

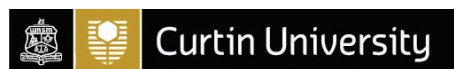
# **Ps-LAMBDA software package**

**Matlab implementation, Version 1.0**

**Sandra Verhagen and Bofeng Li**



Mathematical Geodesy and Positioning, Delft University of Technology



GNSS Research Centre, Curtin University

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Application of Ps-LAMBDA software . . . . .	5
1.2	Disclaimer . . . . .	5
1.3	Outline . . . . .	5
<b>2</b>	<b>Integer estimation methods and their pull-in regions</b>	<b>6</b>
2.1	Parameter estimation . . . . .	6
2.2	Admissible integer estimation . . . . .	9
2.2.1	Z-transformation . . . . .	10
2.2.2	Integer rounding . . . . .	10
2.2.3	Integer bootstrapping . . . . .	11
2.2.4	Integer least squares . . . . .	13
<b>3</b>	<b>Success rate of integer estimation</b>	<b>16</b>
3.1	Definition of success rate . . . . .	16
3.2	Monte Carlo simulation based approximate success rate . . . . .	17
3.3	Success rate and its bounds . . . . .	18
3.3.1	IR success rate and its bounds . . . . .	19
3.3.2	IB success rate and its bounds . . . . .	19
3.3.3	ILS success rate and its bounds . . . . .	20
<b>4</b>	<b>Routines and their usage</b>	<b>24</b>
4.1	Overview of Ps-LAMBDA software . . . . .	24
4.2	The main Ps-LAMBDA routine . . . . .	25
4.2.1	Input arguments . . . . .	25
4.2.2	ILS success rate (SRILS.m) . . . . .	25

4.2.3	IB success rate (SRBoot.m) . . . . .	26
4.2.4	IR success rate (SRRound.m) . . . . .	26
4.3	Routines used by Ps-LAMBDA . . . . .	26
<b>5</b>	<b>Getting started and performance aspects</b>	<b>27</b>
5.1	Getting started . . . . .	27
5.2	Performance aspects . . . . .	28
5.2.1	IR success rate . . . . .	28
5.2.2	IB success rate . . . . .	29
5.2.3	ILS success rate . . . . .	29
5.2.4	Examples with other models . . . . .	31
5.2.5	Which bounds or approximations to use? . . . . .	32
<b>6</b>	<b>Availability, Liability and Updates</b>	<b>34</b>
6.1	Availability . . . . .	34
6.2	Liability . . . . .	34
6.3	Updates . . . . .	34
	<b>Bibliography</b>	<b>34</b>

# Chapter 1

## Introduction

Integer ambiguity resolution is the process of estimating the unknown ambiguities of carrier-phase observables as integers. It applies to a wide range of interferometric applications, of which Global Navigation Satellite System (GNSS) precise positioning is a prominent example. GNSS precise positioning can be accomplished anytime and anywhere on Earth, provided that the integer ambiguities are successfully resolved. In the past two decades, the LAMBDA method has been popularly applied in variety of GNSS applications, stemming from not only its computation efficiency but also its maximizing the probability of correct integer estimation. LAMBDA stands for “Least-squares AMBiguity Decorrelation Adjustment”, which was invented by Teunissen (1993b) and developed at the Delft University of Technology in the 1990s. Recently, the new version of LAMBDA software (version 3.0) is released, which provides more options of integer estimation methods and more efficient search strategy (Verhagen and Li 2012).

Unsuccessful ambiguity resolution, when passed unnoticed, will too often lead to unacceptable errors in the results. Therefore, it is crucial that one is able to evaluate the integer ambiguity estimation. Evaluation of the integer solution is based on the success rate defined as the probability of correct integer estimation. This ambiguity success rate depends on the underlying mathematical model as well as on the integer estimation method used, which is generally difficult to be exactly evaluated. It is therefore necessary to find some easy-to-use approximations of the success rate. So far, a variety of success rate approximations and bounds have been developed for integer least squares (ILS), integer bootstrapping (IB) and integer rounding (IR) (Hassibi and Boyd 1998; Teunissen 1998e; Teunissen 2000; Teunissen 2001b; Verhagen 2005), no standard software however exists to evaluate them.

The Ps-LAMBDA software, as the first implementation in Matlab for evaluation of the ambiguity success rates, was developed in Curtin University of Technology and Delft University of Technology. In the Ps-LAMBDA software, besides the Monte-Carlo simulation based success rate approximation for all integer estimation methods, all the other available approximations and lower and upper bounds of success rate can be easily assessed for each integer estimation method.

## 1.1 Application of Ps-LAMBDA software

Since the success rate can be computed once the float ambiguity variance-covariance (VC)-matrix  $Q_{\hat{a}\hat{a}}$  is known, it can be computed without the need for actual data. As such, the success rate can be used as a very important performance measure for:

- planning purposes (design computations): what is the performance to be expected given a certain measurement set-up at a given time and location;
- deciding whether or not to fix the ambiguities to the integer estimates during the actual data processing (in real-time or post-processing mode);
- research purposes, e.g. to study the impact of receiver noise characteristics, availability of more signals / satellites, baseline length, etcetera.

## 1.2 Disclaimer

This Matlab implementation is intended for educational / research purposes. Readability of the code has therefore been considered more important than computational efficiency. Hence, the code could still be optimized.

## 1.3 Outline

In the next chapter, a brief review of the integer ambiguity resolution theory will be given first, followed by the three integer estimators together with their pull-in regions in the order of complexity. The third Chapter gives the definition of success rate and its evaluations for ILS, IB and IR methods. The routines and their usage are introduced in Chapter 4. Some examples are given in Chapter 5 to show the performance of success rate bounds in Ps-LAMBDA. The final chapter is about the software availability and liability.

## Chapter 2

# Integer estimation methods and their pull-in regions

### 2.1 Parameter estimation

The mixed integer GNSS linear(ized) model can be defined as:

$$\mathbf{y} \sim N(\mathbf{A}\mathbf{a} + \mathbf{B}\mathbf{b}, \mathbf{Q}_{yy}), \quad \mathbf{a} \in \mathbb{Z}^n, \quad \mathbf{b} \in \mathbb{R}^p \quad (2.1)$$

The notation " $\sim$ " is used to describe "distributed as". The  $m$ -vector  $\mathbf{y}$  contains the pseudorange and carrier-phase observables, the  $n$ -vector  $\mathbf{a}$  contains the DD integer ambiguities,  $\mathbf{b}$  is the real-valued parameter vector of length  $p$ , including baseline or position components and possibly tropospheric and ionospheric delay parameters. The coefficient matrices are  $\mathbf{A} \in \mathbb{R}^{m \times n}$  and  $\mathbf{B} \in \mathbb{R}^{m \times p}$ , with  $[\mathbf{A} \ \mathbf{B}]$  of full column rank. The VC-matrix  $\mathbf{Q}_{yy}$  is an  $(m \times m)$  positive definite matrix. In most GNSS applications, the underlying distribution is assumed to be the multivariate normal distribution (denoted by "N").

In general, a three-step procedure is employed to solve model (2.1) based on the least squares criterion. In practice, a user may want to include a validation step after step 1 and step 2.

#### Step 1: Float solution

In the first step, the integer property of the ambiguities  $\mathbf{a}$  is disregarded and the so-called float estimates together with their VC-matrix are computed

$$\begin{bmatrix} \hat{\mathbf{a}} \\ \hat{\mathbf{b}} \end{bmatrix} \sim N \left( \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \begin{bmatrix} \mathbf{Q}_{\hat{\mathbf{a}}\hat{\mathbf{a}}} & \mathbf{Q}_{\hat{\mathbf{a}}\hat{\mathbf{b}}} \\ \mathbf{Q}_{\hat{\mathbf{b}}\hat{\mathbf{a}}} & \mathbf{Q}_{\hat{\mathbf{b}}\hat{\mathbf{b}}} \end{bmatrix} \right) \quad (2.2)$$

with  $\mathbf{Q}_{\hat{\mathbf{a}}\hat{\mathbf{a}}}$  and  $\mathbf{Q}_{\hat{\mathbf{b}}\hat{\mathbf{b}}}$  are the VC-matrices of the float ambiguity and baseline estimators, respectively and  $\mathbf{Q}_{\hat{\mathbf{a}}\hat{\mathbf{b}}} = \mathbf{Q}_{\hat{\mathbf{b}}\hat{\mathbf{a}}}^T$  their covariance matrix.

### Step 2: Integer estimation

In the second step, the float ambiguity estimate  $\hat{a}$  is used to compute the corresponding integer ambiguity estimate, denoted as

$$\check{a} = \mathcal{I}(\hat{a}) \quad (2.3)$$

with  $\mathcal{I} : \mathbb{R}^n \mapsto \mathbb{Z}^n$  the integer mapping from the  $n$ -dimensional space of reals to the  $n$ -dimensional space of integers. In this step, there are different choices of mapping function  $\mathcal{I}$  possible, which correspond to the different integer estimation methods. Popular choices are ILS, IB and IR.

ILS is optimal, as it can be shown to have the largest success rate of all integer estimators (Teunissen 1999b). IR and IB, however, can also perform quite well, in particular after the LAMBDA decorrelation has been applied. Their advantage over ILS is that no integer search is required. Each of the methods will be discussed in more detail in the following sections.

### Step 3: Fixed solution

In the third step, the float solution of the remaining real-valued parameters solved in the first step are updated using the fixed integer parameters,

$$\check{b} = \hat{b} - Q_{\hat{b}\hat{a}}Q_{\hat{a}\hat{a}}^{-1}(\hat{a} - \check{a}) \quad (2.4)$$

Its VC-matrix is obtained by application of the error propagation law:

$$Q_{\check{b}\check{b}} = Q_{\hat{b}\hat{b}} - Q_{\hat{b}\hat{a}}Q_{\hat{a}\hat{a}}^{-1}Q_{\hat{a}\hat{b}} \quad (2.5)$$

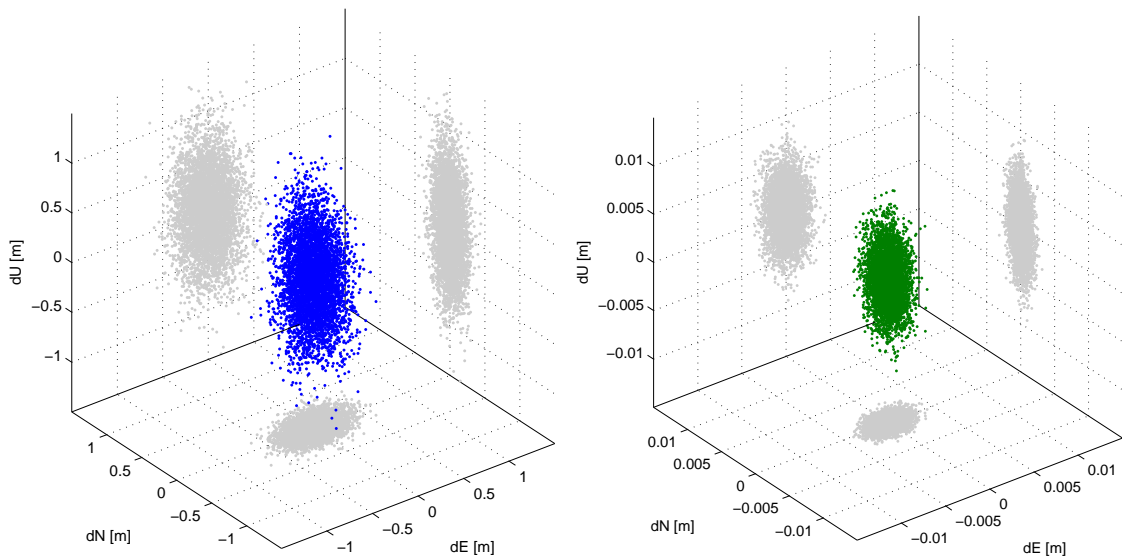


Figure 2.1: Position errors in East (dE), North (dN) and Up (dU) direction in meters for ambiguity float solutions (left panel), ambiguity fixed solutions (right panel) in a short baseline. Note the different scales in the left and right panels.

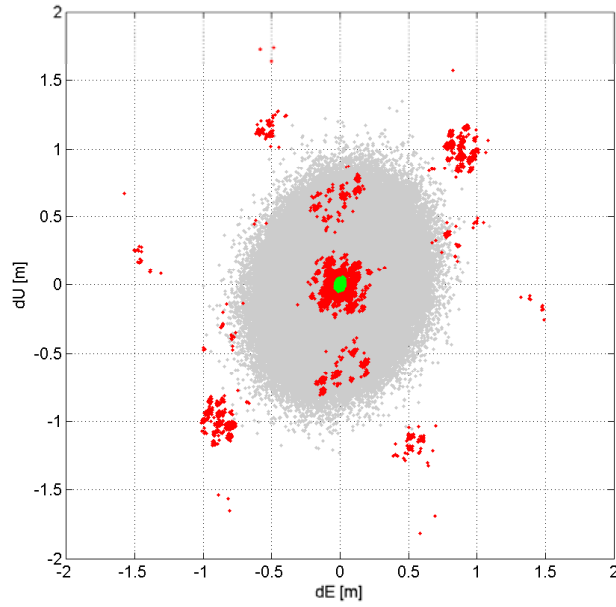


Figure 2.2: Scatterplot of horizontal position errors in meters for float solution (grey dots) and corresponding fixed solution. In this case, 93% of the solutions were correctly fixed (green dots), and 7% was wrongly fixed (red dots).

Note the derivation of VC-matrix based on error propagation law is based on the assumption that the integer ambiguity solution can be assumed to be deterministic. It holds true only when the success rate of ambiguity solution is sufficiently close to 1. In such case, in general  $Q_{\tilde{b}\tilde{b}} \ll Q_{\hat{b}\hat{b}}$ , since after successful ambiguity fixing the carrier-phase measurements start to act as very precise pseudorange measurements. Figure 2.1 shows a scatterplot of the float and fixed position errors based on 10,000 solutions with single epoch, dual-frequency GPS for a short baseline; the success rate is equal to 1. It can be observed that the precision is improved with a factor 100, in agreement with the difference in code and carrier-phase measurement noise.

However, incorrect integer ambiguity estimation may result in the opposite effect in terms of positioning accuracy: rather than a dramatic precision improvement, a wrong ambiguity solution can cause very large position errors, exceeding those of the float solution. This is illustrated in Figure 2.2, which shows a scatterplot of horizontal float position errors for a case where the ambiguities are fixed correctly in only 93% of the cases. The corresponding fixed solutions are shown as either red or green dots: red if the ambiguities are fixed incorrectly, green if they are fixed correctly. It can be seen that in all cases where the ambiguities were fixed correctly, the position errors are very small. However, in case of unsuccessful integer estimation the corresponding position errors tend to be of the same size or even much larger than the corresponding float position errors. The figure shows only the horizontal positioning results, for the vertical component the errors can be as large as 8 meters in this example. This clearly shows that the fixed solution should only be used if the success rate is very high. It is therefore very important to evaluate the success rate of integer solution before use it.



## 2.2 Admissible integer estimation

As previously mentioned there are many ways of computing an integer ambiguity vector  $\tilde{\mathbf{a}}$  from its real-valued counterpart  $\hat{\mathbf{a}}$ . To each such method belongs a different mapping  $\mathcal{I} : \mathbb{R}^n \mapsto \mathbb{Z}^n$ . Due to the discrete nature of  $\mathbb{Z}^n$ , the map  $\mathcal{I}$  will not be one-to-one, but instead a many-to-one map. This implies that different real-valued ambiguity vectors will be mapped to the same integer vector. One can therefore assign a subset  $\mathcal{P}_z \subset \mathbb{R}^n$  to each integer vector  $z \in \mathbb{Z}^n$ :

$$\mathcal{P}_z = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{z} = \mathcal{I}(\mathbf{x})\}, \quad \mathbf{z} \in \mathbb{Z}^n \quad (2.6)$$

The subset  $\mathcal{P}_z$  contains all real-valued ambiguity vectors that will be mapped by  $\mathcal{I}$  to the same integer vector  $z$ . This subset is referred to as the *pull-in region* of  $z$ . It is the region in which all ambiguity float solutions are pulled to the same fixed ambiguity vector  $z$ .

Using the pull-in regions, one can give an explicit expression for the corresponding integer ambiguity estimator. It reads

$$\tilde{\mathbf{a}} = \sum_{\mathbf{z} \in \mathbb{Z}^n} \mathbf{z} \mathcal{P}_z(\hat{\mathbf{a}}), \quad \mathcal{P}_z(\hat{\mathbf{a}}) = \begin{cases} 1 & \text{if } \hat{\mathbf{a}} \in \mathcal{P}_z \\ 0 & \text{otherwise} \end{cases} \quad (2.7)$$

Since the pull-in regions define the integer estimator completely, one can define classes of integer estimators by imposing various conditions on the pull-in regions. One such class is referred to as the class of admissible integer estimators. This class was introduced in Teunissen (1999a) and it is defined as follows.

### Definition

The integer estimator  $\tilde{\mathbf{a}} = \sum_{\mathbf{z} \in \mathbb{Z}^n} \mathbf{z} \mathcal{P}_z(\hat{\mathbf{a}})$  is said to be *admissible* if

- (i)  $\bigcup_{\mathbf{z} \in \mathbb{Z}^n} \mathcal{P}_z = \mathbb{R}^n$
- (ii)  $\text{Int}(\mathcal{P}_{z_1}) \cap \text{Int}(\mathcal{P}_{z_2}) = \emptyset, \quad \forall \mathbf{z}_1, \mathbf{z}_2 \in \mathbb{Z}^n, \quad \mathbf{z}_1 \neq \mathbf{z}_2$
- (iii)  $\mathcal{P}_z = \mathbf{z} + \mathcal{P}_0, \quad \forall \mathbf{z} \in \mathbb{Z}^n$

This definition is motivated as follows. The first condition states that the pull-in regions should not leave any gaps and the second that they should not overlap. The absence of gaps is needed in order to be able to map any float solution  $\hat{\mathbf{a}} \in \mathbb{R}^n$  to  $\mathbb{Z}^n$ , while the absence of overlaps is needed to guarantee that the float solution is mapped to just one integer vector. The last condition of the definition follows from the requirement that  $\mathcal{I}(\mathbf{x} + \mathbf{z}) = \mathcal{I}(\mathbf{x}) + \mathbf{z}, \forall \mathbf{x} \in \mathbb{R}^n, \mathbf{z} \in \mathbb{Z}^n$ . It states that when the float solution is perturbed by  $\mathbf{z}$ , the corresponding integer solution is perturbed by the same amount. This property allows one to apply the *integer remove-restore* technique:  $\mathcal{I}(\hat{\mathbf{a}} - \mathbf{z}) + \mathbf{z} = \mathcal{I}(\hat{\mathbf{a}})$ .

The three popularly used integer estimation methods, IR, IB and ILS, are all examples of admissible integer estimation methods.

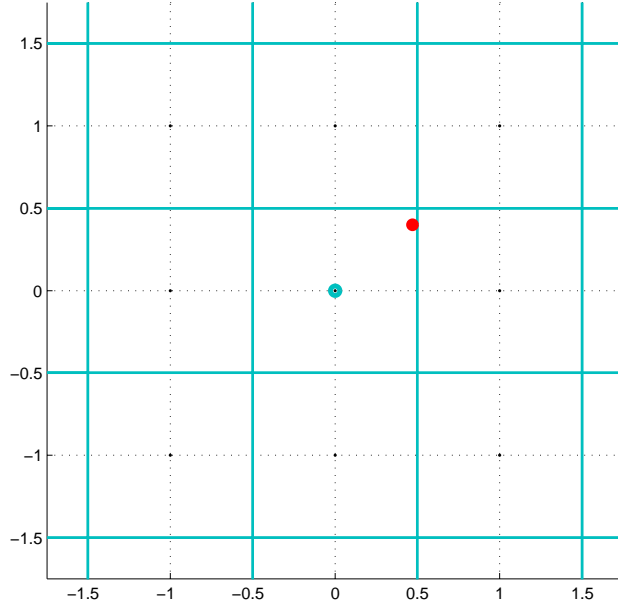


Figure 2.3: 2D IR pull-in regions: unit squares.

### 2.2.1 Z-transformation

It will be explained later that it may be useful to apply a so-called **Z**-transformation to the ambiguity parameters. A matrix is called a **Z**-transformation if it is one-to-one (i.e. invertible) and integer (Teunissen 1995a). Such transformations leave the integer nature of the parameters in tact. If a certain integer estimator is **Z**-invariant it means that if the float solution is **Z**-transformed, the integer solution transforms accordingly. Hence:

$$\tilde{\mathbf{z}} = \mathbf{Z}^T \tilde{\mathbf{a}} \quad \text{if} \quad \hat{\mathbf{z}} = \mathbf{Z}^T \hat{\mathbf{a}} \quad (2.8)$$

A very useful **Z**-transformation is the decorrelating **Z**-transformation (Teunissen 1993a; Teunissen 1994; Teunissen 1995a; Teunissen 1995b). It results in a more diagonal VC-matrix:

$$\mathbf{Q}_{\hat{\mathbf{z}}\hat{\mathbf{z}}} = \mathbf{Z}^T \mathbf{Q}_{\hat{\mathbf{a}}\hat{\mathbf{a}}} \mathbf{Z} \quad (2.9)$$

Once the transformed ambiguities are fixed  $\tilde{\mathbf{z}}$ , one can also apply the back-transformation to recover the integer solution of original ambiguities,  $\tilde{\mathbf{a}} = \mathbf{Z}^{-T} \tilde{\mathbf{z}}$ .

### 2.2.2 Integer rounding

The simplest way to obtain an integer vector from the real-valued float solution is to round each of the entries of  $\hat{\mathbf{a}}$  to its nearest integer. The corresponding integer estimator reads

$$\tilde{\mathbf{a}}_{\text{IR}} = ([\hat{a}_1], \dots, [\hat{a}_n])^T \quad (2.10)$$

where  $[\cdot]$  stands for rounding to the nearest integer.

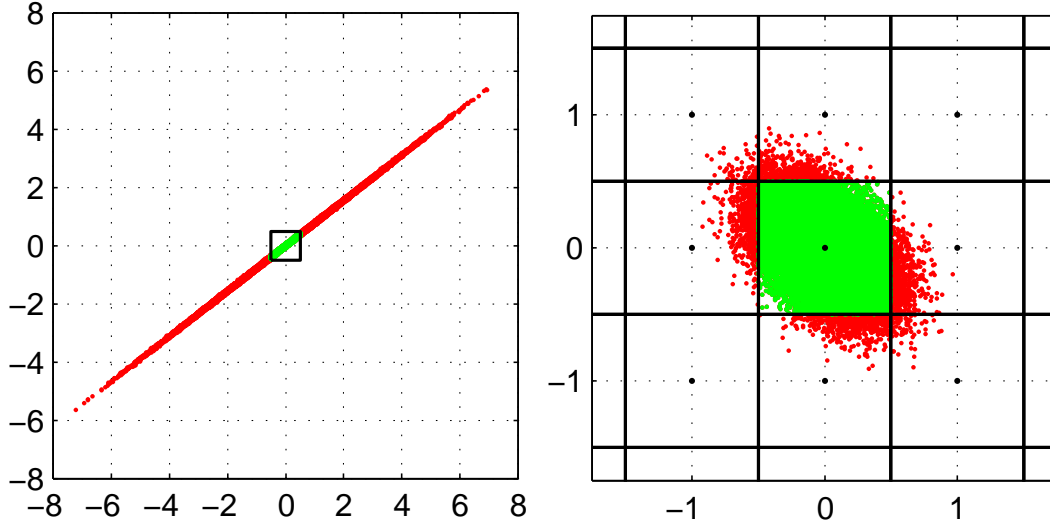


Figure 2.4: 2D IR pull-in regions and 50,000 float solutions. Left: original ambiguities  $\hat{a}$  [cycles]; Right:  $Z$ -decorrelated ambiguities  $\hat{z}$  [cycles].

The pull-in regions for rounding are  $n$ -dimensional unit cubes centred at the integer grid points:

$$\mathcal{P}_{z,IR} = \{\mathbf{x} \in \mathbb{R}^n \mid |\mathbf{c}_i^T(\mathbf{x} - \mathbf{z})| \leq \frac{1}{2}, i = 1, \dots, n\}, \quad \mathbf{z} \in \mathbb{Z}^n \quad (2.11)$$

with  $\mathbf{c}_i$  the unit vector have a 1 as its  $i$ th entry and 0's otherwise. Figure 2.3 shows the 2D pull-in regions for rounding: all float solutions residing in a specific pull-in region will be fixed to the corresponding integer grid point in the centre of the pull-in region, i.e. all are *pulled* to the same integer vector. As an example, one float solution is shown with the red dot, and the corresponding integer solution is depicted with the blue circle.

In general, the rounding estimator is not  $Z$ -invariant, i.e.  $\hat{z}_{IR} \neq Z^T \hat{a}_{IR}$ . Only if  $Z$  is a permutation matrix, and thus the transformation is a simple reordering of the ambiguities, the estimator is  $Z$ -invariant. Note that the IR pull-in regions remain unaffected by the  $Z$ -transformation. Figure 2.4 shows an example for a 2-dimensional (2D) ambiguity vector. 50,000 samples of float ambiguities for a given VC-matrix  $\mathbf{Q}_{\hat{a}\hat{a}}$  were simulated; these are shown as the red and green dots. The left panel shows the original float samples (before  $Z$ -decorrelation), and the pull-in region  $\mathcal{P}_{0,IR}$ , in which all the green samples reside. Hence, for all those samples the  $\mathbf{0}$ -vector is obtained after rounding. The right panel shows the corresponding  $Z$ -decorrelated float ambiguity samples, as well as the surrounding pull-in regions. In this case, many more float samples reside in  $\mathcal{P}_{0,IR}$ : 95% versus 23% before  $Z$ -decorrelation. This shows that the choice for the parameterization of the float ambiguity vector is very important in case of IR.

### 2.2.3 Integer bootstrapping

The IB estimator still makes use of IR, but it takes some of the correlation between the ambiguities into account. The IB estimator follows from a sequential least squares adjustment and it is computed as follows. If  $n$  ambiguities are available, one starts with the most precise ambiguity. Let the  $n$ th ambiguity

be the most precise one, hence we start with rounding  $\hat{a}_n$  to the nearest integer. The remaining float ambiguities are corrected by virtue of their correlation with the last ambiguity. Then the last-but-one, but now corrected, real-valued ambiguity estimate is rounded to its nearest integer and all remaining  $(n - 2)$  ambiguities are then again corrected, but now by virtue of their correlation with this ambiguity. This process is continued until all ambiguities are considered. The components of the bootstrapped estimator  $\check{\mathbf{a}}_{\text{IB}}$  are given as

$$\begin{aligned}\check{a}_{n;\text{IB}} &= [\hat{a}_n] \\ \check{a}_{j;\text{IB}} &= [\hat{a}_{j|J}] = [\hat{a}_j - \underbrace{\sum_{i=j+1}^n \sigma_{\hat{a}_j \hat{a}_{i|I}} \sigma_{\hat{a}_{i|I}}^{-2}}_{l_{i,j}} (\hat{a}_{i|I} - \check{a}_{i;\text{IB}})], \quad \forall j = 1, \dots, n-1\end{aligned}\quad (2.12)$$

The short-hand notation  $\hat{a}_{i|I}$  stands for the  $i$ th ambiguity obtained through a conditioning on the previous  $I = \{i + 1, \dots, n\}$  sequentially rounded ambiguities. The real-valued sequential least squares solution can be obtained by means of the triangular decomposition of the VC-matrix of the ambiguities:  $\mathbf{Q}_{\hat{\mathbf{a}}\hat{\mathbf{a}}} = \mathbf{L}^T \mathbf{D} \mathbf{L}$ , where  $\mathbf{L}$  denotes a unit lower triangular matrix with entries  $l_{i,j}$  (see Eq.(2.12)) and  $\mathbf{D}$  a diagonal matrix with the conditional variances  $\sigma_{\hat{a}_{i|I}}^2$  as its diagonal elements.

The IB pull-in regions are given as:

$$\mathcal{P}_{\mathbf{z},\text{IB}} = \{\mathbf{x} \in \mathbb{R}^n \mid |\mathbf{c}_i^T \mathbf{L}^{-T}(\mathbf{x} - \mathbf{z})| \leq \frac{1}{2}, i = 1, \dots, n\}, \quad \mathbf{z} \in \mathbb{Z}^n \quad (2.13)$$

with  $\mathbf{c}_i$  the unit vector have a 1 as its  $i$ th entry and 0's otherwise. Figure 2.5 shows an example of the IB pull-in regions in the 2D case, which are parallelograms. The float solution is depicted with a red point and its bootstrapped solution is shown with the blue circle.

Like IR, IB suffers as well from a lack of  $\mathbf{Z}$ -invariance, i.e.  $\check{\mathbf{z}}_{\text{IB}} \neq \mathbf{Z}^T \check{\mathbf{a}}_{\text{IB}}$  if  $\hat{\mathbf{z}} = \mathbf{Z}^T \hat{\mathbf{a}}$ . From Eq.(2.12)

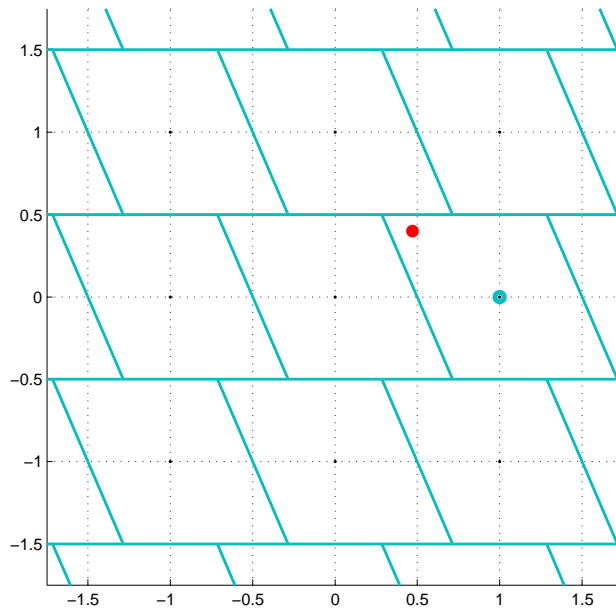


Figure 2.5: 2D IB pull-in regions: parallelograms.

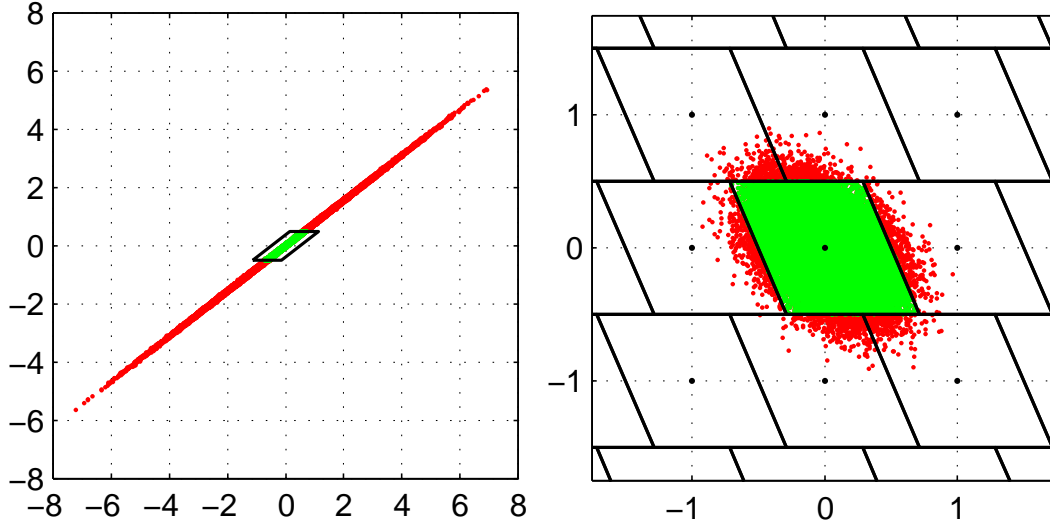


Figure 2.6: 2D IB pull-in regions and 50,000 float solutions. Left: original ambiguities  $\hat{a}$  [cycles]; Right:  $Z$ -decorrelated ambiguities  $\hat{z}$  [cycles].

can be seen that changing the order will already result in a different outcome with bootstrapping. It can be clearly seen from Figure 2.6 how IB is affected by the decorrelating  $Z$ -transformation. Here 96% of the  $Z$ -decorrelated float samples resides in  $\mathcal{P}_{0,IB}$  versus 29% of the original ambiguity samples.

#### 2.2.4 Integer least squares

When solving the GNSS model of Eq.(2.1) in a least squares sense, but now with the additional constraint that the ambiguity parameters should be integer-valued, the integer estimator of the second step in the procedure becomes:

$$\tilde{a}_{ILS} = \arg \min_{z \in \mathbb{Z}^n} \|\hat{a} - z\|_{Q_{\hat{a}\hat{a}}}^2 \quad (2.14)$$

with  $\|\cdot\|_Q^2 = (\cdot)^T Q^{-1}(\cdot)$ . This estimator is known to be optimal, cf. (Teunissen 1999b), which means that the success rate of integer estimation is maximized.

The ILS pull-in region is defined by (Teunissen 1999b):

$$\mathcal{P}_{z,ILS} = \{x \in \mathbb{R}^n \mid |w_z(x)| \leq \frac{1}{2} \|u\|_{Q_{\hat{a}\hat{a}}}, \forall u \in \mathbb{Z}^n\} \quad (2.15)$$

with

$$w_z(x) = \frac{u^T Q_{\hat{a}\hat{a}}^{-1} (x - z)}{\|u\|_{Q_{\hat{a}\hat{a}}}} \quad (2.16)$$

the orthogonal projection of  $(x - z)$  onto the direction vector  $u$ . Hence,  $\mathcal{P}_{z,ILS}$  is the intersection of banded subsets centered at  $z$  and having width  $\|u\|_{Q_{\hat{a}\hat{a}}}$ .

Figure 2.7 shows an example of the 2D ILS pull-in regions. As an example, one float solution is shown with the red dot, and the corresponding integer solution is depicted with the blue circle. For this particular float solution, the same solution as with bootstrapping, see Fig.2.5, is obtained.

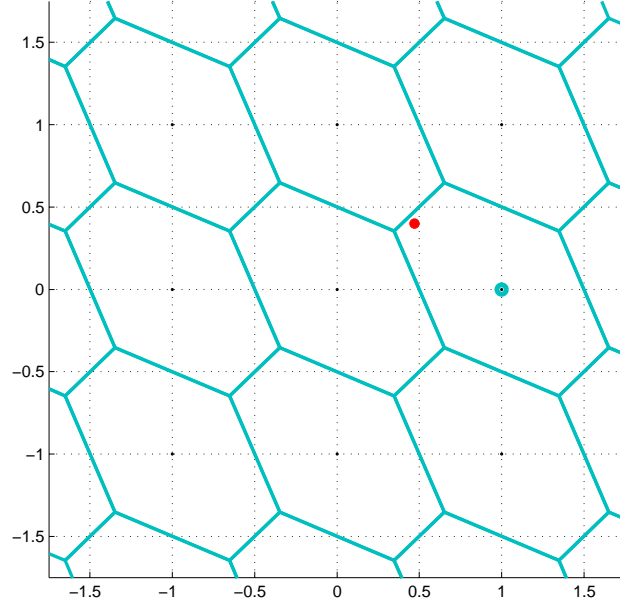


Figure 2.7: 2D ILS Pull-in regions: hexagons.

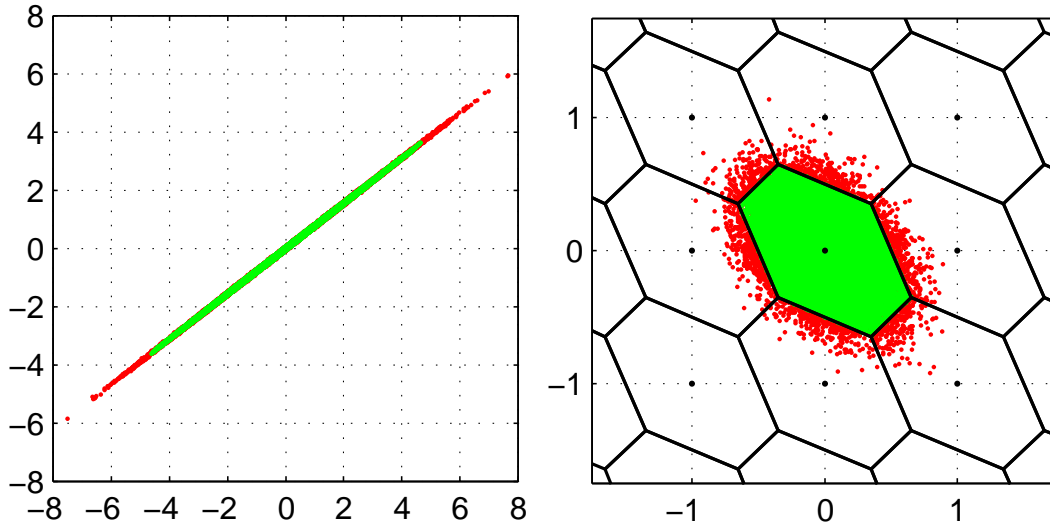


Figure 2.8: 2D ILS pull-in regions and 50,000 float solutions. Left: original ambiguities  $\hat{a}$  [cycles]; Right:  $\mathbf{Z}$ -decorrelated ambiguities  $\hat{z}$  [cycles].

In contrast to IR and IB, the ILS estimator is  $\mathbf{Z}$ -invariant:  $\tilde{z}_{\text{ILS}} = \mathbf{Z}^T \tilde{a}_{\text{ILS}}$  if  $\hat{z} = \mathbf{Z}^T \hat{a}$ . Figure 2.8 shows an example how the decorrelation affects the ILS estimator. For the original VC-matrix  $\mathbf{Q}_{\hat{a}\hat{a}}$  (left panel) the ILS pull-in region follows the distribution of the float samples much better than in case of IR and IB, compare with the corresponding Figures 2.4 and 2.6. Due to the  $\mathbf{Z}$ -invariance the percentage of float samples in  $\mathcal{P}_{0,\text{ILS}}$  (the green dots) is 97% both for the original and  $\mathbf{Z}$ -decorrelated ambiguities. The percentages for all three integer estimators are summarized in Table 2.1.

An integer search is needed to determine  $\tilde{a}_{\text{ILS}}$ . The ILS procedure is efficiently mechanized in the LAMBDA method. A key element of the LAMBDA method is the decorrelating  $\mathbf{Z}$ -transformation, see Section 2.2.1, which results in largely reduced search times. For more information on the LAMBDA

Table 2.1: Percentage of float solutions that is correctly fixed for the three integer estimation methods (corresponding to Figures 2.4 , 2.6 and 2.8).

	IR	IB	ILS
Original ambiguities $\hat{a}$	23	29	97
$Z$ -decorrelated ambiguities $\hat{z}$	95	96	97

method and its wide-spread applications see e.g. (Teunissen 1993a; Teunissen 1995b; Li and Teunissen 2011; Chang et al. 2005; De Jonge and Tiberius 1996a; Hofmann-Wellenhof et al. 2001; Teunissen and Kleusberg 1998; Leick 2004; Strang and Borre 1997; Misra and Enge 2001). It is pointed out that the new version LAMBDA software (version 3.0) has been released recently with more efficient search strategy and more options of integer estimation methods (Verhagen and Li 2012).

## Chapter 3

# Success rate of integer estimation

### 3.1 Definition of success rate

According to the definition of admissible integer estimation in section 2.2, the float ambiguity can be correctly fixed to its integer if and only if it resides its corresponding pull-in region, i.e.,

$$\check{\mathbf{a}} = \mathbf{a} \Leftrightarrow \hat{\mathbf{a}} \in \mathcal{P}_{\mathbf{a}} \quad (3.1)$$

In other words, the probability of correct ambiguity estimation, i.e., success rate  $P_s$ , is equal to the probability that  $\hat{\mathbf{a}}$  resides in the pull-in region  $\mathcal{P}_{\mathbf{a}}$  with  $\mathbf{a}$  the true but unknown ambiguity vector:

$$P_s = P(\check{\mathbf{a}} = \mathbf{a}) = P(\hat{\mathbf{a}} \in \mathcal{P}_{\mathbf{a}}) = \int_{\mathcal{P}_{\mathbf{a}}} f_{\hat{\mathbf{a}}}(\mathbf{x}|\mathbf{a}) d\mathbf{x} \quad (3.2)$$

The probability density function (PDF) of the float ambiguities,  $f_{\hat{\mathbf{a}}}(\mathbf{x}|\mathbf{a})$ , is assumed to be the normal PDF with mean  $\mathbf{a}$ :

$$f_{\hat{\mathbf{a}}}(\mathbf{x}|\mathbf{a}) = \frac{1}{\sqrt{\det(2\pi Q_{\hat{\mathbf{a}}\hat{\mathbf{a}}})}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \mathbf{a})^T Q_{\hat{\mathbf{a}}\hat{\mathbf{a}}}^{-1}(\mathbf{x} - \mathbf{a})\right\} \quad (3.3)$$

As the pull-in regions of the integer estimators are integer-translation invariant, the success rate can also be evaluated as:

$$P_s = \int_{\mathcal{P}_0} f_{\hat{\mathbf{a}}}(\mathbf{x}|\mathbf{0}) d\mathbf{x} \quad (3.4)$$

An illustration is given in Figure 3.1 for the ILS estimator: in the left panel the PDF of a 2D float ambiguity vector is shown, with the corresponding ILS pull-in regions underneath. The right panel shows the probability masses for each integer grid point, equal to the integral of the PDF over the corresponding pull-in regions. In this case, the success rate is equal to the probability mass at  $[0 \ 0]^T$ .

The integration over the pull-in region is very complex in case of ILS and IR and hence it is difficult to evaluate (3.4) exactly. Therefore we need to develop some easy-to-use probabilistic bounds or approximation to the exact success rate. The lower bound can then be used to infer whether ambiguity



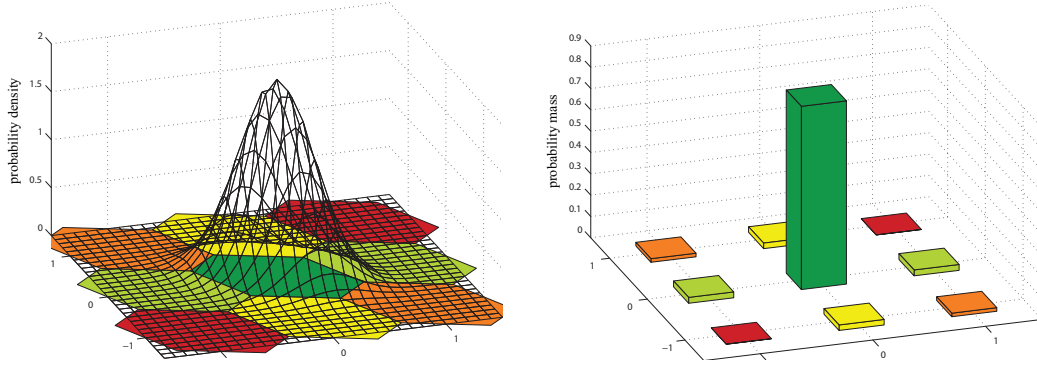


Figure 3.1: Left: PDF and 2D pull-in regions for ILS. Right: corresponding probability mass function.

resolution can be expected to be successful, while the upper bound will show when one can expect successful ambiguity resolution to fail. Thus there is enough confidence that ambiguity resolution will be successful when the lower bound is sufficiently close to one, while no such confidence exists when the upper bound turns out to be too small. These probabilistic bounds have different approximation degrees to the actual success rate, but all of them usually possess efficient numerical computation which is the right motivation of probabilistic bound. In the next section, the available approximations and bounds are described for IR, IB and ILS methods in order of complexity.

### 3.2 Monte Carlo simulation based approximate success rate

The success rate of integer estimation can be approximated by means of Monte Carlo simulation. The procedure is as follows. It is assumed that the float solution is normally distributed  $\hat{\mathbf{a}} \sim N(\mathbf{a}, \mathbf{Q}_{\hat{\mathbf{a}}\hat{\mathbf{a}}})$ , and thus the distribution is symmetric about the mean  $\mathbf{a}$ . Hence, we may shift the distribution over  $\mathbf{a}$  and draw samples from the distribution  $N(\mathbf{0}, \mathbf{Q}_{\hat{\mathbf{a}}\hat{\mathbf{a}}})$ .

The first step is to generate independent random samples with normal distribution  $N(\mathbf{0}, \mathbf{Q}_{\hat{\mathbf{a}}\hat{\mathbf{a}}})$ . One can generate the random sample  $\hat{\mathbf{a}}$  directly from the Matlab built-in function `mvnrnd(0, Qââ)`. Alternatively, one can first generate  $n$  independent samples from the univariate standard normal distribution  $N(0, 1)$ , and then collect these in a vector  $\mathbf{s}$ . This vector is transformed by means of  $\hat{\mathbf{a}} = \mathbf{G}\mathbf{s}$ , with  $\mathbf{G}$  equal to the Cholesky factor of  $\mathbf{Q}_{\hat{\mathbf{a}}\hat{\mathbf{a}}} = \mathbf{G}\mathbf{G}^T$ . The result is a sample  $\hat{\mathbf{a}}$  from  $N(\mathbf{0}, \mathbf{Q}_{\hat{\mathbf{a}}\hat{\mathbf{a}}})$ . The sample  $\hat{\mathbf{a}}$  is used as input for integer estimation. If the output of this estimator equals the null vector, then it is correct, otherwise it is incorrect. This process can be repeated an  $N$  number of times, and one can count how many times the null vector is obtained as a solution, say  $N_s$  times. The approximation of the success rate follows then as:

$$P_s \approx \frac{N_s}{N} \quad (3.5)$$

In the simulation, the number of sample plays an important role for the approximation precision. In order to get good approximations, the number of samples  $N$  must be sufficiently large (Teunissen 1998c). The disadvantage is that it may be very time-consuming to evaluate Eq.(3.5), especially in case of ILS, since for each sample an integer search is required.

Figure 3.2 shows for four GNSS models how the approximation performs depending on the number of

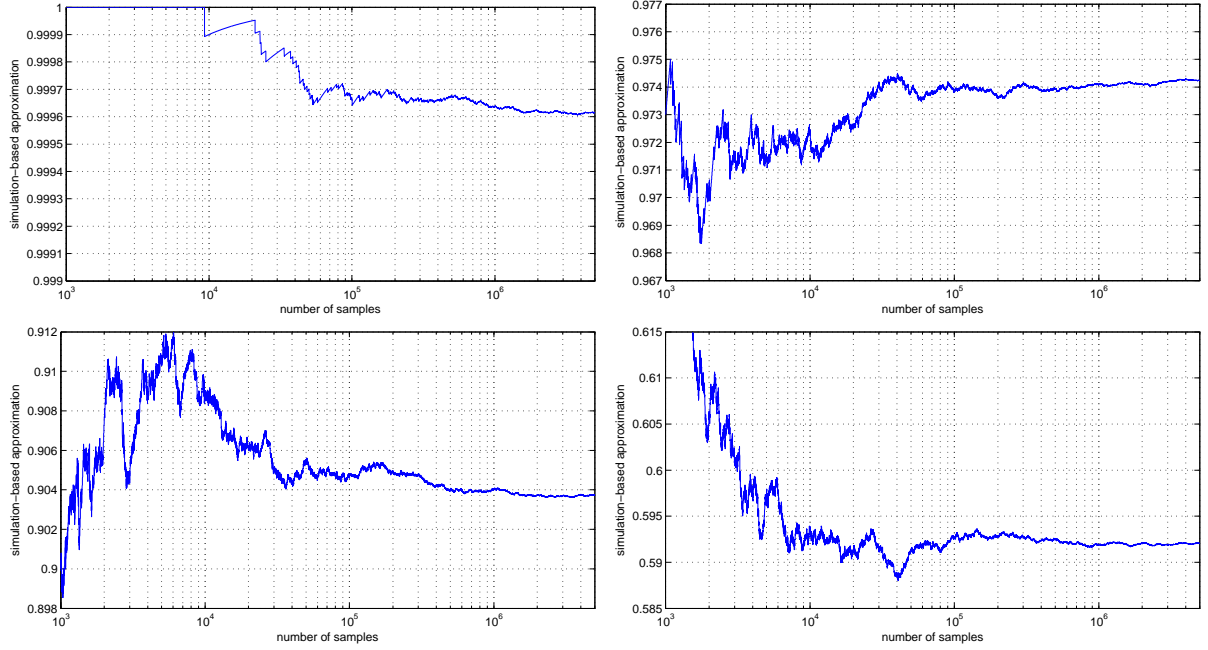


Figure 3.2: Examples of simulation-based success rate as function of number of samples. Each panel shows the results for a different GNSS model.

samples used (similar results were obtained for many other GNSS positioning models). It follows that at least  $10^5$  samples should be used to get a good approximation. At the same time it can be seen that using more samples generally only has a small effect, in the order of  $10^{-3}$ , especially in cases where the success rate is close to 1. With  $10^6$  samples the approximation will be very close to the true value. In the Ps-LAMBDA, it allows to evaluate the simulation based success rates for IR and ILS, where the user may specify the number of samples to be used.

### 3.3 Success rate and its bounds

The probabilistic bounds are developed typically from the following three aspects:

- Simplify the PDF of the float solution by partially capturing its associated VC-matrix. For example, the product of all individual success rate of each scalar integer in a integer vector is used as lower bound of IR success rate of this integer vector.
- Simplify the geometric complexity of pull-in region. The simplified pull-in region has the simpler geometry such that the computation of the PDF integral over this simplified pull-in region is feasible.
- Use the success rate relations between different integer estimation methods. As mentioned, the success rates also depend on the selected integer estimation method, since the pull-in region is different for IR, IB and ILS. In Teunissen (1999b), it was proven that:

$$P(\tilde{\mathbf{a}}_{\text{IR}} = \mathbf{a}) \leq P(\tilde{\mathbf{a}}_{\text{IB}} = \mathbf{a}) \leq P(\tilde{\mathbf{a}}_{\text{ILS}} = \mathbf{a}) \quad (3.6)$$

The ordering is thus the same as the ordering in terms of complexity, since IR is the simplest and ILS the most complex method. This means that if IR or IB provides a very sharp lower bound, a user could decide to use the simpler integer estimation method if their success rate is close to 1 and still obtain (close to) optimal performance.

### 3.3.1 IR success rate and its bounds

#### Lower bound based on the diagonal VC-matrix

The  $n$ -fold integral over the IR pull-in region defined in (2.11) is difficult to evaluate. Only if the VC-matrix  $\mathbf{Q}_{\hat{\mathbf{a}}\hat{\mathbf{a}}}$  is diagonal will the success rate become equal to the  $n$ -fold product of the univariate success rates. In Teunissen (1998e) it was shown that this also provides a lower bound in case  $\mathbf{Q}_{\hat{\mathbf{a}}\hat{\mathbf{a}}}$  is not diagonal:

$$P_{s,\text{IR}} = P(\check{\mathbf{a}}_{\text{IR}} = \mathbf{a}) \geq \prod_{i=1}^n \left( 2\Phi\left(\frac{1}{2\sigma_{\hat{a}_i}}\right) - 1 \right) \quad (3.7)$$

with  $\Phi(x)$  the cumulative normal distribution function:

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x \exp\left\{-\frac{1}{2}t^2\right\} dt$$

Note when the float solution  $\hat{\mathbf{a}}$  is a scalar case, this lower bound is the exact success rate.

#### IR success rate improved by decorrelation

In Section 2.2.2 it was mentioned that IR is not  $\mathbf{Z}$ -invariant. This holds for the IR success rates as well, since the pull-in regions are unaffected by a  $\mathbf{Z}$ -transformation, while the distribution of the transformed ambiguities is changed to  $\hat{\mathbf{z}} \sim N(\mathbf{Z}^T \mathbf{a}, \mathbf{Q}_{\hat{\mathbf{z}}\hat{\mathbf{z}}})$ . If IR is applied to the  $\mathbf{Z}$ -decorrelated ambiguities, the success rate will increase due to the improved precision of the decorrelated ambiguities, i.e.

$$P(\check{\mathbf{z}}_{\text{IR}} = \mathbf{z}) \geq P(\check{\mathbf{a}}_{\text{IR}} = \mathbf{a}) \quad (3.8)$$

### 3.3.2 IB success rate and its bounds

#### IB success rate

In case of bootstrapping the success rate can be evaluated exactly using (Teunissen 1998e):

$$P_{s,\text{IB}} = P(\check{\mathbf{a}}_{\text{IB}} = \mathbf{a}) = \prod_{i=1}^n \left( 2\Phi\left(\frac{1}{2\sigma_{\hat{a}_{i|I}}}\right) - 1 \right) \quad (3.9)$$

where  $\sigma_{\hat{a}_{i|I}}^2$  with  $I = \{i+1, \dots, n\}$  is the conditional variance of the  $i$ th float ambiguity  $\hat{a}_i$  on the float ambiguities from  $(i+1)$  to  $n$ , which is the  $i$ th diagonal element of  $\mathbf{D}$  computed from the Cholesky decomposition on the VC-matrix  $\mathbf{Q}_{\hat{\mathbf{a}}\hat{\mathbf{a}}} = \mathbf{L}^T \mathbf{D} \mathbf{L}$ . To our best knowledge, the IB success rate is one of the most useful success rate evaluations from both computation efficiency and performance.

### Upper bound based on ADOP

In Teunissen (2000) it was proven that such an upper bound is given by:

$$P_{s,IB} \leq \left( 2\Phi\left(\frac{1}{2\text{ADOP}}\right) - 1 \right)^n = P_{\text{ADOP}} \quad (3.10)$$

with ADOP being the Ambiguity Dilution of Precision given by:

$$\text{ADOP} = \sqrt{\det(\mathbf{Q}_{\hat{a}\hat{a}})^{\frac{1}{n}}} \quad (3.11)$$

in units of *cycles*. The ADOP is a diagnostic that captures the main characteristics of the ambiguity precision. It was introduced in Teunissen (1997), described and analyzed in (Teunissen and Odijk 1997; Odijk and Teunissen 2008) and is widely used, see the introduction of Odijk and Teunissen (2008). There is an important property for ADOP that it is  $\mathbf{Z}$ -invariant for the class of admissible ambiguity transformation, i.e.  $\det(\mathbf{Q}_{\hat{a}\hat{a}}) = \det(\mathbf{Q}_{\hat{z}\hat{z}})$ . This merit property makes this upper bound useful, since one does not need apply the transformation and directly compute this upper bound. If it is too small, it can be immediately concluded that IB will be not successful for *any* parameterization of the ambiguities. When the ambiguities are completely decorrelated, the ADOP equals the geometric mean of the standard deviations of the ambiguities, hence it can be considered as a measure of the average ambiguity precision.

### IB success rate improved by decorrelation

The IB success rate is not  $\mathbf{Z}$ -invariant. IB may perform close to optimal if applied to the decorrelated ambiguities  $\hat{z}$  (Teunissen 1998e; Verhagen 2005) and we have:

$$P(\hat{z}_{IB} = z) \geq P(\hat{a}_{IB} = a) \quad (3.12)$$

### 3.3.3 ILS success rate and its bounds

In general it is more difficult to evaluate the ILS success rate than IR and IB since its pull-in region has a rather complicated geometry. So far, a variety of ILS success rate bounds have been developed, of which some are based on bounding the complicated pull-in region with a simple region, while some based on bounding the VC-matrix with a simple one.

### Approximation based on ADOP

It was also proved in Teunissen (1999c) that  $P_{\text{ADOP}}$ , as upper bound (3.10) of IB success rate, can be used as an approximation to the ILS success rate, i.e.,

$$P_{s,ILS} \approx P_{\text{ADOP}} = \left( 2\Phi\left(\frac{1}{2\text{ADOP}}\right) - 1 \right)^n \quad (3.13)$$

### Upper bound based on ADOP

Besides the ADOP based ILS success rate approximation, an upper bound for the ILS success rate based on the ADOP can be given as:

$$P_{s,ILS} \leq P\left(\chi^2(n, 0) \leq \frac{c_n}{\text{ADOP}^2}\right) \quad (3.14)$$

with

$$c_n = \frac{\left(\frac{n}{2}\Gamma\left(\frac{n}{2}\right)\right)^{\frac{2}{n}}}{\pi}$$

and  $\Gamma(\bullet)$  gamma function. This bound was introduced in Hassibi and Boyd (1998), while the proof was given in Teunissen (2000).

### Upper and lower bounds based on bounding the pull-in region

In Teunissen (1998c) lower and upper bounds for the ILS success rate were obtained by bounding the integration region (pull-in region)  $\mathcal{P}_{a,ILS}$ . Obviously, a lower bound is obtained if the integration region is chosen such that it is completely contained by the pull-in region, and an upper bound is obtained if the integration region is chosen such that it completely contains the pull-in region. Let subsets  $L_a$  and  $U_a$  are the integration regions for lower and upper bounds, respectively, it follows that  $L_a \subset \mathcal{P}_{a,ILS}$  and  $U_a \supset \mathcal{P}_{a,ILS}$ . In that case the success rate lies in the interval

$$P(\hat{\mathbf{a}} \in L_a) \leq P(\hat{\mathbf{a}} \in \mathcal{P}_{a,ILS}) \leq P(\hat{\mathbf{a}} \in U_a) \quad (3.15)$$

Both regions of integration should of course be chosen such that the corresponding probabilities are easily evaluated in practice.

Given the definition of the ILS pull-in region  $\mathcal{P}_{a,ILS}$  in Eq.(2.15), it follows that any finite intersection of  $p < n$  banded subsets defined by  $w$  of Eq.(2.16) will enclose  $\mathcal{P}_{a,ILS}$ .

$$U_a = \{\mathbf{x} \in \mathbb{R}^n \mid |w_i(\mathbf{x})| \leq \frac{1}{2}\|\mathbf{c}_i\|_{Q_{\hat{\mathbf{a}}\hat{\mathbf{a}}}}, i = 1, \dots, p\} \supset \mathcal{P}_{a,ILS} \quad (3.16)$$

The idea is illustrated in Figure 3.3 for the 2D case where  $U_a$  is chosen as the intersection of two banded subsets. The probability  $P(\hat{\mathbf{a}} \in U_a)$ , however, cannot be evaluated exactly either, but can be bounded from above to obtain Teunissen (1998c)

$$P_{s,ILS} \leq P(\hat{\mathbf{a}} \in U_a) \leq \prod_{i=1}^p \left(2\Phi\left(\frac{1}{2\sigma_{v_i|I}}\right) - 1\right) = P_{UB,region} \quad (3.17)$$

with the conditional standard deviation  $\sigma_{v_i|I}$  of vector  $\mathbf{v}$ . These are equal to the square root of the diagonal elements of  $\mathbf{D}$  from the  $\mathbf{L}^T \mathbf{D} \mathbf{L}$ -decomposition of  $\mathbf{Q}_{vv}$  with its elements given by:

$$\sigma_{v_i v_j} = \frac{\mathbf{u}_i^T \mathbf{Q}_{\hat{\mathbf{a}}\hat{\mathbf{a}}}^{-1} \mathbf{u}_j}{\|\mathbf{u}_i\|_{Q_{\hat{\mathbf{a}}\hat{\mathbf{a}}}} \|\mathbf{u}_j\|_{Q_{\hat{\mathbf{a}}\hat{\mathbf{a}}}}}, \quad \mathbf{u}_i, \mathbf{u}_j \in \mathbb{Z}^n$$

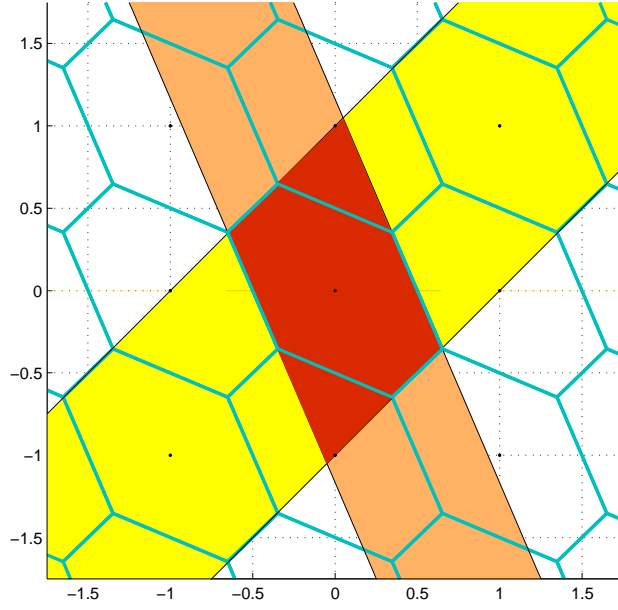


Figure 3.3: Integration region (red) containing  $\mathcal{P}_{0,\text{ILS}}$  and defined by the intersection of two banded subsets.

where the  $\mathbf{u}_i$ ,  $i = 1, \dots, p$  need to be linearly independent to guarantee the full-rank of VC-matrix  $\mathbf{Q}_{vv}$ . The procedure for computation of this upper bound is as follows. LAMBDA is used to find the  $q \gg n$  closest integers  $\mathbf{u}_i \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$  for  $\hat{\mathbf{a}} = \mathbf{0}$ . These  $q$  integer vectors are ordered by increasing distance to the zero vector, measured in the metric  $\mathbf{Q}_{\hat{\mathbf{a}}\hat{\mathbf{a}}}$ . Start with  $\mathbf{U} = \mathbf{u}_1$  so that  $\text{rank}(\mathbf{U}) = 1$ . Then find the first candidate  $\mathbf{u}_j$ . ( $j = 2, \dots, q$ ) for which  $\text{rank}([\mathbf{u}_1 \ \mathbf{u}_j]) = 2$ . Continue with  $\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_j]$ , and find the next candidate that results in an increase in rank. Continue this process until  $\text{rank}(\mathbf{U}) = n$ . It cannot guarantee that we can find  $n$  independent integer vectors from  $q$  integer vectors. In Ps-LAMBDA software, we initially set  $q = 100n$  and if we obtain that  $\text{rank}(\mathbf{U}) < n$  from these  $q$  integer vectors, we augment  $\mathbf{U}$  until its rank equal to  $n$  by trying  $c_i$ ,  $i = 1, \dots, n$  with all elements of 0s except 1 at  $i$ th slot. For more information, one refers to Teunissen (1998c) and Verhagen (2005).

Note that in the higher dimensional case many subsets are necessary to obtain a tight upper bound, and selection of the subset is rather complicated. In addition, it is computationally demanding, since the determination of the subset involves the evaluation of many integer candidates to be obtained with LAMBDA.

For the lower bound, one can find an ellipsoid  $L_a$  to expand the pull-in region  $\mathcal{P}_{a,\text{ILS}}$  from inside Teunissen (1998c)

$$L_a = \left\{ \mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x} - \mathbf{a}\|_{\mathbf{Q}_{\hat{\mathbf{a}}\hat{\mathbf{a}}}}^2 \leq \frac{1}{4} \min_{\mathbf{u} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}} \|\mathbf{u}\|_{\mathbf{Q}_{\hat{\mathbf{a}}\hat{\mathbf{a}}}}^2 \right\}$$

The concept is illustrated in Figure 3.4 for two different pull-in regions, corresponding to different VC-matrices  $\mathbf{Q}_{\hat{\mathbf{a}}\hat{\mathbf{a}}}$ . Therefore lower bound  $P(\hat{\mathbf{a}} \in L_a)$  can be evaluated based on the  $\chi^2$ -distribution:

$$P_{s,\text{ILS}} \geq P(\hat{\mathbf{a}} \in L_a) = P\left(\chi^2(n, 0) \leq \frac{1}{4} \min_{\mathbf{u} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}} \|\mathbf{u}\|_{\mathbf{Q}_{\hat{\mathbf{a}}\hat{\mathbf{a}}}}^2\right) = P_{\text{LB,region}} \quad (3.18)$$

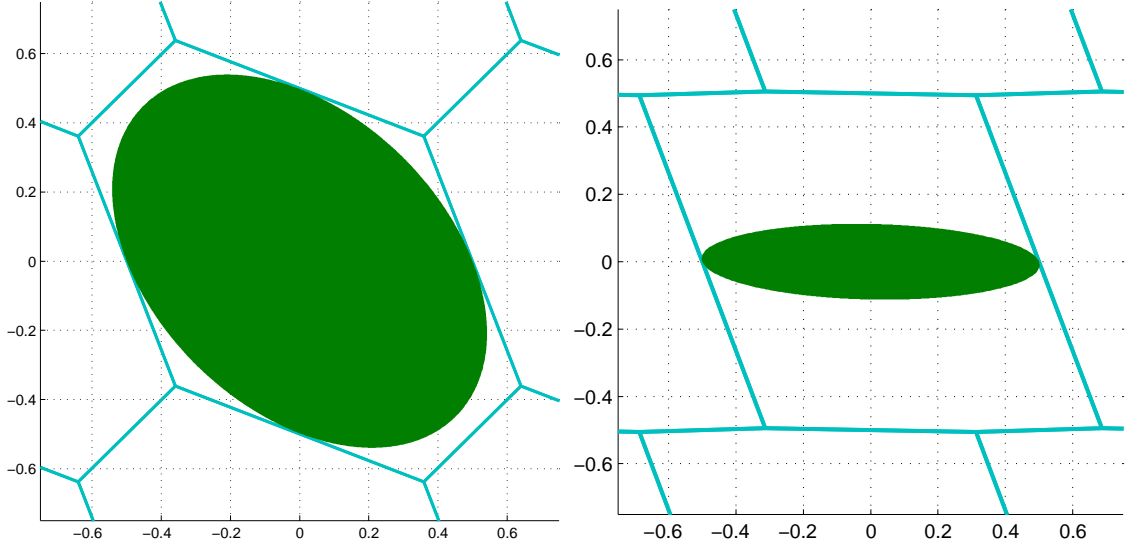


Figure 3.4: Two examples of the ellipsoidal region (green) contained by the pull-in region  $\mathcal{P}_{0,\text{ILS}}$  (different shape of pull-in regions is due to different VC-matrices  $\mathbf{Q}_{\hat{a}\hat{a}}$ ).

### Upper and lower bounds based on bounding VC-matrix

It is also possible to obtain a lower and an upper bound by bounding the actual VC-matrix from above and below by diagonal matrices, and then to computing the ILS success rate belonging to these diagonal matrices becomes straightforward (Teunissen 1998c). The simplest way of bounding the actual ambiguity VC from above and below, is to make use of unit matrix scaled by its maximum and minimum eigenvalues. This gives

$$\lambda_{\min} \mathbf{I}_n \leq \mathbf{Q}_{\hat{z}\hat{z}} \leq \lambda_{\max} \mathbf{I}_n \quad (3.19)$$

where  $\lambda_{\min}$  and  $\lambda_{\max}$  are the minimum and maximum eigenvalues of  $\mathbf{Q}_{\hat{z}\hat{z}}$ , and  $\mathbf{I}_n$  is an identity matrix of order  $n$ . The ILS success rate bounds follow as:

$$P_{\text{LB, eigen}} = \left( 2\Phi\left(\frac{1}{2\sqrt{\lambda_{\max}}}\right) - 1 \right)^n \leq P_{s,\text{ILS}} \leq \left( 2\Phi\left(\frac{1}{2\sqrt{\lambda_{\min}}}\right) - 1 \right)^n = P_{\text{UB, eigen}} \quad (3.20)$$

Note that the two bounds coincide when the two extreme eigenvalues coincide. This is the case when the ambiguity VC-matrix itself is a scaled unit matrix. In the real GNSS applications, these two extreme eigenvalues will differ considerably when the VC-matrix of the original DD ambiguities is used. In that case the above two bounds would become too loose to be useful. When using the decorrelated ambiguities as produced by the LAMBDA method, the elongation of the ambiguity search space is considerably reduced and the ratio of the two extreme eigenvalues is pushed towards its minimum of one. Hence, the above bounds are much sharper using the eigenvalues of the transformed ambiguity VC-matrix than using the eigenvalues of the original DD ambiguity VC-matrix. In Ps-LAMBDA software, the decorrelated ambiguity VC-matrix is employed.

## Chapter 4

# Routines and their usage

### 4.1 Overview of Ps-LAMBDA software

Figure 4.1 gives an overview of the structure of Ps-LAMBDA software. The options of approximations and bounds of success-rate are shown for ILS, IB and IR methods.

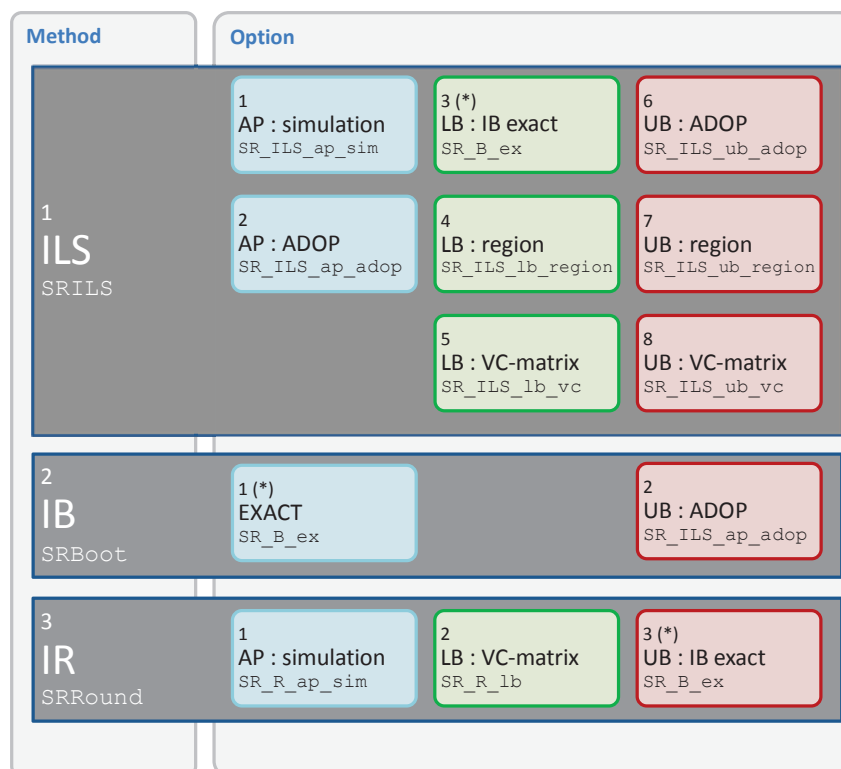


Figure 4.1: Ps-LAMBDA: overview of available methods and options in routine SuccessRate. Default option is indicated with (\*). Names of underlying routines are shown as well. AP=approximation (blue), LB=lower bound (green), UB=upper bound (red).



## 4.2 The main Ps-LAMBDA routine

The main routine is

```
Ps = SuccessRate(Qa, method, opt, decor, nsamp)
```

The VC-matrix Qa should be square ( $n \times n$ ), symmetric and positive-definite. The input and output arguments will be described first, followed by some examples that which subroutine can be used if the user is interested in the different methods.

### 4.2.1 Input arguments

The following input arguments are needed, depending on the method of choice:

```
Qa : VC-matrix of ambiguities
method : 1 - ILS [DEFAULT] (SRILS.m)
         2 - IB           (SRBoot.m)
         3 - IR           (SRRound.m)
opt : Approximation / bound to compute, depending on method
decor : 1 - decorrelation [DEFAULT]
        1 - no decorrelation
nsamp : number of samples only used for simulation-based approximation.
```

The choice for “decor” is only relevant for IR and IB, since these estimators are not  $\mathcal{Z}$ -invariant. Decorrelation is always applied for ILS to ensure computation efficiency.

The output “Ps” is the computed success rate.

### 4.2.2 ILS success rate (SRILS.m)

The routine for computing the approximation / bound of ILS success rate is

```
Ps = SRILS(Qa,opt,nsamp);
```

Here there are 8 options for approximation / bounds of ILS success rate:

```
opt : 1 - Monte-Carlo simulation based approximation (SRsm_ILS.m)
      2 - ADOP based approximation                  (SRa_adop.m)
      3 - Lower bound of IB success rate [DEFAULT]   (SRe_boot.m)
      4 - Lower bound by bounding pull-in region     (SRlb_region.m)
      5 - Lower bound by bounding covariance matrix  (SRlb_vcv.m)
      6 - Upper bound based on ADOP                  (SRub_adop.m)
      7 - Upper bound by bounding pull-in region     (SRub_region.m)
      8 - Upper bound by bounding covariance matrix  (SRub_vcv.m)
      9 - All
```

One can invoke this subroutine by specifying `method=1` in the main routine. Alternatively, one can also directly invoke this subroutine. But now one needs to first apply the decorrelation transformation on  $Q_{\hat{a}\hat{a}}$  to ensure the computation efficiency.

### 4.2.3 IB success rate (SRBoot.m)

The routine for computing the approximation / bound of IB success rate is

```
Ps = SRBoot(Qa,opt,decor);
```

Here there are 2 options of success rate approximation / bounds:

```
opt : 1 - Exact success-rate [DEFAULT] (SRe_boot.m)
      2 - ADOP-based upper bound      (SRa_adop.m)
      3 - All
```

For subroutine “SRa\_adop”, it is independent on the ambiguity transformation, referring to subsection 3.3.2, while “SRe\_boot” depends on the decorrelation. If the input “decor” is true,  $Q_{\hat{a}\hat{a}}$  will be transformed firstly and then used for “SRe\_boot”.

### 4.2.4 IR success rate (SRRound.m)

The routine for computing the approximation / bound of IR success rate is

```
Ps = SRRound(Qa,opt,decor,nsamp);
```

It has 4 options for success rate approximations / bounds:

```
opt : 1 - Monte-Carlo simulation based approximation (SRsm_round.m)
      2 - Lower bound based on diagonal VC-matrix   (SRlb_round.m)
      3 - Upper bound based on IB success-rate       (SRe_boot.m)
      4 - All
```

All of these subroutines are dependent on the decorrelation. If “decor” is true, all subroutines will be based on the decorrelated VC-matrix.

## 4.3 Routines used by Ps-LAMBDA

Two subroutines, `decorrel.m` and `ssearch.m`, used for ambiguity resolution in LAMBDA software are used in Ps-LAMBDA software. Subroutine `decorrel.m` is used to conduct the decorrelating  $Z$ -transformation; while `ssearch.m` is used to implement the integer search with “search-and-shrink” technique. For more information, one can refer to the manual of new version LAMBDA software (Verhagen and Li 2012).

## Chapter 5

# Getting started and performance aspects

### 5.1 Getting started

It is most convenient to include the folder with the Ps-LAMBDA routines in your Matlab path. The folder contains a demonstration routine SR\_GUI with examples of how the program can be used. Open the routine in your editor; the comments will guide you through the different options.

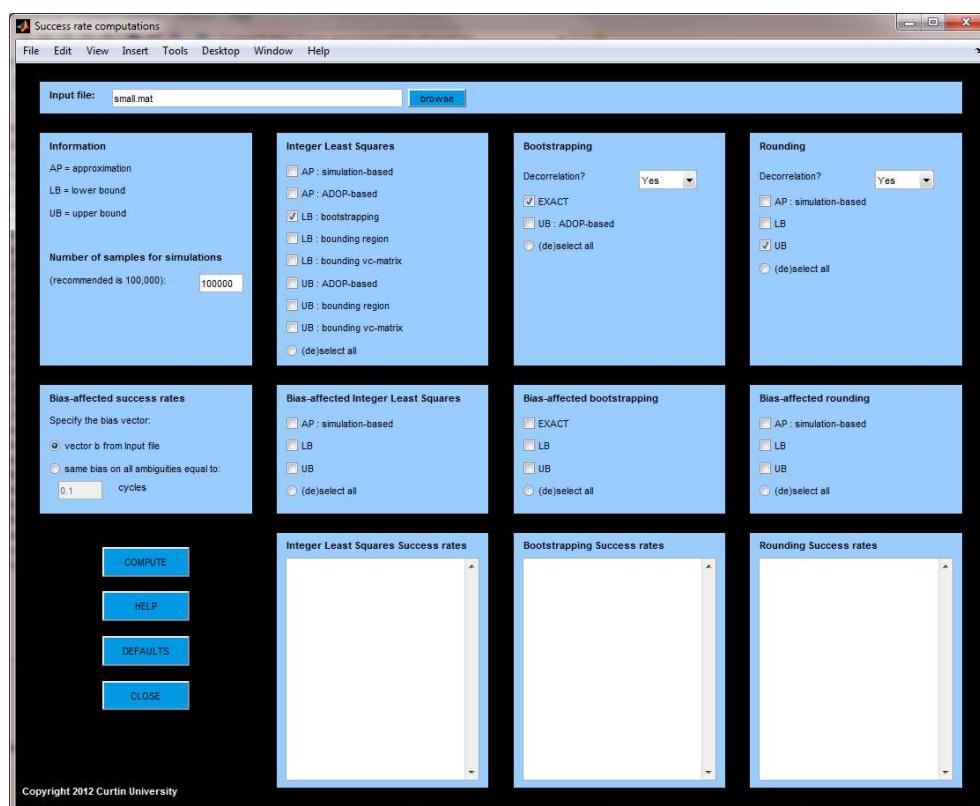


Figure 5.1: Graphical User Interface of Ps-LAMBDA software.

The toolbox also includes a Graphical User Interface (see Figure 5.1) which allows the user to select an input file which contains the VC-matrix  $Q_{\hat{a}\hat{a}}$  and to compute all the desired bounds and approximations

for different integer estimation methods simultaneously.

## 5.2 Performance aspects

In this section the bounds and approximations for ILS, IB and IR methods, will be briefly assessed for different GNSS models, where the different factors affecting the float ambiguity precision are varied as shown in Table 5.1. An exponential elevation-dependent weighting is applied (more noise is assumed for observations from low-elevation satellites) to the standard deviations of the observations and of the ionosphere corrections. The scale factors applied to the VC-matrix  $Q_{\hat{a}\hat{a}}$  can either be interpreted as representing a different number of epochs, or a different measurement precision due to different receiver quality. In following, the Monte-Carlo simulation based success rate is referred as actual success rate with number of samples  $10^6$ .

Table 5.1: Measurement scenarios (standard deviations (STD) apply to zenith direction).

system	GPS - combined GPS+Galileo
times	49 different epochs
frequencies	L5 - L1+L5 - L1+L5+L2/E5b
STD of undifferenced observations	code: 15 cm; phase: 1 mm
VC-matrix scale factors	0.25 - 0.5 - 1 - 2 - 4
STD of ionosphere corrections	5 - 15 mm

### 5.2.1 IR success rate

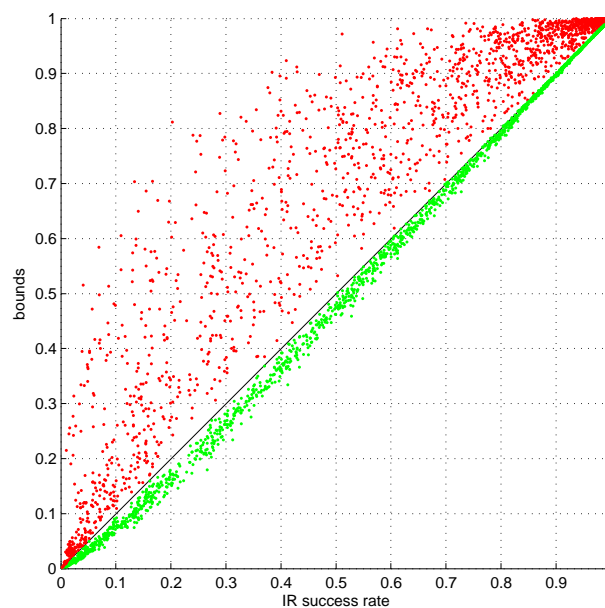


Figure 5.2: IR success rates: upper bound based on IB (red; Eq 3.9) and lower bound based on diagonal VC-matrix (green; Eq 3.7) versus the actual IR success rate for the models from Table 5.1.

Figure 5.2 shows the lower bound and upper bound versus the actual IR success rates (all for the decorrelated ambiguities). It can be seen that the lower bound is very tight, whereas the upper bound based on the IB success rate is not as tight, thus indicating that IB may still significantly outperform IR.

### 5.2.2 IB success rate

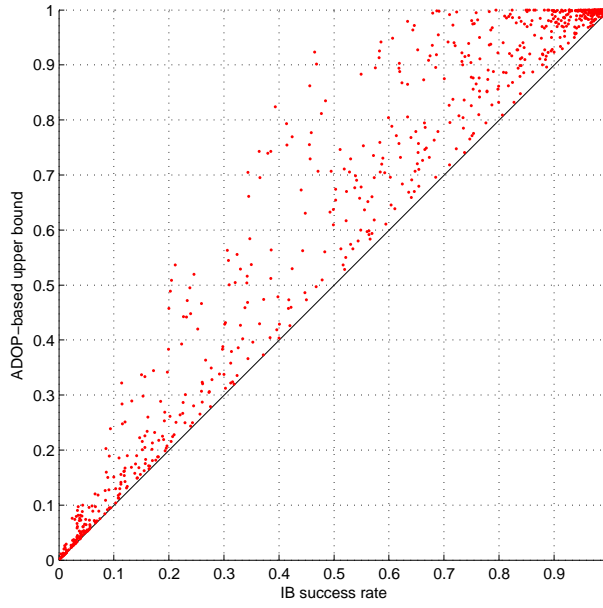


Figure 5.3: IB success rates: ADOP-based upper bound (Eq 3.10) versus the exact IB success rate (Eq 3.9) for the models from Table 5.1.

Figure 5.3 shows that the ADOP-based upper bound is in these cases often significantly higher than the exact IB success rate  $P(\tilde{z}_{IB} = z)$ . Better bounding performance is obtained for lower dimensions  $n$ , which is due to the replacement of the  $n$  conditional standard deviations in Eq.(3.9) by a single value equal to ADOP.

### 5.2.3 ILS success rate

Figure 5.4 (Left) shows how the IB success rate performs as a lower bound for ILS. In practice, the IB success rate is commonly used as the best known lower bound, and these results confirm that especially if the success rate is high, this is indeed the case. At the same time, it can be seen how ILS may still significantly outperform IB for lower success rates. For these cases the ADOP-based upper bound often gives a too optimistic value compared to the actual success rate. As is shown later, however, the bounding performance improves for lower dimensions (cf. Figure 5.7). A similar conclusion can be given for the ADOP-based approximation of the ILS success rate as shown in Figure 5.4 (Right). Only in some of these cases can it be used as a coarse approximation. The approximation improves in case of lower dimensions (cf. Figure 5.7).

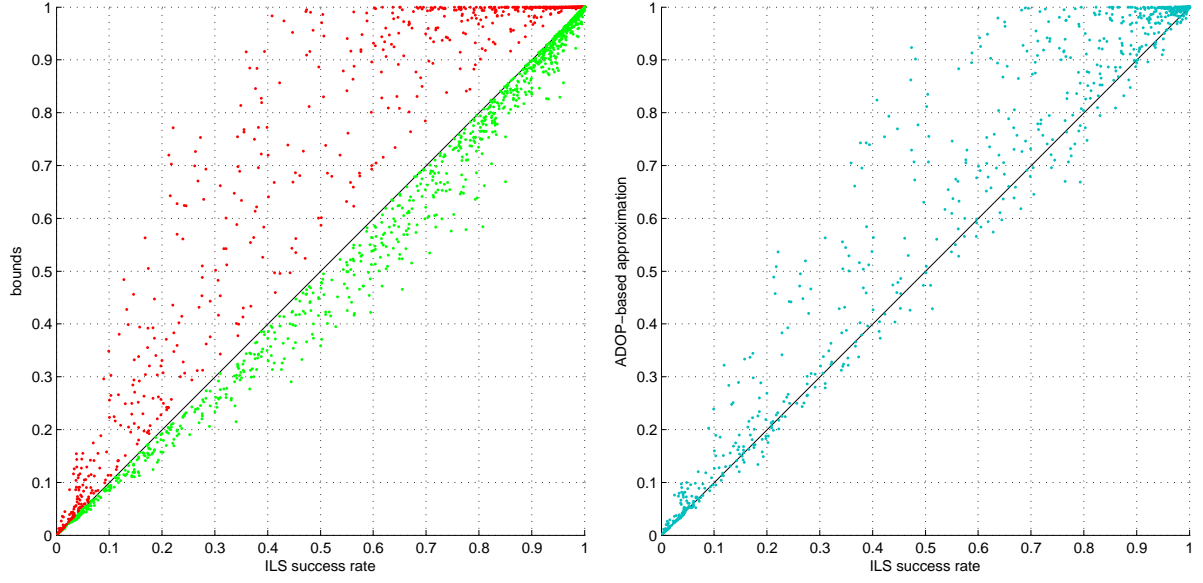


Figure 5.4: ILS success rates: lower bound based on IB (green; Eq 3.9) and upper bound based on ADOP (red; Eq 3.14) versus the actual ILS success rate (Left); ADOP-based approximation (Eq 3.10) versus the actual ILS success rate (Right) for the models from Table 5.1.

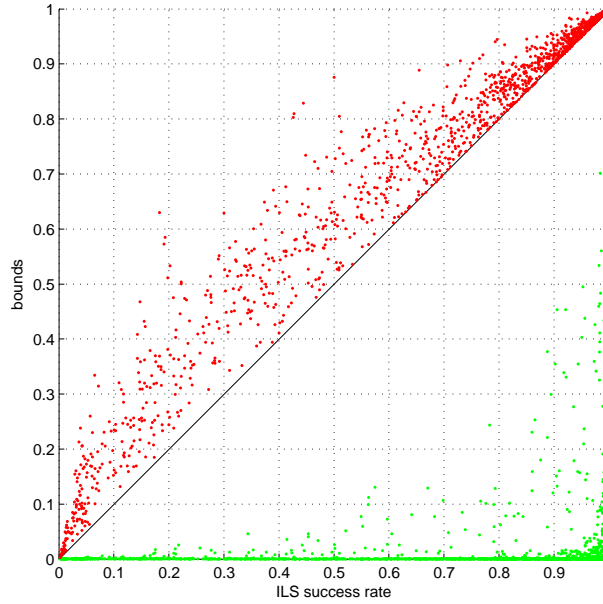


Figure 5.5: ILS success rates: lower (Eq 3.17) and upper bounds (Eq 3.17) based on bounding the pull-in region versus the actual ILS success rate for the models from Table 5.1.

Figure 5.5 shows the lower and upper bound of the ILS success rate based on bounding the pull-in region. It can be seen that the upper bound performs reasonably well, whereas the lower bound is generally not tight at all - it will be close to zero unless the success rate is very close to 1. The bad performance can be explained based on the 2D example on the right-hand side of Figure 3.4: the ellipsoidal region may leave a large part of the ILS pull-in region uncovered. This will be the case when there is a large variation in the variances  $\sigma_{\hat{z}_i \hat{z}_i}$  (making the ellipsoidal region elongated).

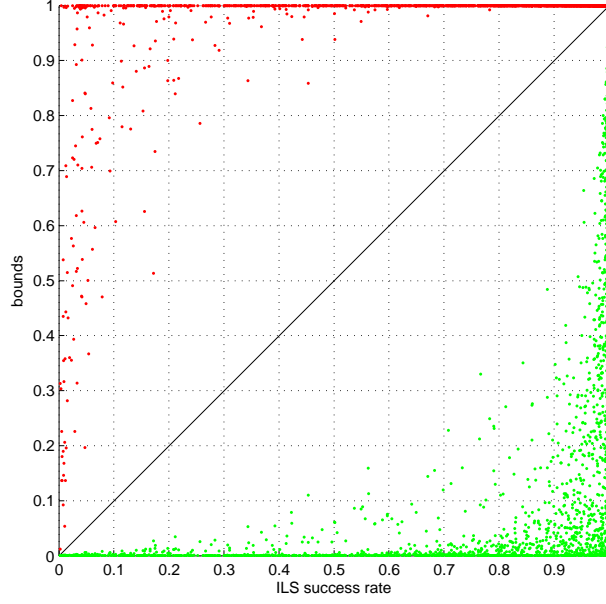


Figure 5.6: ILS success rates: lower and upper bounds (Eq 3.20) based on bounding the VC-matrix versus the actual ILS success rate for the models from Table 5.1.

Figure 5.6 shows the lower and upper bound of the ILS success rate based on bounding the VC-matrix. It can be seen that both bounds perform poorly. Similarly as with the ADOP-based approximation of the ILS success rate, this is especially true for large  $n$  due to the replacement of the  $n$  conditional standard deviations in Eq.(3.9) by the square root of the minimum or maximum eigenvalue, respectively.

#### 5.2.4 Examples with other models

So far, the performance of the success rate bounds and approximations was analyzed based on the linearized DD GNSS model parameterized in terms of the baseline unknowns. However, the geometry-free model is used for example for integrity monitoring or as a first step in the data processing. Here, we will show an example based on a dual-frequency GPS model for one satellite-receiver pair (i.e. one DD code and phase observation per frequency). The undifferenced code and phase standard deviations were set to 15 cm and 1.5 mm, respectively. The float ambiguity VC-matrix (units are *cycles*<sup>2</sup>) obtained in this way is:

$$\mathbf{Q}_{\hat{a}\hat{a}} = \begin{bmatrix} 1.2429 & 0.9683 \\ 0.9683 & 0.7547 \end{bmatrix} \quad (5.1)$$

In addition, a scaling is applied to analyze the performance for different precisions:

$$\mathbf{Q}_{\hat{a}\hat{a},f} = f \times \mathbf{Q}_{\hat{a}\hat{a}} \quad (5.2)$$

The ILS success rate approximations and bounds are shown in Figure 5.7 as a function of the scale factor  $f$ . The lower bound based on the exact IB success rate is very sharp. Interestingly, this also holds for the ADOP-based upper bound and approximation (the orange line is hardly visible, as it is plotted below the graph of the simulation-based success rate). In this case the bounds based on bounding the

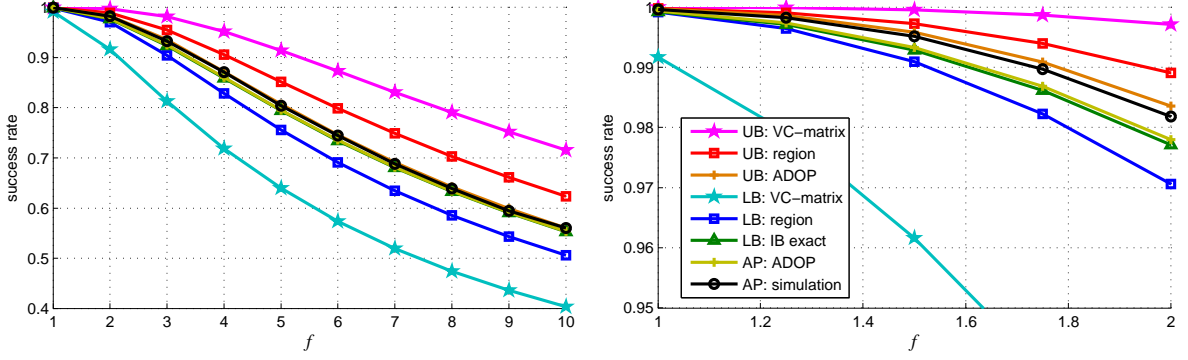


Figure 5.7: ILS success rate bounds for 2-frequency geometry-free model with 2 ambiguities,  $f$  is the scale factor applied to the VC-matrix (right panel shows same results, but only for smaller  $f$ ).

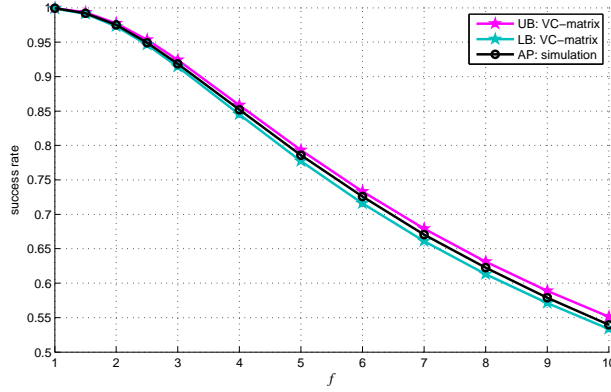


Figure 5.8: ILS success rate bounds based on bounding the  $(2 \times 2)$  scaled VC-matrix with both variances equal to 0.02, and covariance equal to 0.0005. The scale factor is equal to  $f$ .

integration region are quite sharp if the success rate is high, but become less tight as the scale factor increases, and consequently the success rate decreases.

In all results shown so far, the bounds based on bounding the VC-matrix  $Q_{\hat{a}\hat{a}}$  are generally not tight at all. An example where also these bounds will work well is when all variances are equal to a certain value  $v$  and all the covariances equal to a value  $c$ , with  $v \gg c$ :

$$\sigma_{\hat{a}_i \hat{a}_i}^2 = v, \quad \sigma_{\hat{a}_i \hat{a}_j} = c, \quad \forall i, j = 1, \dots, n; \quad i \neq j \quad (5.3)$$

Figure 5.8 shows the bounds for an example with  $n = 2$ ,  $v = 0.02$  and  $c = 0.0005$ . Again the scaling according to Eq.(5.2) is applied.

### 5.2.5 Which bounds or approximations to use?

The results in this section show that the success rate bounds and approximations differ in their performance. The simulation-based approximations of the IR and ILS success rates work well if enough samples are used. However, they may not be suitable for real-time applications as their computation time may be long. Computation time will also be an issue for real-time applications if the upper bound



of the ILS success rate based on bounding the pull-in region is considered. For design and research purposes, as well as for post-processing, computation time will not be an issue. All other bounds and approximations can be used in real-time.

For the IR success rate, the lower bound was shown to perform well. For the ILS success rate, the lower bound based on the exact IB success rate, and the upper bound based on bounding the pull-in region generally perform very well for the GNSS models considered here. Furthermore, it was shown that the other bounds and approximations may work well for certain applications where the dimension is lower or the structure of the VC-matrix  $Q_{\hat{a}\hat{a}}$  is different, see for example Figures 5.7 and 5.8.

## Chapter 6

# Availability, Liability and Updates

### 6.1 Availability

The Matlab implementation of the Ps-LAMBDA software *Version 1.0* is available on request.

### 6.2 Liability

Use of the accompanying Ps-LAMBDA software is allowed, but no liability for the use of the software will be accepted by the authors or their employer. Giving proper credits to the authors is the only condition posed upon the use of the Ps-LAMBDA software. We ask you to refrain from passing the software to third parties. Instead you are asked to pass our (e-mail / website) address to them, so we can send the software upon their request or they can download freely from website. The reason is that in this way we have a complete overview of the users of the software, enabling us to keep everyone informed of further developments.

### 6.3 Updates

We welcome any suggestion for improvement of the code, the in-source documentation and the description in the report. We also would like to encourage you to communicate to us about results obtained with the Ps-LAMBDA method, and comparisons made with other methods. We would also be much obliged if you inform us in case you decide to use the method commercially. As said before, there are no restrictions on that, other than properly acknowledging the designers of the method and their employer. If you are planning to make a version in another language and would like to make it public, we would like you to contact us, in order to coordinate the efforts.

# Bibliography

- Boon F, Ambrosius B (1997). Results of real-time applications of the LAMBDA method in GPS based aircraft landings. In *Proc. of the International Symposium on Kinematic Systems in Geodesy, Geomatics and Navigation, Banff, Canada*, pp. 339–345.
- Boon F, De Jonge PJ, Tiberius CCJM (1997). Precise aircraft positioning by fast ambiguity resolution using improved troposphere modelling. In *Proc. of ION GPS-1997, Kansas City MO*, pp. 1877–1884.
- Chang X, Yang X, Zhou T (2005). MLAMBDA: a modified LAMBDA method for integer least-squares estimation. *Journal of Geodesy*, 79: 552–565.
- De Jonge PJ, Tiberius C (1996a). The LAMBDA method for integer ambiguity estimation: implementation aspects, LGR-Series, No 12. Technical report, Delft University of Technology.
- De Jonge PJ, Tiberius CCJM (1996b). Integer ambiguity estimation with the LAMBDA method. In *Proc. of IAG Symposium No. 115, GPS trends in terrestrial, airborne and spaceborne applications, G. Beutler et al. (eds), Springer Verlag*, pp. 280–284.
- De Jonge PJ, Tiberius CCJM, Teunissen PJG (1996). Computational aspects of the LAMBDA method for GPS ambiguity resolution. In *Proc. of ION GPS-1996, Kansas City MO*, pp. 935–944.
- Hassibi A, Boyd S (1998). Integer parameter estimation in linear models with applications to GPS. *IEEE Transactions on Signal Processing*, 46(11): 2938 –2952.
- Hofmann-Wellenhof B, Lichtenegger H, Collins J (2001). *Global positioning system: theory and practice*, 5th edn. Springer Berlin Heidelberg, New York.
- Jonkman NF (1998). *Integer GPS ambiguity estimation without the receiver-satellite geometry*. Delft Geodetic Computing Centre, LGR series No.18, Delft University of Technology, 95pp.
- Joosten P, Tiberius CCJM (2000). Fixing the ambiguities: are you sure they're right. *GPS World*, 11(5): 46–51.
- Joosten P, Tiberius CCJM (2002). LAMBDA: FAQs. *GPS Solutions*, 6(1-2): 109 – 114.
- Leick A (2004). *GPS satellite surveying*. 3rd edn. John Wiley, New York.
- Li B, Teunissen PJG (2011). High dimensional integer ambiguity resolution: A first comparison between LAMBDA and Bernese. *The Journal of Navigation*, 64: S192–S210.
- Misra P, Enge P (2001). *Global Positioning System: Signals, Measurements, and Performance*. Ganga-Jamuna Press, Lincoln MA.
- Odijk D, Teunissen PJG (2008). ADOP in closed form for a hierarchy of multi-frequency single-baseline GNSS models. *Journal of Geodesy*, 82: 473–492.

- Strang G, Borre K (1997). *Linear Algebra, Geodesy, and GPS*. Wellesley-Cambridge Press, Wellesley MA.
- Teunissen P (1999a). The probability distribution of the GPS baseline for a class of integer ambiguity estimators. *Journal of Geodesy*, 73: 275–284.
- Teunissen PJG (1993a). Least squares estimation of the integer GPS ambiguities. In *Invited lecture, Section IV Theory and Methodology, IAG General Meeting, Beijing*.
- Teunissen PJG (1993b). Least-squares estimation of the integer GPS ambiguities. invited lecture. In *Section IV Theory and Methodology, IAG General Meeting, August, Beijing, China*.
- Teunissen PJG (1994). A new method for fast carrier phase ambiguity estimation. In *Proceedings IEEE Position, Location and Navigation Symposium PLANS'94, Las Vegas, NV*, pp. 562–573.
- Teunissen PJG (1995a). The invertible GPS ambiguity transformations. *Manuscripta Geodaetica*, 20: 489–497.
- Teunissen PJG (1995b). The least-squares ambiguity decorrelation adjustment: a method for fast GPS integer ambiguity estimation. *Journal of Geodesy*, 70: 65–82.
- Teunissen PJG (1997). A canonical theory for short GPS baselines. Part IV: Precision versus reliability. *Journal of Geodesy*, 71: 513–525.
- Teunissen PJG (1998a). A class of unbiased integer GPS ambiguity estimators. *Artificial Satellites*, 33(1): 4–10.
- Teunissen PJG (1998b). *GPS carrier phase ambiguity fixing concepts*. In: PJG Teunissen and Kleusberg A, *GPS for Geodesy*, Springer-Verlag, Berlin.
- Teunissen PJG (1998c). On the integer normal distribution of the GPS ambiguities. *Artificial Satellites*, 33(2): 49–64.
- Teunissen PJG (1998d). Some remarks on GPS ambiguity resolution. *Artificial Satellites*, 32(3): 119–130.
- Teunissen PJG (1998e). Success probability of integer GPS ambiguity rounding and bootstrapping. *Journal of Geodesy*, 72: 606–612.
- Teunissen PJG (1999b). An optimality property of the integer least-squares estimator. *Journal of Geodesy*, 73(11): 587–593.
- Teunissen PJG (1999c). An optimality property of the integer least-squares estimator. *Journal of Geodesy*, 73: 587–593.
- Teunissen PJG (2000). ADOP based upperbounds for the bootstrapped and the least-squares ambiguity success rates. *Artificial Satellites*, 35(4): 171–179.
- Teunissen PJG (2001a). GNSS ambiguity bootstrapping: Theory and applications. In *Proc. KIS2001, International Symposium on Kinematic Systems in Geodesy, Geomatics and Navigation, June 5-8, Banff, Canada*, pp. 246–254.
- Teunissen PJG (2001b). Integer estimation in the presence of biases. *Journal of Geodesy*, 75: 399–407.
- Teunissen PJG (2001c). Integer estimation in the presence of biases. *Journal of Geodesy*, 75: 399–407.

- Teunissen PJG (2001d). Statistical GNSS carrier phase ambiguity resolution: a review. In *Proc. of 2001 IEEE Workshop on Statistical Signal Processing, August 6-8, Singapore*, pp. 4–12.
- Teunissen PJG (2002). The parameter distributions of the integer GPS model. *Journal of Geodesy*, 76(1): 41–48.
- Teunissen PJG (2003). Theory of carrier phase ambiguity resolution. *Wuhan University Journal of Natural Sciences*, 8: 471–484.
- Teunissen PJG (2010). Mixed integer estimation and validation for next generation GNSS. In W. Freeden, M. Nashed, and T. Sonar (Eds.), *Handbook of Geomathematics*, pp. 1101–1127. Springer Berlin Heidelberg.
- Teunissen PJG, De Jonge PJ, Tiberius CCJM (1996). The volume of the GPS ambiguity search space and its relevance for integer ambiguity resolution. In *Proc. of ION GPS-1996, Kansas City MO*, pp. 889–898.
- Teunissen PJG, De Jonge PJ, Tiberius CCJM (1998). Performance of the LAMBDA method for fast GPS ambiguity resolution. *Navigation*, 44(3): 373–383.
- Teunissen PJG, Joosten P, Tiberius CCJM (2000). Bias robustness of GPS ambiguity resolution. In *Proc. of ION GPS-2000, Salt Lake City UT*, pp. 104–112.
- Teunissen PJG, Kleusberg A (1998). *GPS for geodesy, 2nd edn*. Springer Berlin Heidelberg New York.
- Teunissen PJG, Odijk D (1997). Ambiguity Dilution of Precision: definition, properties and application. In *Proc. of ION GPS-1997, Kansas City MO*, pp. 891–899.
- Teunissen PJG, Verhagen S (2008). GNSS Carrier Phase Ambiguity Resolution: Challenges and Open Problems. In *M Sideris (ed) Observing our changing Earth, International Association of Geodesy, Volume 133*, pp. 785–792. Springer Verlag, Berlin.
- Tiberius CCJM, De Jonge PJ (1995). Fast positioning using the LAMBDA method. In *Proc. of DSNS'95, Bergen, Norway*, pp. paper no.30. The Nordic Institute of Navigation, Oslo.
- Verhagen S (2005). On the reliability of integer ambiguity resolution. *Navigation*, 52(2): 99–110.
- Verhagen S, Joosten P (2004). Analysis of integer ambiguity resolution algorithms. *European Journal of Navigation*, 2(4): 38–50.
- Verhagen S, Li B (2012). Lambda software package: Matlab implementation, version 3.0. Technical report, Delft University of Technology and Curtin University.