r':
10        lineptr[*n] = '\n';
11        lineptr[(*)+1] = '\0';
12        fgetc_function(stream);
13        return (*n);
14        break;
15       case '\n':
16        lineptr[*n] = rd;
17        lineptr[(*)+1] = '\0';
18        return (*n);
19        break;
20       case '\0':
21        lineptr[*n] = rd;
22        return (*n);
23        break;
24       case EOF:
25        if ((*n) == 0) return -1;
26        else return (*n);

```



```
27     break;
28     default:
29         lineptr[*n] = rd;
30         (*n)++;
31         if ((*n) >= MAX_INPUT_LINE - 3) return -1; /*Line too long,
32         quit reading to avoid buffer overflow*/
33         break;
34     }
35 } while (1);
36 return -1;
37 }
```

Listing A.12: gLAB 6.0.0 function for reading a line from a data stream.

In code snippet [A.12](#) there are the following changes:

- The *while* loop is an infinite loop (condition “1” is always true), therefore there is no need to check for any conditions during the loop execution.
- The *if* conditions have been changed to a single *switch* statement (therefore a check is carried out one per byte instead of four per byte).
- The read “fgetc” function is changed to “fgetc” function”. The latter is not a function, but a string to be replaced by the preprocessor with the required system function name to be used. In Windows and Linux, there is the “non locking” version of “fgetc” function (with a different name in each OS). “Non locking” means that every time the function enters and tries to read the file stream, it will not execute the code to lock the file (the lock is for preventing multiple threads of the same process to read the same file stream concurrently in order to avoid race conditions). As gLAB always reads in single-thread mode (even if multi-thread is enabled), there is no need to lock the file. In the case of Mac OS, the “Non-locking” of “fgetc” function is not available, so it will use the standard ‘fgetc’ function. The usage of the “non-locking” version of “fgetc” will only be enabled if the “READNOLOCK” is provided to the compiler and it is not a Mac OS. The preprocessor commands for setting this behaviour are shown in code snippet [A.13](#).
- The usage of “register” keyword, which indicates to the processor to use this variable only on CPU registers and not in memory, for faster access time and skipping copying its value to Random Access Memory (RAM). The compiler may not always be able to achieve this goal, depending on the complexity of the code (in that case the behaviour will be like a normal variable).

```

1 #if (defined READNOLOCK) && (!defined __APPLE__)
2   #if defined __WIN32__ || defined _WIN32
3     #define fgetc_function _fgetc_nolock
4     #define feof_function feof
5     #define ferror_function ferror
6     #define fflush_function fflush
7   #else
8     #define fgetc_function fgetc_unlocked
9     #define feof_function feof_unlocked
10    #define ferror_function ferror_unlocked
11    #define fflush_function fflush_unlocked
12  #endif
13 #else
14   #define fgetc_function fgetc
15   #define feof_function feof
16   #define ferror_function ferror
17   #define fflush_function fflush
18 #endif
19 }

```

Listing A.13: Preprocessor commands for detecting “READNOLOCK” flag and OS. If both are found, the “non locking” functions are used. Note that all possible “non locking” functions are defined.

Moreover, the code in A.12 has the advantage that, with slight modifications, it can be used to read and separate by words (with any single character word delimiter) in a single iteration (the usual procedure is to read the data in a buffer, and then parse the buffer to separate by words), thus halving the processing times. In code snippet A.14 is shown the code for reading and separating the words by blank space characters. If multiple consecutive blank spaces are found, they will be considered as a single word separator, instead of multiple empty words:

```

1 int getLWords (int *n, int *numWords, int *lenWords, FILE *
2 stream) {
3
4   register int rd;
5   /*The "register" keyword is to indicate the compiler to try
6    to save this variable in a CPU
7    register instead of in memory, so it has faster access
8    times*/
9   *n = 0;
10  *numWords = 0;
11  lenWords[*numWords] = 0;
12  do {
13    rd=fgetc_function(stream);
14    switch(rd) {
15      case '\r':
16        fgetc_function(stream);
17        wordlst[*numWords][lenWords[*numWords]]='\0';
18        return (*n);
19      break;

```

```

17         case '\n':
18             wordlst[*numWords][lenWords[*numWords]]='\0';
19             return (*n);
20             break;
21         case '\0':
22             wordlst[*numWords][lenWords[*numWords]]='\0';
23             return (*n);
24             break;
25         case EOF:
26             wordlst[*numWords][lenWords[*numWords]]='\0';
27             if ((*n)==0) return -1;
28             else return (*n);
29             break;
30         case ' ':
31             if (lenWords[*numWords]==0) continue; /*
Multiple consecutive spaces are considered a single column
separator*/
32             (*n)++;
33             wordlst[*numWords][lenWords[*numWords]]='\0';
34             (*numWords)++;
35             lenWords[*numWords]=0;
36             break;
37         default:
38             (*n)++;
39             wordlst[*numWords][lenWords[*numWords]]=rd;
40             lenWords[*numWords]++;
41             if ((*n)>=MAX_INPUT_LINE-3) return -1; /*Line
too long, quit reading to avoid buffer overflow*/
42             break;
43     }
44     } while (1);
45
46     return -1;
47 }

```

Listing A.14: gLAB 6.0.0 function for reading a line modified for separating words by blank spaces.

The limitation of code in A.14 is that the word delimiter is hard coded, but for most of the cases (especially in GNSS) this is not a problem. If it is necessary to parse files with different word separators, the simple solution is to duplicate the function and modify the word delimiter. For instance, to process the Galileo eXtensible Markup Language (XML) almanac files, the code is shown in A.15:

```

1 int getLWordsXML (int *n, int *numWords, int *lenWords, FILE *
stream) {
2
3     register int rd;
4     /*The "register" keyword is to indicate the compiler to try
to save this variable in a CPU
5     register instead of in memory, so it has faster access
times*/
6     *n = 0;

```

```

7   *numWords = 0;
8   lenWords[*numWords] = 0;
9   do {
10    rd=fgetc_function(stream);
11    switch(rd) {
12        case '\r':
13        fgetc_function(stream);
14        wordlst[*numWords][lenWords[*numWords]]='\0';
15        return (*n);
16        break;
17        case '\n':
18        wordlst[*numWords][lenWords[*numWords]]='\0';
19        return (*n);
20        break;
21        case '\0':
22        wordlst[*numWords][lenWords[*numWords]]='\0';
23        return (*n);
24        break;
25        case EOF:
26        wordlst[*numWords][lenWords[*numWords]]='\0';
27        if ((*n)==0) return -1;
28        else return (*n);
29        break;
30        case '<': case '>':
31        (*n)++;
32        wordlst[*numWords][lenWords[*numWords]]='\0';
33        (*numWords)++;
34        lenWords[*numWords]=0;
35        break;
36        default:
37        (*n)++;
38        wordlst[*numWords][lenWords[*numWords]]=rd;
39        lenWords[*numWords]++;
40        if((*n)>=MAX_INPUT_LINE-3) return -1; //Line
too long, quit reading to avoid buffer overflow*/
41        break;
42    }
43 } while (1);
44
45 return -1;
46 }

```

Listing A.15: gLAB 6.0.0 function for reading a line modified for separating words by '<' and '>' (XML files).

A.9 Micro-optimizations

Micro-optimizations are optimizations that reduce processing time in a single or few CPU cycles. If a micro-optimization is performed several times in a function, and the function is called several millions times (for instance the gLAB function

that reads a line from a file, shown in section A.8) the total saved time can be of the order of a fraction of a second (or even greater). If multiple micro-optimizations are added together, the total saved processing time can be quite noticeable (at human perception level). The most common micro-optimizations are related to RAM access, where the CPU can simply skip reading or writing to RAM and use only its internal registers or caches.

A.9.1 Inlining functions

Each time a function is called, the CPU needs to save the value of its internal registers in the stack and write the functions parameters to the stack, then call the function, and when it returns, restore the internal register from the stack. If the function is small enough (the maximum size is decided by the compiler), the compiler can simply embed the code of the called function inside the current function. This is known as inlining functions.

To inline functions, the keyword “inline” must be used in both the declaration and definition of a function. The usage of this keyword does not guarantee that the compiler actually inlines the function, as the compiler will not do it unless the required conditions are met (in which case it will be treated as any other function).

In gLAB, the “inline” keyword is used only in function “m2v`opt” (since version 2.0.0), show in code snippet A.16. This function is also defined as “static”, to explicitly indicate that the function will only be used in the current translation unit (only used in the source file where the function is located), which will increase the chances that the compiler can inline the function, since sharing a function between translation units can be a reason for not inlining a function.

```
1 static inline int m2v_opt (int i, int j) {  
2     return i*(i+1)/2+j;  
3 }
```

Listing A.16: gLAB inline function definition.

A.9.2 Register keyword

The “register” keyword in a variable declaration is used to indicate to the compiler that it should try to emit code where the variable is only used inside the CPU registers, which may not always be possible (in which case it will be treated as any other variable). The advantage of this approach is that CPU can skip loading and saving the variable value into RAM.

There are the following limitations when a variable is only used on registers:

- They do not have a memory address (therefore no pointers can be used to access this variable).
- They cannot be global variables. They only exist inside the function where they are declared (but its value can be passed as a parameter to another function or be used as the return value for the current function).
- There must be enough registers in the CPU to perform the rest of the computations, as each variable only in registers will permanently occupy one register.

An example of “register” keyword is shown in code snippet [A.12](#).

A.9.3 Restrict keyword

When a function has several input parameters two or more of which are pointers to the same type of variable (e.g. pointer to integers arrays), the compiler has no way of knowing whether the pointers of the same type are pointing to the same memory region (memory overlap). For instance, each pointer may refer to different indices of the same array, which allows the whole array to be accessed with either of the pointers. This is known as memory aliasing.

As the compiler does not know whether there is memory aliasing, the compiler is forced to assume that there is memory aliasing in order to guarantee the correctness of the code result, which implies a forced value refresh (from RAM) of the memory pointed to by the current pointer whenever any of the pointers of the same type modify a value.

When the “restrict” keyword is used, it indicates to the compiler that there is no memory aliasing, therefore the forced value refreshed is no longer executed if not necessary.

Note that if the user uses the “restrict” keyword and there is indeed memory aliasing, the results will be incorrect. As an additional comment, the optimizations performed by the compilers when the “register” keyword is used occurs only when the compiler optimization flag ‘-O3’ is enabled, and therefore an incorrect usage of this keyword can remain undetected in non-optimized builds (usually the non production version). A full explanation of this keyword is provided in the following video: [mCoding \[2021\]](#).

An example of “restrict” keyword is shown in code snippet [A.17](#) with the declaration of the standard function “memcpy”, which is used to copy memory regions and it assumes that both memory regions do not overlap:

```
1 void* memcpy(void *restrict out, const void *restrict in,  
    size_t n);
```

Listing A.17: “memcpy” function declaration.

A.9.4 Const keyword

In C, the “const” keyword does not mean a constant is declared, but it sets the memory region indicated used by the variable as read only (which is not technically the same, since a constant is a hard-coded value, and therefore there are small differences).

The “const” keyword meaning can be different depending on its position and whether the variable is a pointer. There are the following cases:

1. The variable is not a pointer. In this case, using “const” prior or after the type specifier does not make any difference, as there is only the memory region where the variable is stored. An example is provided in code snippet [A.18](#):

```
1 const int x; /*x is an integer which is constant*/  
2 int const x; /*x is a constant integer*/
```

Listing A.18: “const” keyword with non pointer variables.

2. The variable is a pointer and the “const” keyword is prior to the type specifier. In this case, the pointer itself can be modified, but the memory region which the pointer points to is set as read only. An example is provided in code snippet [A.19](#):

```
1 const int *x; /*x is a pointer to an integer which is  
    constant*/
```

Listing A.19: “const” keyword prior to type specifier with pointer variables.

3. The variable is a pointer and the “const” keyword is after the type specifier. In this case, the pointer itself cannot be modified (as the memory region where the pointer is saved is set as read only), but the memory region pointed by the point can be modified. An example is provided in code snippet [A.20](#):

```
1 int const *x; /*x is a constant pointer to an integer*/
```

Listing A.20: “const” keyword prior after the type specifier with pointer variables.

Note that, as shown in code snippets [A.18](#), [A.19](#) and [A.20](#), reading the attributes from right-hand to left-hand makes it easier to understand the effect of the “const” keyword.

When a non-pointer variable is declared as “const”, the compiler may decide to convert the variable to a hard-coded value. When used in any other variable declaration, the compiler may be able to generate better code (e.g. less memory accesses to this variable).

The “const” keyword can be used in the declaration of function parameters. In this case, the “const” keyword will only take effect for the duration of the function. A typical use is for parameters pointing to arrays, since knowing that the array content cannot be modified may allow advanced techniques such as vectorization. An example of “const” usage can be found in code snippet [A.17](#).

A.9.5 Variable declaration in for loop definition

When a local variable is declared, its lifetime is also indirectly determined. A local variable declared inside a function will only last for the current execution of a function.

Likewise, when a *for* loop is used, if the control variable for the *for* loop is declared in the *for* definition, then the lifetime of this local variable will be only during the execution of the current *for* loop. On the one hand, this variable will not be accessible before or after the *for* loop, and on the other hand, it allows the compiler to skip the update of the variable value in RAM, as it will only be able to use the variable inside the CPU registers (especially in the last iteration of the loop). This optimization is not possible if the variable needs to be used after the loop ends.

In code snippet [A.21](#) is shown the case where the variable is declared outside the *for* loop and in code snippet [A.22](#) the variable is declared in the *for* definition.

```
1 int i;
2 /*i variable is accessible
   before loop starts*/
3 for(i=0;i<10;i++) {
4   /*code execution*/
5 }
6 /*i variable is accessible
   after loop ends*/
```

Listing A.21: Variable defined prior to *for* loop.

```
1 /*i variable is not
   accessible before loop
   starts*/
2 for(int i=0;i<10;i++) {
3   /*code execution*/
4 }
5 /*i variable is no longer
   accessible after loop
   ends*/
```

Listing A.22: Variable defined in *for* loop definition.

A.9.6 Variable declaration inside a loop

When a local variable is declared, its lifetime is also indirectly determined. A local variable declared inside a function will only last for the current execution of a function.

Likewise, when a *for* or *while* loop is used, if a variable is declared inside the loop, the lifetime of the variable will be a single iteration of the loop. Therefore, in each iteration, the value of the variable in each iteration will be undefined until a value is set, and it will not be the last value saved in the previous iteration of the loop. This optimization is not possible if the variable value needs to be used across iterations or be used after the loop ends.

As the variable lasts for only one iteration of the loop, the CPU, if possible, can skip saving the variable value into RAM and use only the internal registers. In code snippet A.23 is shown the case where the variable is declared prior to the loop and on code snippet A.24 the variable is declared inside the loop:

```
1 int i,j;
2 /*j variable is accessible
3 before loop starts*/
4 for(i=0;i<10;i++) {
5     /*code execution with j
6     variable*/
7     /*Variable j value is
8     saved across iterations*/
9 }
10 /*j variable is accessible
11 after loop ends*/
```

Listing A.23: Variables defined prior to *for* loop.

```
1 /*j variable is not
2 accessible before loop
3 starts*/
4 int i;
5 for(i=0;i<10;i++) {
6     int j;
7     /*code execution with j
8     variable*/
9     /*Variable j value is not
10    saved across iterations*/
11 }
12 /*j variable is no longer
13 accessible after loop
14 ends*/
```

Listing A.24: Variable defined inside loop.

A.10 Other possible optimizations

There are other optimization techniques which are not implemented in gLAB but of which the thesis author is aware, which may be used in the future:

- Branchless coding.
- Rewrite code sections in order to reduce CPU cache misses.
- Usage of CPU vectorization instructions.

- Usage of compiler attributes to provide hints to the CPU branch predictor.
- Usage of compiler attributes in variables (GNU [2023c]) or functions declarations (GNU [2023a]).

Appendix B

gLAB downloads per country

In this annex the numbers of downloads per country (from all gLAB versions) in the period from 30th April 2019 to 12th of March 2023 are provided in table [B.1](#) (ordered by number of downloads, starting with the most downloaded):

Table B.1: Full list of gLAB number of downloads per country (from the 30th April 2019 to 12th of March 2023) ordered by number of downloads.

Order	Number of downloads	Percentage	Country code	Country name
1	183785	82.5904%	CN	China
2	4447	1.9984%	US	United States
3	2917	1.3109%	SG	Singapore
4	2731	1.2273%	IR	Iran
5	2243	1.0080%	ID	Indonesia
6	2109	0.9478%	IN	India
7	2091	0.9397%	DE	Germany
8	1764	0.7927%	NG	Nigeria
9	1252	0.5626%	ES	Spain
10	1207	0.5424%	BR	Brazil
11	1061	0.4768%	JP	Japan
12	854	0.3838%	MA	Morocco

Continued on next page

Table B.1: Full list of gLAB number of downloads per country (from the 30th April 2019 to 12th of March 2023) ordered by number of downloads. (Continued)

13	820	0.3685%	FR	France
14	766	0.3442%	DZ	Algeria
15	762	0.3424%	RU	Russia
16	755	0.3393%	EG	Egypt
17	718	0.3227%	CO	Colombia
18	652	0.2930%	PL	Poland
19	604	0.2714%	GH	Ghana
20	598	0.2687%	IT	Italy
21	514	0.2310%	UG	Uganda
22	510	0.2292%	GR	Greece
23	508	0.2283%	GB	United Kingdom
24	472	0.2121%	CL	Chile
25	462	0.2076%	VN	Vietnam
26	348	0.1564%	NP	Nepal
27	344	0.1546%	RO	Romania
28	330	0.1483%	TW	Taiwan
29	304	0.1366%	TR	Turkey
30	295	0.1326%	IQ	Iraq
31	289	0.1299%	ET	Ethiopia
32	283	0.1272%	NL	Netherlands
33	232	0.1043%	KE	Kenya
34	215	0.0966%	MX	Mexico
35	198	0.0890%	AU	Australia
36	173	0.0777%	EC	Ecuador
37	170	0.0764%	CZ	Czechia
38	168	0.0755%	TH	Thailand
39	165	0.0741%	KR	South Korea
40	160	0.0719%	CA	Canada
41	155	0.0697%	MY	Malaysia

Continued on next page

Table B.1: Full list of gLAB number of downloads per country (from the 30th April 2019 to 12th of March 2023) ordered by number of downloads. (Continued)

42	153	0.0688%	AE	United Arab Emirates
43	146	0.0656%	IL	Israel
44	140	0.0629%	SA	Saudi Arabia
45	135	0.0607%	OM	Oman
46	130	0.0584%	PT	Portugal
47	128	0.0575%	LK	Sri Lanka
48	126	0.0566%	PE	Peru
49	125	0.0562%	BG	Bulgaria
50	123	0.0553%	BD	Bangladesh
51	122	0.0548%	BO	Bolivia
52	118	0.0530%	ZW	Zimbabwe
53	99	0.0445%	FI	Finland
54	95	0.0427%	PH	Philippines
55	92	0.0413%	AR	Argentina
56	90	0.0404%	NO	Norway
57	87	0.0391%	AT	Austria
58	85	0.0382%	UA	Ukraine
59	84	0.0377%	SE	Sweden
60	83	0.0373%	SN	Senegal
61	81	0.0364%	BE	Belgium
62	74	0.0333%	RS	Serbia
63	74	0.0333%	CH	Switzerland
64	69	0.0310%	TZ	Tanzania
65	69	0.0310%	CI	Ivory Coast
66	68	0.0306%	SO	Somalia
67	64	0.0288%	HR	Croatia
68	64	0.0288%	BI	Burundi
69	58	0.0261%	CM	Cameroon
70	57	0.0256%	SY	Syria

Continued on next page

Table B.1: Full list of gLAB number of downloads per country (from the 30th April 2019 to 12th of March 2023) ordered by number of downloads. (Continued)

71	56	0.0252%	EE	Estonia
72	55	0.0247%	TT	Trinidad and Tobago
73	54	0.0243%	PK	Pakistan
74	53	0.0238%	ZA	South Africa
75	52	0.0234%	BA	Bosnia and Herzegovina
76	49	0.0220%	VE	Venezuela
77	47	0.0211%	MM	Myanmar
78	45	0.0202%	QA	Qatar
79	44	0.0198%	SI	Slovenia
80	43	0.0193%	SD	Sudan
81	41	0.0184%	DO	Dominican Republic
82	38	0.0171%	LB	Lebanon
83	36	0.0162%	YE	Yemen
84	33	0.0148%	HU	Hungary
85	31	0.0139%	MV	Maldives
86	31	0.0139%	LT	Lithuania
87	27	0.0121%	SK	Slovakia
88	27	0.0121%	JM	Jamaica
89	25	0.0112%	JO	Jordan
90	24	0.0108%	LY	Libya
91	23	0.0103%	XK	Kosovo
92	23	0.0103%	AL	Albania
93	22	0.0099%	KZ	Kazakhstan
94	21	0.0094%	NZ	New Zealand
95	21	0.0094%	IE	Ireland
96	21	0.0094%	CR	Costa Rica
97	18	0.0081%	TL	Timor Leste
98	18	0.0081%	DK	Denmark

Continued on next page

Table B.1: Full list of gLAB number of downloads per country (from the 30th April 2019 to 12th of March 2023) ordered by number of downloads. (Continued)

99	17	0.0076%	LU	Luxembourg
100	15	0.0067%	PG	Papua New Guinea
101	15	0.0067%	PA	Panama
102	14	0.0063%	CU	Cuba
103	14	0.0063%	BY	Belarus
104	12	0.0054%	UY	Uruguay
105	11	0.0049%	PY	Paraguay
106	11	0.0049%	MW	Malawi
107	11	0.0049%	KG	Kyrgyzstan
108	11	0.0049%	BF	Burkina Faso
109	10	0.0045%	AG	Antigua and Barbuda
110	9	0.0040%	MO	Macao
111	9	0.0040%	LV	Latvia
112	8	0.0036%	BW	Botswana
113	7	0.0031%	FK	Falkland Islands
114	7	0.0031%	CY	Cyprus
115	6	0.0027%	PF	French Polynesia
116	6	0.0027%	MN	Mongolia
117	6	0.0027%	IS	Iceland
118	5	0.0022%	MZ	Mozambique
119	5	0.0022%	MU	Mauritius
120	5	0.0022%	MG	Madagascar
121	5	0.0022%	BT	Bhutan
122	4	0.0018%	TN	Tunisia
123	4	0.0018%	NI	Nicaragua
124	4	0.0018%	MK	North Macedonia
125	4	0.0018%	HN	Honduras
126	4	0.0018%	GT	Guatemala
127	4	0.0018%	BB	Barbados

Continued on next page

Table B.1: Full list of gLAB number of downloads per country (from the 30th April 2019 to 12th of March 2023) ordered by number of downloads. (Continued)

128	3	0.0013%	SV	El Salvador
129	3	0.0013%	SL	Sierra Leone
130	3	0.0013%	RE	Reunion
131	3	0.0013%	KW	Kuwait
132	3	0.0013%	CD	Democratic Republic of the Congo
133	3	0.0013%	BH	Bahrain
134	2	0.0009%	TG	Togo
135	2	0.0009%	GE	Georgia
136	2	0.0009%	CV	Cabo Verde
137	1	0.0004%	VU	Vanuatu
138	1	0.0004%	SR	Suriname
139	1	0.0004%	NA	Namibia
140	1	0.0004%	MS	Montserrat
141	1	0.0004%	ME	Montenegro
142	1	0.0004%	KH	Cambodia
143	1	0.0004%	GA	Gabon
144	1	0.0004%	BZ	Belize
145	1	0.0004%	BS	Bahamas
146	1	0.0004%	BN	Brunei
147	1	0.0004%	AM	Armenia

Appendix C

Multi-day SBAS availability maps computation

In this annex, the gLAB manual pages with SBAS availability maps description and multiple day addition are attached (starting on the next page). They are attached exactly as in the gLAB manual, as this is the format and text that the user will encounter when reading the gLAB manual.

9 SBAS MAPS DESCRIPTION

SBAS plots mode is a processing mode (available only in command line) where, instead of computing the navigation solution, gLAB loops through a grid of points (in longitude and latitude) in the map (by default the EGNOS coverage area), during a 24 hour period. In each point, gLAB checks how many satellites are available for SBAS solution (by computing its SBAS corrections). If there are 4 or more satellites with SBAS corrections (and the geometry matrix can be inverted), the protection levels are computed. If the protection levels are below the alarm limits, then there is PA solution available for the current point and epoch. When the selected GEO is below the elevation mask or has no PA (Precision Approach) solution, if GEO switching is enabled, gLAB will loop through all GEO until a GEO is over the elevation mask and provides a PA solution or no unprocessed GEO remain.

In order to compute the geometry matrix, receiver a priori coordinates are needed. The a priori coordinates are the longitude and latitude of the current point in the grid, and the height is a fixed for all points (by default 0 metres -over the WGS84 geoid-). The geometry matrix also contains the measurements weights provided by the SBAS corrections.

In SBAS plots mode, no RINEX observation file is used (as it is not necessary to compute the navigation solution). In order to select which satellites are available at each point in each epoch, gLAB has to loop through all available PRN (for GPS, PRNs 1 to 32) and try to compute its SBAS corrections.

The number of epochs to be processed for the SBAS maps depends on the data files and the software used. gLAB computes daily (24 hours) SBAS map, with the possibility to generate hourly maps (use option `'-sbasplots:hourlymaps'`).

Two data files are necessary for computing a daily SBAS map (24 hours): a SBAS corrections file and a navigation file. In order to account for the convergence time (in this case, wait for the SBAS data buffers get filled), the SBAS data file must contain additional corrections from the previous day. The minimum amount of data is 15 minutes, but gLAB accepts up to two hours of data of convergence time. That is, if the SBAS data file starts at 22h or later, gLAB will automatically assume that all the data from 22h (or later) until 23:59h is data for convergence time and will read all this corrections at once for filling its internal SBAS data memory buffers. Once it reaches 0:00h of the following day it will start the data processing until 23:59h. Since version 5.4.0, gLAB shows an INFO message with the amount of convergence time detected in the SBAS file. If it less than 15 minutes, gLAB warns the user by stating that there is not enough convergence time.

Regarding the navigation file, it is necessary for the user to create a consolidated navigation data file using data from many different stations (for instance, all IGS stations). This is due to each station does not have all of the navigation messages (selected by the IODE) used by SBAS (this can even occur with the 'brdc' consolidated file). One way to check that the navigation file has missing messages is to do a standard SBAS processing (using a RINEX observation file from any station) and enable the "SBASUNSEL" output message. In this message it should appear messages with the text "No broadcast block with IOD...". Since version 5.4.0, gLAB prints an INFO message about the need to create a consolidated file (except if the navigation filename's first 4 characters are "mixn" or "gage", as these are the navigation files created by the gAGE research group). Aside, for the navigation file, it is necessary to include 2 hours of navigation messages from the previous day and 2 hours from the following day. This is due as some of the navigation messages used by SBAS at the start or end of the day are saved in the previous or following day. Since version 5.4.0, gLAB checks the start and end time of the navigation file for these 2 hours of data from the previous and

following day. If not included, a warning message is printed. Furthermore, it is recommended to apply a data cleansing algorithm to the navigation messages when creating the consolidated navigation message file, in order to remove data logging errors. An algorithm for data cleansing is described in chapter 3.3 of the thesis "Safe satellite navigation with multiple constellations: global monitoring of GPS and GLONASS signal-in-space anomalies", available at "<https://web.stanford.edu/group/scpnt/gpslab/pubs/theses/LHengThesisFinalSignedSecured.pdf>"

As stated above, gLAB can only process a single day in each run. When the user wants to process a range of days, it will need to process each day independently and join the results following the instructions below. Processing only a single day per run has the following advantages and disadvantages:

Advantages:

- More efficient code as it does not need to deal with day changes.
- Multiple instances of gLAB can be run independently in parallel (one for each CPU available). It is much more efficient than running a single instance with multi-threading for multiple days.
- If the results are not as expected, it is easier to identify the day(s) with problems (e.g. an ionospheric storm), as each day is an independent run.
- If one processed day fails (e.g. missing data), only that day has to be reprocessed.
- Data files remain the same size independently of the data range size processed.

Disadvantages:

- More efficient code as it does not need to deal with day changes.
- Data files with the convergence time have to be created for each day.
- An external script to run gLAB for each day must be generated by the user.
- Percentiles for DOP can only be computed for a single day, as DOP samples are not printed. Nevertheless, the DOP percentile cannot be computed for a long range of days, as for each day, each DOP component (HDOP, PDOP or GDOP) needs around 1GB of memory. For instance, computing HDOP, PDOP and GDOP percentiles for 10 days would need around 30GB of memory.
- Data for each day has to be added up in order to get the statistics for the whole period (explained below).

In order to join the results from multiple days, it is necessary to build a script that reads the gLAB output files and adds everything up. The user can generate this script in many ways (e.g. a Python script). Below is provided some Bash scripts (which can be executed in Linux, Mac or in Windows -using the Cygwin terminal-). In these scripts, the following assumptions are made:

- The output files have the day of year (padded with zeros) and the year in its filenames. This is very important, as this allows to select all the files at once in chronological order (as the bash interpreter automatically orders the selected files). In the examples the filenames used are (user can set the output filenames using the appropriate parameters):

Daily files:

AvailMap_001_2017.txt, RiskMap_002_2017.txt, RiskMarMap_003_2017.txt,
lonoMap_004_2017.txt, ...

HDOPMap_001_2017.txt, PDOPMap_002_2017.txt,
GDOPMap_003_2017.txt, Discontinuities_List_004_2017.txt, ...

Hourly files:

AvailMap_001_2017_00h.txt, RiskMap_002_2017_01h.txt,
RiskMarMap_003_2017_10h.txt, lonoMap_004_2017_11h.txt, ...

HDOPMap_001_2017_12h.txt, PDOPMap_002_2017_13h.txt,
GDOPMap_003_2017_14h.txt, Discontinuities_List_004_2017_23h.txt, ...

- Data files are saved in a different folder for each day. The folder structure is "YYYY/DoY", being "YYYY" the year with four digits and "DoY" the day of year padded with zeros.
- This example will join data from two years (2017 and 2018).

9.1 ADDING UP SBAS AVAILABILITY MAPS

SBAS Availability map data is easy to add up, as the output file contains the number of epochs with PA solution under both alarm limits (horizontal and vertical) and the number of epochs processed. These values are independent from day to day, therefore it is only needed to add to total number of epochs with solution and the total number of epochs processed for each day and location, and divide these values in order to get the availability percentage for the whole period. The following bash script does this computation:

```
cat ./201?/???/AvailMap_???_201?.txt|awk -v OFMT="%9.5f" '$1=="#MINLAT"
{getline;getline;getline} {avail[$1] "$2")+=$4;numepochs[$1] "$2")+=$5} END
{for (i in avail) {print i,avail[i],numepochs[i],numepochs[i]==0?0:100*avail[i]/numepochs[i]}}'|awk
'{printf "%6.2f %7.2f %9.5f %8d %8d\n",$1,$2,$5,$3,$4}'|sort -g -k1 -k2 >
Avail_added_no_header.txt
```

NOTES:

- For selecting all the hourly maps, change the filename in the previous command from './201?/???/AvailMap_???_201?.txt' to './201?/???/AvailMap_???_201?_??h.txt'
- For selecting some hourly maps (not necessarily continuous hours), a comma separated list of hours (padded with zeros) inside curly braces must be given in the filenames.

For instance, for selecting hours "1,3,16,17,18,23", the filename should be: './201?/???/AvailMap_???_201?_{01,03,16,17,18,23}h.txt'

The previous command line creates a new file in the gLAB format with all the values added, but with no header. Any header from any of the output files for availability maps can be used. The following command reads the header from the file "AvailMap_001_2017.txt" and copy it to the previous file:

```
head -3 ./2017/001/AvailMap_001_2017.txt|cat - Avail_added_no_header.txt >
Avail_added_with_header.txt
```

The file created, 'Avail_added_with_header.txt' can now be processed by the gLAB plotting tool as a normal SBAS availability plot file. The user will have to change the plot title to state that the file has data for the two years (using the '-t' parameter).

9.2 ADDING UP SBAS IONOSPHERE AVAILABILITY MAPS

The process is identical as the SBAS availability maps. The same scripts are used, but changing the filenames:

```
cat ./201?/???/IonoMap_???_201?.txt|awk -v OFMT="%9.5f" '$1=="#MINLAT"
{getline;getline;getline} {avail[$1] " "$2)+=$4;numepochs[$1] " "$2)+=$5} END
{for (i in avail) {print
i,avail[i],numepochs[i],numepochs[i]==0?0:100*avail[i]/numepochs[i]}}'|awk
'{printf "%6.2f %7.2f %9.5f %8d %8d\n",$1,$2,$5,$3,$4}'|sort -g -k1 -k2 >
IonoAvail_added_no_header.txt
```

NOTES:

- For selecting all the hourly maps, change the filename in the previous command from './201?/???/IonoMap_???_201?.txt' to './201?/???/IonoMap_???_201?_??h.txt'
- For selecting some hourly maps (not necessarily continuous hours), a comma separated list of hours (padded with zeros) inside curly braces must be given in the filenames.

For instance, for selecting hours "1,3,16,17,18,23", the filename should be: './201?/???/IonoMap_???_201?_{01,03,16,17,18,23}h.txt'

The previous command line creates a new file in the gLAB format with all the values added, but with no header. Any header from any of the output files for availability maps can be used. The following command reads the header from the file "IonoAvailMap_001_2017.txt" and copy it to the previous file:

```
head -3 ./2017/001/IonoAvailMap_001_2017.txt|cat -
IonoAvail_added_no_header.txt > IonoAvail_added_with_header.txt
```

The file created, 'IonoAvail_added_with_header.txt' can now be processed by the gLAB plotting tool as a normal SBAS ionosphere availability plot file. The user will have to change the plot title to state that the file has data for the two years (using the '-t' parameter).

9.3 ADDING UP SBAS HDOP MAPS

SBAS HDOP output files contain the mean value, the total HDOP and the number of epochs with PA solution. Note that all epochs with protection levels over the alarm limits are not counted for the DOP percentile or mean value, as most of them will be outliers (this is the behaviour since version 5.4.0, whereas in version 5.3.0 all epochs with PA solution were

included in the mean computation). The mean value for each day is independent from each day, therefore we just need to add the total HDOP value for each day and position and the total number of epochs with PA solution, and compute the global mean for the two years. In some positions (in the edges), the number of epochs with PA solution may be 0, so it is necessary to check for this value in order to avoid dividing by zero:

```
cat ./2017/???/HDOPMap_???_2017.txt|awk -v OFMT="%9.5f" '$1=="#MINLAT"
{getline;getline;getline} {mean[$1] "$2)+=$3;numepochs[$1"
"$2)+=$5;total[$1" "$2)+=$4;percent[$1" "$2)+=$6} END {for (i in total)
{print
i,total[i],numepochs[i],numepochs[i]==0?0:total[i]/numepochs[i],percent[i
]}}'|awk '{printf "%6.2f %7.2f %15.2f %17.2f %8d
%10.2f\n", $1,$2,$5,$3,$4,$6}'|sort -g -k1 -k2 > HDOP_added_no_header.txt
```

NOTES:

- For selecting all the hourly maps, change the filename in the previous command from './2017/???/HDOPMap_???_2017.txt' to './2017/???/HDOPMap_???_2017_???.txt'
- For selecting some hourly maps (not necessarily continuous hours), a comma separated list of hours (padded with zeros) inside curly braces must be given in the filenames.

For instance, for selecting hours "1,3,16,17,18,23", the filename should be: './2017/???/HDOPMap_???_2017_{01,03,16,17,18,23}h.txt'

The previous command line creates a new file in the gLAB format with all the values added, but with no header. Any header from any of the output files for HDOP maps can be used. The following command reads the header from the file "HDOPMap_001_2017.txt" and copy it to the previous file:

```
head -3 ./2017/001/HDOPMap_001_2017.txt|cat - HDOP_added_no_header.txt >
HDOP_added_with_header.txt
```

The file created, 'HDOP_added_with_header.txt' can now be processed by the gLAB plotting tool as a normal SBAS HDOP plot file. The user will have to change the plot title to state that the file has data for the two years (using the '-t' parameter).

9.4 ADDING UP SBAS PDOP MAPS

Identical to HDOP maps, but changing the filenames to PDOP files:

```
cat ./2017/???/PDOPMap_???_201?.txt|awk -v OFMT="%9.5f" '$1=="#MINLAT"
{getline;getline;getline} {mean[$1] " "$2]+=$3;numepochs[$1"
"$2]+=$5;total[$1" "$2]+=$4;percent[$1" "$2]+=$6} END {for (i in total)
{print
i,total[i],numepochs[i],numepochs[i]==0?0:total[i]/numepochs[i],percent[i
]}}'|awk '{printf "%6.2f %7.2f %15.2f %17.2f %8d
%10.2f\n",$1,$2,$5,$3,$4,$6}'|sort -g -k1 -k2 > PDOP_added_no_header.txt
```

NOTES:

- For selecting all the hourly maps, change the filename in the previous command from './2017/???/PDOPMap_???_201?.txt' to './2017/???/PDOPMap_???_201?_??h.txt'
- For selecting some hourly maps (not necessarily continuous hours), a comma separated list of hours (padded with zeros) inside curly braces must be given in the filenames.

For instance, for selecting hours "1,3,16,17,18,23", the filename should be: './2017/???/PDOPMap_???_201?_{01,03,16,17,18,23}h.txt'

The previous command line creates a new file in the gLAB format with all the values added, but with no header. Any header from any of the output files for PDOP maps can be used. The following command reads the header from the file "PDOPMap_001_2017.txt" and copy it to the previous file:

```
head -3 ./2017/001/PDOPMap_001_2017.txt|cat - PDOP_added_no_header.txt >
PDOP_added_with_header.txt
```

The file created, 'PDOP_added_with_header.txt' can now be processed by the gLAB plotting tool as a normal SBAS PDOP plot file. The user will have to change the plot title to state that the file has data for the two years (using the '-t' parameter).

9.5 ADDING UP SBAS GDOP MAPS

Identical to HDOP maps, but changing the filenames to GDOP files:

```
cat ./2017/???/GDOPMap_???_2017.txt|awk -v OFMT="%9.5f" '$1=="#MINLAT"
{getline;getline;getline} {mean[$1" "$2]+=$3;numepochs[$1"
"$2]+=$5;total[$1" "$2]+=$4;percent[$1" "$2]+=$6} END {for (i in total)
{print
i,total[i],numepochs[i],numepochs[i]==0?0:total[i]/numepochs[i],percent[i
]}}'|awk '{printf "%6.2f %7.2f %15.2f %17.2f %8d
%10.2f\n",$1,$2,$5,$3,$4,$6}'|sort -g -k1 -k2 > GDOP_added_no_header.txt
```

NOTES:

- For selecting all the hourly maps, change the filename in the previous command from './2017/???/GDOPMap_???_2017.txt' to './2017/???/GDOPMap_???_2017_?h.txt'
- For selecting some hourly maps (not necessarily continuous hours), a comma separated list of hours (padded with zeros) inside curly braces must be given in the filenames.

For instance, for selecting hours "1,3,16,17,18,23", the filename should be: './2017/???/GDOPMap_???_2017_{01,03,16,17,18,23}h.txt'

The previous command line creates a new file in the gLAB format with all the values added, but with no header. Any header from any of the output files for GDOP maps can be used. The following command reads the header from the file "GDOPMap_001_2017.txt" and copy it to the previous file:

```
head -3 ./2017/001/GDOPMap_001_2017.txt|cat - GDOP_added_no_header.txt >
GDOP_added_with_header.txt
```

The file created, 'GDOP_added_with_header.txt' can now be processed by the gLAB plotting tool as a normal SBAS GDOP plot file. The user will have to change the plot title to state that the file has data for the two years (using the '-t' parameter).

9.6 ADDING UP SBAS AVIATION CONTINUITY RISK MAPS (SLIDING WINDOW)

When computing the total continuity risk for multiple continuous days from several single day files (or hourly files), data files are not independent from each other, as each discontinuity (defined as the transition from an epoch with PA solution -under both alarm limits- to an epoch without PA solution or with PA solution over any alarm limits) found affects previous samples. In the case of the sliding window, every time a discontinuity is found, the epochs with PA solution (under alarm limits) previous to the current discontinuity are counted as "in risk" or affected by the current discontinuity.

The number of samples "in risk" is the size of the window if the number of continuous epochs with PA solution under alarm limits (prior to the previous discontinuity) is equal or greater than the window size, otherwise, only the number of continuous epochs with PA solution under alarm limits (prior to the previous discontinuity) are "in risk". The total discontinuity risk is finally

computed by dividing the total number of epochs "in risk" by the total number of epochs with PA solution under alarm limits. Discontinuities can occur, for instance, at the beginning of the day, which would affect samples from the previous day. To account for these cases, a script (attached at the end of this manual) has to read the discontinuities (from the discontinuity list file generated by gLAB) and count the number of samples which were "in risk" by discontinuities but not counted as "in risk" due to the day or hour change.

In order to understand how to count these samples, two facts about how gLAB computes the number of discontinuities for the sliding window:

- For daily and hourly files, if the processing ended with discontinuity, the epochs "in risk" for this discontinuity are accounted in the statistics.
- For hourly files, at the beginning of each hour, internal counters are all reset, therefore, discontinuities from each hour do not affect the previous hour or the following one.

An example is shown below on how to compute the continuity risk, using a sliding window of 5 epochs (this is a very small window size, used for explanatory purposes. Default aviation sliding window size is 15 seconds):

NOTE: the following symbols are used:

- '|' is a separator for a day or hour transition
- 'X' is an epoch with PA solution under alarm limits (or in short, "epoch with PA solution")
- 'O' is an epoch without PA solution or with PA limits over any alarm limit (or in short, "epoch without PA solution")

```

O O O X X X X X X X X O O O X X X X O O O O O X X X O O O X X
      |   |   |   |   |   |   |
      - - - - -   - - - - -   - - -
      |           |           |
      v           v           v
5 epochs "in risk"  5 epochs "in risk"  3 epochs "in risk"

```

Total epochs "in risk": 13 epochs

Total epochs with PA solution: 18 epochs

Continuity Risk = $13/18 = 0.72$

The following cases can occur during a day or hour transition:

- Case 1: Day/hour ends without PA solution

```

X O O || O O O
or  X X O || X X O   (all epochs "in risk" are correctly accounted)
or  X X O || X X X

```

As stated above, when the day or hour ends with discontinuity, the previous epochs with PA solution are counted for the number of epochs "in risk". No additional epochs "in risk" have to be accounted to the global counter.

- Case 2: Day/hour ends with PA solution and the next day/hour file starts without PA solution

```

x x x || 0 0 0   or   x x x || 0 0 x
|   |
- - -
|
v

```

These epochs are not accounted as "in risk" due to the day/hour change

The discontinuity at the beginning of the next day/hour affects the epochs with PA solution from the previous day. The number of epochs "in risk" from the previous day will be the size of the window (if the number of continuous epoch with PA solution is equal or larger than the window size) or the number of continuous epoch with PA solution if it is smaller than the window size.

- Case 3: Day/hour ends with PA solution, the next day/hour starts with PA solution and a discontinuity occurs in an epoch which is smaller than the window size:

```

x x x || x x x 0   or   0 x x || x x x 0
|   |
- - -
|
v

```

These epochs are not accounted as "in risk" due to the day/hour change

- When a day/hour ends with PA solution, the following day starts with PA solution, and a discontinuity occurs, if the epoch of the discontinuity is equal or larger than the window size, all epochs "in risk" will be accounted (as they will all be from the next day), but if the epoch of the discontinuity is smaller than the window size, some of the epochs "in risk" will be in the previous day. Precisely, the number of epochs in "in risk" from the previous day that need to be added to the global counter will be the difference between the window size and the number of epochs with PA solution in the current day. If this difference is greater than the number of continuous epochs until the end of the day in the previous day, then the number of epochs "in risk" to be added to the global counter will be the number of continuous epochs until the end of the previous day.

To count the total continuity risk (sliding window) for the whole period, it has to be done in four steps. First, add the total number of epochs "in risk" and the total number of epochs with PA solution. This can be done with the following command line instruction:

```
cat ./201?/???/RiskMap_???_201?.txt| awk -v OFMT="%9.5f" '$1=="#MINLAT"
{getline;getline;getline} {numdiscont[$1 " "$2]+=$4;numepochs[$1 " "$2]+=$5}
END {for (i in numdiscont) {print
i,numdiscont[i],numepochs[i],numepochs[i]==0?1:numdiscont[i]/numepochs[i]
}}'|awk '{printf "%6.2f %7.2f %9.5f %8d %8d\n",$1,$2,$5,$3,$4}'|sort -g -
k1 -k2 > Risk_added_no_header.txt
```

NOTES:

- For selecting all the hourly maps, change the filename in the previous command from './201?/???/RiskMap_???_201?.txt' to './201?/???/RiskMap_???_201?_??h.txt'
- For selecting some hourly maps (not necessarily continuous hours), a comma separated list of hours (padded with zeros) inside curly braces must be given in the filenames.
For instance, for selecting hours "1,3,16,17,18,23", the filename should be: './201?/???/RiskMap_???_201?_{01,03,16,17,18,23}h.txt'

The previous command line creates a new file in the gLAB format with all the values added, but with no header. The second step is to count the number of epochs "in risk" missing in the global counter due to the day/hour transitions.

The AWK script 'count_samples_sliding_window.awk' at the end of this help) does the counting. To run it, execute the following command line:

```
cat ./201?/???/Discontinuities_List_???_201?.txt|awk -f
count_samples_sliding_window.awk > epochs_to_add_in_risk.txt
```

NOTES:

- For selecting all the hourly maps, change the filename in the previous command from './201?/???/Discontinuities_List_???_201?.txt' to './201?/???/Discontinuities_List_???_201?_??h.txt'
- For selecting some hourly maps (not necessarily continuous hours), a comma separated list of hours (padded with zeros) inside curly braces must be given in the filenames.
For instance, for selecting hours "1,3,16,17,18,23", the filename should be: './201?/???/Discontinuities_List_???_201?_{01,03,16,17,18,23}h.txt'
- The script 'count_samples_sliding_window.awk' will automatically recognize if files are daily or hourly and count epochs accordingly

The script 'count_samples_sliding_window.awk' creates an output file with the same format as the file 'Risk_added_no_header.txt'.

The third step is to add the epochs "in risk" from the 'epochs_to_add_in_risk.txt' file (generated by the AWK) script to the total epoch counter in 'Risk_added_no_header.txt' file. It can be done with the following command:

```
awk -v OFMT="%9.5f" 'NR==FNR {numdiscont[$1 " "$2]+=$4;numepochs[$1 "
"$2]+=$5;next} {numdiscont[$1 " "$2]+=$3} END{for (i in numdiscont) {print
i,numdiscont[i],numepochs[i],numepochs[i]==0?1:numdiscont[i]/numepochs[i]
}}' Risk_added_no_header.txt epochs_to_add_in_risk.txt|awk '{printf "%6.2f
```

```
%7.2f %9.5f %8d %8d\n", $1, $2, $5, $3, $4}'|sort -g -k1 -k2 >
Risk_added_allriskepochs_no_header.txt
```

The previous command line creates a new file in the gLAB format with all the values added, but with no header. Any header from any of the output files for continuity risk maps can be used. The following command reads the header from the file "RiskMap_001_2017.txt" and copy it to the previous file:

```
head -3 ./2017/001/RiskMap_001_2017.txt|cat -
Risk_added_allriskepochs_no_header.txt >
Risk_added_allriskepochs_header.txt
```

The file created, 'Risk_added_allriskepochs_header.txt' can now be processed by the gLAB plotting tool as a normal SBAS Continuity Risk plot file. The user will have to change the plot title to state that the file has data for the two years (using the '-t' parameter).

NOTE: To ensure the steps were done correctly, it is recommended to execute the previous commands twice. Once using only the daily files, and the next time using only the hourly files. The resulting files created by both executions should be equal, as the total number of epochs "in risk" has to be equal independently if the total number of epochs "in risk" are counted using daily or hourly files.

9.7 ADDING UP SBAS MARITIME CONTINUITY RISK MAPS (FIXED WINDOW)

The continuity risk for the fixed window is computed as the number of discontinuities multiplied by the window size and divided by the total number of epochs in the processed range (independently if any epoch had PA solution under alarm limits or not). For instance, if the processed range is two non-gap years, the total number of epochs is $2 \times 365 \times 86400 = 63072000$ (or 63158400 if there is a leap year). When adding up the total number of discontinuities for several days/hours, it can occur that some discontinuities are not accounted or other are counted twice. Prior to explain what these cases are, it is necessary first to detail three facts about how gLAB computes the discontinuities for the fixed window:

- For daily and hourly files, if the processing ended with discontinuity, this last discontinuity is accounted in the statistics.
- For hourly files, at the beginning of each hour, internal counters are all reset, therefore, discontinuities from each hour do not affect the previous hour or the following one.
- If one day or hour has no epochs with PA solution, the number of discontinuities and number of epochs will be 0.

An example is shown below on how to compute the continuity risk, using a fixed window of 5 epochs (this is a very small window size, used for explanatory purposes. Default maritime fixed window size is 900 seconds or 15 minutes):

NOTE: the following symbols are used:

- '|' is a separator for a day or hour transition
- 'X' is an epoch with PA solution under alarm limits (or in short, "epoch with PA solution")
- 'O' is an epoch without PA solution or with PA limits over any alarm limit (or in short, "epoch without PA solution")

```

O O O X X X X X X X X O O O X X X X O O O O O X X X O O O X X
| |           | |           |           | | |
- - -       - - - -       - - - - -       - - -
|           |           |           |
v           v           v           v
1 discontinuity      1 discontinuity      1 discontinuity      1 discontinuity

```

```

Total number of discontinuities:      4 discontinuities
Total number of epochs processed:    34 epochs
Window Size:                          5 epochs
Continuity Risk =  $4 \times 5 / 34 = 0.59$ 

```

The following cases can occur during a day or hour transition:

- Case 1: Day/hour ends without PA solution and the next epoch with PA solution is at the beginning of day/hour:

```

0 x 0 || x x 0
or  x x 0 || x x x      (All discontinuities are correctly accounted)
or  x x 0 || 0 0 0 || x x x

```

When the day or hour ends with discontinuity, the discontinuity is accounted in the global counter of total discontinuities. As the following day starts with PA solution (or if the discontinuity spans for multiple days but the first epoch with PA solution is at the beginning of the day/hour), only one discontinuity is accounted (as days without any epoch with PA solution have zero discontinuities). Therefore, no discontinuities have to be subtracted from the global counter.

- Case 2: Day/hour ends without PA solution and the following day starts without PA solution:

```

0 x 0 || 0 x 0   or   x x 0 || 0 x x   or   x x 0 || 0 0 0 || 0 x x   or   x 0 0 || 0 0 0 || 0 0 x
| |               | |               |           |               |           |
- -               - -               - - - - -   - - - - -
|                 |                 |           |               |           |
v                 v                 v           v               v           v

```

The discontinuity is counted twice due to the day/hour change (at the end and at the beginning of next day)

In this case, there is one discontinuity that start spans in multiple days/hours. This discontinuity is accounted at the end of the day/hour and at the beginning of the next day/hour with PA solution (if the discontinuity spans for multiple days/hours, days without any epochs with PA solution have zero discontinuities). That is, it is counted twice. Therefore it is necessary to subtract one discontinuity to the global counter of discontinuities.

- Case 3: At the beginning of the data range, a discontinuity lasts a full day/hour and the next day starts with PA solution:

```

|| 0 0 0 || x x 0   or   || 0 0 0 || x x x   or   || 0 0 0 || 0 0 0 || x x x   or   || 0 0 0 || 0 0 0 || x x 0
| |                 | |                 |           |               |           |
- - -               - - -               - - - - -   - - - - -
|                   |                   |           |               |           |
v                   v                   v           v               v           v

```

The discontinuity is not counted due to it starts with the first epoch processed and ends at the end of a day

If in the first day/hour of the data range processed (or every time there is a reset due to the data considered is not a range of continuous days or hours), the discontinuity lasts for the full day (and it may span to multiple days) and the first epoch with PA solution is at the beginning of the day, then this initial discontinuity is not accounted in the global counter. It is necessary to add one to the global counter.

- Case 4: A discontinuity lasts a full day/hour (or several days/hours) and the next day starts with PA solution:

```

X X X || 0 0 0 || X X X   or   X X X || 0 0 0 || 0 0 0 || X X X
      | |                   |           |
      - - -                 - - - - -
      |                       |
      v                       v

```

The discontinuity is not counted due to it starts at the beginning of a day and ends at the end of a day

If a discontinuity lasts for multiple days/hours, and it starts at the beginning of a day/hour and ends at the end of a day/hour, this discontinuity will not be accounted in the global counter of discontinuities, as days without PA solution have zero discontinuities. It is necessary to add one to the global counter.

- Case 5: At the end of the data range, a discontinuity starts at the beginning of day/hour and last until the end of the data range:

```

X X X || 0 0 0 ||   or   0 X X || 0 0 0 || 0 0 0 ||
      | |                   |           |
      - - -                 - - - - -
      |                       |
      v                       v

```

The discontinuity is not counted due to it starts at the beginning of a day and ends in the last epoch processed

At the end of the data range, if a discontinuity starts at the beginning of day and lasts for a full day or hour (or multiple full days or hours) until the end of data, then this discontinuity is not accounted in the global counter of discontinuities, as days without PA solution have zero discontinuities. Therefore it is necessary to add one discontinuity to the global counter of discontinuities.

To count the total continuity risk (fixed window) for the whole period, it has to be done in four steps. First, add the total total number of discontinuities. This can be done with the following command line instruction:

```
cat ./201?/???/RiskMarMap_???_201?.txt| awk -v totalepochs=63072000 -v
windowsize=900 -v OFMT="%9.5f" '$1=="#MINLAT" {getline;getline;getline}
{numdiscont[$1] "$2"+=$4;numepochs[$1] "$2"+=$5} END {for (i in numdiscont)
{print
i,numdiscont[i],numepochs[i],numepochs[i]==0?1:windowsize*numdiscont[i]/t
otalepochs}}'|awk '{printf "%6.2f %7.2f %9.5f %8d
%8d\n",$1,$2,$5,$3,$4}'|sort -g -k1 -k2 > RiskMar_added_no_header.txt
```

NOTES:

- The size of the window must be set (in seconds). By default is 900. To change it, change the text 'windowsize=900' to another value.

For instance, to set the window size to 15, the text should be 'windowsize=15'

- The total number of epochs of the processed must be set (in this example it is two years, which are 63072000 epochs). To change it, change the text 'totalepochs=63072000' to another value.
- For selecting all the hourly maps, change the filename in the previous command from './201?/???/RiskMarMap_???_201?.txt' to './201?/???/RiskMarMap_???_201?_??h.txt'
- For selecting some hourly maps (not necessarily continuous hours), a comma separated list of hours (padded with zeros) inside curly braces must be given in the filenames.

For instance, for selecting hours "1,3,16,17,18,23", the filename should be: './201?/???/RiskMarMap_???_201?_{01,03,16,17,18,23}h.txt'

The second step is to count the number of discontinuities counted twice and the number of discontinuities not counted due to the day/hour transitions. As the cases of discontinuities not being counted only occur if there were any epochs with PA solution and are less frequent than the number of discontinuities counted twice, the AWK script 'count_samples_fixed_window.awk' will count both cases and provide the number of discontinuities to be subtracted from the global counter of discontinuities, as there will be more case of discontinuities counted twice. To run the script, execute the following command line:

```
cat ./201?/???/Discontinuities_List_???_201?.txt|awk -f
count_samples_fixed_window.awk > epochs_to_subtract_riskmar.txt
```

NOTES:

- For selecting all the hourly maps, change the filename in the previous command from './201?/???/Discontinuities_List_???_201?.txt' to './201?/???/Discontinuities_List_???_201?_??h.txt'
- For selecting some hourly maps (not necessarily continuous hours), a comma separated list of hours (padded with zeros) inside curly braces must be given in the filenames.

For instance, for selecting hours "1,3,16,17,18,23", the filename should be: './201?/???/Discontinuities_List_???_201?_{01,03,16,17,18,23}h.txt'

- The script 'count_samples_fixed_window.awk' will automatically recognize if files are daily or hourly and count discontinuities accordingly

The script 'count_samples_fixed_window.awk' creates an output file with the same format as the file 'RiskMar_added_no_header.txt'.

The third step is to subtract the discontinuities from the 'epochs_to_subtract_riskmar.txt' file (generated by the AWK) script to the total epoch counter in 'RiskMar_added_no_header.txt' file. It can be done with the following command:

```
awk -v totalepochs=63072000 -v windowsize=900 -v OFMT="%9.5f" 'NR==FNR
{numdiscont[$1" "$2]+=$4;numepochs[$1" "$2]+=$5;next} {numdiscont[$1"
"$2]-=$3} END{for (i in numdiscont) {print
i,numdiscont[i],numepochs[i]==0?0:totalepochs,numepochs[i]==0?1:windowsize*
numdiscont[i]/63072000}}' RiskMar_added_no_header.txt
epochs_to_subtract_riskmar.txt|awk '{printf "%6.2f %7.2f %9.5f %8d
%8d\n",$1,$2,$5,$3,$4}'|sort -g -k1 -k2 >
RiskMar_subtracted_alldiscontinuities_no_header.txt
```

NOTES:

- The size of the window must be set (in seconds). By default is 900. To change it, change the text 'windowsize=900' to another value.
For instance, to set the window size to 15, the text should be 'windowsize=15' (the default for aviation is 15 seconds, for maritime is 900 seconds or 15 minutes).
- The total number of epochs of the processed must be set (in this example it is two years, which are 63072000 epochs). To change it, change the text 'totalepochs=63072000' to another value.

The previous command line creates a new file in the gLAB format with all the values added, but with no header. Any header from any of the output files for maritime continuity risk maps can be used. The following command reads the header from the file "RiskMarMap_001_2017.txt" and copy it to the previous file:

```
head -3 ./2017/001/RiskMarMap_001_2017.txt|cat -
RiskMar_subtracted_alldiscontinuities_no_header.txt >
RiskMar_subtracted_alldiscontinuities_header.txt
```

The file created, 'RiskMar_subtracted_alldiscontinuities_header.txt' can now be processed by the gLAB plotting tool as a normal SBAS Maritime Continuity Risk plot file. The user will have to change the plot title to state that the file has data for the two years (using the '-t' parameter).

NOTE: To ensure the steps were done correctly, it is recommended to execute the previous commands twice. One time using only the daily files, and the second time using only the hourly files. The resulting files created by both executions should be equal, as the total number of discontinuities has to be equal independently if the total number of discontinuities are counted using daily or hourly files.

9.8 AWK SCRIPT FOR ADDING SEVERAL DAYS OF CONTINUITY RISK MAPS WITH SLIDING WINDOW

AWK script 'count_samples_sliding_window.awk' for counting the number of samples that have to be added to the total number of epochs with continuity risk when computing the total number of epochs with continuity risk for a multiple day period with a sliding window.

NOTES:

- Before running the script, the window size has to be set in the second line of the script.
- This script works in gLAB version 5.4.0 and onwards.

```
-----
BEGIN {
    WinSize=15 #Set the size of the window size (in seconds)
    #Default window size of SBAS Aviation: 15
    #Default window size of SBAS Maritime: 900
    FirstFile=0
    Firstday=1
    HourlyMap=0
}

$1=="#MINLAT" {
    #Check if we are joining hourly files
    if ($NF=="HOURLY") {
        HourlyMap=1
    }
    getline
    Hour23to0=0
    if (HourlyMap==1 && ($9!=DoY || $10!=Year)) {
        if (Hour==23 && $NF==0 && ($9==(DoY+1) || (DoY==1 && $10==(Year+1)))) {
            Offset=3600*23
            Hour23to0=1
            #Reset lastValidEpoch values that are previous to 23 hour
            for (i in lastValidEpochHour) {
                if (lastValidEpochHour[i]<23) {
                    lastValidEpoch[i]=0
                    lastValidEpochHour[i]=0
                }
            }
        }
    }
}
```

```
    }
  } else {
    #New day found. Don't take into account discontinuities
    #from previous day as when joining hourly maps we do not
    #join the whole day (except when hour 23 and hour 0 is used)
    FirstFile=0
    Firstday=1
    Offset=3600
    #Reset lastValidEpoch values
    for (i in lastValidEpoch) {
      lastValidEpoch[i]=0
      lastValidEpochHour[i]=0
    }
  }
} else if (HourlyMap==1 && $9==DoY && $10==Year) {
  if ((Hour+1)!=$NF) {
    #For the case we are joining hourly files but hourly files
    #are not consecutive hours
    FirstFile=0
    Firstday=1
    #Reset lastValidEpoch values
    for (i in lastValidEpoch) {
      lastValidEpoch[i]=0
      lastValidEpochHour[i]=0
    }
  }
  Offset=3600
} else if (HourlyMap==0) {
  if ($9==(DoY+1) || (DoY==1 && $10==(Year+1))) {
    #New day, which is the following one
  } else {
    #New day, but it is not a continuous day. Don't take into account
    #discontinuities from the previous days
    FirstFile=0
    Firstday=1
  }
}
```

```

        #Reset lastValidEpoch values
        for (i in lastValidEpoch) {
            lastValidEpoch[i]=0
            lastValidEpochHour[i]=0
        }
    }
}
DoY=$9
Year=$10
if (HourlyMap==1) {
    Hour=$NF
    TotalEpochs=3600*Hour
} else {
    TotalEpochs=86400
    Hour=0
    Offset=0
}
getline
for (i in TodayAlreadyComputed) {
    TodayAlreadyComputed[i]=0
}
FirstFile++
if (FirstFile>1) {
    Firstday=0
}
next
}

($3-3600*Hour)<WinSize && Firstday==0 {
    NumAffectedToday=$3-3600*Hour
    StablePeriod=$6
    NumToAffectYesterday=WinSize-NumAffectedToday
    if (Hour23to0==1) {
        LastEpochsAvailYesterday=86400-lastValidEpoch[$1" "$2]
    } else if (HourlyMap==1 && lastValidEpoch[$1" "$2]>TotalEpochs) {

```

```

    LastEpochsAvailYesterday=WinSize
  } else {
    LastEpochsAvailYesterday=TotalEpochs-lastValidEpoch[$1 " $2]
  }
  #Check that last epoch of previous day was not in a discontinuity
  if (TodayAlreadyComputed[$1 " $2]==0 && lastValidEpoch[$1 "
"$2]!=(TotalEpochs+Offset)) {
    #Check if the continuity duration was from the start of the file
    if (NumAffectedToday<WinSize) {
      if (LastEpochsAvailYesterday>0) {
        #Check that discontinuity started at epoch 0 or that is is the first
        #discontinuity found in the file
        if ((NumAffectedToday-StablePeriod)==0) {
          if (LastEpochsAvailYesterday>=NumToAffectYesterday) {
            TotalExtraAffected[$1 " $2]+=NumToAffectYesterday
          } else {
            TotalExtraAffected[$1 " $2]+=LastEpochsAvailYesterday
          }
        }
      }
    }
  }
  TodayAlreadyComputed[$1 " $2]=1
}

{
  lastValidEpoch[$1 " $2]=$4+1 #Add 1 as the available epoch starts the next epoch
  lastValidEpochHour[$1 " $2]=Hour
}

END {
  for (i in TotalExtraAffected) {
    split(i,coord)
    printf "%6.2f %7.2f %6d\n",coord[1],coord[2],TotalExtraAffected[i]
  }
}

```

```
}

```

9.9 AWK SCRIPT FOR ADDING SEVERAL DAYS OF CONTINUITY RISK MAPS WITH FIXED WINDOW

AWK script 'count_samples_fixed_window.awk' for counting the number of samples that have to be subtracted to the total number of discontinuities when computing the total number of discontinuities for a multiple day period with a fixed window.

NOTES:

- Before running the script, the window size has to be set in the second line of the script.
- This script works in gLAB version 5.4.0 and onwards.

```
-----
BEGIN {
    WinSize=900 #Set the size of the window size (in seconds)
    #Default window size of SBAS Aviation: 15
    #Default window size of SBAS Maritime: 900
    FirstFile=0
    Firstday=1
    HourlyMap=0
}

$1=="#MINLAT" {
    #For the case that the files has no discontinuity data
    if (NumLinesReadFile<=3) {
        for (i in SubtractDiscon) {
            if (SubtractDiscon[$1 " "$2]==1) {
                EpochAvailStartFile[$1 " "$2]=1
            }
        }
    }
    NumLinesReadFile=1
    #Check if we are joining hourly files
    if ($NF=="HOURLY") {
        HourlyMap=1
    }
}
```

```
getline
if (HourlyMap==1 && ($9!=DoY || $10!=Year)) {
  if (Hour==23 && $NF==0 && ($9==(DoY+1) || (DoY==1 && $10==(Year+1)))) {
    Offset=86400
  } else {
    #New day found. Don't take into account discontinuities
    #from previous day as when joining hourly maps we do not
    #join the whole day (except when hour 23 and hour 0 is used)
    FirstFile=0
    Firstday=1
    Offset=0
    #Reset lastValidEpoch values
    for (i in lastValidEpoch) {
      lastValidEpoch[i]=0
      DataAvailAndEndedWithDiscontinuity[i]=0
      SubtractDiscon[i]=0
      EpochAvailStartFile[i]=0
    }
  }
} else if (HourlyMap==1 && $9==DoY && $10==Year) {
  if ((Hour+1)!=NF) {
    #For the case we are joining hourly files but hourly files
    #are not consecutive hours
    FirstFile=0
    Firstday=1
    #Reset lastValidEpoch values
    for (i in lastValidEpoch) {
      lastValidEpoch[i]=0
      DataAvailAndEndedWithDiscontinuity[i]=0
      SubtractDiscon[i]=0
      EpochAvailStartFile[i]=0
    }
  }
}
Offset=0
} else if (HourlyMap==0) {
```

```

if ($9==(DoY+1) || (DoY==1 && $10==(Year+1))) {
    #New day, which is the following one
} else {
    #New day, but it is not a continuous day. Don't take into account
    #discontinuities from the previous days
    FirstFile=0
    Firstday=1
    #Reset lastValidEpoch values
    for (i in lastValidEpoch) {
        lastValidEpoch[i]=0
        DataAvailAndEndedWithDiscontinuity[i]=0
        SubtractDiscon[i]=0
        EpochAvailStartFile[i]=0
    }
}
}
DoY=$9
Year=$10
if (HourlyMap==1) {
    Hour=$NF
    TotalEpochs=3600*Hour
    MaxDisconDuration=3600
} else {
    TotalEpochs=86400
    MaxDisconDuration=86400
    Hour=0
    Offset=0
}
getline
for (i in TodayAlreadyComputed) {
    TodayAlreadyComputed[i]=0
}
FirstFile++
if (FirstFile>1) {
    Firstday=0
}

```



```

    }
    next
}

{
    NumLinesReadFile++
}

#If the first day has no available epochs, we need to check if the first
#day with valid data, the valid data starts at the first hour or day
(((($3-3600*Hour-$6)==0 || ($3-$6)==0)) && $6>0 {
    if (SubtractDiscon[$1" "$2]==1) {
        EpochAvailStartFile[$1" "$2]=1
    }
}

SubtractDiscon[$1" "$2]==1 && $5!=MaxDisconDuration {
    SubtractDiscon[$1" "$2]=0
}

($3-3600*Hour)==0 && Firstday==0 {
    #Discontinuities that are counted double are only in the case that previous day ends
    #with a discontinuity and the next day starts with a discontinuity. Also, the
    #discontinuity from the previous day must be one that did not last the hole day,
    #as in this case, the number of discontinuities (NUMDISCONT) is set to 0, as no PA
    #epochs were available, no discontinuity events occurred
    if (lastValidEpoch[$1" "$2]==(TotalEpochs+Offset) && TodayAlreadyComputed[$1"
"$2]==0 && $5!=MaxDisconDuration) {
        #if (HourlyMap==0 || (HourlyMap==1 && DataAvailAndEndedWithDiscontinuity[$1"
"$2]==1) ) {
            if (DataAvailAndEndedWithDiscontinuity[$1" "$2]==1) {
                if (EpochAvailStartFile[$1" "$2]==0) {
                    TotalExtraAffected[$1" "$2]++
                }
                SubtractDiscon[$1" "$2]=0
                EpochAvailStartFile[$1" "$2]=0
            }
        }
    }
}

```

```

    }
    DataAvailAndEndedWithDiscontinuity[$1" "$2]=0
  }
  TodayAlreadyComputed[$1" "$2]=1
}

{
  if ($5==MaxDisconDuration) {
    #if (Firstday==1) {
      if (lastValidEpoch[$1" "$2]!=86400) {
        #If the discontinuity lasts the whole day and the previous day ended without
        discontinuity,
        #and the next day starts with valid data, this discontinuity will not be counted
        #Therefore we need to subtract one from the total count when necessary
        #The check for if the following day starts with data is done later
        if ((lastValidEpoch[$1" "$2]!=(3600*Hour) && Hour>0) || (lastValidEpoch[$1" "$2]==0
        && Hour==0)) {
          SubtractDiscon[$1" "$2]=1
        }
      }
    }
    #Discontinuity lasts the hole day or hour
    if (HourlyMap==0) {
      lastValidEpoch[$1" "$2]=86400
    } else {
      lastValidEpoch[$1" "$2]=3600*(Hour+1)
    }
  } else if ((HourlyMap==1 && ($4+1)==(3600*(Hour+1))) || (HourlyMap==0 &&
  ($4+1)==86400)) {
    lastValidEpoch[$1" "$2]=$4+1 #Add 1 as the avail epoch starts the next epoch
    DataAvailAndEndedWithDiscontinuity[$1" "$2]=1
    HasHadDataAvail[$1" "$2]=1
  } else {
    lastValidEpoch[$1" "$2]=$4+1 #Add 1 as the avail epoch starts the next epoch
    DataAvailAndEndedWithDiscontinuity[$1" "$2]=0
    HasHadDataAvail[$1" "$2]=1
  }
}

```

```
}  
END {  
  for (i in SubtractDiscon) {  
    #If the last day(s) or hour(s) ends had discontinuity for the hole day/hour and the  
    #last available epoch was at the end of a day/hour, then the discontinuity that lasted  
    #all the hour(s) or day(s) is not accounted for, so it is necessary to subtract one  
    #to the total count so it is taken into account  
    if (SubtractDiscon[i]==1 && TotalExtraAffected[i]>0 && HasHadDataAvail[i]==1) {  
      TotalExtraAffected[i]--  
    }  
  }  
  for (i in TotalExtraAffected) {  
    split(i,coord)  
    printf "%6.2f %7.2f %6d\n",coord[1],coord[2],TotalExtraAffected[i]  
  }  
}
```


Bibliography

- Alonso MT (2023) Galileo Broadcast Ephemeris and Clock Errors, and Observed Fault Probabilities for ARAIM. PhD thesis
- Alonso MT, Ferigato C, Ibáñez D, Perrotta D, Rovira-García A, Sordini E (2021) Analysis of 'Pre-Fit' Datasets of gLAB by Robust Statistical Techniques. *Stats* 4(2):400–418, doi: 10.3390/stats4020026, URL <https://doi.org/10.3390/stats4020026>
- Aragon-Angel A, Rovira-García A, Arcediano-Garrido E, Ibáñez Segura D (2021) Galileo Ionospheric Correction Algorithm Integration into the Open-Source GNSS Laboratory Tool Suite (gLAB). *Remote Sensing* 13(2):191, doi: 10.3390/rs13020191, URL <https://doi.org/10.3390/rs13020191>
- Ashby, N (2003) Relativity in the Global Positioning System. *Living Reviews in Relativity* 6, doi: 10.12942/lrr-2003-1, URL <https://link.springer.com/article/10.12942/lrr-2003-1>
- Beutler G, Dach R, Hugentobler U, Montenbruck O, Weber G, Brockmann E (2014) GLONASS April - What Went Wrong? *GPS World* 25:14–16, URL <https://www.gpsworld.com/the-system-glonass-in-april-what-went-wrong/>
- Blanch J, Walter T, Enge P, Lee Y, Pervan B, Rippl M, Spletter A (2012) Advanced RAIM user Algorithm Description: Integrity Support Message Processing, Fault Detection, Exclusion, and Protection Level Calculation. In: *Proceedings of the 25th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2012)*, pp 2828–2849
- Buehler S, Ibáñez D (2020) Multi-band EGNOS file format description document. URL http://www.egnos-pro.esa.int/ems/EMS_UID_Multi-band.pdf, accessed: 2023-07-06
- Caamaño M (2022) Network-based ionospheric gradient monitoring to support ground based augmentation systems. PhD thesis, URL <http://hdl.handle.net/10803/674794>

- Cabinet Office, Government Of Japan (2021) Quasi-zenith satellite system interface specification satellite positioning, navigation and timing service (IS-QZSS-PNT-004). URL <https://qzss.go.jp/en/technical/ps-is-qzss/ps-is-qzss.html>, accessed: 2023-07-06
- Cabinet Office, Government Of Japan (2023a) QZSS constellation status. <https://sys.qzss.go.jp/dod/en/constellation.html>, accessed: 2023-07-06
- Cabinet Office, Government Of Japan (2023b) QZSS transmission signals. https://qzss.go.jp/en/overview/services/sv03_signals.html, accessed: 2023-07-06
- Cabinet Office, Government Of Japan (2023c) Quasi-Zenith satellite orbit (QZO). <https://qzss.go.jp/en/technical/technology/orbit.html>, accessed: 2023-07-06
- Crustal Dynamics Data Information System (2023) URL https://cddis.nasa.gov/Techniques/GNSS/GNSS_Overview.html
- CSNO (2017) BeiDou-SIS-ICD. Tech. rep., URL <http://en.beidou.gov.cn/SYSTEMS/ICD/201806/P020180608519640359959.pdf>, BeiDou Navigation Satellite System Signal in Space. Interface Control Document. Open Service Signal B1C (Version 1.0)
- CSNO (2019) BeiDou-SIS-ICD. Tech. rep., URL <http://en.beidou.gov.cn/SYSTEMS/ICD/201902/P020190227702348791891.pdf>, BeiDou Navigation Satellite System Signal in Space. Interface Control Document. Open Service Signal B1I (Version 3.0)
- CSNO (2023a) BeiDou constellation status. <http://www.csno-tarc.cn/en/system/constellation>, accessed: 2023-07-06
- CSNO (2023b) BeiDou system introduction. <http://www.csno-tarc.cn/en/system/introduction>, accessed: 2023-07-06
- Cygwin (2023) Cygwin project. URL <https://www.cygwin.com/index.html>, accessed: 2023-07-06
- DB-IP (2023) DB-IP: IP geolocation API and database. URL <https://db-ip.com>, accessed: 2023-07-06
- de Jonge, Paul and Tiberius, C (1998) The LAMBDA method for integer ambiguity estimation: implementation aspects. Delft Geodetic Computing Centre LGR Series 12
- EC (2016) Ionospheric Correction Algorithm for Galileo Single Frequency Users. URL https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_Ionospheric_Model.pdf

- EC (2017) European Union Public Licence v1.2. URL <https://eupl.eu>, accessed: 2023-07-06
- ECAC (2016) ECAC Member States. <https://www.ecac-ceac.org/about-ecac/member-states>
- ESA (2023) European Space Agency Community License - v2.4 Permissive (Type 3). URL <https://essr.esa.int/license/european-space-agency-community-license-v2-4-permissive-type-3>, accessed: 2023-07-06
- ESSP (2019) EGNOS Service Provision Yearly Report (April 2018-March 2019). URL https://egnos-user-support.essp-sas.eu/sites/default/files/documents/EGNOS%20Service%20Provision%20Yearly%20Report%202018%20-%202019_0.pdf
- EU (2017) EGNOS OS SDD version 2.3. URL https://egnos-user-support.essp-sas.eu/sites/default/files/documents/egnos_os_sdd_in_force.pdf
- EU (2021) Galileo Open Service Signal In Space Control Document (OS SIS ICD), Issue 2.0. URL https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf, accessed: 2023-07-06
- EUROCAE (2017) Minimum Operational Performance Standards for Galileo / Global Positioning System / Satellite-based Augmentation System Airborne Equipment. EUROCAE Document ED-259 Version 0.3.13
- EUROCAE (2018) Minimum Operational Performance Standards for Galileo / Global Positioning System / Satellite-based Augmentation System Airborne Equipment. EUROCAE Document ED-259 Version 0.8
- EUROCONTROL (2001) PEGASUS. URL <https://www.eurocontrol.int/tool/pegasus>
- European Organization For Nuclear Research, OpenAIRE (2013) Zenodo. doi: 10.25495/7GXXK-RD71, URL <https://www.zenodo.org/>
- European Union (2016) European GNSS (Galileo) Open Service-Ionospheric Correction Algorithm for Galileo Single Frequency Users v1.2. URL https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_Ionospheric_Model.pdf, accessed: 2023-07-06
- EUSPA (2019) EGNOS Maritime Service Performance Assessment and Test Campaign. EUSPA Specific Contract 1 under framework contract GSA/OP/12/16/Lot1 "Integration of Fundamental Elements", Task 2-6, 2017-2019.

- EUSPA (2022a) About EGNOS. URL <https://egnos-user-support.essp-sas.eu/egnos-system/about-egnos>, accessed: 2023-07-06
- EUSPA (2022b) EO & GNSS market report. Luxembourg, doi: 10.2878/94903, URL https://www.euspa.europa.eu/sites/default/files/uploads/euspa_market_report_2022.pdf
- EUSPA (2022c) What is SBAS? URL <https://www.euspa.europa.eu/european-space/eu-space-programme/what-sbas>, accessed: 2023-07-06
- EUSPA (2023a) Galileo constellation status. <https://www.gsc-europa.eu/system-service-status/constellation-information>, accessed: 2023-07-06
- EUSPA (2023b) Galileo programme. <https://www.euspa.europa.eu/european-space/galileo/programme>, accessed: 2023-07-06
- EUSPA (2023c) Galileo services. <https://www.gsc-europa.eu/galileo/services>, accessed: 2023-07-06
- EUSPA (2023d) Galileo High Accuracy Service (HAS). URL <https://www.gsc-europa.eu/galileo/services/galileo-high-accuracy-service-has>, accessed: 2023-07-06
- FAA SatNav News (2007) WAAS now available in canada and mexico. URL https://www.abreco.com.mx/gps/satnavnews_november2007.pdf
- Federal Communications Commission (2022) FCC Online Table of frequency allocations. <https://transition.fcc.gov/oet/spectrum/table/fcctable.pdf>, accessed: 2023-07-06
- Fernández I, Damy S, Susi M, Martini I, Winkel J, Cancela S, Chamorro A, Calle D, Javier de Blas F, Simón J, Blonski D, Ibáñez D (2023) Galileo Authentication and High Accuracy: Getting to the Truth. Inside GNSS (February 2023) URL <https://insidegnss.com/galileo-authentication-and-high-accuracy-getting-to-the-truth>
- gAGE/UPC (2020) gLAB changelog. URL <https://gage.upc.edu/en/learning-materials/software-tools/glab-tool-suite/glab-release-notes>, accessed: 2023-07-06
- GAWK (2023) GNU AWK. URL <https://www.gnu.org/software/gawk/manual/gawk.html>, accessed: 2023-07-06
- GNU (1997) GNU Profiler. URL https://ftp.gnu.org/old-gnu/Manuals/gprof-2.9.1/html_mono/gprof.html, accessed: 2023-07-06

- GNU (2023a) Declaring attributes of functions. URL <https://gcc.gnu.org/onlinedocs/gcc/Function-Attributes.html>, accessed: 2023-07-06
- GNU (2023b) GNU Compiler Collection. URL <https://gcc.gnu.org>, accessed: 2023-07-06
- GNU (2023c) Specifying attributes of variables. URL <https://gcc.gnu.org/onlinedocs/gcc/Variable-Attributes.html>, accessed: 2023-07-06
- GPSgov (2023a) GPS Performance. URL <https://www.gps.gov/systems/gps/performance/>, accessed: 2023-07-06
- GPSgov (2023b) GPS Space Segment. URL <https://www.gps.gov/systems/gps/space>, accessed: 2023-07-06
- GPS World Staff (2018) Japan's QZSS service now officially available. GPS World URL <https://www.gpsworld.com/japans-qzss-service-now-officially-available>
- Gunning K, Walter T, Enge P (2017) Characterization of GLONASS broadcast clock and ephemeris: Nominal performance and fault trends for ARAIM. doi: 10.33012/2017.14926, URL https://www.researchgate.net/publication/313879576_Characterization_of_GLONASS_Broadcast_Clock_and_Ephemeris_Nominal_Performance_and_Fault_Trends_for_ARAIM
- Hilla S (2016) The extended standard product 3 orbit format (SP3-d). URL <https://files.igs.org/pub/data/format/sp3d.pdf>
- IAC (2023) GLONASS constellation status. <https://www.glonass-iac.ru/en/sostav0G>, accessed: 2023-07-06
- IALA (2015) Guideline Number 1112 On Performance and Monitoring of DGNSS Services in the Frequency Band 283.5-325 KHz. URL <https://www.iala-aism.org/product/performance-and-monitoring-of-dgnss-services-in-the-frequency-band-283-5-325khz-1112/?download=true>, accessed: 2023-07-06
- Ibáñez D (2014) Implementation of GNSS ionospheric models in gLAB. Master's thesis, UPC, Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona, URL <http://hdl.handle.net/2099.1/24770>
- Ibáñez D (2016) gLAB upgrade with EGNOS data processing. Master's thesis, UPC, Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona, URL <http://hdl.handle.net/2117/88479>
- Ibáñez D, Rovira-García A, Sanz J, Juan J, Gonzalez-Casado G, Jimenez-Baños D, López-Echazarreta C, Lapin I (2018) The GNSS Laboratory

- Tool Suite (gLAB) updates: SBAS, DGNSS and Global Monitoring System. In: Proceedings of the 9th ESA Workshop on Satellite Navigation Technologies (NAVITEC 2018), ESA/ESTEC, Noordwijk, The Netherlands, doi: 10.1109/NAVITEC.2018.8642707, URL <https://ieeexplore.ieee.org/document/8642707>
- Ibáñez D, Rovira-Garcia A, Sanz J, Juan J, González-Casado G, Alonso M, López-Salcedo JA, Jia H, Pancorbo FJ, García C, Martín I, Rodrigo A, López M (2019) A Kinematic campaign to evaluate EGNOS 1046 maritime service, ION Publications (The Institute of navigation), pp 840–854. doi: 10.33012/2019.16941, URL <https://doi.org/10.33012/2019.16941>
- Ibáñez D, Rovira Garcia A, Alonso MT, Sanz J, Juan JM, González Casado G, López Martínez M (2020) EGNOS 1046 Maritime Service Assessment. *Sensors* 20(1):276, doi: 10.3390/s20010276, URL <https://doi.org/10.3390/s20010276>
- ICAO (2006) International Standards and Recommended Practices, Annex 10 to the Convention on International Civil Aviation, Volume 1 Radio Navigation Aids
- ICAO (2015) Report of the fifth meeting of Ionospheric Studies Task Force (ISTF/5), Okinawa, Japan. URL <https://www.icao.int/APAC/Meetings/2015%20ISTF5/ISTF5%20Report%20-%20final.pdf>
- IERS (2006) SINEX - solution (software/technique) independent exchange format version 2.02. URL https://www.iers.org/SharedDocs/Publikationen/EN/IERS/Documents/ac/sinex/sinex_v202_pdf.pdf?__blob=publicationFile&v=2, accessed: 2023-07-06
- IMO (1974) International Convention for the Safety of Life at Sea (SOLAS). URL <https://www.ifrc.org/docs/idrl/I456EN.pdf>, accessed: 2023-07-06
- IMO (2011) Resolution A.1046(27) World-Wide Radio Navigation System (WWRNS); International Maritime Organization Assembly. URL [https://wwwcdn.imo.org/localresources/en/KnowledgeCentre/IndexofIMOResolutions/AssemblyDocuments/A.1046\(27\).pdf](https://wwwcdn.imo.org/localresources/en/KnowledgeCentre/IndexofIMOResolutions/AssemblyDocuments/A.1046(27).pdf), accessed: 2023-07-06
- Intel (2022) Intel Core i9-12900KS Processor Specifications. URL <https://ark.intel.com/content/www/us/en/ark/products/225916/intel-core-i912900ks-processor-30m-cache-up-to-5-50-ghz.html>
- Intel (2023) Intel one API DPC++/C++ Compiler. URL <https://www.intel.com/content/www/us/en/developer/tools/oneapi/dpc-compiler.html>, accessed: 2023-07-06

- International GNSS Service Data (2023) URL <https://igs.org/data>
- ISO/IEC (1999) ISO/IEC 9899:1999 (E). URL https://www.dii.uchile.cl/~daespino/files/Iso_C_1999_definition.pdf, accessed: 2023-07-06
- ISRO (2017) The IRNSS signal-in-space interface control document ICD, version 1.1. URL https://www.isro.gov.in/media_isro/pdf/SateliteNavigation/irnss_sps_icd_version1.1-2017.pdf, accessed: 2023-07-06
- ISRO (2018) PSLV-C41 Successfully Launches IRNSS-1I Navigation Satellite. <https://www.isro.gov.in/2018press11.html>, accessed: 2023-07-06
- ISRO (2022) The IRNSS Signal-in-Space Interface Control Document ICD. <https://www.isro.gov.in/IRNSSSignal.html>, accessed: 2023-07-06
- ISRO (2023) IRNSS constellation status. https://en.wikipedia.org/wiki/Indian_Regional_Navigation_Satellite_System#List_of_satellites, accessed: 2023-07-06
- JagranJosh (2015) GPS Aided Geo Augmented Navigation (GAGAN) System fully operationalised. URL <https://www.jagranjosh.com/current-affairs/gps-aided-geo-augmented-navigation-gagan-system-fully-operationalised-1432719480-1>
- Johnston G, Riddell A, Hausler G (2017) The International GNSS Service, 1st edn. Springer, doi: 10.1007/978-3-319-42928-1, URL <https://link.springer.com/book/10.1007/978-3-319-42928-1>
- Juan J, Sanz J, Rovira-Garcia A, González-Casado G, Ibáñez D, Orús R (2018) AATR an ionospheric activity indicator specifically based on GNSS measurements. *Journal of Space Weather and Space Climate* 8(A14):1–11, doi: 10.1051/swsc/2017044, URL <https://doi.org/10.1051/swsc/2017044>
- Juan J, Sanz J, González-Casado G, Rovira-García A, Timote C, Orús R (2022) Applying the geodetic detrending technique for investigating the consistency of GPS I2p(y) in several receivers. *Journal of geodesy* 96(11, article 85), doi: 10.1007/s00190-022-01672-3, URL <http://hdl.handle.net/2117/376213>
- Juan JM, Hernández-Pajares M, Sanz J, Ramos-Bosch P, Aragon-Angel A, Orús R, Ochieng W, Feng S, Jofre M, Coutinho P, Samson J, Tossaint M (2012) Enhanced precise point positioning for GNSS users. *IEEE Transactions on Geoscience and Remote Sensing* 50(10):4213–4222, doi: 10.1109/TGRS.2012.2189888

- Kalman RE (1960) A New Approach to Linear Filtering and Prediction Problems. Transactions of the ASME–Journal of Basic Engineering 82(Series D):35–45, URL <https://www.cs.unc.edu/~welch/kalman/media/pdf/Kalman1960.pdf>
- Klausmeier, E (2021) Filtering Bots and Crawlers from Access.log. URL <https://eklausmeier.goip.de/blog/2021/12-13-filtering-bots-and-crawlers-from-access-log/>
- Klobuchar J (1987) Ionospheric Time-Delay Algorithms for Single-Frequency GPS Users. *IEEE Transactions on Aerospace and Electronic Systems* AES-23(3):325–331, doi: 10.1109/TAES.1987.310829, URL <https://doi.org/10.1109/TAES.1987.310829>
- Lu J, Guo X, Su C (2020) Global capabilities of BeiDou Navigation Satellite System. *Satellite Navigation*, August 2020 doi: 10.1186/s43020-020-00025-9, URL <https://doi.org/10.1186/s43020-020-00025-9>
- Magdaleno S, de la Casa C, Lacarra E, López M (2019) EGNOS SIS performance based on IMO Res. A.1046. IALA input paper ENG8-11.7
- Mannucci AJ, Wilson BD, Yuan DN, Ho CH, Lindqwister UJ, Runge TF (1998) A global mapping technique for GPS-derived ionospheric total electron content measurements. *Radio Science* 33(3):565–582, doi: 10.1029/97RS02707, URL <https://doi.org/10.1029/97RS02707>
- MathWorks (2023) MATLAB program. URL <https://www.mathworks.com/products/matlab.html>
- McCarthy D, Petit G (2004) IERS Conventions (2003). IERS Technical Note 32. IERS Convention Center., Frankfurt am Main
- mCoding (2021) The ONLY C keyword with no C++ equivalent. URL <https://www.youtube.com/watch?v=TBGu3NNpF1Q>
- Microsoft (2023) Visual Studio C/C++ Compiler. URL <https://visualstudio.microsoft.com/vs/features/cplusplus/>, accessed: 2023-07-06
- Mitchell J, Rubinov E, Marshall C (2019) SBAS test-bed demonstration project. URL <https://frontiersi.com.au/wp-content/uploads/2018/08/SBAS-Test-bed-Overview-Report.pdf>, accessed: 2023-07-06
- Montenbruck O, Hauschild A, Steigenberger P, Langley RB (2010) The System: Three's the Challenge. *GPS World* URL <https://www.gpsworld.com/threes-challenge/>

- Montenbruck O, Steigenberger P, Wang N, Hauschild A (2022) Characterization and performance assessment of BeiDou-2 and BeiDou-3 satellite group delays. *NAVIGATION: Journal of the Institute of Navigation* 69(3), doi: 10.33012/navi.526, URL <https://navi.ion.org/content/69/3/navi.526>, <https://navi.ion.org/content/69/3/navi.526.full.pdf>
- MSC (2015) Performance Standards for Multi-System Shipborne Radionavigation Receivers, Annex 17, Resolution MSC.401(95). URL [https://wwwcdn.imo.org/localresources/en/KnowledgeCentre/IndexofIMOResolutions/MSCResolutions/MSC.401\(95\).pdf](https://wwwcdn.imo.org/localresources/en/KnowledgeCentre/IndexofIMOResolutions/MSCResolutions/MSC.401(95).pdf), accessed: 2023-07-06
- National Weather Service (2015) St. Patrick's Day 2015 Geomagnetic Storm. URL https://sites.nationalacademies.org/cs/groups/ssbsite/documents/webpage/ssb_160880.pdf
- Navipedia (2011a) EGNOS General Introduction. http://www.navipedia.net/index.php/EGNOS_General_Introduction
- Navipedia (2011b) MSAS General Introduction. http://www.navipedia.net/index.php/MSAS_General_Introduction
- Navipedia (2011c) WAAS General Introduction. http://www.navipedia.net/index.php/WAAS_General_Introduction
- OpenMP (2023) Open Multi-Processing (OpenMP). URL <https://www.openmp.org>, accessed: 2023-07-06
- PennState College of Earth and Mineral Sciences (2016) GLONASS Satellites and Signals. URL <https://www.e-education.psu.edu/geog862/node/1874>, accessed: 2023-07-06
- Pi X, Mannucci A, Lindqwiste U, Ho C (1997) Monitoring of global ionospheric irregularities using the Worldwide GPS Network. *Geophysical Research Letters* 24(18):2283–2286, doi: 10.1029/97GL02273, URL <https://doi.org/10.1029/97GL02273>
- Prange L, Arnold D, Dach R, Kalarus MS, Schaer S, Stebler P, Villiger A, Jäggi A (2020) CODE product series for the IGS MGEX project. doi: 10.7892/boris.75882.3, URL <http://dx.doi.org/10.7892/boris.75882.3>
- Rawer K (1963a) Propagation of Decameter Waves (HF-band) in Meteorological and Astronomical Influences on Radio Wave Propagation. Ed. Landmark, B. Pergamon Press, New York
- Rawer K (1963b) Propagation of decameter waves (h.f. band). In: Landmark B (ed) *Meteorological and Astronomical Influences on Radio Wave Propagation*, p 221, URL <https://ui.adsabs.harvard.edu/abs/1963mair.conf..221R>, provided by the SAO/NASA Astrophysics Data System

- Ray J, Gurtner W (2010) RINEX extensions to handle clock information. URL https://files.igs.org/pub/data/format/rinex_clock302.txt, accessed: 2023-07-06
- Riani M, Perrotta D, Torti F (2012) FSDA: A MATLAB toolbox for robust analysis and interactive data exploration. *Chemometrics and Intelligent Laboratory Systems* 116:17–32, doi: 10.1016/j.chemolab.2012.03.017, URL <https://www.sciencedirect.com/science/article/pii/S0169743912000974>
- RISDE (2008) GLONASS ICD v5.1. Tech. rep., URL <http://gauss.gge.unb.ca/GLONASS.ICD.pdf>, accessed: 2023-07-06
- Romero I (2020) The Receiver Independent Exchange Format Version 3.05. URL <https://files.igs.org/pub/data/format/rinex305.pdf>, accessed: 2023-07-06
- Romero I (2021) The Receiver Independent Exchange Format Version 4.00. URL https://files.igs.org/pub/data/format/rinex_4.00.pdf, accessed: 2023-07-06
- Rostoker G (1972) Geomagnetic indices. *Reviews of Geophysics* 10(4):935–950, doi: 10.1029/RG010i004p00935, URL <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/RG010i004p00935>
- Rothacher M, Schmid R (2010) ANTEX: The antenna exchange format, version 1.4. URL <https://files.igs.org/pub/data/format/antex14.txt>, accessed: 2023-07-06
- Rousseeuw P (1984) Least median of squares regression. *Journal of the American Statistical Association* 79(388):871–880, doi: 10.1080/01621459.1984.10477105, URL <https://www.tandfonline.com/doi/abs/10.1080/01621459.1984.10477105>
- Rovira-Garcia A, Juan J, Sanz J, González-Casado G (2015) A Worldwide Ionospheric Model for Fast Precise Point Positioning. *Geoscience and Remote Sensing, IEEE Transactions on* 53(8):4596–4604, doi: 10.1109/TGRS.2015.2402598, URL <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7053952>
- Rovira-Garcia A, Juan J, Sanz J, González-Casado G (2016a) Fast Precise Point Positioning: A System to Provide Corrections for Single and Multi-frequency Navigation. *Navigation-Journal of the Institute of Navigation* doi: 10.1002/navi.148, URL <http://dx.doi.org/10.1002/navi.148>
- Rovira-Garcia A, Juan J, Sanz J, González-Casado G, Ibáñez D (2016b) Accuracy of ionospheric models used in GNSS and SBAS: methodology and

- analysis. *Journal of geodesy* 90:229–240, doi: 10.1007/s00190-015-0868-3, URL <http://hdl.handle.net/2117/84674>
- Rovira-Garcia A, Ibáñez D, Orús R, Juan J, Sanz J, González-Casado G (2019) Assessing the quality of ionospheric models through GNSS positioning error: methodology and results. *GPS solutions* 24:4:1–4:12, doi: 10.1007/s10291-019-0918-z, URL <https://doi.org/10.1007/s10291-019-0918-z>
- Rovira García A, Ibáñez Segura D, Li M, Alonso Alonso MT, Sanz Subirana J, Juan Zornoza JM, González Casado G (2022) gLAB hands-on education on satellite navigation, Universitat Politècnica de Catalunya. doi: 10.5821/conference-9788419184405.010, URL <http://hdl.handle.net/2117/368624>
- RTCA-MOPS (2006) Minimum Operational Performance Standards for Global Positioning System/Wide Area Augmentation System Airborne Equipment. RTCA Document 229-D
- RTCM SC-104 (2010) Radio Technical Commission for Maritime Services: Special Committee No.104. Recommended Standards for Differential Navstar GPS Service, Version 2.3, Paper 130-2010/SC104-STD
- RTCM SC-104 (2022) Radio Technical Commission for Maritime Services: Special Committee No.104. Recommended Standards for Differential Navstar GPS Service, Version 3.3, Paper RTCM Paper 141-2016-SC104-STD
- Sanz J, Juan M, Prats X (2003) EGNOS Tutorial. URL https://server.gage.upc.edu/TEACHING_MATERIAL/SLIDES/EGNOS_tutorial_gAGE.pdf
- Sanz J, Juan J, Hernández-Pajares M (2013a) *GNSS Data Processing, Vol. I: Fundamentals and Algorithms*. ESA Communications, ESTEC. TM-23/1., Noordwijk, the Netherlands, URL <https://gage.upc.edu/en/learning-materials/library/gnss-books/gnss-data-processing-book>
- Sanz J, Juan J, Hernández-Pajares M (2013b) *GNSS Data Processing, Vol. II: Laboratory Exercises*. ESA Communications, ESTEC. TM-23/2., Noordwijk, the Netherlands, URL <https://gage.upc.edu/en/learning-materials/library/gnss-books/gnss-data-processing-book>
- Sanz J, Juan J, Rovira-Garcia A, González-Casado G (2017) GPS differential code biases determination: methodology and analysis. *GPS solutions* 21(4):1549–1561, doi: 10.1007/s10291-017-0634-5, URL <https://link.springer.com/article/10.1007/s10291-017-0634-5>
- Sanz J, Juan J, López C, González-Casado G, Rovira-García A, Alonso M, Ibáñez D (2018) Automated GPS Signal-in-Space anomalies monitoring over more

- than 11 years, ION Publications (The Institute of navigation), pp 2172–2185. doi: 10.33012/2018.15853, URL <http://hdl.handle.net/2117/128470>
- SC-159 RF (2005) GNSS-based Precision Approach Local Area Augmentation System (LAAS) Signal-in-space Interface Control Document (ICD). RTCA, URL <https://books.google.es/books?id=dQWbngEACAAJ>
- Schaer S, Gurtner G, Feltens J (1998) IONEX: The IONosphere Map EXchange Format Version 1.0
- Schaer, S (2018) SINEX BIAS - solution (software/technique) independent exchange for GNSS biases version 1.00. URL https://files.igs.org/pub/data/format/sinex_bias_100.pdf, accessed: 2023-07-06
- Shi J, Gao Y (2014) A comparison of three PPP integer ambiguity resolution methods. GPS Solutions 18:519–528, doi: 10.1007/s10291-013-0348-2, URL <https://doi.org/10.1007/s10291-013-0348-2>
- Starlink (2023) Starlink aviation. URL <https://www.starlink.com/aviation>, accessed: 2023-07-06
- Suard N, Werner G, Estey L (2003) Proposal for a new RINEX-type exchange file for GEO SBAS broadcast data. URL https://files.igs.org/pub/data/format/geo_sbas.txt, accessed: 2023-07-06
- Tange O (2020) GNU parallel 20201022 ('samuelpaty'). doi: 10.5281/zenodo.4118697, URL <https://doi.org/10.5281/zenodo.4118697>, GNU Parallel is a general parallelizer to run multiple serial command line programs in parallel without changing them.
- TechTarget (2021) High availability (HA). URL <https://www.techtarget.com/searchdatacenter/definition/high-availability>, accessed: 2023-07-06
- The LLVM Project (2023) Clang Compiler. URL <https://clang.llvm.org/index.html>, accessed: 2023-07-06
- The Qt Company (2023) Qt Framework. URL <https://www.qt.io/product/framework>, accessed: 2023-07-06
- Timoté C (2023) Contributions to real-time monitoring of ionosphere using GNSS signals. PhD thesis
- Todd Walter (1999) WAAS MOPS: Practical Examples. URL https://www.researchgate.net/publication/2314663_WAAS_MOPS_Practical_Examples

- Torti F, Perrotta D, Atkinson A, Riani M (2012) Benchmark testing of algorithms for very robust regression: FS, LMS and LTS. *Computational Statistics & Data Analysis* 56(8):2501–2512, doi: 10.1016/j.csda.2012.02.003, URL <https://www.sciencedirect.com/science/article/pii/S0167947312000680>
- Tossaint M, Samson J, Toran F, Ventura-Traveset J, Hernández-Pajares M, Juan J, Sanz J, Ramos-Bosch P (2007) The Stanford-ESA Integrity Diagram: A New Tool for The User Domain SBAS Integrity Assessment. *NAVIGATION* 54(2):153–162, doi: <https://doi.org/10.1002/j.2161-4296.2007.tb00401.x>, URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2161-4296.2007.tb00401.x>
- U R Rao Satellite Centre (2019) IRNSS Indian Regional Navigation Satellite System. <https://www.ursc.gov.in/navigation/irnss.jsp>, accessed: 2023-07-06
- USA Navigation Center (2022) GPS Interface Specification IS-GPS-200. Revision N. Tech. rep., URL <https://www.gps.gov/technical/icwg/IS-GPS-200N.pdf>
- USA Navigation Center (2023) GPS constellation status. <https://www.navcen.uscg.gov/gps-constellation>, accessed: 2023-07-06
- Walter, T and Enge, P (1995) Weighted RAIM for Precision Approach. In: *Proceedings of the 8th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1995)*, pp 1995–2004
- Webb F, Zumberge J (1993) *An Introduction to GIPSY/OASIS-II*. Jet Propulsion Laboratory, Pasadena, CA, USA
- Werner G, Estey L (2007a) The Receiver Independent Exchange Format Version 2.11. URL <https://files.igs.org/pub/data/format/rinex211.txt>, accessed: 2023-07-06
- Werner G, Estey L (2007b) The Receiver Independent Exchange Format Version 3.00. URL <https://files.igs.org/pub/data/format/rinex300.pdf>, accessed: 2023-07-06
- Wikipedia (2023) ARM architecture family. URL https://en.wikipedia.org/wiki/ARM_architecture_family, accessed: 2023-07-06
- Wu J, Wu S, Hajj G, Bertiguer W, Lichten S (1993) Effects of Antenna Orientation on GPS Carrier Phase Measurements. *Manuscripta Geodaetica* 18:91–98
- Zhang Q, Liu X, Liu Z, Hu Z, Zhao Q (2022) Performance evaluation of BDS-3 ionospheric delay correction models (BDSK and BDGIM):

First year for full operational capability of global service. *Advances in Space Research* 70(3):687–698, doi: <https://doi.org/10.1016/j.asr.2022.05.009>, URL <https://www.sciencedirect.com/science/article/pii/S0273117722003702>

Zumberge JF, Heflin MB, Jefferson DC, Watkins MM, Webb FH (1997) Precise Point Positioning for the efficient and robust analysis of GPS data from large networks. *Journal of Geophysical Research: Solid Earth* 102(B3):5005–5017, doi: 10.1029/96JB03860, URL <http://dx.doi.org/10.1029/96JB03860>

Index

- 3D Error, [39](#), [46](#)
- AATR, [103](#)
- AATR and SBAS, [103](#)
 - AATR application, [104](#)
 - AATR definition, [103](#)
 - Planetary AATR index, [105](#)
- Acronyms list, [XXIII](#)
- Advanced gLAB algorithms, [59](#)
- ANTEX, [18](#), [27](#), [41](#), [81](#)
- Apache licence, [3](#), [15](#)
- APC, [41](#), [79](#)
- APV, [106](#)
- ARAIM, [133](#)
- ARM, [15](#)
- ARP, [36](#)
- Availability, [54](#)
- AWGN, [24](#)

- BDS, [71](#)

- CDMA, [11](#)
- Challenges in gLAB upgrade, [71](#)
- Clang, [138](#)
- CNAV1, [22](#)
- CODE, [41](#)
- CoM, [41](#)
- Conclusion, [129](#)
- Continuity fixed window, [92](#)
- Continuity risk sliding window, [54](#)
- Continuity sliding window, [93](#)
- CONUS, [49](#)
- Convergence time, [45](#), [66](#)

- CTI, [92](#)
- Cycle-slip detection, [78](#)

- DAT, [3](#)
- DGNSS, [2](#), [33](#), [133](#)
- Divergence Free, [73](#)
- DOP, [19](#), [31](#)
- DPC, [3](#)

- EGNOS, [92](#), [106](#)
- EGNOS for maritime, [91](#)
 - Maritime availability, [95](#)
 - Maritime continuity, [96](#)
 - Maritime positioning, [98](#)
 - Maritime requirements, [91](#)
 - Performance study, [94](#)
- EMS, [21](#)

- F/NAV, [17](#), [22](#)
- Fast-PPP, [2](#), [14](#), [25](#), [82](#), [133](#)
- FDMA, [11](#)
- FDSA, [117](#)
- Formal error, [45](#)
- Future Work, [133](#)

- gAGE, [13](#), [18](#)
- Galileo, [71](#)
- GBAS, [48](#)
- GCC, [138](#)
- GDOP, [39](#), [46](#)
- GEO, [29](#), [32](#), [71](#)
- Geometry Free, [73](#), [79](#)
- gLAB description, [13](#)
 - DPC, [26](#)

- gLAB positioning example, 28
- GUI, 16
- Main gLAB features, 14
- gLAB downloads per country, 155
- gLAB source code optimizations, 135
 - Avoiding multiple recomputations of the same elements, 136
 - Compiler flags, 142
 - Convert if chains to switch statements, 138
 - Join multiple if conditions in a single switch statement, 139
 - Line reading optimization, 143
 - Memory allocation, 141
 - Micro-optimizations, 148
 - Const keyword, 151
 - Inlining functions, 149
 - Register keyword, 149
 - Restrict keyword, 150
 - Variable declaration in for loop definition, 152
 - Variable declaration inside a loop, 153
 - Other possible optimizations, 153
 - Precise orbit and clocks management, 137
 - String generation for printing, 135
- gLAB standard algorithms
 - DGNSS, 57
 - PPP, 40
 - SBAS 1F, 46
 - SPP, 35
- gLAB state of the art algorithms
 - Fast-PPP, 64
 - SBAS DFMC, 67
 - Uncombined PPP, 59
- gLAB state-of-the-art algorithms, 35
- GLONASS, 71
- GMS, 100
- GNSS, 7
 - BDS, 8, 11, 35, 40
 - Galileo, 7, 11, 35, 40
 - GLONASS, 7, 11, 35, 40
 - GPS, 7, 10, 35, 40
 - IRNSS, 8, 11, 35, 40
 - QZSS, 8, 11, 35, 40
- GNSS Introduction, 7
- GPL, 3, 15
- GPS, 71
- GPS signal in space anomalies, 118
 - Introduction, 118
 - Methodology and data set, 118
 - Results, 119
- GUI, 3
- HAS, 45
- HDOP, 39, 46
- Horizontal Error, 39, 46
- I/NAV, 17, 22
- IALA, 92
- IAR, 133
- ICD, 40
- If chain, 138
- IGS, 29, 32
- IMO, 92
- IMP, 92
- Impact of gLAB tool, 123
- Intel oneAPI, 138
- Introduction
 - Architecture, 8
 - Segments, 9
- IOD, 17, 23
- IONEX, 37, 65
- Iono Geometry Free, 73, 79
- Ionosphere correction assessment, 108
 - Assessing ionosphere corrections, 108
 - Introduction, 108
- Ionosphere Free, 73, 79
- IRNSS, 71
- Kalman, 20

- Klobuchar, 37
- KML, 20
- LAMBDA, 133
- LLI, 27
- LMS, 117
- LNAV, 22
- Maritime, 92
- MATLAB, 117
- Measurement selection, 71
 - Selection Methodology, 73
- Melbourne-Wübbena, 73
- MSC, 92
- Multi day SBAS availability plots, 82
- Multi-day SBAS availability maps computation, 161
- Multi-threading, 86
- NEAD, 54, 93
- NeQuick-G, 37
- NeQuick-G integration into gLAB, 112
 - Integration, 112
 - Introduction, 112
- NEU, 31
- OpenMP, 15
- Optimization, 84
- Output prints, 80
- PDOP, 39, 46
- PEGASUS, 21
- Positioning error, 45
- PPP, 14, 65, 82, 133
- Prefit outlier detection, 116
- Purpose of the thesis, 1
- Qt, 13, 15, 33
- QZSS, 71
- RAIM, 116, 133
- Receiver APC correction, 88
- RINEX, 17, 22–24, 133
- RINEX-B, 21
- ROTI, 103
- RTCM, 17, 22, 27, 33, 133
- RTK, 133
- Saint Patrick storm, 106
- SBAS, 33, 48, 92, 105
 - Architecture, 50
 - Data, 48
 - Integrity, 52
 - Alarm Limit, 52
 - Protection Level, 52, 53
 - Modes, 51
 - Operation modes, 52
 - HMI, 52
 - MI, 52
 - Normal operation, 52
 - System unavailable, 53
 - Requirements, 46
 - Requirements Values, 51
 - Systems, 49
 - ANGA, 50
 - BDSBAS, 50
 - EGNOS, 49
 - GAGAN, 50
 - KASS, 50
 - MSAS, 49
 - SDCM, 49
 - SouthPAN, 50
 - WAAS, 49
- SBAS 1F, 2
- SBAS availability maps, 25, 27
- SBAS DFMC, 2
- Scientific studies with gLAB, 91
- Second Iono Free, 73
- Signal, 9
 - Components, 9
 - Carrier, 9
 - Frequencies, 10
 - Navigation data, 10
 - PRN, 9
 - Ranging code, 9
- SINEX, 32
- SINEX BIAS, 65, 133
- SNR, 19, 27

- SOLAS, [92](#)
- SP3, [17](#), [22](#), [23](#), [41](#)
- SPP, [14](#), [116](#)
- Standard development, [5](#)
- SVN, [27](#), [80](#)
- Switch statement, [138](#)

- TDOP, [39](#), [46](#)
- Thesis context, [3](#)
- Thesis introduction, [1](#)
- Thesis outline, [3](#)

- Thesis publications, [4](#)
- TTA, [92](#)

- UN, [92](#)
- Uncombined PPP, [2](#), [82](#)

- VDOP, [39](#), [46](#)
- Vertical Error, [39](#), [46](#)
- Visual Studio, [138](#)

- WAAS, [106](#)
- WIR, [33](#)

Research group of Astronomy and GEomatics (gAGE)
Universitat Politècnica de Catalunya (UPC)
23rd of October 2023

