Master thesis

Protection Levels for High Integrity Localization for Autonomous Driving

Anthony Welte

Supervisor: Philippe XU, Philippe BONNIFAIT
Tutor: Luc JAULIN

August 20, 2017

Contents

List of Figures

Abstract

To develop autonomous vehicle, many challenges need to be tackled. The localization system is essential to the rest of the algorithm. Therefore, it needs to be both accurate and reliable. Part of the localization engine is of course a Global Navigation Satellite System. Recently new methods such as Precise Point Positioning (PPP) have been developed to provide a more accurate localization. By using precise orbit and clock as well as estimate errors it removes part of the measurement errors. Those systems have shown to reach an accuracy comparable to RTK systems, but their application to low cost hardware is not established yet. To provide reliable localization, the integrity of the position needs to be considered. This is taken care of in part by computing Protection Level. Conventional methods, relying on the position standard deviation, have shown unreliability issues in urban area due to the multipath on GNSS signals not being considered. Several methods have been developed to solve this problem. The Isotropy Based Protection Level (IBPL) method provides protection levels for position estimated by Least Square Estimation, whereas the Kalman Integrated Protection Level (KIPL) method applies for Kalman filter based estimations. Both the simulation and experimental tests conducted show the robustness of those two methods to multipath. However, to obtain usable position estimates, efforts needs to be made to decrease estimation errors.

Keywords

Autonomous driving, GNSS, Integrity, PPP, Protection Level

Thanks

I would like to thank the entire Heudiasyc laboratory for welcoming me and integrating me to the project. I also thank the other interns with whom it was a delight to work with. Finally, I thank in particular Philippe XU and Philippe BONNIFAIT for offering me the opportunity to work on this project and for their guidance and helpful commentary all along this internship.

Chapter 1

Introduction

Autonomous driving will surely be one of the most important technological progress in the automotive industry. Early technologies such as drive assist and self parking have already shown that autonomous systems can improve and facilitate the driving experience. As a result every major car manufacturer is working on fully autonomous cars. However, a lot of technical challenges need to be solved to reach this goal.

Localization is one of those challenge. It is one of the main components used to take decisions, therefore it needs to satisfy strict requirements. The localization engine has to provide pose estimates, i.e., position and heading, with high accuracy and high integrity, meaning that it needs to be able to warn the user when its error exceed a certain level. To reach these goals, Global Navigation Satellite System (GNSS) will be essential. Their unique capacity to provide a global localization anywhere on Earth has made them an already widely used tool for all transport means. The civil aviation has faced similar challenges in terms of accuracy and integrity of the localization. Norms and methods have therefore been developed to solve those issues and used on commercial flights. However, the environment in which cars drive makes it hard to apply the same algorithms. Indeed, with the proximity to the ground new sources of error emerge. Multipath and Non-Line-of-Sight (NLOS) become significant sources of error in an urban environment. The algorithms developed for civil aviation are not robust to such disturbances and require new approaches.

The European Safety Critical Applications Positioning Engine (ESCAPE) aims to provide solutions to those problems. The project is initiated by the European Global Navigation Satellite Systems Agengy (GSA) and mobilize several private companies with different skills to produce and demonstrate a localization engine designed for autonomous driving applications. It will use of multi-constellation and multi-frequency GNSS. It will take advantage of the new Galileo Constellation and its commercial services to provide more accurate satellite information. Maps, cameras and other sensors will be used to improve the accuracy of the localization. The localization engine will also provide an integrity layer. My internship objective was to study, explain and test some of those solutions.

My work focuses on methods used or developed by the company GMV which will be responsible for the GNSS side of the ESCAPE localization engine. After researching the Precise Point Positioning method, most of my internship has been dedicated to integrity methods, specifically, ways to compute protection levels. After researching the Isotropy Based Protection Level (IBPL) and Kalman Integrated Protection Level (KIPL) methods, I have worked on finding out their mathematical justifications. Simulations and experimental tests have also been undertaken to verify the correctness and robustness of these methods.

Chapter 2

Global navigation satellite system

2.1 Overview

Global navigation Satellite System (GNSS) is based on constellations of satellites[6] providing a global localization solution to its users. Originally composed of a single constellation, the GPS constellation (American), it now includes GLONASS (Russian), and GALILEO (European). Each constellation is composed of at least 24 satellites distributed on three to six orbital planes. This positioning ensures that at least four satellites are always in view. The satellites contain synchronized atomic clocks that are used to broadcast very accurate times to the ground.

Computation of a position. To find their position, GNSS receivers use Time Difference of Arrival (TDOA). The satellites broadcast their clock data as well as orbit information. The receiver is therefore aware of the satellites emission time and their position at that time. Using the difference between the emission time and the receiver time, one can compute the range between the satellites and the receiver. However, since the receiver does not have precise clock the reception time is inaccurate and induces an error on the range obtained. This issue is resolved in the estimation process by estimating the error in addition of the 3 dimensional position.

Finally, the receiver position can be estimated by solving equation (2.1).

$$\mathbf{y} = h\left(x, y, z, d\right) = \begin{pmatrix} h_1\left(x, y, z, d\right) \\ h_2\left(x, y, z, d\right) \\ \vdots \\ h_m\left(x, y, z, d\right) \end{pmatrix}, \tag{2.1}$$

with

$$h_X\left(x, y, z, d\right) = \sqrt{\left(x - x^X\right)^2 + \left(y - y^X\right)^2 + \left(z - z^X\right)^2} + d,$$

where

$\mathbf{y}$	measured pseudo-ranges,
(x, y, z)	receiver coordinates,
$\left(x^X, y^X, z^X\right)$	satellite coordinates,
d	receiver clock error.

Additional features. GNSS has additional capabilities. More than one frequency are used by the satellites to transmit data. This enables multi-frequency capable receivers to correct for some measurement errors. Also, using Doppler effects on the signal, it is possible to estimate the speed of the receiver and to smooth the pseudorange measurements.

2.2 Galileo particularity

The Galileo project initiated by the European Union through the European Space Agency (ESA) and the European GNSS Agency (GSA) is set to be fully operational in 2020. It will come with unique features, some particularly interesting for autonomous driving applications.

- The GALILEO Commercial service will provide encrypted data broadcasted on the E6 band (1278,75 MHz). The service will be developed by service providers which will need to buy the rights to use the service. They will share a bandwidth of 448bit/sec (~80bit/sec per provider) which is enough to provide more precise orbits and clocks information thus reaching higher accuracy. Authentication data will also be sent allowing receivers to verify the origin of the signal thus countering spoofing attacks.
- An open Safety of Life (SOL) service will provide integrity messages and warnings when the signals are not guaranteed to follow the specifications.
- Public regulated navigation will provide government users with a more robust signal, less prone to jamming or interferences.
- Search and Rescue service will be able to detect emergency beacons and provide accurate localization to rescue teams. This service will also allow to send information to the beacon.

2.3 Classical observables correction

To obtain an accurate localization GNSS receiver take into account several effects. Indeed a lot of error sources can affect the GNSS signals and result in inaccurate measurements. This section focuses on effects that can be corrected by the receiver without other external information than those send by GNSS satellites. table 2.1 list the error sources that can affect the pseudoranges and phase measurements. Some errors are not corrected in classical receivers, either because they are too small to be relevant or because additional information would be required.

Table 2.1: Error sources and associated typical range of the errors.

Error source	Typical range error
Satellite clock (dt^s)	1m
Satellite Ephemeris (d_{orb})	1m
Group Delay (T_{gd})	2.5m
Ionosphere (d_{iono})	10m
Troposphere (d_{tropo})	1m
Relativity (d_{rel})	0.5m
Multi-path	0.5m
Phase wind-up (d_w)	0.01m

Equation 2.2 and 2.3 show the two measured values by a single antenna receiver and the effects that need to be dealt with to obtain an accurate position. Measuring the carrier phase is helpful because it can be measured more precisely, however, to be used, its ambiguity (N_1) needs to be resolved.

$$\rho_1 = R_r^s + c(dt_r - dt^s + T_{gd}) + d_{orb} + d_{trop} + d_{ion} + d_{rel} + \epsilon(\rho_1), \tag{2.2}$$

$$C_1 = R_r^s + c(dt_r - dt^s) + d_{orb} + d_{trop} - d_{ion} + d_{rel} + d_{w1} + \lambda N_1 + \epsilon(C_1), \tag{2.3}$$

where

ρ_1 is the measured pseudorange (m),
C_1 is the measured carrier phase (m),
R_r^s is the geometric range (m),
c is the speed of light (m/s),
dt_r is the receiver clock error (s),
dt^s is the satellite clock error (s),
T_{gd} is the Group Delay (s),
d_{orb} is the satellite orbit error (m),
d_{trop} is the tropospheric delay (m),
d_{ion} is the ionospheric delay (m),
d_{rel} is the relativistic effects (m),
d_{w1} is the phase wind-up (m) (see section 2.4.3),
λ is the wavelength of L1 (m),
N_1 is the phase ambiguity.

2.3.1 Satellite Group Delay

The group delay T_{gd} is a delay caused by the component the signal has to go through before being emitted. It depends on the frequency of the signal and of each satellite components. To correct this effect the group delay is sent by the GPS satellite. Until 1999, this delay was determined using factory measurements. Now, the Jet Propulsion Laboratory (JPL) provides up to date corrections broadcasted by the GPS satellites.

2.3.2 Relativistic effects

Because of the speed of the satellite relative to the receiver, relativistic effects have to be considered. Indeed, because of relativistic effects, the clock speed of the satellites and of the receivers are different which will eventually cause ever increasing errors. To correct this effect the clock speed of the satellite is adjusted in the factory to make sure that when the satellite is in orbit, its clock speed is the same as the receivers. Another error is due to the slight eccentricity of the satellite's orbit. This starts to become relevant when trying to obtain centimeter level accuracy. Therefore, the delay due to relativistic effects must be corrected using Equation (2.4).

$$d_{rel} = -\frac{2\mathbf{x_{sat}} \cdot \dot{\mathbf{x}}_{\mathbf{sat}}}{c}, \tag{2.4}$$

where $\mathbf{x_{sat}}$ and $\dot{\mathbf{x}}_{\mathbf{sat}}$ are the position and velocity of the satellite either in a Conventional Celestial Reference System (CRS) or in a Conventional Terrestrial Reference System (TRS).

2.3.3 Atmospheric delays

GPS localization is based on the travel time of signals sent by satellites, therefore the precision of the localization is greatly affected by the variation of speed at which the signals travel. The different atmospheric layers (figure 2.1) have an effect on this speed and can consequently affect the localization of the receiver. Two layers are considered when trying to compensate for atmospheric effects, the ionosphere and the troposphere.

2.3.3.1 The ionospheric effects

In order to compensate for the ionospheric effect, several methods exist.

Dual frequency solution Because of the dispersive property of the ionosphere, the refraction of a signal depends on its frequency. Therefore, by combining the measurement of GPS signals at two different frequencies, it is possible to remove most of the effect of the ionosphere. By applying Function (2.5) and (2.6), users of dual frequency receivers are able to remove ionospheric effects.

$$\rho_{ionofree} = \frac{f_1^2 \rho_1 - f_2^2 \rho_2}{f_1^2 - f_2^2}, \tag{2.5}$$

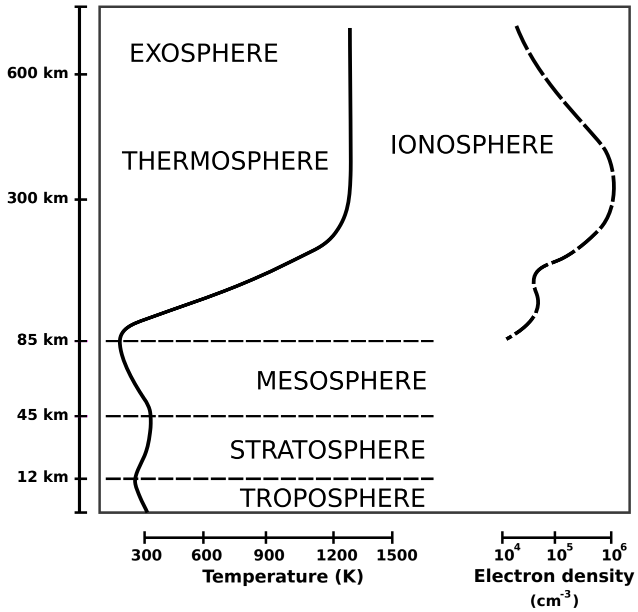


Figure 2.1: Atmospheric layers.

$$C_{ionofree} = \frac{f_1^2 C_1 - f_2^2 C_2}{f_1^2 - f_2^2}. \tag{2.6}$$

Ionospheric models When using a single frequency receiver, models have to be used to remove (or at least reduce) ionospheric effects. The following models exist :

- GRAPHIC

The most simple way to remove ionospheric delays is to use the GRoup And Phase Ionospheric Calibration (GRAPHIC). This method takes advantage of the opposing sign of the group ionospheric delay and phase ionospheric delay; by taking the mean of the two components, the delays cancel out.

$$G_1 = \frac{\rho_1 + C_1}{2} = R_r^s + cdt_r + \frac{T_{gd}}{2} + d_{trop} + \frac{d_{w1}}{2} + \frac{\lambda N_1}{2} + \epsilon(G_1). \tag{2.7}$$

However, this method has the disadvantage of introducing the carrier phase ambiguity in the equation, making the parameters harder to estimate. Indeed, [5] has shown that over half an hour is required for the estimated ambiguities to converge.

- The Klobuchar ionospheric model

The Klobuchar ionospheric model was designed to use as little memory and processing power as possible. That way its parameters can be broadcasted by GNSS satellites, and receivers can easily perform the necessary computations. It uses the user's latitude Φ_u and longitude λ_u and the satellite elevation angle E and azimuth A combined with 8 parameters (four α_n and four β_n) included in the GPS navigation message to compute the time delay caused by the ionosphere. The detailed algorithm is described in Appendix A.1. Using the time delay, it is trivial to compute the corresponding range.

2.3.3.2 The tropospheric delay

The signal is also delayed when going through the troposphere. This delay can be decomposed into two parts, a wet and a dry part. The dry part corresponds to the delay induced by the gas in the atmosphere (O_2, N_2). It depends on the receiver location, the pressure and the temperature of the atmosphere. Even though the pressure and temperature are hard to estimate, they cause very little variation making this component easily

Table 2.2: Niell Mapping Coefficient (hydrostatic).

Coefficient	Latitude (ϕ)				
ξ	15°	30°	45°	60°	75°
	Average				
a	$1.2769934e-3$	$1.2683230e-3$	$1.2465397e-3$	$1.2196049e-3$	$1.2045996e-3$
b	$2.9153695e-3$	$2.9152299e-3$	$2.9288445e-3$	$2.9022565e-3$	$2.9024912e-3$
c	$62.610505e-3$	$62.837393e-3$	$63.721774e-3$	$63.824265e-3$	$64.258455e-3$
	Amplitude				
a	0.0	$1.2709626e-5$	$2.6523662e-5$	$3.4000452e-5$	$4.1202191e-5$
b	0.0	$2.1414979e-5$	$3.0160779e-5$	$7.2562722e-5$	$11.723375e-5$
c	0.0	$9.0128400e-5$	$4.3497037e-5$	$84.795348e-5$	$170.37206e-5$
	Height Correction				
a_{ht}	$2.53e-5$				
b_{ht}	$5.49e-3$				
c_{ht}	$1.14e-3$				

modeled. To model this delay, the Zenith Hydrostatic Delay (ZHD) is computed using the Saastamoinen model, see equation (2.8).

$$ZHD = \frac{0.0022767P}{1 - 0.00266\cos(2\varphi) - 0.00028H}. \tag{2.8}$$

The Niell mapping function is commonly used to obtain d_{tropo}^{dry}:

$$M_{dry}(E, H) = m(E, a_d, b_d, c_d) + \Delta m(E, H), \tag{2.9}$$

with

$$\Delta m(E, H) = \left(\frac{1}{\sin(E)} - m(E, a_{ht}, b_{ht}, c_{ht})\right) \times H, \tag{2.10}$$

$$m(E, a, b, c) = \frac{1 + \frac{a}{1+\frac{b}{1+c}}}{\sin(E) + \frac{a}{\sin(E)+\frac{b}{\sin(E)+c}}}, \tag{2.11}$$

$$d_{tropo}^{dry} = M_{dry}(E, H) \times ZHD, \tag{2.12}$$

where

P	surface pressure (hPa),
φ	ellipsoidal latitude,
H	surface height above the ellipsoid (km),
E	elevation.

The a_d, b_d, c_d and a_{ht}, b_{ht}, c_{ht} are interpolated using table 2.2.

Concerning the wet part of the tropospheric delay, models exist, but they are not accurate enough. Therefore, the Zenith Wet Delay (ZWD) is usually estimated along with the position and the clock delay of the receiver and a mapping function as shown in equation (2.13).

$$d_{tropo}^{wet} = M_{wet} * ZWD, \tag{2.13}$$

$$M_{wet} = m(E, a_w, b_w, c_w), \tag{2.14}$$

where a_w, b_w, c_w are interpolated using table 2.3.

Table 2.3: Niell Mapping Coefficient (wet).

Coefficient	Latitude (ϕ)				
ξ	15°	30°	45°	60°	75°
a	$5.8021897e-4$	$5.6794847e-4$	$5.8118019e-4$	$5.9727542e-4$	$6.1641693e-4$
b	$1.4275268e-3$	$1.5138625e-3$	$1.4572752e-3$	$1.5007428e-3$	$1.7599082e-3$
c	$4.3472961e-2$	$4.6729510e-2$	$4.3908931e-2$	$4.4626982e-2$	$5.4736038e-2$

2.3.4 Carrier phase ambiguities

The carrier phase can be measured much more precisely than the pseudorange. Therefore, using it would increase the accuracy of the localization. However, unlike the code measurement, the carrier phase measurement is ambiguous. Indeed, to get the distance between the satellite and the receiver, it is not only required to measure the phase of the signal, the number of cycles the signal has gone through is also needed. This number is an integer, however, it is complicated to obtain such accurate value. Depending on the method used, these values for each satellite can be approximated [2] using equation (where n correspond to a number of measurements).

$$\hat{N} = \frac{\rho_1 - C_1 - cT_{gd} - 2d_{ion} + d_{w1}}{-\lambda_1}, \tag{2.15}$$

$$\bar{N} = \frac{\sum^n \hat{N}}{n}. \qquad \text{(where } n \text{ correspond to a number of measurements)}$$

This sliding average method has the advantage of providing fairly fast ambiguity resolution, however, it is not an accurate method and it might result in unusable data. For this reason PPP algorithm usually prefer to estimate these parameters in the Kalman filter. Estimating the ambiguities has the disadvantage of requiring a significant convergence time (see [7],[10]) before being able to use these values. Therefore the most accurate localization is only reached after around an hour of acquisition.

Moreover the loss of a satellite will result in the loss of the ambiguity resolution, thus a decrease in accuracy. This implies that losing a satellite will require the ambiguity estimation to restart for that satellite, which will again require time to converge.

2.4 Precise Point Positioning (PPP)

In order to obtain sub-meter precision, several methods exist. Real Time Kinematic (RTK) satellite navigation offers centimeter to millimeter level of precision, but rely on local correction send by a nearby reference station. Even though many stations exist, most areas are not covered by such systems. Precise Point Positioning (PPP) methods tackle that issue by using precise orbits and clock in order to obtain decimeter to centimeter levels of precision anywhere on Earth. These methods also take into account the disturbances caused by the atmosphere (ionosphere/troposphere) and correct other effects that introduce errors (table 2.1). To apply the necessary corrections, outside information is needed, that is why PPP services either require an Internet access or another satellite link to obtain the information in real-time.

2.4.1 Precise Orbit and Clock

To obtain precise orbit and clock of the satellites, PPP providers rely on a network of reference stations around the world that record the satellite signals and stream this data online. Servers centralize this information and estimate the precise position of the satellites. To do so, the algorithm takes into account several effects that disturb the measurements.

- First, it is needed to know the exact positions of the reference station. To obtain this information it is needed to compensate for tidal effects that move the reference station. Indeed, solid Earth tides have to be considered to ensure the precise positioning of the reference station. They can affect the position of the sites by up to 30 cm. Also, in coastal regions, ocean loading can affect the elevation of the ground

Table 2.4: IGS Satellite orbits and clocks products.

Type		**Accuracy**	**Latency**	**Sample Interval**
Broadcast	orbits	~100cm	real-time	daily
	clocks	~5ns RMS ~2.5ns SDev		
Ultra-Rapid (predicted half)	orbits	~5cm	real-time	15min
	clocks	~3ns RMS ~1.5ns SDev		
Ultra-Rapid (observed half)	orbits	~3cm	3-9 hours	15min
	clocks	~150ps RMS ~50ps SDev		
Rapid	orbits	~2.5cm	17-41 hours	15min
	clocks	~75ps RMS ~25ps SDev		5min
Final	orbits	~2.5cm	12-18 days	15min
	clocks	~75ps RMS ~25ps SDev		Sat: 30s Stn: 5min

and has to also be considered. The elevation of a reference station can change by a few centimeters due to this effect. To account for these effects, accurate models of the Earth deformation [9] are used.

- The phase measurement are affected by Phase Wind-up. This phenomenon occurs because of the satellite rotation relative to the receiver. Indeed, GPS satellites need to face the center of the Earth while at the same time having their solar panel facing the sun. This forces the satellites to change their attitude as they rotate around the Earth. This change of attitude can cause a phase shift, this effect is presented extensively in section 2.4.3.
- Finally, the satellite's antenna lever-arm has to be considered. Indeed, when trying to locate a satellite, the estimated position corresponds to its center of mass and not to its antenna. This distance can be around 1m and has to be considered using the satellite attitude to obtain precise orbits.

Other atmospheric disturbances have to be also considered.

Several services such as the International GNSS Service (IGS) offer predicted orbits and clock for real-time applications. The IGS provides different categories of products (see table 2.4) that vary in accuracy and latency. For real-time applications, the ultra-rapid products are available.

A file containing the orbits and clock for both the GPS and GLONASS constellations can be accessed on the FTP site of the IGS[1].

2.4.2 Ionospheric delay

To correct for the ionospheric delay, multi-frequency receivers are usually used. This method removes almost entirely the ionospheric delay. However, for single frequency receivers, the methods used by classical techniques cannot correct this delay as well. To solve this problem additional information can be used [14, 8], namely the Global Ionospheric Model (GIM) provides helpful data to correct ionospheric delay.

The Center for Orbit Determination in Europe (CODE) provides predicted maps of the ionosphere Total Electron Content (TEC) that can be used to estimate the delay caused by the ionosphere. figure 2.2 shows a representation of a map provided by CODE[2].

By interpolating the provided maps[11], an approximation of the total electron content at the Ionospheric Pierce Points (IPP, calculation in Appendix A.2) can be obtained. Using Equation 2.16 we obtain the ionospheric delay for a satellite at the zenith position. We can then use a mapping function (see Equation 2.17) that takes into account the increase path length due to the elevation of the satellite.

[1] `ftp://ftp.igs.org/pub/product/`

[2] `ftp://ftp.unibe.ch/aiub/CODE/`

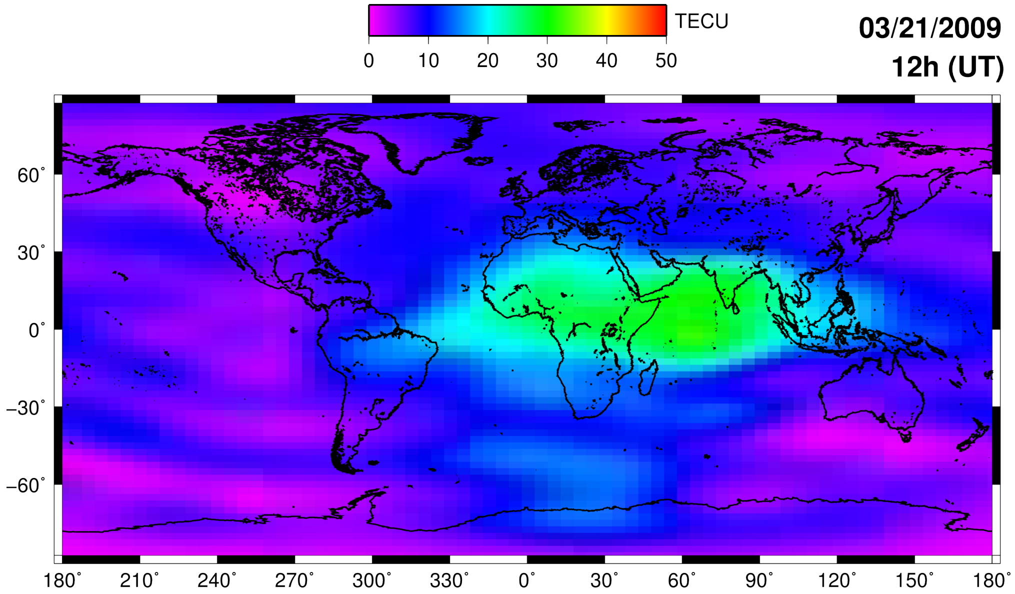


Figure 2.2: Representation of a TEC map provided by CODE.

$$d_{ion,zenith} = \frac{40.3}{f^2} \times TEC(\lambda, \beta, t), \tag{2.16}$$

$$F_{pp} = \left(1 - \left(\frac{R_{earth}\cos(\phi)}{R_{earth} + h_I}\right)\right)^{-\frac{1}{2}}. \tag{2.17}$$

Taking into account the elevation, the ionospheric path delay becomes

$$d_{ion} = F_{pp}\frac{40.3}{f^2} \times TEC(\lambda, \beta, t), \tag{2.18}$$

where

ϕ	elevation,
f	signal frequency (Hz),
R_{earth}	earth radius,
h_I	height of the maximum electron density (usually 350km).

2.4.3 Carrier phase wind-up

Carrier phase wind up has a small effect on the measurements. However, when trying to reach sub-decimeter precision this effect is sometimes considered.

Because of the polarized nature of the GPS signal, its phase is affected by the orientation of emitting and receiving antennas (see figure 2.3). Indeed, if one antenna rotates by 180°, the signal phase changes by the same amount. This error caused by this effect can be up to 10 cm (half of the wavelength) on the phase measurement. The satellite rotates in order to face the sun as much as possible, thus it slowly rotates around its antenna to be in the optimal orientation. This slow effect is important for the base station that monitors the satellite and helps provide the accurate orbits. However, for a fixed receiver turned on for a few hours this effect is usually negligible. In the case of a moving receiver, it might be interesting to correct this effect depending on the orientation of the receiver.

Using the relative orientation between the satellite and receiver and the satellite-receiver unit vector $\hat{k}$, the phase delay can be computed with equation (2.20).

$$d_{w1} = \frac{\Delta\phi}{2\pi}\lambda, \tag{2.19}$$

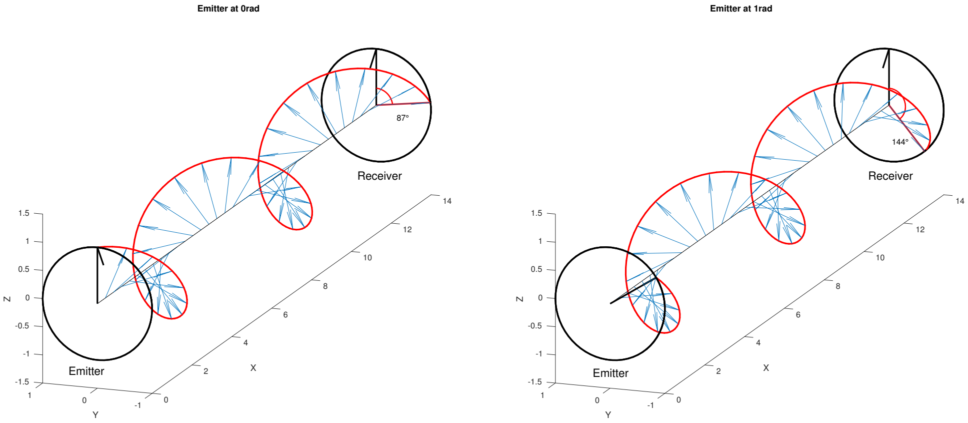


Figure 2.3: Wind-up effect.

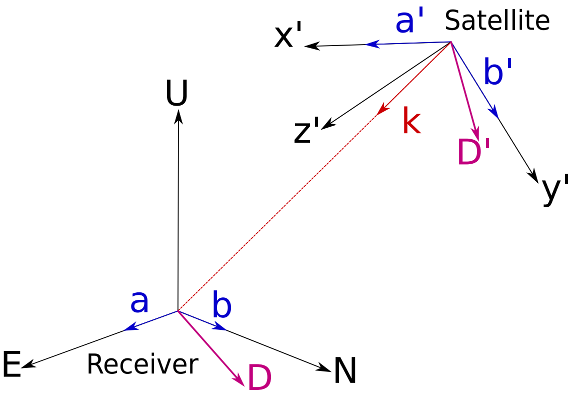


Figure 2.4: Wind Up Reference Frame

$$\Delta\phi = sign(\zeta)\arccos\left(\frac{\overrightarrow{D'}\cdot\overrightarrow{D}}{\left\|\overrightarrow{D'}\right\| * \left\|\overrightarrow{D}\right\|}\right), \tag{2.20}$$

with

$$\zeta = \hat{k}\cdot(\overrightarrow{D'}\times\overrightarrow{D}),$$

$$\overrightarrow{D} = \hat{a} - \hat{k}(\hat{k}\cdot\hat{a}) + \hat{k}\times\hat{b},$$

$$\overrightarrow{D'} = \hat{a}' - \hat{k}(\hat{k}\cdot\hat{a}') + \hat{k}\times\hat{b}',$$

where $\hat{k}$ is the unitary vector pointing from the satellite to the receiver; $\hat{a}$, $\hat{b}$ are unitary vectors pointing to the East and North; $\hat{b}'$ is the unitary vector orthogonal to $\hat{k}$ and pointing to the sun; and $\hat{a}' = \hat{b}'\times\hat{k}$ (see figure 2.4).

2.4.4 Practical implementation

Precise Point Positioning methods can be implemented in many ways depending on what is known, what is modeled, and what has to be estimated. Both Least Square Estimation and Kalman Filtering can be used to estimate the parameters. figure 2.5 shows the steps needed for one implementation of the PPP method.

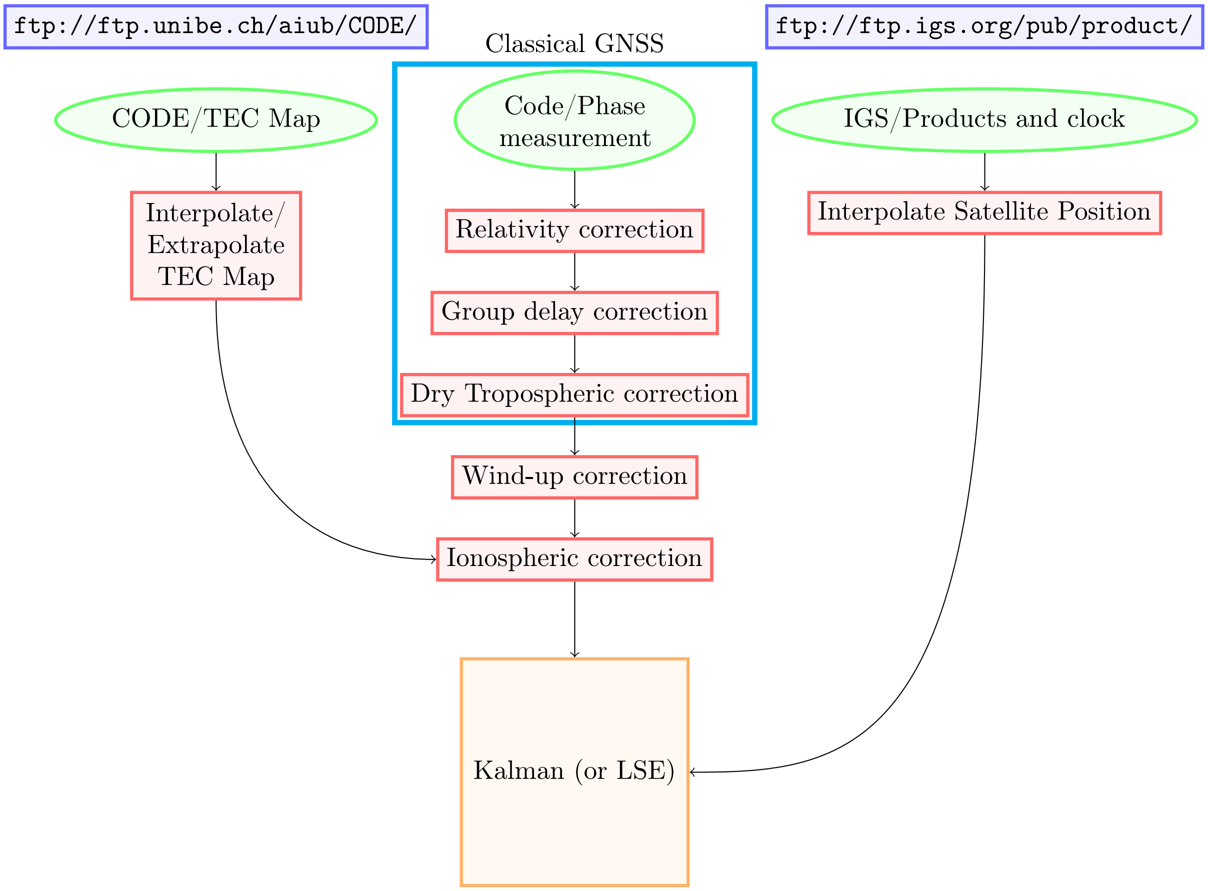


Figure 2.5: Possible PPP corrections steps.

Both the Least Square Estimation and the Kalman Filter require a linear system. The following equations describe the observation system linearized around $\begin{pmatrix} x_0 & y_0 & z_0 \end{pmatrix}^T$.

State vector:

$$X = \begin{pmatrix} dx & dy & dz & cdt_r & ZWD & N_1 & \dots & N_n \end{pmatrix}^T. \tag{2.21}$$

Observation/Design matrix:

$$H = \begin{bmatrix} \frac{x_0-x^1}{\rho^1} & \frac{y_0-y^1}{\rho^1} & \frac{z_0-z^1}{\rho^1} & 1 & M^1_{wet} & 0 & \dots & 0 & \dots & 0 \\ \frac{x_0-x^1}{\rho^1} & \frac{y_0-y^1}{\rho^1} & \frac{z_0-z^1}{\rho^1} & 1 & M^1_{wet} & \lambda & \dots & 0 & \dots & 0 \\ & & & & \vdots & & & & & \\ \frac{x_0-x^k}{\rho^k} & \frac{y_0-y^k}{\rho^k} & \frac{z_0-z^k}{\rho^k} & 1 & M^k_{wet} & 0 & \dots & 0 & \dots & 0 \\ \frac{x_0-x^k}{\rho^k} & \frac{y_0-y^k}{\rho^k} & \frac{z_0-z^k}{\rho^k} & 1 & M^k_{wet} & 0 & \dots & \lambda & \dots & 0 \\ & & & & \vdots & & & & & \\ \frac{x_0-x^n}{\rho^n} & \frac{y_0-y^n}{\rho^n} & \frac{z_0-z^n}{\rho^n} & 1 & M^n_{wet} & 0 & \dots & 0 & \dots & 0 \\ \frac{x_0-x^n}{\rho^n} & \frac{y_0-y^n}{\rho^n} & \frac{z_0-z^n}{\rho^n} & 1 & M^n_{wet} & 0 & \dots & 0 & \dots & \lambda \end{bmatrix}. \tag{2.22}$$

Observation matrix:

$$Y = \begin{pmatrix} R_1 & \Phi_1 & R_2 & \Phi_2 & \cdots & R_n & \Phi_n \end{pmatrix}^T = H * X. \tag{2.23}$$

2.4.5 PPP accuracy and limitation

2.4.5.1 Expected accuracy of PPP methods

The accuracy of PPP methods greatly depends on the setup used for the experiment. Indeed, dual frequency users [13, 12] report centimeter level accuracy after a convergence time of a few hours. Single frequency users [7, 15] however, report decimeter level accuracy in just a few epochs. Nevertheless it is a significant improvement compared to classical GNSS.

2.4.5.2 Limitations

PPP methods cannot solve every positioning problems yet.

Multipath and Non Line-of-Sight (NLOS). These methods do not provide any way to deal with multipath or non line-of-sight signals. These effects appear when signals bounce off structures. The receiver then either receives the signal once, when it has bounced of something (NLOS), or twice when it also receive the direct signal from the satellite (multipath). These errors can however be mitigated by increasing the number of available satellites, making any one satellite less critical to the localization. Using multiple constellations can therefore make the system less sensitive to this problem. Moreover, the new Galileo constellation offers a signal (E5) less prone to multipath by using a new modulation method.

Loss of signal. PPP methods are also very sensitive to loss of signals. Indeed, when a satellite or all satellites are lost (when going through a tunnel for instance) the convergence of the solution is lost and the process has to be restarted. The convergence time is due to the time it takes to estimate the tropospheric delay and the phase ambiguity. When losing the signal, the phase ambiguity has to be solved again and it again requires time to converge to a constant value. Dual frequency users are more affected by this effect because the loss of just one of the two frequencies causes the convergence to be lost. To be affected by the loss of the signal, dual frequency users can also run a single frequency PPP to be able to keep the convergence when one of the frequency is lost.

2.4.6 Conclusion

PPP methods offer improvements over classical GNSS localization. Using external information provide a significant advantages and allows to correct most of the measurement errors. Moreover the new Galileo constellation will offer services providing such information. This will surely increase the number of PPP users and help the development of new technologies such as autonomous driving.

However work still needs to be done concerning ambiguity resolution to obtain a system robust to loss of signals. Moreover, solution to deal with Multipath and Non Line-of-Sight are needed to use such techniques in urban environment where these sources of error are important.

To implement this method, tools have been searched. The RTKlib library[3] implement most of the aforementioned corrections, however, it does not yet provide real-time single-frequency PPP which is required with the current available hardware. Moreover, for the ESCAPE project, it was decided to use a multi-frequency receiver making PPP implementation much easier. The implementation of this method was therefore not undertaken, but should be done when more developed libraries are available.

[3] See `https://github.com/tomojitakasu/RTKLIB`

Chapter 3

Integrity methods

3.1 Introduction

Positioning Integrity

Definition. Integrity is a measure of the trust that can be placed on the correctness of the information provided by the navigation system. It includes the capacity of the system to provide timely warnings to the users when the system should not be used for navigation.[1]

Integrity is not to be mistaken for accuracy. Accuracy usually refers to smaller percentile (90%, 95%), whereas integrity is more interested in the far end of the tail of the error distribution ($1 - 10^{-5}$, $1 - 10^{-7}$). Accuracy focuses on guaranteeing that "most" values are close to the real value, whereas integrity focuses on guaranteeing that very few values are dangerously far from the actual values. Moreover, accuracy and integrity differ in their use-case. Accuracy is used to characterize an error distribution, whereas integrity is used to characterize the usability of the results by the system.

Integrity requirements of a system can be defined using several parameters:

- Alert Limit: the maximum error at which point the information cannot be used. In civil aviation, alert limits depend on the phase of flight and are of the order of several kilometers when cruising. A targeted alert limit for self driving application is of about one meter, thus guaranteeing localization on the right lane.
- Time to Alert: the maximum delay between the moment the system is out of tolerance and the moment the user is warned.
- The integrity risk or Target Integrity Risk (TIR): the probability for which integrity is not achieved. Applications that involve safety of life require a small Integrity Risk, typically 10^{-7} in civil aviation. Integrity in autonomous driving is still at an early stage, thus no Target Integrity Risk has been decided yet.
- Protection Level (PL): bound on the position error that satisfies the integrity risk. Since the position error cannot be known without knowing the actual position, a bound on this error is used. An alert is raised when the bound exceeds the alert limit. A protection level must satisfy $P\left(Error > PL\right) \leq TIR$.

Integrity Systems Integrity was originally developed for civil aviation applications and resulted in three architectures providing integrity information.

- Satellite-Based Augmentation Systems (SBAS): provide integrity-related information through dedicated satellites in geostationary orbit. Moreover the same satellites can be used has GNSS satellites, increasing positioning availability.

[1] this definition was proposed in [1]

- Ground-Based Augmentation Systems (GBAS): provide integrity-related information through high frequency broadcasts. These are used at airports to increase the positioning integrity (necessary to satisfy the smaller Alert Limits).
- Receiver Autonomous Integrity Monitoring (RAIM): integrity monitoring based on standalone GNSS data (built within the internal algorithms of the receiver), sometimes with additional embedded sensors.

During this internship, I focused on two RAIM algorithms that aim to compute protection levels. Both IBPL and KIPL are interesting for autonomous driving application since they claim to be robust to multipath and Non Line of Sight (NLOS) problems. These problems arise in ground applications, specifically in urban environment and are not considered by methods developed for the civil aviation. The following methods aim to provide protection levels that are robust to such measurement errors and are therefore usable for autonomous driving applications.

3.2 Isotropy Based Protection Level (IBPL)[3]

3.2.1 Introduction

3.2.1.1 Notation

Notation	Description	Dimension
$\mathbf{x}$	state vector	n
$\mathbf{y}$	observation vector	m
H	observation matrix	$m \times n$
$\boldsymbol{\epsilon}$	error on the observations (**unknown**)	m
$\hat{\mathbf{x}}$	state estimate vector	n
$\mathbf{r}$	residual of the least square estimation	m
$\mathbf{e}$	error on the estimation (**unknown**)	n
$\mathbf{s}$	image though H of $\mathbf{e}$ ($\mathbf{s} = H \cdot \mathbf{e}$)	m

The scalar product $\mathbf{x}^T \cdot \mathbf{x}$ will be noted $\mathbf{x}^2$.

3.2.1.2 Linearized localization problem

The Isotropy based Protection Level (IBPL) is a method used to compute protection level when estimating parameters using the Least Square Estimator. The Least Square Estimator is used by following these steps.

Given the linear observation equation:

$$\mathbf{y} = H \cdot \mathbf{x} + \boldsymbol{\epsilon}; \tag{observation}$$

the state estimate is computed as

$$\hat{\mathbf{x}} = \left(H^T H\right)^{-1} H^T \mathbf{y}. \tag{estimation}$$

The residual can be computed using the measurement and the estimated state as

$$\mathbf{r} = \mathbf{y} - H \cdot \hat{\mathbf{x}}. \tag{residual}$$

The estimation error can also be defined but is usually unknown in practical applications

$$\mathbf{e} = \hat{\mathbf{x}} - \mathbf{x}. \tag{error}$$

3.2.1.3 Properties of the errors

The Least Square Estimation has useful properties that link the measurement error $\boldsymbol{\epsilon}$, the residual $\mathbf{r}$ and $\mathbf{s} = H \cdot \mathbf{e}$ the image of the estimation error $\mathbf{e} = \hat{\mathbf{x}} - \mathbf{x}$ through H.

Proposition 1. $\boldsymbol{\epsilon} = \mathbf{r} + \mathbf{s}$.

Proposition 2. $\mathbf{s}^T \cdot \mathbf{r} = 0$.

Proposition 3. $\boldsymbol{\epsilon}^2 = \mathbf{r}^2 + \mathbf{s}^2$.

The proofs for these propositions can be found in Appendix B.2.

3.2.2 The main idea

The Isotropy based Protection Level (IBPL) is a method developed and patented by GMV. It is only intended to be used with **Least Square Estimation** methods specifically in satellite based localization systems.

- It is based on the **isotropy** hypothesis on the measurement error, meaning the measurement error vector can point in any direction of the measurement space with the same probability.
- Unlike existing methods it does not make any assumption on the distribution on individual measurements (no Gaussian assumption).

The protection level PL provided by this method is defined by equation (3.1).

$$PL = k \frac{\|\mathbf{r}\|}{\sqrt{\lambda_{min}^{H^T H}}} = k \, \|\mathbf{r}\| \, \sqrt{\lambda_{max}^{(H^T H)^{-1}}}, \tag{3.1}$$

where

k is a precomputed parameter (see Appendix B.1),

$\mathbf{r}$ is the residual vector of the least square estimation,

$\lambda_{min}^{H^T H}$ / $\lambda_{max}^{\left(H^T H\right)^{-1}}$ are respectively the smallest and largest eigenvalues of $H^T H$ and $(H^T H)^{-1}$.

Note:

- $\lambda_{min}^{H^T \cdot H}$ is the smallest eigenvalue of $H^T H$ therefore $\lambda_{max}^{\left(H^T \cdot H\right)^{-1}} = \frac{1}{\lambda_{min}^{H^T \cdot H}}$ is the biggest Eigenvalue of $\left(H^T H\right)^{-1}$(because all eigenvalues are positive).
- When searching for a Horizontal Protection Level the biggest eigenvalue of the horizontal components of the matrix $\left(H^T H\right)^{-1}$ can be used

$$\sqrt{\lambda_{hmax}^{(H^T \cdot H)^{-1}}} = \sqrt{\frac{h_{XX} + h_{YY}}{2} + \sqrt{\left(\frac{h_{XX} - h_{YY}}{2}\right)^2 + \left(\frac{h_{XY} + h_{YX}}{2}\right)^2}}, \tag{3.2}$$

with

$$\left(H^T H\right)^{-1} = \begin{bmatrix} h_{XX} & h_{XY} & h_{XZ} & \cdots \\ h_{YX} & h_{YY} & h_{YZ} & \cdots \\ h_{ZX} & h_{ZY} & h_{ZZ} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

3.2.3 Justification

During my internship, I work on finding the justification of this method based on the information from the patent[3]. The patent describes some of the main aspects of the method, without necessarily explaining the reasoning behind them. The following work is the result of my understanding of these aspects and of the justifications I have found for them.

Definition 4. Finding a protection level consists in finding a number PL which for a target integrity risk α satisfies:

$$P\left(\|\mathbf{e}\| \geq PL\right) \leq \alpha. \tag{3.3}$$

3.2.3.1 Rephrasing the problem

The first step is to write the problem using the same variables used in the Least Square Estimation (i.e., r and s). To achieve this, let us show that if there is a number k so that $P\left(\mathbf{s}^2 \geq k^2 \cdot \mathbf{r}^2\right) = \alpha$ ($\boldsymbol{s}$ being equal to $H \cdot \mathbf{e}$), then the protection level $PL = \frac{k \cdot \|\mathbf{r}\|}{\sqrt{\lambda_{min}^{H^T H}}}$ satisfies $P\left(\|\mathbf{e}\| \geq PL\right) \leq \alpha$.

Proof. Suppose that there is a number k so that $P\left(\mathbf{s}^2 \geq k^2 \cdot \mathbf{r}^2\right) = \alpha$.

$$P\left(\mathbf{s}^2 \geq k^2 \cdot \mathbf{r}^2\right) = \alpha, \tag{3.4}$$

$$P\left(\mathbf{e}^T \cdot H^T \cdot H \cdot \mathbf{e} \geq k^2 \cdot \mathbf{r}^2\right) = \alpha. \tag{3.5}$$

Using theorem 5 (see section §B.3), it is known that $\mathbf{e}^T \cdot H^T \cdot H \cdot \mathbf{e} \geq \|\mathbf{e}\|^2 \lambda_{min}^{H^T \cdot H}$. Moreover, theorem 6 (see section §B.3) ensures that $P\left(\|\mathbf{e}\|^2 \lambda_{min}^{H^T \cdot H} \geq k^2 \cdot \mathbf{r}^2\right) \leq P\left(\mathbf{e}^T \cdot H^T \cdot H \cdot \mathbf{e} \geq k^2 \cdot \mathbf{r}^2\right)$. Therefore,

$$P\left(\|\mathbf{e}\|^2 \lambda_{min}^{H^T \cdot H} \geq k^2 \cdot \mathbf{r}^2\right) \leq \alpha, \tag{3.6}$$

$$P\left(\|\mathbf{e}\|^2 \geq \frac{k^2 \cdot \mathbf{r}^2}{\lambda_{min}^{H^T \cdot H}}\right) \leq \alpha. \tag{3.7}$$

Therefore, if there is k so that $P\left(\mathbf{s}^2 \geq k^2 \cdot \mathbf{r}^2\right) = \alpha$, the protection level $PL = \frac{k \cdot \|\mathbf{r}\|}{\sqrt{\lambda_{min}^{H^T \cdot H}}}$ satisfies $P\left(\|\mathbf{e}\| \geq PL\right) \leq \alpha$. □

3.2.3.2 Finding k that satisfies $P\left(\mathbf{s}^2 \geq k^2 \cdot \mathbf{r}^2\right) = \alpha$

First, it is useful to remember a few properties of Least Square Estimation linking $\boldsymbol{\epsilon}$, $\mathbf{r}$ and $\mathbf{s}$ (see proofs in Appendix B.2):

- $\boldsymbol{\epsilon} = \mathbf{r} + \mathbf{s}$,
- $\mathbf{s}^T \cdot \mathbf{r} = 0$,
- $\boldsymbol{\epsilon}^2 = \mathbf{r}^2 + \mathbf{s}^2$.

Normalization of the problem. The absolute value of $\mathbf{s}$ and $\mathbf{r}$ are not relevant to find k, what is of interest is the **ratio** $\left(\frac{\mathbf{s}^2}{\mathbf{r}^2}\right)$. The problem can therefore be normalized to only focus on this aspect.

$$P\left(\mathbf{s}^2 \geq k^2 \cdot \mathbf{r}^2\right) \quad = \alpha, \tag{3.8}$$

$$\Longleftrightarrow P\left(\frac{\mathbf{s}^2}{\boldsymbol{\epsilon}^2} \geq k^2 \cdot \frac{\mathbf{r}^2}{\boldsymbol{\epsilon}^2}\right) = \alpha. \tag{3.9}$$

In the following part we consider only the normalized problem by defining

$$\widetilde{\boldsymbol{\epsilon}} = \frac{\boldsymbol{\epsilon}}{\|\boldsymbol{\epsilon}\|}, \qquad \widetilde{\mathbf{r}} = \frac{\mathbf{r}}{\|\boldsymbol{\epsilon}\|}, \qquad \widetilde{\mathbf{s}} = \frac{\mathbf{s}}{\|\boldsymbol{\epsilon}\|}. \tag{3.10}$$

Hence,

$$\widetilde{\mathbf{r}}^2 + \widetilde{\mathbf{s}}^2 = 1. \tag{3.11}$$

The problem becomes finding k that satisfies the equation

$$P\left(\widetilde{\mathbf{s}}^2 \geq k^2 \cdot \widetilde{\mathbf{r}}^2\right) = \alpha. \tag{3.12}$$

By construction $\widetilde{\boldsymbol{\epsilon}}$ is a vector of the m-dimensional (because there are m measurements) unit circle. $\widetilde{\mathbf{r}}$ and $\widetilde{\mathbf{s}}$ form an orthogonal decomposition of $\widetilde{\boldsymbol{\epsilon}}$ where $\widetilde{\mathbf{s}} \in Im(H)$ and $\widetilde{\mathbf{r}} \in Im(H)^{\perp}$. Therefore, the problem can be represented as in figure 3.1.

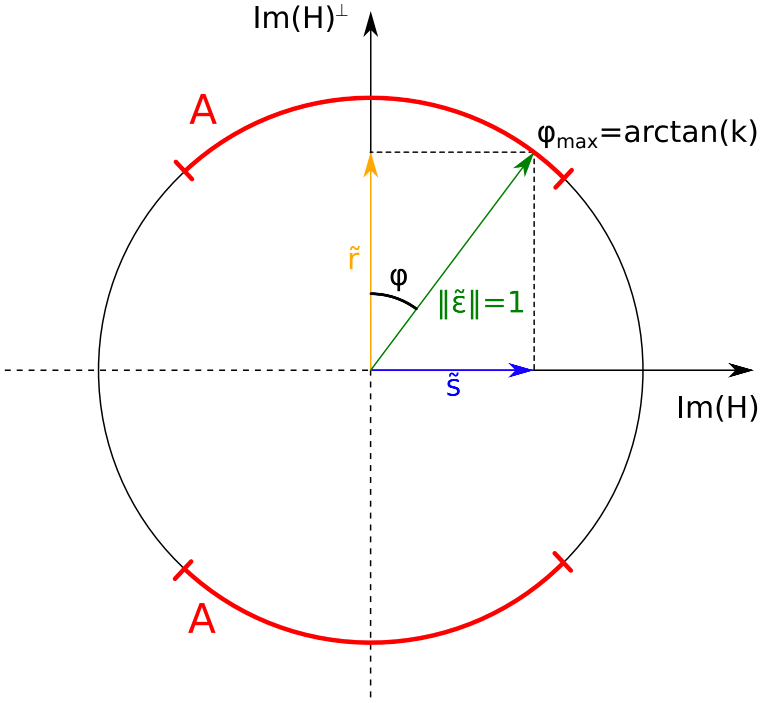


Figure 3.1: Orthogonal representation of $\widetilde{\boldsymbol{\epsilon}}$.

One-dimensional problem. To better understand the method, it is useful to consider an example. Let take the case of localization in a **one-dimensional space** using **two measurements**. Because the space has one dimension, $Im\,(H)$ has also one dimension. Moreover, $Im\,(H)^T$ has $m - n = 2 - 1 = 1$ dimension. The problem can therefore be represented has in figure 3.1. k can be found by solving the following equation:

$$P\left(\frac{\|\widetilde{\mathbf{s}}\|}{\|\widetilde{\mathbf{r}}\|} \leq k\right) = 1 - \alpha. \tag{3.13}$$

In the 2-dimensional measurement space $\widetilde{\boldsymbol{\epsilon}}$ is a vector of the unit circle. Therefore, the area defined by $\frac{\|\widetilde{\mathbf{s}}\|}{\|\widetilde{\mathbf{r}}\|} \leq k$ can be represented by an arc (in red on figure 3.1). The measurement error $\widetilde{\boldsymbol{\epsilon}}$ is supposed to be **isotropic**. Therefore, the probability of $\widetilde{\boldsymbol{\epsilon}}$, to be anywhere on the unit circle is uniform. For that reason the probability of $\widetilde{\boldsymbol{\epsilon}} \in A$ (where A is a region of the circle) correspond to the ratio between the length of that region and the length of the entire space (the unit circle).

Hence, k can be computed in 1-dimensional space with 2-dimensional measurement space by solving

$$\frac{l_{arc}}{l_{circle}} = \frac{2 \cdot \varphi_{max}}{\pi} = \frac{2 \cdot \arctan(k)}{\pi} = 1 - \alpha. \tag{3.14}$$

Therefore, for $\alpha = 10^{-3}$,

$$k\left(\alpha = 10^{-3}, m = 2, n = 1\right) = 636.6192. \tag{3.15}$$

Using simple example, it is easier to see the role of the parameter k in the protection level:

- If a small integrity risk is sought, the more measurement error (not in term of absolute value but in term of distribution in the measurement space) needs to be considered by the Protection Level. Therefore, the area defined by $\frac{\|\widetilde{\mathbf{s}}\|}{\|\widetilde{\mathbf{r}}\|} \leq k$ needs to be large, so k needs to be large.
- For small values of k (small protection level) the estimation that will satisfy the protection level are those which contain most of the measurement error in the residual $\mathbf{r}$. Those measurements have smaller $\mathbf{s}$ (so smaller estimation error $\mathbf{e}$). This means that as k increases, worse and worse estimates are included in the Protection Level (the best values are included first).

Finding k in the general case As in one dimension, $\frac{\|\widetilde{\mathbf{s}}\|}{\|\widetilde{\mathbf{r}}\|} \leq k$ defines an area A (m-dimensional area). Because of the **isotropy assumption** $P\left(\widetilde{\boldsymbol{\epsilon}} \in A\right)$ can be expressed by equation (3.16) which is the m-dimensional equivalent to equation (3.14). When in one dimension, a point had to be integrated over an arc, in m-dimension, a $(m-n)$-dimensional sphere has to be integrated over an n-dimensional cap.

$$P\left(\widetilde{\boldsymbol{\epsilon}} \in A\right) = \frac{\int_{\widetilde{\boldsymbol{\epsilon}} \in A} d\widetilde{\boldsymbol{\epsilon}}}{Surface\left(Sphere_{m-1}(1)\right)} = 1 - \alpha, \tag{3.16}$$

where

A	corresponds to the area where $\frac{\|\mathbf{s}\|}{\|\mathbf{r}\|} \leq k$,
$Sphere_{m-1}(1)$	corresponds to the sphere in m-dimensional space of radius 1 ($Sphere_1$ is a circle in two-dimensional space).

- The surface of a sphere of radius τ in an n-dimensional space is given by
$$Surface\left(Sphere_{m-1}(\tau)\right) = \frac{2 \cdot \pi^{\frac{m}{2}}}{\Gamma\left(\frac{m}{2}\right)} \cdot \tau^{m-1}. \tag{3.17}$$
- The integration over A can be decomposed in two parts: $A_{Im(H)}$, the integration over the dimensions of $Im(H)$; and $A_{Im(H)^{\perp}}$, the integration over the rest of the dimensions. The integral of $d\widetilde{\boldsymbol{\epsilon}}$ over $A_{Im(H)^{\perp}}$ corresponds to the surface of a sphere in $(m-n)$-dimensional space with a radius of $\|\widetilde{\mathbf{r}}\|$ (indeed, $\|\widetilde{\mathbf{r}}\| = \tau$ defines a sphere of $(m-n)$-dimensional space with a radius of τ). Therefore, to obtain the surface area of A, the sphere in $(m-n)$-dimensional space with a radius of $\|\widetilde{\mathbf{r}}\|$ needs to be integrated over the rest of the dimensions (i.e., $A_{Im(H)}$).

$$\int_{\widetilde{\boldsymbol{\epsilon}} \in A} d\widetilde{\boldsymbol{\epsilon}} = \overbrace{\int \cdots \int}^{A_{Im(H)}} \overbrace{\int \cdots \int}^{A_{Im(H)^T}} d\widetilde{\boldsymbol{\epsilon}} \qquad \left(\overbrace{\int \cdots \int}^{A_{Im(H)}} \text{ will be noted } \int_{\widetilde{\boldsymbol{\epsilon}}|_{A_{Im(H)}}} \right) \tag{3.18}$$

$$= \int_{\widetilde{\boldsymbol{\epsilon}}|_{A_{Im(H)}}} Surface\left(Sphere_{m-n-1}\left(\|\widetilde{\mathbf{r}}\|\right)\right) d\widetilde{\boldsymbol{\epsilon}} \tag{3.19}$$

$$= \int_{\widetilde{\boldsymbol{\epsilon}}|_{A_{Im(H)}}} Surface\left(Sphere_{m-n-1}\left(\sqrt{1-\widetilde{\mathbf{s}}^2}\right)\right) d\widetilde{\boldsymbol{\epsilon}} \tag{3.20}$$

$$= \int_{\widetilde{\boldsymbol{\epsilon}}|_{A_{Im(H)}}} \frac{2 \cdot \pi^{\frac{m-n}{2}}}{\Gamma\left(\frac{m-n}{2}\right)} \cdot \left(1 - \widetilde{\mathbf{s}}^2\right)^{\frac{m-n-1}{2}} d\widetilde{\boldsymbol{\epsilon}}. \tag{3.21}$$

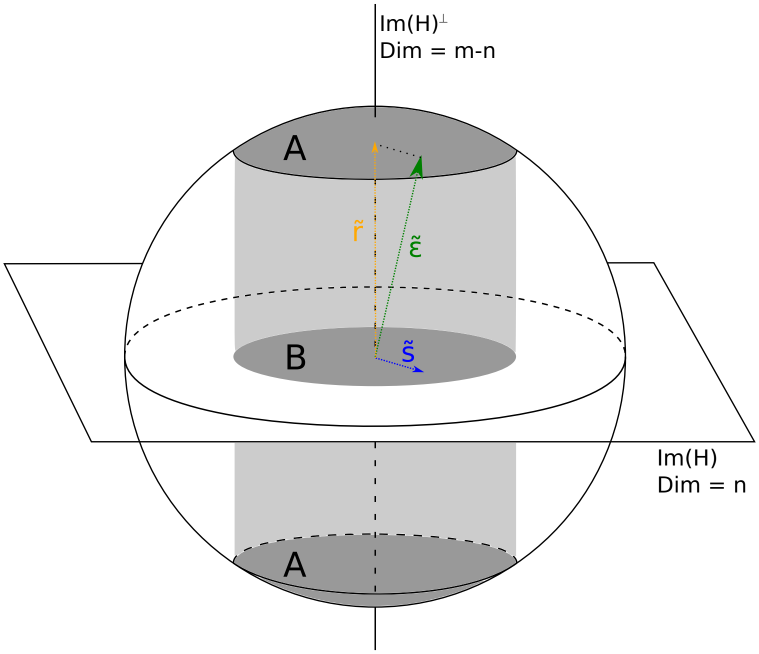


Figure 3.2: Integration in the m-dimensional measurement space.

Instead of having to integrate over the "curved" area A, it is preferred to integrate over "flat" area B. That way we can directly integrate using $\widetilde{\mathbf{s}}$ without having to change its expression (to introduce $\widetilde{\boldsymbol{\epsilon}}$). To apply this change, a variable change is required (see figure 3.3). This projection from A to B introduces the term $\frac{1}{\sqrt{1-\widetilde{\mathbf{s}}^2}}$ inside the integral.

$$\int_{\widetilde{\boldsymbol{\epsilon}}\in A} d\widetilde{\boldsymbol{\epsilon}} = \int_{\widetilde{\mathbf{s}}\in B} \frac{1}{\sqrt{1-\widetilde{\mathbf{s}}^2}} \frac{2\cdot\pi^{\frac{m-n}{2}}}{\Gamma\left(\frac{m-n}{2}\right)} \cdot \left(1-\widetilde{\mathbf{s}}^2\right)^{\frac{m-n-1}{2}} d\widetilde{\mathbf{s}} \tag{3.22}$$

$$= \int_{\widetilde{\mathbf{s}}\in B} \frac{2\cdot\pi^{\frac{m-n}{2}}}{\Gamma\left(\frac{m-n}{2}\right)} \cdot \left(1-\widetilde{\mathbf{s}}^2\right)^{\frac{m-n-2}{2}} d\widetilde{\mathbf{s}}. \tag{3.23}$$

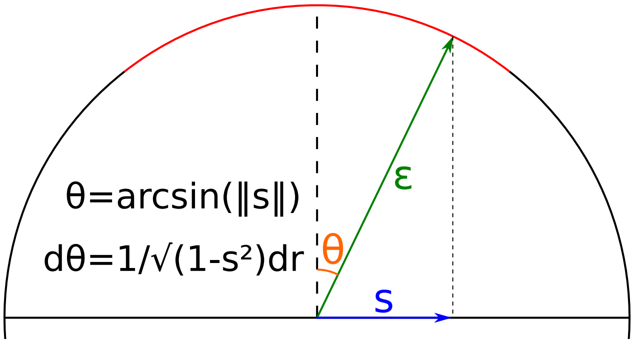


Figure 3.3: Variable change.

Finally,

$$P\left(\widetilde{\boldsymbol{\epsilon}} \in A\right) = \pi^{-\frac{n}{2}} \frac{\Gamma\left(\frac{m}{2}\right)}{\Gamma\left(\frac{m-n}{2}\right)} \int_{\widetilde{\mathbf{s}}\in B} \left(1-\widetilde{\mathbf{s}}^2\right)^{\frac{m-n-2}{2}} d\widetilde{\mathbf{s}}. \tag{3.24}$$

Therefore, k is found by solving the equation

$$\Gamma\left(\frac{m}{2}\right)\cdot\int_{\widetilde{\mathbf{s}}\in B}\left(1-\widetilde{\mathbf{s}}^2\right)^{\frac{m-n-2}{2}}d\widetilde{\mathbf{s}}=\Gamma\left(\frac{m-n}{2}\right)\cdot\pi^{\frac{n}{2}}\cdot(1-\alpha)\,, \tag{3.25}$$

where

$$B=\left\{\widetilde{\mathbf{s}}\mid\frac{\widetilde{\mathbf{s}}^2}{\widetilde{\mathbf{r}}^2}\le k^2\right\}=\left\{\widetilde{\mathbf{s}}\mid\frac{\widetilde{\mathbf{s}}^2}{1-\widetilde{\mathbf{s}}^2}\le k^2\right\}=\left\{\widetilde{\mathbf{s}}\mid\widetilde{\mathbf{s}}\le\sqrt{\frac{k^2}{1+k^2}}\right\}. \tag{3.26}$$

3.3 IBPL Implementation

3.3.1 Non linear least square estimation

GPS localization is a non-linear problem. Indeed, in its simplest form, the observation equation for m satellites is:

$$\mathbf{y}=h\left(x,y,z,d\right)=\left(\begin{array}{c}h_1\left(x,y,z,d\right)\\h_2\left(x,y,z,d\right)\\\vdots\\h_m\left(x,y,z,d\right)\end{array}\right), \tag{3.27}$$

with

$$h_X\left(x,y,z,d\right)=\sqrt{\left(x-x^X\right)^2+\left(y-y^X\right)^2+\left(z-z^X\right)^2}+d, \tag{3.28}$$

where

(x,y,z)	receiver coordinates,
$\left(x^X,y^X,z^X\right)$	satellite coordinates,
d	receiver clock error.

To solve the non-linear problem, the least square estimation is going to be used iteratively on the system linearized around a state $\mathbf{x_k}=\left(x_k,y_k,z_k,d\right)^T$. The first step is to choose the initial $\mathbf{x_k}$, the Jacobian H of the system can then be computed for that state:

$$H=\left[\begin{array}{cccc}\frac{x_k-x^1}{R^1}&\frac{y_k-y^1}{R^1}&\frac{z_k-z^1}{R^1}&1\\\frac{x_k-x^2}{R^2}&\frac{y_k-y^2}{R^2}&\frac{z_k-z^2}{R^2}&1\\&\vdots&&\\\frac{x_k-x^m}{R^m}&\frac{y_k-y^m}{R^m}&\frac{z_k-z^m}{R^m}&1\end{array}\right], \tag{3.29}$$

where

$$R^X=\sqrt{\left(x_k-x^X\right)^2+\left(y_k-y^X\right)^2+\left(z_k-z^X\right)^2}. \tag{3.30}$$

Then the error on the measurement is computed as

$$\mathbf{dy}=\mathbf{y}-h\left(\mathbf{x_k}\right). \tag{3.31}$$

Using the linear least square method $\mathbf{dx}=\mathbf{x_{k+1}}-\mathbf{x_k}$ can be found by using either the non-weighted method (3.32) or the weighted method (3.33).

$$\mathbf{dx}=\left(H^TH\right)^{-1}H^T\mathbf{dy}, \tag{3.32}$$

$$\mathbf{dx}=\left(H^TWH\right)^{-1}H^TW\mathbf{dy}\text{ (where }W\text{ is the weigthting matrix).} \tag{3.33}$$

The next state is then computed as

$$\mathbf{x_{k+1}} = \mathbf{x_k} + \mathbf{dx}. \tag{3.34}$$

The process repeats until a precise enough result is found.

This method outputs the last estimated position $\mathbf{x_k}$, the last Jacobian H and the residual vector $\mathbf{r} = \mathbf{y} - h\left(\mathbf{x_k}\right)$.

3.3.2 Computing the Horizontal Protection Level (HPL)

The isotropy based protection level (IBPL) can be computed for a given target integrity risk α, using a precomputed coefficient k, the residual of the least square estimation $\mathbf{r}$ and the Dilution of Precision matrix $\left(H^T H\right)^{-1}$ according to equation (3.35). Here, the method to compute the Protection Level is presented in the East North Up (ENU) frame but it is not limited to this frame.

$$HPL = k * \|\mathbf{r}\| * \sqrt{\lambda^h_{max}\left(\left(H^T H\right)^{-1}\right)}, \tag{3.35}$$

where

$$\left(H^T H\right)^{-1} = \begin{bmatrix} h_{EE} & h_{EN} & h_{EU} & \cdots \\ h_{NE} & h_{NN} & h_{NU} & \cdots \\ h_{UE} & h_{UN} & h_{UU} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{3.36}$$

and

$$\sqrt{\lambda^h_{max}\left(\left(H^T \cdot H\right)^{-1}\right)} = \sqrt{\frac{h_{EE}+h_{NN}}{2} + \sqrt{\left(\frac{h_{EE}-h_{NN}}{2}\right)^2 + \left(\frac{h_{EN}+h_{NE}}{2}\right)^2}}, \tag{3.37}$$

and k is solution of

$$\Gamma\left(\frac{m}{2}\right) \cdot \int_{\mathbf{z}\in\mathbb{R}^n, 0\leq\mathbf{z}^2\leq\frac{k^2}{1+k^2}} \left(1-\mathbf{z}^2\right)^{\frac{m-n-2}{2}} \cdot \mathbf{dz} = \Gamma\left(\frac{m-n}{2}\right) \cdot \pi^{\frac{n}{2}} \cdot (1-\alpha), \tag{3.38}$$

where

α is the confidence level,

n is the number of parameters to estimate (4: GPS, 5: GPS+GLONASS),

m is the number of satellites.

3.3.3 Computing k

3.3.3.1 General method to compute k

k can be computed by solving equation (3.38), therefore it is only function of the confidence level α, the number of measurements m and the number of parameters to estimate n. Because k does not depend on the result of the estimation it can be precomputed for common values of n, m and α and be stored in a lookup table to be used in real time.

Equation (3.38) can be rewritten to isolate the integral,

$$\int_{\mathbf{z}\in\mathbb{R}^n, 0\leq\mathbf{z}^2\leq\frac{k^2}{1+k^2}} \left(1-\mathbf{z}^2\right)^{\frac{m-n-2}{2}} \cdot \mathbf{dz} = \frac{\Gamma\left(\frac{m-n}{2}\right)}{\Gamma\left(\frac{m}{2}\right)} \cdot \pi^{\frac{n}{2}} \cdot (1-\alpha). \tag{3.39}$$

- The right hand side of the equation can easily be computed.
- The left hand side of the equation can be computed using the spherical coordinate $(r, \phi_1, \phi_2, \ldots, \phi_{n-1})$.

In this coordinate system

$$\mathbf{z} = r \cdot \mathbf{r}\left(\phi_1, \phi_2, \ldots, \phi_{n-1}\right), \tag{3.40}$$

$$\mathbf{dz} = r^{n-1} \cdot \sin^{n-2}\left(\phi_1\right) \cdot \sin^{n-3}\left(\phi_2\right) \cdot \ldots \cdot \sin\left(\phi_{n-2}\right) \cdot dr \cdot d\phi_1 \cdot d\phi_2 \cdot \ldots \cdot d\phi_{n-1}, \tag{3.41}$$

where $\mathbf{r}$ is the unit vector oriented in the same direction as $\mathbf{z}$.

Therefore, using this system the integrations over the different dimensions are independent and can be easily computed (see next paragraph for an example in 4 dimensions).

3.3.3.2 Example in dimension 4

Using the aforementioned spherical coordinates system with $n = 4$,

$$\int_{\mathbf{z}\in\mathbb{R}^4, 0\leq\mathbf{z}^2\leq\frac{k^2}{1+k^2}} \left(1-\mathbf{z}^2\right)^{\frac{m-6}{2}} \cdot \mathbf{dz} = \int_0^{2\pi}\int_0^{\pi}\int_0^{\pi}\int_0^{\sqrt{\frac{k^2}{1+k^2}}} \left(1-r^2\right)^{\frac{m-6}{2}} \cdot r^3 \cdot \sin^2\left(\phi_1\right) \cdot \sin\left(\phi_2\right) \cdot dr \cdot d\phi_1 \cdot d\phi_2 \cdot d\phi_3 \tag{3.42}$$

$$= \frac{\pi}{2} \cdot 2 \cdot 2\pi \cdot \int_0^{\sqrt{\frac{k^2}{1+k^2}}} r^3 \cdot \left(1-r^2\right)^{\frac{m-6}{2}} \cdot dr \tag{3.43}$$

$$= 2\pi^2 \cdot \int_0^{\sqrt{\frac{k^2}{1+k^2}}} r^3 \cdot \left(1-r^2\right)^{\frac{m-6}{2}} \cdot dr. \tag{3.44}$$

k can be found by numerically solving[2] the equation

$$\int_0^{\sqrt{\frac{k^2}{1+k^2}}} r^3 \cdot \left(1-r^2\right)^{\frac{m-6}{2}} \cdot dr = \frac{1}{2}\frac{\Gamma\left(\frac{m-4}{2}\right)}{\Gamma\left(\frac{m}{2}\right)} \cdot (1-\alpha). \tag{3.45}$$

Using the integration by parts (section §B.4), the integral can be removed entirely and k can be found by solving

$$\alpha = \frac{m-2}{2} \cdot \left(1+k^2\right)^{\frac{4-m}{2}} - \frac{m-4}{2} \cdot \left(1+k^2\right)^{\frac{2-m}{2}}. \tag{3.46}$$

3.4 IBPL Simulation Results

To verify the theoretical results, several simulations have been created. One dimensional simulations have been tested first. They comprise of a wall and a mobile. The mobile is place randomly and must localize itself using the distance with the wall (6 measurements are used). The method has been extensively tested for integrity risks (α) from 10^{-2} to 10^{-5}. A few tests have been done for more demanding integrity risks however to obtain statistically significant results the number of epochs has to be increased dramatically.

Stanford Diagram. To analyze the results, a simplified Stanford diagram is used. It presents the distribution of the computed protection level relative to the actual error by a 2D histogram. The red line corresponds to the points for which the protection level matches the actual error. The points below that line are those with a protection level smaller than the actual error. The probability for any point to fall below that line should not exceed the target integrity risk.

Figure 3.4 shows the result for 10 000 epochs. The IBPL method produces protection levels that exactly satisfy the TIR. Indeed, in one dimension $P\left(\|\mathbf{e}\| \geq PL\right) = \alpha$ because the approximation made in section 3.2.3.1 is not required.

[2] In Matlab, the function fzero was used: `https://www.mathworks.com/help/matlab/ref/fzero.html`

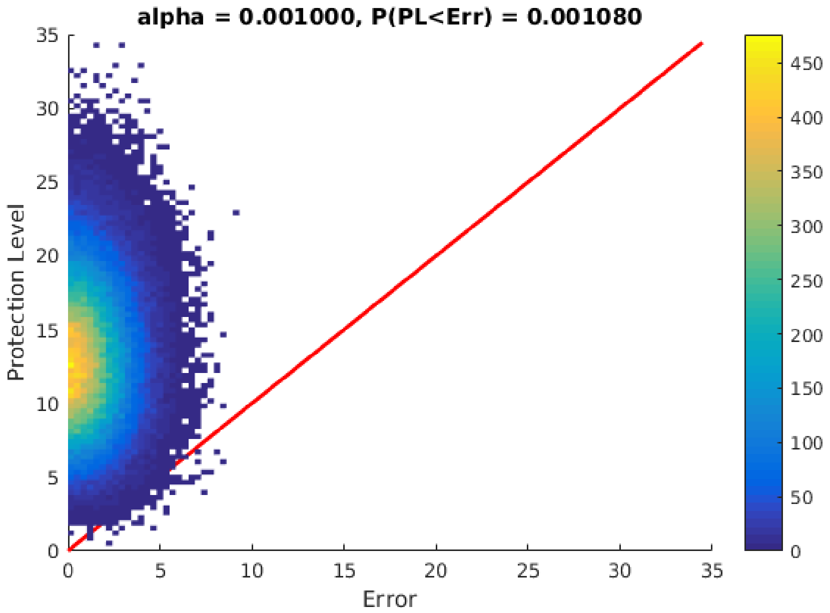


Figure 3.4: One-dimensional IBPL Stanford diagram.

Multipath (i.e. outliers) has been added to the simulation to test the robustness of the algorithm. The results on figure 3.5 show that $P(\|\mathbf{e}\| \geq PL) \leq \alpha$ therefore the algorithm computes consistent protection levels.

A 4D simulation similar to the GNSS localization problem has also been programmed. The result of this test is presented in figure 3.6 for an integrity risk $\alpha = 0.01$ with 1 000 000 epochs and a random number of satellites (from 6 to 10 satellites). In 4D the equality $P(\|\mathbf{e}\| \geq PL) = \alpha$ is lost. The protection levels are higher than they need to be which make them still consistent, but also increase the risk of having protection levels exceeding the Alert Limit. Protection Levels too high will result in an increase of the number of alerts and therefore make the system unavailable.

3.4.1 Conclusion

The IBPL method is interesting for several reasons. It provides protection levels that seem robust to multipath without making assumption on the error distribution other than the isotropy. During my intership, I have managed to show how, starting from the Least Square Estimation, the IBPL method is constructed and what are the steps and hypothesis used to justify this method. The result from the simulation are promising, but experimental tests are required to verify its robustness on real data.

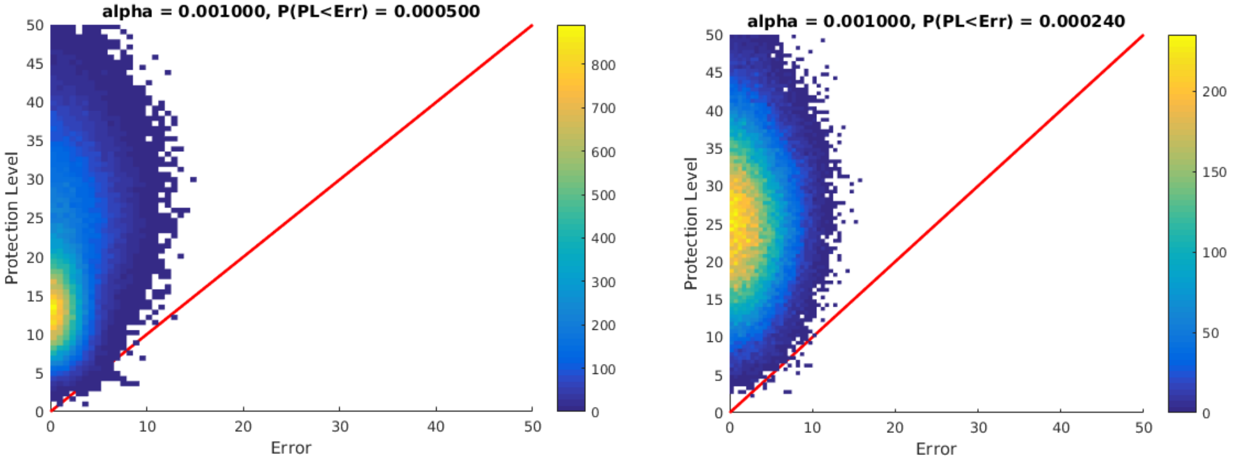
However, since this method is based on Least Square Estimation, it does not take advantage of the correlation between the previous positions. A Kalman based method would solve that issue, that is why the Kalman Integrated Protection Level (KIPL) has also been studied.

3.5 Kalman Integrated Protection Level (KIPL)[4]

3.5.1 Kalman Filter

The Kalman Integrated Protection Level (KIPL) is designed to apply for estimation using Kalman filtering. Kalman filtering can be split into two steps, a prediction step and an update step.

Prediction step:



(a)

(b)

Figure 3.5: One dimensional IBPL Stanford diagram with multipath. (a) Multipath for half the epochs. (b) Multipath for all the epochs.

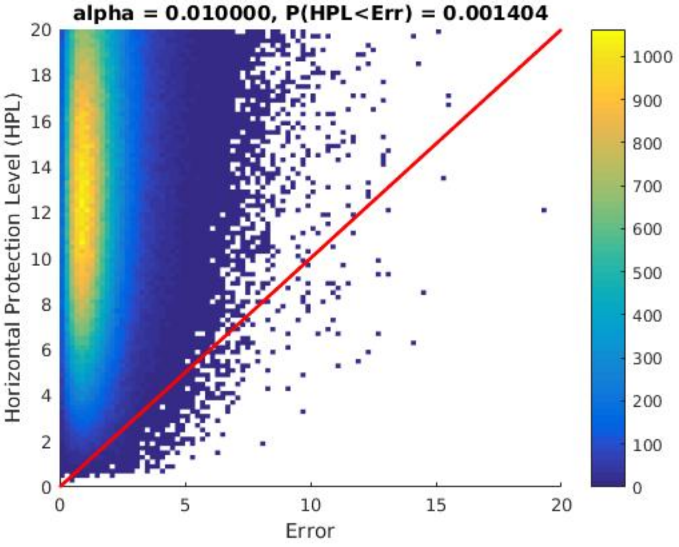


Figure 3.6: 4D IBPL Stanford diagram without multipath.

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1}, \qquad \text{(Predicted state estimate)}$$
$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k. \qquad \text{(Predicted estimated covariance)}$$

Update step:

$$\tilde{y}_k = z_k - H_k \hat{x}_{x|k-1}, \qquad \text{(Measurement residuals)}$$
$$S_k = H_k P_{k|k-1} H_k^T + R_k, \qquad \text{(Residuals covariance)}$$
$$K_k = P_{k|k-1} H_k^T S_k^{-1}, \qquad \text{(Kalman gain)}$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k, \qquad \text{(Update state estimate)}$$
$$P_{k|k} = (I - K_k H_k) P_{k|k-1}. \qquad \text{(Update estimated covariance)}$$

3.5.2 The main idea

The KIPL algorithm works by modeling the estimation error by a zero-mean multivariate Student distribution. This distribution can be described by three parameters, a dimension d, a degree of freedom N_m and a covariance matrix R_m. Its density function is

$$f_{t_{N_m(R_m)}}(\boldsymbol{\theta}) = \frac{\Gamma\left(\frac{N_m+d}{2}\right)}{\Gamma\left(\frac{N_m}{2}\right) \cdot N_m^{\frac{d}{2}} \cdot \pi^{\frac{d}{2}} \cdot \det(R_m)^{\frac{1}{2}}} \cdot \left(1 + \frac{1}{N_m} \boldsymbol{\theta}^T R_m^{-1} \boldsymbol{\theta}\right)^{-\frac{N_m+d}{2}}. \tag{3.47}$$

This distribution is chosen instead of a Gaussian distribution to improve the robustness against outliers. Using this distribution, a bound on the error can be easily computed by integrating the tail of the distribution.

To obtain a Student distribution on the estimate error, two things need to be considered. First the measurement error needs to be modeled. Then, the error needs to be propagated. The algorithm proposes a method to model the measurement error that will be detailed in section 3.5.4.

The state update of the Kalman filter can be written as

$$\hat{x}_{k|k} = K_k z_k + (I - K_k H_k) F_k \hat{x}_{k-1|k-1}. \tag{3.48}$$

The same formula is used to update the estimate error.

3.5.3 Dealing with different measurement types

This method is designed to work with multiple measurement sources. However, it assumes that all measurements have the same variance which is not true when the measurements come from different sources. To solve this issue, the method computes a partial bound B_m for each measurement type and add them at the end. To compute such bounds, blocks of matrices will be used (for H_k and S_k) corresponding to appropriate measurement types. These matrices will be noted H_m and S_m. Even though the patent[4] does not detail why this can be done, looking at a simple example shows that this approach is justified.

Pseudorange & Doppler example For instance, in the case of pseudorange and Doppler measurements for GPS localization, B_{pseudo} will be computed using the observation matrix mapping the state to the pseudorange observations ($H_{k,pseudo}$) and the appropriate S_{pseudo} matrix. Similarly $B_{doppler}$ will be computed using the observation matrix mapping the state to the Doppler observations ($H_{k,doppler}$) and the appropriate $S_{doppler}$. It is worth noting that the two measurement types are assumed to have uncorrelated errors.

$$z_k = \begin{bmatrix} z_{k,pseudo} \\ z_{k,doppler} \end{bmatrix}, \tag{3.49}$$

$$H_k = \begin{bmatrix} H_{k,pseudo} \\ H_{k,doppler} \end{bmatrix}, \tag{3.50}$$

$$S_k = \begin{bmatrix} S_{k,pseudo} & 0 \\ 0 & S_{k,doppler} \end{bmatrix}. \tag{3.51}$$

Therefore,

$$K_k = P \cdot \left[\begin{array}{ccc} H^T_{k,pseudo} \cdot S^{-1}_{k,pseudo} & ; & H^T_{k,doppler} \cdot S^{-1}_{k,doppler} \end{array} \right]. \tag{3.52}$$

The state can be expressed using the previous state and the observation.

$$\hat{x}_{k|k} = K_k z_k + (I_d - K_k H_k) F_k \hat{x}_{k-1|k-1}, \tag{3.53}$$

therefore

$$e_{x,k} = \underbrace{K_k e_{z,k}}_{A} + \underbrace{(I_d - K_k H_k) F_k e_{x,k-1}}_{B}. \tag{3.54}$$

Rewritten using the previous notation

$$e_{x,k} = P \cdot H^T_{k,pseudo} \cdot S^{-1}_{k,pseudo} \cdot e_{z,k,pseudo} + P \cdot H^T_{k,doppler} \cdot S^{-1}_{k,doppler} \cdot e_{z,k,doppler} + (I_d - K_k H_k) F_k e_{x,k-1}, \tag{3.55}$$

or

$$e_{x,k,pseudo} = P \cdot H^T_{k,pseudo} \cdot S^{-1}_{k,pseudo} \cdot e_{z,k,pseudo} + (I_d - K_k H_k) F_k e_{x,k-1,pseudo}; \tag{3.56}$$

$$e_{x,k,doppler} = P \cdot H^T_{k,pseudo} \cdot S^{-1}_{k,pseudo} \cdot e_{z,k,pseudo} + (I_d - K_k H_k) F_k e_{x,k-1,doppler}; \tag{3.57}$$

$$e_{x,k} = e_{x,k,pseudo} + e_{x,k,doppler}. \tag{3.58}$$

As a result, an upper bound on $\|e_{x,k}\|$ can be computed by summing the bounds of $\|e_{x,k,pseudo}\|$ and $\|e_{x,k,doppler}\|$. This example with two types of measurement can be extended to any number of measurements. Therefore, a protection level can be computed as the sum of partial bounds

$$PL = \sum_m B_m. \tag{3.59}$$

3.5.4 Modeling the error

To model the errors, the state estimation error and the measurement errors are assumed to be distributed according to a zero-mean multivariate Student t-distribution with a degree of freedom N_m and a covariance matrix R_m.

$$e_{x,k} = \left(\hat{x}_{k|k} - x\right) \sim t_{N_m}(R_m). \tag{3.60}$$

The updated error distribution for a measurement type m is :

$$e_{x,k,m} = \underbrace{K_{k,m} e_{z,k,m}}_{A} + \underbrace{(I_d - K_k H_k) F_k e_{x,k-1,m}}_{B}. \tag{3.61}$$

It is worth noting that the left term A involves the partial Kalman gain $K_{k,m}$, but the right term involves the full Kalman gain K_k and observation matrix H_k.

For readability the index k will be ignored in the next parts.

Distribution of A The measurement error distribution is computed iteratively. Also little explanation has been found on the way the degree of freedom is computed, the covariance matrix seems to correspond to the covariance matrix for studentized residual.

$$A \sim t_{N_{m1}}\left(R_{m1}\right), \tag{3.62}$$

with

$$N_{m1} = n_m + \beta N'_{m1}, \tag{3.63}$$

$$n_m = n_{OBS,m} - tr\left(H_m K_m\right), \qquad (n_{OBS,m} \text{ being the number of measurements of type m})$$

and

$$R_{m1} = K_m\left(r_m^2 S_m\right) K_m^T, \tag{3.64}$$

$$r_m^2 = \left(y_m^T S_m^{-1} y_m + \beta N'_{m1}\left(r'_m\right)^2\right)/N_{m1}, \tag{3.65}$$

Distribution of B Since $e_{x,k-1} \sim t_{N'_m}\left(R'_m\right)$ (where N'_m and R'_m are the parameters of the Student distribution describing $\left(\hat{x}_{k-1|k-1} - x\right)$),

$$B \sim t_{N_{m2}}\left(R_{m2}\right), \tag{3.66}$$

with

$$N_{m2} = N'_{m,} \tag{3.67}$$

$$R_{m2} = U R'_m U^T, \tag{3.68}$$

$$U = \left(I_d - KH\right) F. \tag{3.69}$$

This results comes from the properties of the multiplication of a multivariate t-distribution by a matrix. Indeed, when multiplying a t-distribution by a matrix U, the degree of freedom stays unchanged while the covariance matrix Γ becomes $U\Gamma U^T$.

Sum of A and B The sum of two t-distributions is not in the general case a student distribution. However it can be approximated by:

$$e_{x,k} \sim t_{N_m}\left(R_m\right), \tag{3.70}$$

with

$$R_m = R_{m1} + R_{m2}, \tag{3.71}$$

and N_m solution of

$$N_m^{\frac{d-2}{2}} t_m^{2-d}\left(1+t_m^{-2}\right)^{-\frac{N_m+d-2}{2}} = N_{m1}^{\frac{d-2}{2}} t_{m1}^{N_{m1}} \exp\left\{\frac{\left(N_{m1}+N_{m2}\right) N_{m1}}{2N_{m2}} t_{m2}^2\right\} + N_{m2}^{\frac{d-2}{2}} t_{m2}^{N_{m2}} \exp\left\{\frac{\left(N_{m1}+N_{m2}\right) N_{m2}}{2N_{m1}} t_{m1}^2\right\}, \tag{3.72}$$

with

$$t_{m1} = \left[\frac{N_{m1} tr\left(R_{m1}\right)}{tr\left(S\right)}\right]^{\frac{1}{2}}, \tag{3.73}$$

$$t_{m2} = \left[\frac{N_{m2} tr\left(R_{m2}\right)}{tr\left(S\right)}\right]^{\frac{1}{2}}, \tag{3.74}$$

$$t_{m} = \left[\frac{N_{m} tr\left(R_{m}\right)}{tr\left(S\right)}\right]^{\frac{1}{2}}, \tag{3.75}$$

$$S = N_{m1}R_{m1} + N_{m2}R_{m2} + Z, \tag{3.76}$$

$$Z = \omega \cdot \left(N_{m1}R_{m1} + N_{m2}R_{m2}\right), \tag{3.77}$$

ω being a tuning parameter ($\omega \sim [1;3]$ for small confidence levels, $\omega \sim 10$ for high confidence levels, according to [4]).

3.5.5 Computing the integrity bound

After the last step, the state estimate error is modeled by a n-dimensional t-distribution.

$$e_X \sim t_{N_m}\left(R_m\right), \tag{3.78}$$

with
R_m covariance matrix,
N_m is the degree of freedom.

The patent then describes without justifying, how, using N_m and R_m, the integrity bound can be computed. In the following part I propose a reasoning leading to the result. For that I make an hypothesis that is not clearly stated in the patent but that I needed to reach the final result. There may be a way to reach the result without making that assumption.

3.5.5.1 Hypothesis

To compute the protection level, the covariance matrix is assumed to be diagonal with the same variance for all dimensions.

$$R_m = \sigma_X^2 \cdot I_d. \tag{3.79}$$

This assumption results in a spherically symmetric distribution.

$$\forall \boldsymbol{\theta_1}, \boldsymbol{\theta_2} \in \mathbb{R}^n, \qquad \|\boldsymbol{\theta_1}\| = \|\boldsymbol{\theta_2}\| \implies f_{t_\nu}\left(\boldsymbol{\theta_1}\right) = f_{t_\nu}\left(\boldsymbol{\theta_2}\right). \tag{3.80}$$

Differently said, no direction is privileged, therefore is only function of the norm of $\boldsymbol{\theta}$.

3.5.5.2 Integration

The density function of $t_\nu\left(\mu, R_m\right)$ is

$$f_{t_{N_m}}\left(\boldsymbol{\theta}\right) = \frac{\Gamma\left(\frac{N_m+d}{2}\right)}{\Gamma\left(\frac{N_m}{2}\right) \cdot N_m^{\frac{d}{2}} \cdot \pi^{\frac{d}{2}} \cdot \det\left(R_m\right)^{\frac{1}{2}}} \cdot \left(1 + \frac{1}{N_m}\boldsymbol{\theta}^T R_m^{-1}\boldsymbol{\theta}\right)^{-\frac{N_m+d}{2}}. \tag{3.81}$$

Because of the previous assumption, this function can be written using only the norm of $\boldsymbol{\theta}$.

$$f_{t_{N_m}}\left(\boldsymbol{\theta}\right) = \frac{\Gamma\left(\frac{N_m+d}{2}\right)}{\Gamma\left(\frac{N_m}{2}\right) \cdot N_m^{\frac{d}{2}} \cdot \pi^{\frac{d}{2}} \cdot \det\left(R_m\right)^{\frac{1}{2}}} \cdot \left(1 + \frac{1}{N_m}\frac{1}{\sigma_X^2} \cdot \|\boldsymbol{\theta}\|^2\right)^{-\frac{N_m+d}{2}}. \tag{3.82}$$

Using this expression we can compute the probability density of the norm of $\boldsymbol{\theta}$ ($r = \|\boldsymbol{\theta}\|$).

$$f_{\|\mathbf{\Theta}\|}\left(r\right)=\int_{\|\boldsymbol{\theta}\|=r} f_{t_{N_m}}\left(\boldsymbol{\theta}\right)\cdot d\boldsymbol{\theta}. \tag{3.83}$$

Moreover, because the density function is constant for a given radius, the probability density function of the norm of $\boldsymbol{\theta}$ can be easily computed as

$$f_{\|\mathbf{\Theta}\|}\left(r\right)=\underbrace{\frac{2\pi^{\frac{d}{2}}}{\Gamma\left(\frac{d}{2}\right)}\cdot r^{d-1}}_{\text{Surface of a sphere}}\cdot\underbrace{\frac{\Gamma\left(\frac{N_m+d}{2}\right)}{\Gamma\left(\frac{N_m}{2}\right)\cdot N_m^{\frac{d}{2}}\cdot\pi^{\frac{d}{2}}\cdot\det\left(R_m\right)^{\frac{1}{2}}}\cdot\left(1+\frac{1}{N_m}\frac{1}{\sigma_X^2}\cdot\|\boldsymbol{\theta}\|^2\right)^{-\frac{N_m+d}{2}}}_{\text{probability density on the sphere}}, \tag{3.84}$$

with $r=\|\boldsymbol{\theta}\|$.
Therefore,

$$f_{\|\mathbf{\Theta}\|}\left(r\right)=\frac{2\cdot\Gamma\left(\frac{N_m+d}{2}\right)}{\Gamma\left(\frac{d}{2}\right)\Gamma\left(\frac{N_m}{2}\right)}\cdot\frac{r^{d-1}}{N_m^{\frac{d}{2}}\cdot\det\left(R_m\right)^{\frac{1}{2}}}\cdot\left(1+\frac{1}{N_m\sigma_X^2}\cdot r^2\right)^{-\frac{N_m+d}{2}} \tag{3.85}$$

$$=\frac{2}{Beta\left(\frac{d}{2},\frac{N_m}{2}\right)}\cdot\frac{r^{d-1}}{N_m^{\frac{d}{2}}\cdot\left(\sigma_X^2\right)^{\frac{d}{2}}}\cdot\left(1+\frac{1}{N_m\sigma_X^2}\cdot r^2\right)^{-\frac{N_m+d}{2}}. \tag{3.86}$$

The protection level can then be computed.

$$\alpha=\int_{B_m}^{\infty} f_{\|\mathbf{\Theta}\|}\left(r\right)\cdot dr \tag{3.87}$$

$$=\frac{2}{Beta\left(\frac{d}{2},\frac{N_m}{2}\right)}\int_{B_m}^{\infty}\frac{r^{d-1}}{N_m^{\frac{d}{2}}\cdot\left(\sigma_X^2\right)^{\frac{d}{2}}}\cdot\left(1+\frac{1}{N_m\sigma_X^2}\cdot r^2\right)^{-\frac{N_m+d}{2}}dr. \tag{3.88}$$

Using the variable change $y=\frac{r}{\sqrt{N_m\sigma_X^2}}$, $dy=\frac{1}{\sqrt{N_m\sigma_X^2}}dr$

$$\alpha=\frac{2}{Beta\left(\frac{d}{2},\frac{N_m}{2}\right)}\int_{\frac{B_m}{\sqrt{N_m\sigma_X^2}}}^{\infty}\frac{y^{d-1}\cdot\left(N_m\sigma_X^2\right)^{\frac{d-1}{2}}}{N_m^{\frac{d}{2}}\cdot\left(\sigma_X^2\right)^{\frac{d}{2}}}\cdot\left(1+y^2\right)^{-\frac{N_m+d}{2}}\cdot\left(N_m\sigma^2\right)^{\frac{1}{2}}\cdot dy \tag{3.89}$$

$$=\frac{2}{Beta\left(\frac{d}{2},\frac{N_m}{2}\right)}\int_{\frac{B_m}{\sqrt{N_m\sigma_X^2}}}^{\infty}\frac{y^{d-1}}{\left(1+y^2\right)^{\frac{N_m+d}{2}}}\cdot dy. \tag{3.90}$$

Finally, by defining $k=\frac{B_m}{\sqrt{N_m\sigma_X^2}}$, the protection level is computed as

$$B_m=k\sqrt{N_m\sigma_X^2}, \tag{3.91}$$

with k solution of

$$\alpha=\frac{2}{Beta\left(\frac{d}{2},\frac{N_m}{2}\right)}\int_{k}^{\infty}\frac{y^{d-1}}{\left(1+y^2\right)^{\frac{N_m+d}{2}}}\cdot dy. \tag{3.92}$$

3.5.6 Computing the protection level

The aforementioned method works in parallel of the Kalman filter without interfering with it. It uses the filter's residual (y_m), the measurement covariance matrix (S_m) and the estimate covariance matrix to build a t-distribution that best model the estimate error. Knowing this distribution's parameters enable us to compute bounds for each measurement type.

Finally, after having computed the bounds for all measurement types, the protection level is computed by adding them.

$$PL = \sum_{m} B_m. \tag{3.93}$$

3.6 KIPL Simulation Results

This method was first tested with a one-dimensional example. The simulation setup was the same as for the IBPL method with this time the mobile following a sinusoidal trajectory. Again the method has been tested for integrity risks (α) from 10^{-2} to 10^{-5}.

The simulation shows that after a convergence time of a few hundred epochs (see figure 3.8), the protection level stays almost constant to a certain value. This behavior is probably due to the measurement error generated by the simulation having a constant standard deviation. Therefore, once the Kalman filter has converged to a solution, the covariance matrices are almost constant which in turn result in a protection level that converges to a constant value.

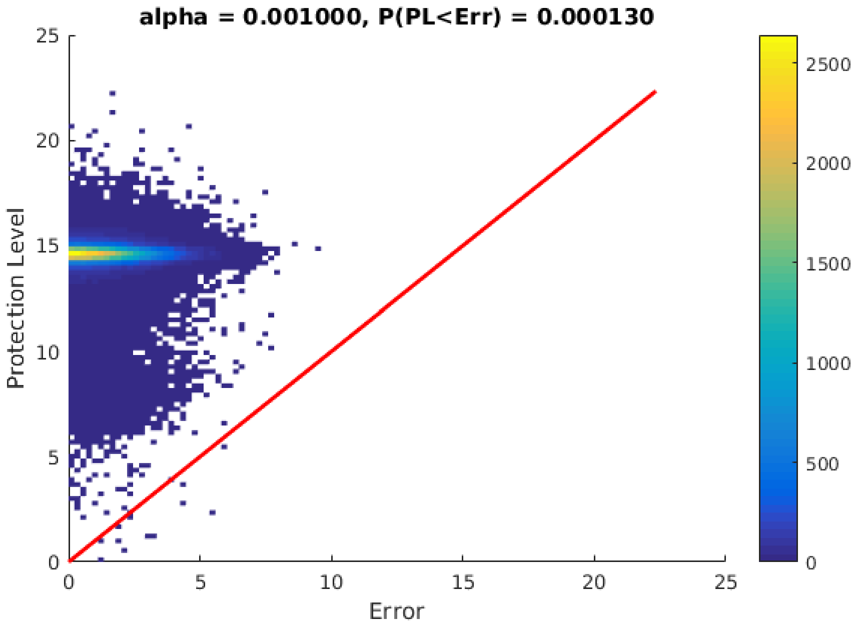


Figure 3.7: One-dimensional KIPL Stanford diagram.

Multipath has also been added to the simulation. figure 3.9 (a) shows the result for multipath applied to the second half of the simulation. The two phases can be clearly distinguished and the same convergence behavior is observed.

3.6.1 Conclusion

Just like the IBPL method, this technique provides Protection Levels that satisfy $P\left(Error > PL\right) \leq TIR$. However, no improvement has been observed concerning the size of those protection levels. Even though I have been able to understand and justify the general idea of the method there are still a lot that is unexplained. In particular the inventors of the method claim it can be used to provide different protection levels for every dimensions. This aspect would be worth investigating since in autonomous driving application the cross track protection level is more critical than the along track.

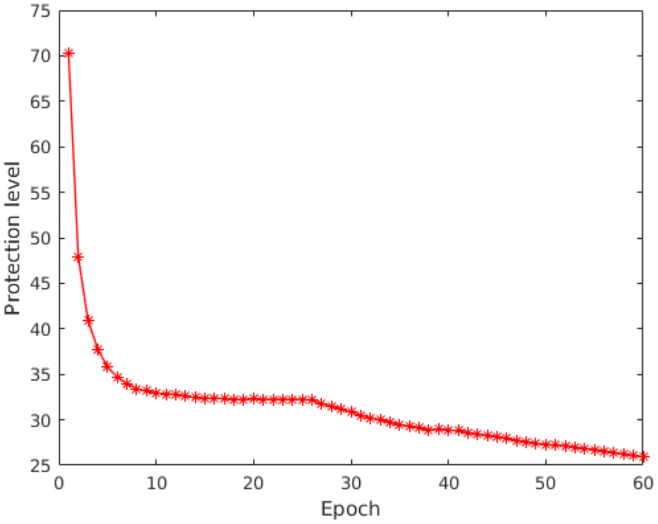
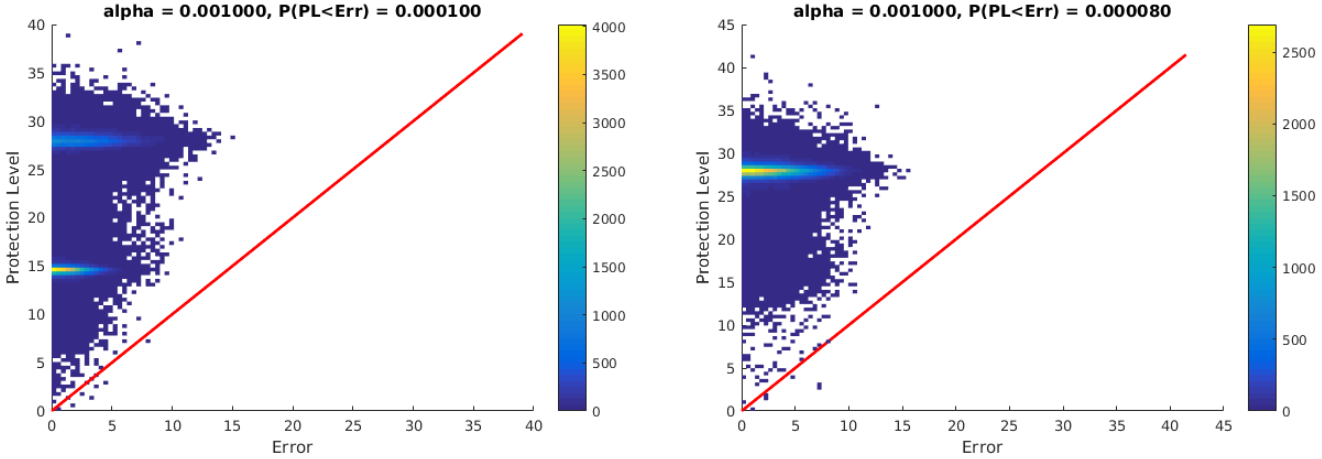


Figure 3.8: Convergence of One-dimensional KIPL



(a)

(b)

Figure 3.9: One-dimensional KIPL Stanford diagram with multipath. (a) Multipath for half the epochs. (b) Multipath for all the epochs.

Chapter 4

Experimental Results

To verify the integrity of the algorithms, they have been applied to experimental data. This data was recorded using the two Renault Zoe from the Heudiasyc laboratory. The experiments were conducted on July 27th 2017 on public roads in Compiègne, France. The algorithms were then used in post processing on the datasets.

Figure 4.1: Heudiasyc laboratory APAChE electric cars.

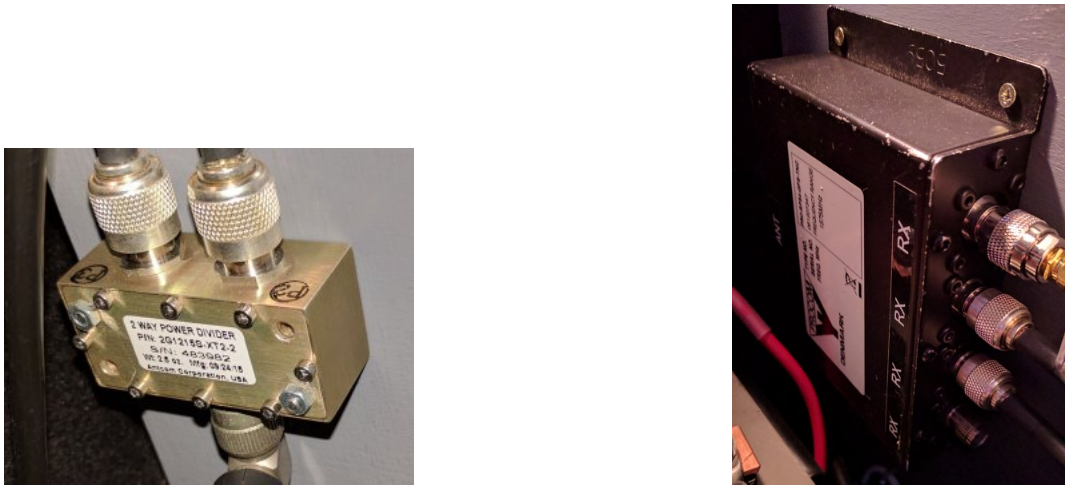
4.1 Platform description

The Heudiasyc laboratory has two Renault Zoe equipped for autonomous driving. They have LiDARs, cameras and GNSS receivers.

As a ground truth, a SPAN-CPT from Novatel was used. It combines a GNSS receiver and Inertial Measurement Unit (IMU). The GNSS receiver was used in RTK mode with corrections transmitted through the 4G mobile network guarantying the most accurate positioning. The IMU uses fiber optic gyros (FOG) and Electromechanical System (MEMS). This system provided the ground truth with a frequency of 100Hz.

The GNSS unit in production car will be much cheaper than the SPAN-CPT system. For these experiments, a Ublox EVK-M8T was used. It is capable of processing multi-GNSS signals which significantly increases the number of available satellites (usually above 15). This unit provided measurements at 2Hz.

To have the same antenna for both systems (and thus the same position), splitters have been used. Tests were conducted to determine the effect of adding the devices. The Signal-to-Noise Ratio (SNR) was used to measure the quality of the signal during static recordings on the laboratory's test track. The results show no influence of the splitters on the quality of the signal they were therefore used for the recordings.

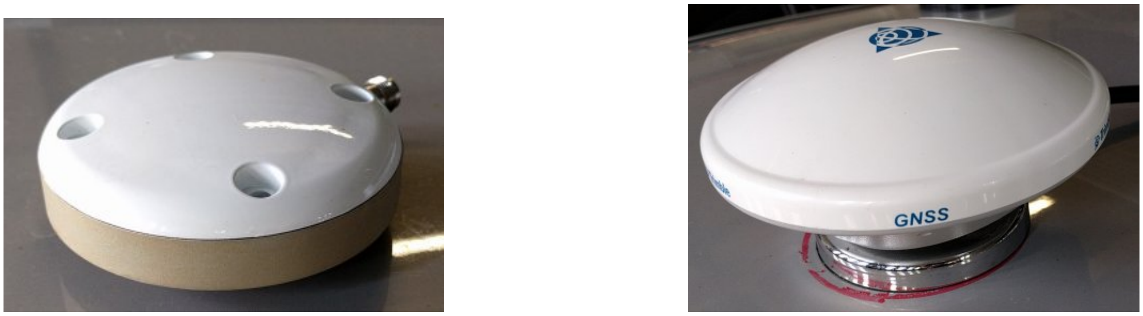


(a) Splitter 1.

(b) Splitter 2.

Figure 4.2: GNSS signal splitters.

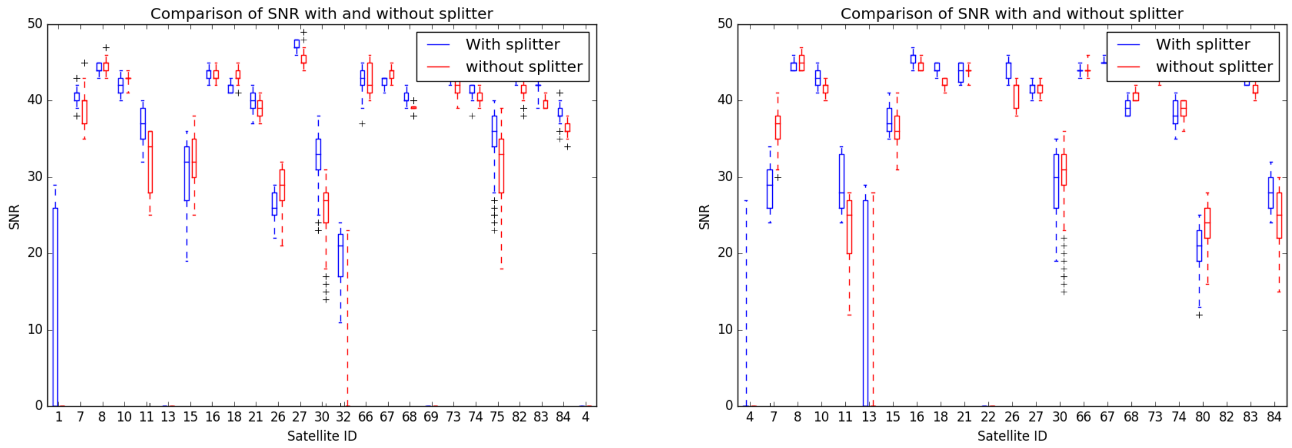
The same testing method was used to select between the Antcom antenna and the Trimble antenna. The results show a signal-to-noise ratio equivalent for both antenna designs. However, during testing, the Trimble antenna add noticeably higher SNRs than the other and figure 4.5 shows both higher SNRs for some satellites and worse for others. This result might be due to the antenna design being more robust to multipath signal. The Trimble antenna was therefore used for the recordings.



(a) Antcom antenna.

(b) Trimble antenna.

Figure 4.4: GNSS antennas.



(a) Splitter 1.

(b) Splitter 2.

Figure 4.3: Comparison of the SNR with and without the two splitters

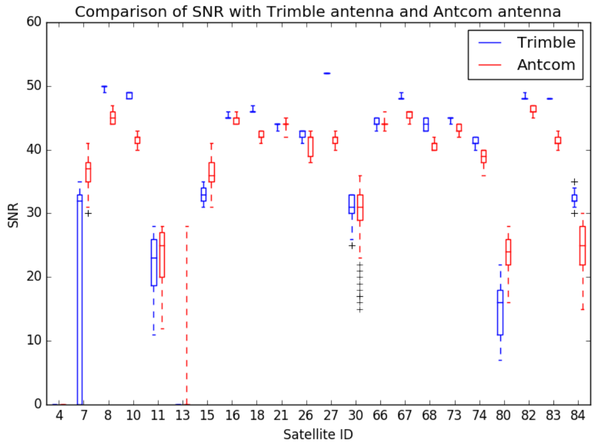


Figure 4.5: Comparison of the SNR for the two antennas.

4.2 Robot Operating System (ROS)

4.2.1 Overview

To handle the software development, the middleware ROS is used. It allows for modular development of programs by handling the interfaces between the blocks of the program. Each node in the program communicates using messages sent over a publisher/subscriber framework. This architecture allows for easy recording and replaying of sensory inputs which make possible the development and testing of nodes offline.

The main advantage of ROS compared to other middlewares is its community. Hundreds of programs made by the community are available online which makes it easy to develop for common component.

4.2.2 Experimental setup

To acquire the two GPSs data, existing nodes have been used. The ublox node was modified to parse additional binary messages from the Ublox receiver. To record the datasets, only the sensors, visualization and diagnostic nodes were running. The two nodes computing the protection levels for the IBPL and KIPL methods were used in a post processing phase using the recorded data.

4.3 Computed Integrity Bounds

On July 27th several routes have been driven in order to produce a variety of datasets. The resulting datasets of both cars have been combined into two datasets on which the algorithms have been applied. They were not separated depending on the type of environment (open sky or high multipath) since the number of available epochs does not make those characteristic distinguishable. However, the data from the two cars have been processed separately since the localization quality on one of the cars was significantly worse than on the other.

4.3.1 The IBPL method

To compute the Protection Levels using the IBPL method, the norm of the residual vector of the Least Square Estimation is required. However, the Ublox receiver uses Kalman filtering to provide the receiver position. I therefore started by recomputing the position using the raw measurements. To get the satellite's position, the IGS predicted orbits were used (see section 2.4.1). Unfortunately, I did not manage to obtain a coherent position using the raw measurements. The Isotropy Based Protection Levels where therefore computed using the residual from the ublox Kalman filter. The settings were chosen in order to apply the least amount of filtering to obtain Protection Levels as consistent as possible. This approximation should only make the protection levels larger than they are supposed to be as the Least Square Estimation would have minimized the residual norm.

The result of the experiment is shown in figure 4.6.

4.3.2 The KIPL method

The KIPL method is not designed to fully integrate to the Kalman filter, but rather works in parallel using its variables. Therefore, it is possible to compute protection levels without having to start from raw measurements. Moreover the Ublox-M8T provides proprietary messages that contain all the information needed to apply the algorithm.

4.3.3 The 4.4σ method

To show the advantages two previous methods offer compared to traditional methods, a third protection level was computed using the standard deviation of the localization solution. A protection level of 4.417173σ corresponds to a percentile of 99.999% therefore it should also satisfy $P\left(Error > PL\right) \leq \alpha$.

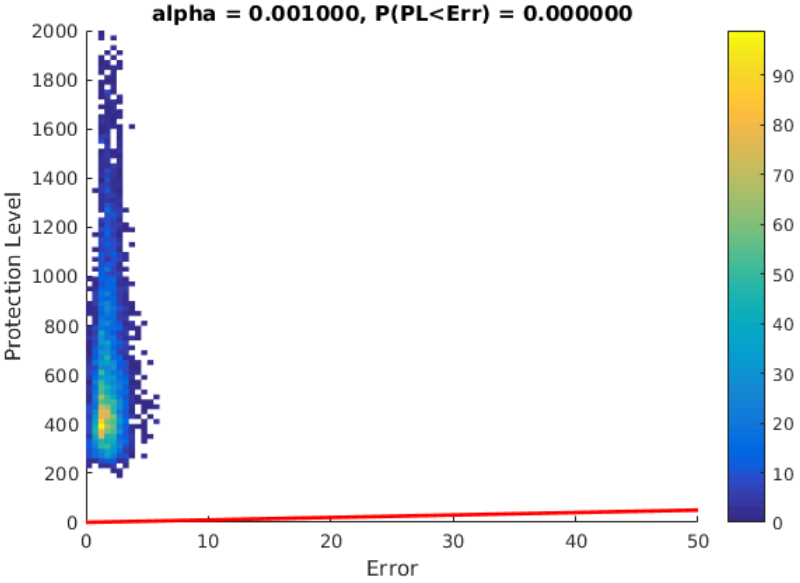
4.3.4 Comparison

The results presented in figure 4.6 shows that applying the IBPL method with residuals obtained using Kalman filtering is not feasible. Therefore, using this method would require using a Least Square Estimator which would result in a worse localization.

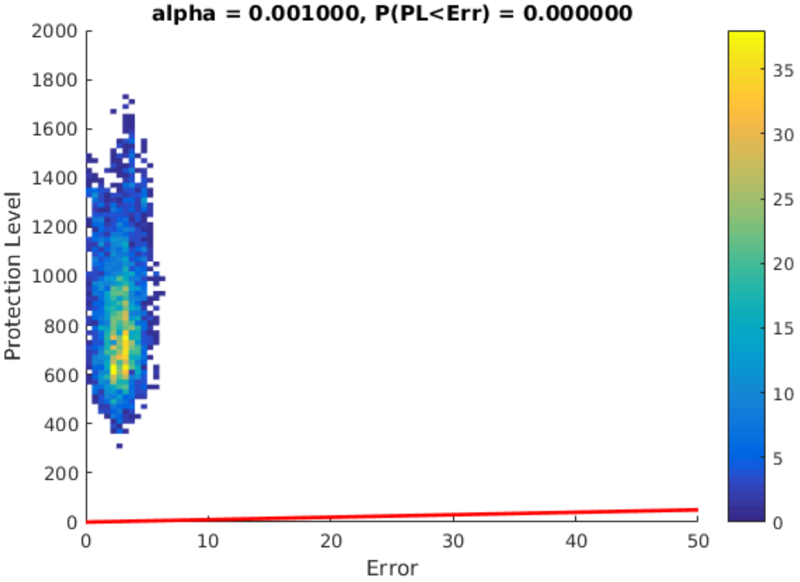
The KIPL method, however (see figure 4.7), shows satisfying results. More data would be required to guaranty that it verifies $P\left(Error > PL\right) \leq \alpha$, but the results from the dataset indicated that it does. The protection levels computed are a bit pessimistic, hence there is still room for improvement, but they are reasonably small to work with.

The last method presented in figure 4.8 is a clear example that traditional methods to compute protection levels are not usable for autonomous driving application. Indeed the method does not guarantee the integrity of the localization ($P\left(Error > PL\right) > \alpha$). An localization accuracy too optimistic will result in protection levels that cannot be trusted.

Note. Even though the two cars drove in the same environment, a noticeable difference in their accuracy can be seen. The cause of this difference is yet to be determined. It is possibly due to the Velodyne LiDAR installed on the roof of the Gray car blocking or reflecting some GNSS signals.

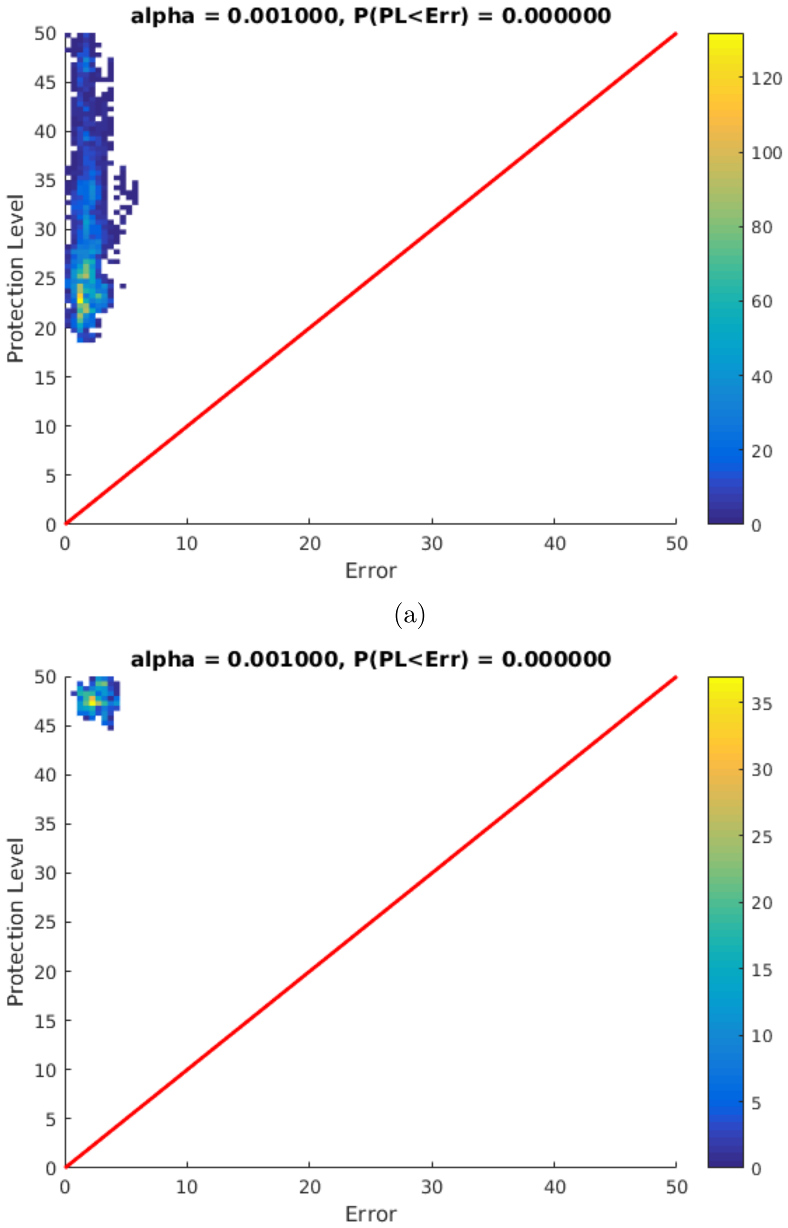


(a)



(b)

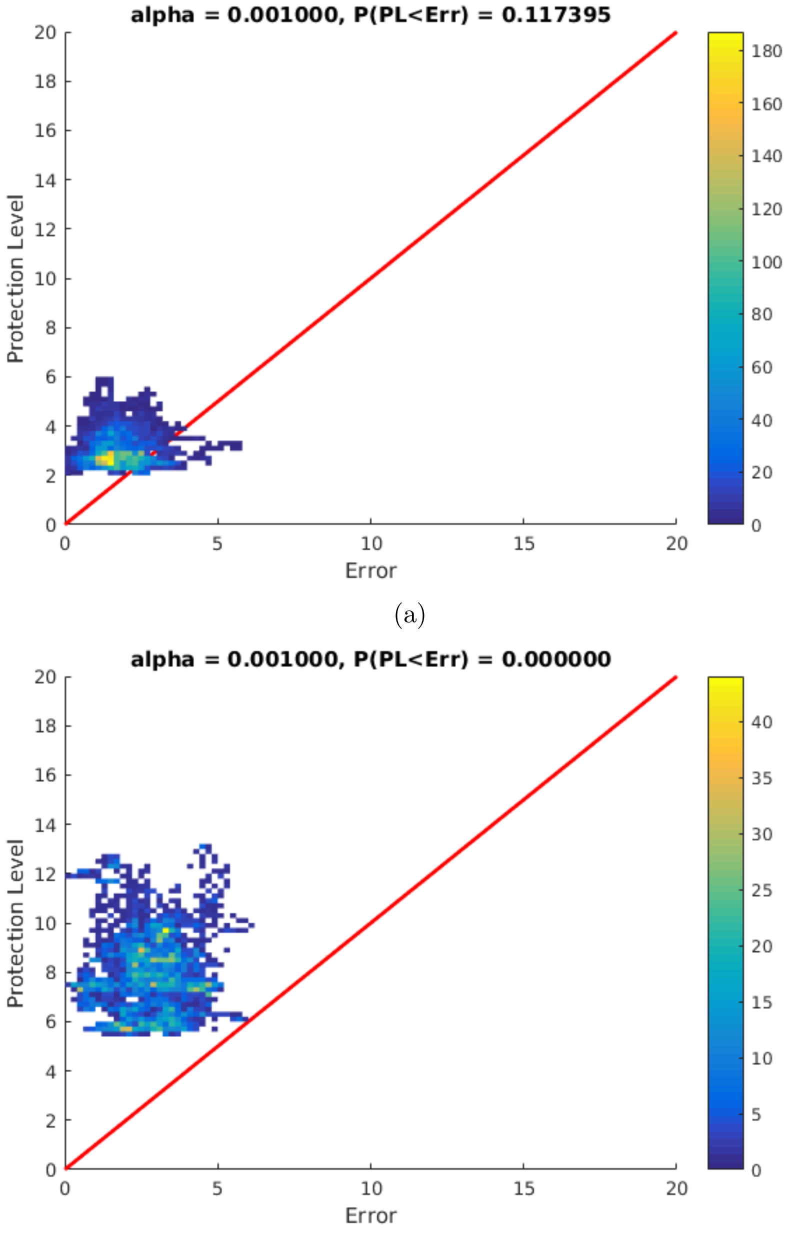
Figure 4.6: IBPL Stanford diagrams for experimental tests. (a) White vehicle. (b) Gray vehicle.



(a)

(b)

Figure 4.7: KIPL Stanford diagrams for experimental tests. (a) White vehicle. (b) Gray vehicle.



(a)

(b)

Figure 4.8: Stanford diagrams for experimental tests with 4.4σ. (a) White vehicle. (b) Gray vehicle.

Chapter 5

Conclusion

The objectives of this internship were to study and test solutions to accuracy and integrity problems for localization. In particular, I have worked on a method to increase the localization accuracy, Precise Point Positioning (PPP). My work also aimed to explain and test two integrity methods, Isotropy Based Protection Level (IBPL) and Kalman Integrated Protection Level (KIPL).

The result of this work is first an understanding of the Precise Point Positioning method and its advantages and limitations. Furthermore, I have managed to explain the reasoning behind the Isotropy Based Protection Level (IBPL) and was able to implement this method and test it in simulations. The simulation results show that it is a robust technique, however, this is not yet confirmed by experimental results. I have also reached a general understanding of the Kalman Integrated Protection Level (KIPL). I was able to implement and test this method both in simulation and on experimental data. This work has shown the robustness of this method and its interest, since the provided protection level are of reasonable size.

Even though work still needs to be done to fully justify every aspect of the KIPL method, the objectives of this internship have been reached. The methods are understood and the algorithms have been implemented and tested.

Future work. Several approaches can be considered to continue this work. The integrity methods were not tested using PPP. The implementation of this method would allow for better accuracy, therefore it would probably result in smaller protection levels.

The IBPL results could also be improved. Processing multiple epochs at once would introduce a form of filtering, which might help reduce the size of the protection levels.

Finally, the ability of the KIPL method to process multiple measurement types could be exploited in order to improve its results. In particular adding camera or lane detecting sensors could reduce significantly the size of the protection level.

Bibliography

[1] 2008 us federal radionavigation plan, dot-vntsc-rita-08-02/dod-4650.5. Technical report, Department of Defense, Department of Homeland Security, and Department of Transportation, 2008.

[2] C. Burgi A. Somieski, E. Favey. Precise point positioning with single-frequency mass market receivers. *Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008)*, pages 1429–1436, September 2008.

[3] GMV AEROSPACE and DEFENCE S A. Method for autonomous determination of protection levels for gnss positioning based on navigation residuals and an isotropic confidence ratio, April 2008.

[4] GMV AEROSPACE and DEFENCE S A. Device and method for computing an error bound of a kalman filter based gnss position solution, April 2016.

[5] Y. Chen, J. Liu, L. Ruotsalainen, L. Chen, T. Kröger, H. Kuusniemi, L. Pei, R. Chen, T. Tenhunen, and Y. Wang. Lane detection based on a visual-aided multiple sensors platform. In *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, pages 740–747, April 2012.

[6] Christopher J. Hegarty Elliott D. Kaplan. *Understanding GPS, Principles and Applications (second edition)*. Artech House, 2005.

[7] V. L. Knoop, P. F. de Bakker, C. C. J. M. Tiberius, and B. van Arem. Lane determination with gps precise point positioning. *IEEE Transactions on Intelligent Transportation Systems*, PP(99):1–11, 2017.

[8] Chen Kongzhe and Gao Yang. Real-time precise point positioning using single frequency data,. *Proceedings of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2005)*, pages 1514–1523, September 2005.

[9] Dennis D. McCarthy and Gérard Petit. Iers technical note no. 32. Technical report, International Earth Rotation and Reference Systems Service (IERS), 2003.

[10] Junbo Shi. *Precise Point Positioning Integer Ambiguity Resolution with Decoupled Clocks*. PhD thesis, Department of Geomatics Engineering, University of Calgary, December 2012.

[11] Joachim Feltens Stefan Schaer, Werner Gurtner. *IONEX: The IONosphere Map EXchange Format Version 1*. Astronomical Institute, University of Berne, Switzerland & ESA/ESOC, Darmstadt, Germany, February 1998.

[12] G. Tobías, J.D. Calle, A.J. García, D. Luque, and I. Rodríguez. magicgnssŕtcm-based service, a leap forward towards multi-gnss high accuracy real-time processing. *Proceedings of the 28th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2015)*, September 2015.

[13] G. Tobías, J.D. Calle, P. Navarro, I. Rodríguez, and D. Rodríguez. magicgnssŕeal-time pod and ppp multi-gnss service,. *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014)*, pages 1046–1055, September 2014.

[14] Rikkert Jan Wienia. *Use of Global Ionospheric Maps for Precise Point Positioning*. PhD thesis, Delft University of Technology, December 6 2008.

[15] Gao Y., Zhang Y., and Chen K. Development of a real-time single-frequency precise point positioning system and test results. *Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2006)*, pages 2297–2303, September 2006.

Appendix A

Precise Point Positioning (PPP)

A.1 Klobuchar ionospheric delay[6]

Algorithm A.1 Klobuchar ionospheric delay.

```
$\psi = \frac{0.0137}{E + 0.11} - 0.022$   (semicircles)   ▷ Earth centred angle (elevation E in semicircles)
$\phi_I = \phi_u + \psi \cos A$   (semicircles)   ▷ Latitude of the IPP
if $\Phi_I > +0.416$ then
    $\Phi_I = +0.416$
else if $\Phi_I < -0.416$ then
    $\Phi_I = -0.416$
end if
$\lambda_I = \lambda_u + \frac{\psi \sin A}{\cos \phi_I}$   (semicircles)   ▷ Longitude of the IPP
$\phi_m = \phi_I + 0.064 \cos(\lambda_I - 1.617)$   (semicircles)   ▷ Geomagnetic latitude of the IPP
$t = 43\,200\,\lambda_I + t_{GPS}$(seconds)   ▷ Local time of the IPP
while $t < 0$ do
    $t = t + 86400$
end while
while $t \geq 86400$ do
    $t = t + 86400$
end while
$A_I = \sum_{n=0}^{3} \alpha_n \phi_m^n$   (seconds)   ▷ Amplitude of the ionospheric delay
if $A_I < 0$ then
    $A_I = 0$
end if
$P_I = \sum_{n=0}^{3} \beta_n \phi_m^n$   (seconds)   ▷ Period of the ionospheric delay
if $P_I < 72\,000$ then
    $P_I = 72\,000$
end if
$X_I = \frac{2\pi(t - 50\,400)}{P_I}$   (radians)   ▷ Phase of the ionospheric delay
$F = 1.0 + 16.0\,(0.53 - E)^3$   ▷ Slant factor
$I_{L1_{GPS}} = \begin{cases} \left[5 \cdot 10^{-9} + \sum_{n=0}^{3} \alpha_n \phi_m^n \cdot \left(1 - \frac{X_I^2}{2} + \frac{X_I^4}{24}\right)\right] \cdot F & ; |X_I| \leq 1.57 \\ 5 \cdot 10^{-9} \cdot F & ; |X_I| \geq 1.57 \end{cases}$   ▷ Ionospheric time delay
```

A.2 Ionospheric Pierce Point (IPP)[6]

Earth central angle between the user and the Ionospheric Pierce Point.

$$\psi_{pp} = \frac{\pi}{2} - E - sin^{-1}\left(\frac{R_e}{R_e + h} \cdot \cos(E)\right). \tag{A.1}$$

Latitude of the Ionospheric Pierce Point:

$$\phi_{pp} = \sin^{-1}\left(\sin(\phi_u)\cos(\psi_{pp}) + \cos(\phi_u)\sin(\psi_{pp})\cos(A)\right). \tag{A.2}$$

Longitude of the Ionospheric Pierce Point:

$$\text{if } \phi_u > 70° \text{ and } \tan(\psi_{pp})\cos(A) > \tan\left(\frac{\pi}{2} - \phi_u\right), \tag{A.3}$$

$$\text{or if } \phi_u\text{<-70° and tan(}\psi_{pp}\text{)cos(A)<tan}\left(\frac{\pi}{2} - \phi_u\right), \tag{A.4}$$

$$\lambda_{pp} = \lambda_u + \sin^{-1}\left(\frac{\sin(\psi_{pp}) \cdot \sin(A)}{\cos(\phi_{pp})}\right)\text{, otherwise,} \tag{A.5}$$

$$\lambda_{pp} = \lambda_u + \pi - \sin^{-1}\left(\frac{\sin(\psi_{pp}) \cdot \sin(A)}{\cos(\phi_{pp})}\right), \tag{A.6}$$

where,

ψ_{pp}	Earth central angle between the user and the Ionospheric Pierce Point,
E	elevation of the satellite,
A	azimuth of the satellite,
R_e	radius of the Earth,
h	altitude of the IPP,
ϕ_u	latitude of the user,
λ_u	longitude of the user.

Appendix B

IBPL

B.1 Values of k for a four-dimensional problem

The following array has been computed using Matlab. I solved the equation (3.39) by using the fzero function for different values of Target Integrity Risk (α) and for different numbers of satellites (m).

m$\backslash\alpha$	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}
5	14.9442913	149.994444	1499.99949	15000.2806	150000.0	1500000.0	15000000.0
6	4.29963172	14.0889582	44.7045844	141.416052	447.211918	1414.21303	4472.13578
7	2.66899343	6.18707237	13.5203729	29.2162233	62.9848554	135.722836	293.240772
8	2.02663502	3.99712707	7.30998147	13.1099496	23.3749635	41.6018915	73.9992717
9	1.67814093	3.01886443	4.98678297	8.02629677	12.7971816	20.3300751	32.2511600
10	1.45619661	2.46958849	3.82304776	5.73984077	8.51051784	12.5494746	18.4592747
11	1.30066666	2.11749785	3.13487240	4.48629698	6.32487958	8.85309245	12.3475619
12	1.18457285	1.87164054	2.68249739	3.70761619	5.03861106	6.78858815	9.10430891
13	1.09395918	1.68945559	2.36270266	3.18092185	4.20443259	5.50294813	7.16288418
14	1.02084992	1.54846229	2.12440594	2.80214573	3.62448091	4.63776444	5.89705703
15	0.960338620	1.43568801	1.93964532	2.51693013	3.19978254	4.02094927	5.01776094

B.2 Least Square Estimation properties

If $\boldsymbol{\epsilon}$ is the measurement error, $\mathbf{r}$ the least square estimation residual and $\mathbf{s} = H \cdot \mathbf{e}$ the image of the estimation error $\mathbf{e} = \hat{\mathbf{x}} - \mathbf{x}$ through H.

Proposition. $\boldsymbol{\epsilon} = \mathbf{r} + \mathbf{s}$.

Proof.

$$\begin{aligned}
\mathbf{r} &= \mathbf{y} - H \cdot \hat{\mathbf{x}} && \text{(B.1)}\\
&= H \cdot \mathbf{x} + \boldsymbol{\epsilon} - H \cdot \hat{\mathbf{x}} && \text{(B.2)}\\
&= \boldsymbol{\epsilon} - H \cdot (\hat{\mathbf{x}} - \mathbf{x}) && \text{(B.3)}\\
&= \boldsymbol{\epsilon} - H \cdot \mathbf{e} && \text{(B.4)}\\
&= \boldsymbol{\epsilon} - \mathbf{s}. && \text{(B.5)}
\end{aligned}$$

□

Proposition. $\mathbf{s}^T \cdot \mathbf{r} = 0$.

Proof.

$$\mathbf{s}^T \cdot \mathbf{r} = (\hat{\mathbf{x}} - \mathbf{x})^T \cdot H^T \cdot (\mathbf{y} - H \cdot \hat{\mathbf{x}}) \tag{B.6}$$

$$= (\hat{\mathbf{x}} - \mathbf{x})^T \cdot H^T \cdot (I - H \cdot (H^T \cdot H)^{-1} \cdot H^T) \cdot \mathbf{y} \tag{B.7}$$

$$= (\hat{\mathbf{x}} - \mathbf{x})^T \cdot H^T \cdot \mathbf{y} - (\hat{\mathbf{x}} - \mathbf{x})^T \cdot H^T \cdot \mathbf{y} \tag{B.8}$$

$$= 0. \tag{B.9}$$

□

Proposition. $\boldsymbol{\epsilon}^2 = \mathbf{r}^2 + \mathbf{s}^2$.

Proof. Because $\mathbf{r}$ and $\mathbf{s}$ are orthogonal, Pythagoras theorem assures $\boldsymbol{\epsilon}^2 = \mathbf{r}^2 + \mathbf{s}^2$. □

B.3 Useful theorems

Theorem 5. *If A is a real symmetric matrix and $\lambda_{min}(A)$ is its smallest eigenvalue.*

$$\forall \mathbf{x} \in \mathbb{R}^m, \qquad \mathbf{x}^T A \mathbf{x} \geq \lambda_{min}(A) \|\mathbf{x}\|^2 . \tag{B.10}$$

Proof. If A is a real symmetric matrix, $\exists U$ so that $U^T U = I$ and $U A U^T = \Lambda$ where Λ is diagonal.

$$\begin{aligned}
\mathbf{x}^T A \mathbf{x} &= \langle \mathbf{x}, A\mathbf{x} \rangle \\
&= \left\langle \mathbf{x}, U^T \Lambda U \mathbf{x} \right\rangle \\
&= \langle U\mathbf{x}, \Lambda U \mathbf{x} \rangle && (\|U\mathbf{x}\| = \|\mathbf{x}\|) \\
&= \sum_k \lambda_k \left[U\mathbf{x}\right]_k^2 \\
&\geq \lambda_{min} \sum_k \left[U\mathbf{x}\right]_k^2 \\
&\geq \lambda_{min} \|U\mathbf{x}\|^2 \\
&\geq \lambda_{min} \|\mathbf{x}\|^2 .
\end{aligned}$$

□

Theorem 6. *If X is a random variable, and $Y = u(X)$, with $\forall x,\ u(x) \leq x$, then*

$$F_X(x) \leq F_Y(x), \tag{B.11}$$

or

$$P(X \leq x) \leq P(Y \leq x). \tag{B.12}$$

Proof. Let $Z = X - Y = X - u(X)$, because $\forall x,\ u(x) \leq x$, the random variable Z is positive.
Therefore, $P(Y \leq x) = P(X - Z \leq x) = P(X \leq x + Z) \geq P(X \leq x)$. □

B.4 Integration by parts

The following equations describe the steps used to simplify equation (3.45) and remove the integration.

$$\int_0^{\sqrt{\frac{k^2}{1+k^2}}} r^3 \cdot \left(1-r^2\right)^{\frac{m-6}{2}} \cdot dr = \int_0^{\sqrt{\frac{k^2}{1+k^2}}} r^2 \cdot \frac{-2r}{-2} \cdot \frac{\frac{m-4}{2}}{\frac{m-4}{2}} \cdot \left(1-r^2\right)^{\frac{m-6}{2}} \cdot dr \tag{B.13}$$

$$= \int_0^{\sqrt{\frac{k^2}{1+k^2}}} r^2 \cdot \frac{-2r}{-2} \cdot \frac{\frac{m-4}{2}}{\frac{m-4}{2}} \cdot \left(1-r^2\right)^{\frac{m-6}{2}} \cdot dr \tag{B.14}$$

$$= \left[r^2 \cdot \frac{\left(1-r^2\right)^{\frac{m-4}{2}}}{-2 \cdot \frac{m-4}{2}} \right]_0^{\sqrt{\frac{k^2}{1+k^2}}} - \int_0^{\sqrt{\frac{k^2}{1+k^2}}} \text{-2r} \cdot \frac{\left(1-r^2\right)^{\frac{m-4}{2}}}{m-4} \cdot dr \tag{B.15}$$

$$= -\frac{\frac{k^2}{1+k^2} \cdot \left(1-\frac{k^2}{1+k^2}\right)^{\frac{m-4}{2}}}{m-4} + \frac{2}{m-4} \int_0^{\sqrt{\frac{k^2}{1+k^2}}} r \cdot \left(1-r^2\right)^{\frac{m-4}{2}} \cdot dr \tag{B.16}$$

$$= -\frac{\frac{k^2}{1+k^2} \cdot \left(1-\frac{k^2}{1+k^2}\right)^{\frac{m-4}{2}}}{m-4} + \frac{2}{m-4} \left[\frac{\left(1-r^2\right)^{\frac{m-2}{2}}}{-2 \cdot \frac{m-2}{2}} \right]_0^{\frac{k^2}{1+k^2}} \tag{B.17}$$

$$= -\frac{\frac{k^2}{1+k^2} \cdot \left(1-\frac{k^2}{1+k^2}\right)^{\frac{m-4}{2}}}{m-4} + \frac{2}{(m-4)(m-2)} - \frac{2\left(\frac{1}{1+k^2}\right)^{\frac{m-2}{2}}}{(m-4)(m-2)}. \tag{B.18}$$

So,

$$\int_0^{\sqrt{\frac{k^2}{1+k^2}}} r^3 \cdot \left(1-r^2\right)^{\frac{m-6}{2}} \cdot dr = \frac{1}{2} \frac{\Gamma\left(\frac{m-4}{2}\right)}{\Gamma\left(\frac{m}{2}\right)} \cdot (1-\alpha) \tag{B.19}$$

$$\Longleftrightarrow -\frac{\frac{k^2}{1+k^2} \cdot \left(1-\frac{k^2}{1+k^2}\right)^{\frac{m-4}{2}}}{m-4} + \frac{2}{(m-4)(m-2)} - \frac{2\left(\frac{1}{1+k^2}\right)^{\frac{m-2}{2}}}{(m-4)(m-2)} = \frac{1}{2} \frac{\Gamma\left(\frac{m-4}{2}\right)}{\Gamma\left(\frac{m}{2}\right)} \cdot (1-\alpha) \tag{B.20}$$

$$\Longleftrightarrow -\frac{\frac{k^2}{1+k^2} \cdot \left(1-\frac{k^2}{1+k^2}\right)^{\frac{m-4}{2}}}{m-4} + \frac{2}{(m-4)(m-2)} - a\frac{2\left(\frac{1}{1+k^2}\right)^{\frac{m-2}{2}}}{(m-4)(m-2)} = \frac{1}{2} \cdot \frac{1}{\left(\frac{m-4}{2}\right)\left(\frac{m-2}{2}\right)} \cdot (1-\alpha) \tag{B.21}$$

$$\Longleftrightarrow -\frac{(m-2)\frac{k^2}{1+k^2} \cdot \left(1-\frac{k^2}{1+k^2}\right)^{\frac{m-4}{2}}}{2} + 1 - \left(\frac{1}{1+k^2}\right)^{\frac{m-2}{2}} = (1-\alpha) \tag{B.22}$$

$$\Longleftrightarrow \frac{(m-2)\frac{k^2}{1+k^2} \cdot \left(\frac{1}{1+k^2}\right)^{\frac{m-4}{2}}}{2} + \left(1+k^2\right)^{\frac{2-m}{2}} = \alpha \tag{B.23}$$

$$\Longleftrightarrow \frac{(m-2) \cdot k^2 \cdot \left(1+k^2\right)^{\frac{2-m}{2}}}{2} + \left(1+k^2\right)^{\frac{2-m}{2}} = \alpha \tag{B.24}$$

$$\Longleftrightarrow \left(1+k^2\right)^{\frac{2-m}{2}} \cdot \left(\frac{m-2}{2} \cdot k^2 + 1\right) = \alpha \tag{B.25}$$

$$\Longleftrightarrow \left(1+k^2\right)^{\frac{2-m}{2}} \cdot \left(\frac{m-2}{2} \cdot \left(1+k^2\right) - \frac{m-4}{2}\right) = \alpha \tag{B.26}$$

$$\Longleftrightarrow \frac{m-2}{2} \cdot \left(1+k^2\right)^{\frac{4-m}{2}} - \frac{m-4}{2} \cdot \left(1+k^2\right)^{\frac{2-m}{2}} = \alpha. \tag{B.27}$$