LAMBDA toolbox

Version 4.0 – Documentation

Massarweh, L., Verhagen, S., and Teunissen, P.J.G.

Delft University of Technology

This page intentionally left blank

Table of Contents

Changelog

Date	Author(s)	Version	Comments
01/06/2024	Massarweh, L., Verhagen, S., Teunissen, P.J.G.	v1.0	Technical documentation for LAMBDA 4.0 toolbox.
…			

List of Figures

List of Tables

1. Introduction

The Least-squares AMBiguity Decorrelation Adjustment method, or LAMBDA method, has been introduced in Teunissen (1993) and it has been adopted over almost 30 years. The method deals with mixed-integer models' estimation problems, with some of the *unknown* parameters subject to integer constraints. An example can be found in Global Navigation Satellite System (GNSS) applications, where by resolving integer-constrained ambiguities in the carrier-phase measurements it becomes possible to exploit their millimeter-level precision. This is only *one* example of LAMBDA supported applications, and others will be discussed later in this documentation.

1.1. History of LAMBDA developments

The LAMBDA toolbox has originally been developed in the 1990s at the Delft University of Technology. A timeline of past and current implementations is given in Figure 1.

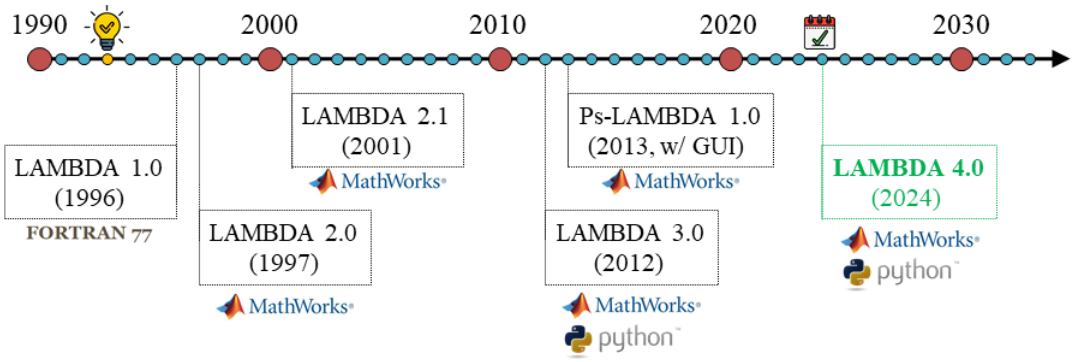


Figure 1: Timeline of the LAMBDA software implementations, and programming languages considered for each release version.

A first release of LAMBDA software, Version 1.0, was provided in Fortran-77 and main implementation aspects are extensively described in de Jonge and Tiberius (1996). This was later translated by Kai Borre to MATLAB Version 2.0, see Section 15.6 in Strang and Borre (1997), while further improved to Version 2.1 by Joosten (2001). In Verhagen et al. (2012), a LAMBDA Version 3.0 has been presented for MATLAB environment, and later translated to Python language by Psychas et al. (2019). In addition to LAMBDA toolbox developments, dealing with integer estimation and validation, an additional Ps-LAMBDA toolbox, Version 1.0, was implemented by Verhagen et al. (2013), which focuses on the statistical evaluation of the ambiguity success rate.

1.1.1. Improvements in Version 2.1

The main features and improvements, with respect to earlier versions, are:

- Good readability of the software.
- A modular approach with a limited set of subroutines, enabling one to perform for example only the decorrelation-step, only the search-step, or both steps.
- Possibility to find a number of candidates larger than 2, which can be very useful for research purposes.
- The size of the search ellipsoid is based on bootstrapping instead of rounding (if requested number of candidate vectors is smaller than $n + 1$, with n the dimension of the float ambiguity vector).

1.1.2. Improvements in Version 3.0

The main features and improvements, with respect to earlier versions, are:

- An alternative search strategy is implemented, based on searching in an alternating way around the conditional estimates and shrinking the search ellipsoid. The latter was proposed in Teunissen (1993), see de Jonge and Tiberius (1996). Chang et al. (2005) implemented the technique in their MLAMBDA package.
- The bootstrapping and rounding estimators can be applied in case one does not want to apply Integer Least-Squares (ILS).
- It is possible to output the integer bootstrapping success rate with decorrelated ambiguities. This success rate, following decorrelation, is known to be a tight lower bound of the ILS success rate (Teunissen 1999).
- Partial ambiguity resolution can be applied, based on fixing a selected subset of the decorrelated ambiguities such that the success rate will be larger or equal to a minimum required success rate.
- The Ratio Test can be applied to decide on acceptance of the fixed solution. The model-driven Ratio Test with Fixed Failure rate can be applied or the Ratio Test with a fixed (user-defined) threshold value.

1.1.3. Improvements in Version 4.0

The main features and improvements, with respect to earlier versions, are:

- Implementation of the main algorithms has been improved, now computationally more suitable for high dimensional problems.

- Additional functionalities are implemented, as well as new (classes of) estimators are also introduced, such as
  - Vectorial Integer Bootstrapping (VIB), member of the Integer class and introduced in Teunissen et al. (2021);
  - Integer Aperture Bootstrapping (IAB), member of the Integer Aperture class and introduced in Teunissen (2005a);
  - Best Integer Equivariant (BIE), member of the Integer Equivariant class and introduced in Teunissen (2003a);

- A novel integer search algorithm is adopted, firstly introduced in Ghasemmehdi and Agrell (2011).

- An alternative partial ambiguity resolution (PAR) approach is included now, while making use of the BIE estimator.

- The look-up tables from Hou et al. (2016) have been adopted for the Fixed Failure-rate Ratio Test (FFRT).

- In the MATLAB implementation, no additional toolboxes are needed for any of the LAMBDA and/or Ps-LAMBDA functionalities.

This new LAMBDA 4.0 toolbox merges LAMBDA 3.0 and Ps-LAMBDA 1.0 capabilities, where many Ps-LAMBDA functionalities are enhanced by the algorithms' improvements of the new version, in particular for what concerns numerical simulations. These ones are now available for several estimators and different classes, thus providing an approximation for their success and/or failure rate.

An overview of all main LAMBDA 4.0 functionalities will be given later in Section 3.

1.2. Disclaimer

This new LAMBDA 4.0 implementation, as also older versions, is mainly intended for research purposes. Hence, even though the LAMBDA method itself is improved in terms of computational efficiency, the code could still be optimized for specific problems. In fact, given the large applicability of the LAMBDA method to different mixed-integer problems, more tailored implementations might be possible and this should be taken into account when making a performance comparison against other methods.

Lastly, in addition to the two LAMBDA and Ps-LAMBDA main scripts, we provide the users with open source access to several toolbox functionalities that can also be used as standalone functions. This further supports the integrability of LAMBDA 4.0 routines in current or future software. We refer the reader to LAMBDA 4.0 User Manual for further information and examples.

1.3. Outline

This documentation is organized as follows:

- In Section 2, an overview of the LAMBDA method is provided, introducing mixed-integer models, discussing the 'ambiguity decorrelation' step and different classes of estimators.

- In Section 3, a summary description of the LAMBDA 4.0 toolbox is given, where we focus on all functionalities available for both LAMBDA (for *estimation* purposes) and Ps-LAMBDA (for *evaluation* purposes).

- In Section 4, some practical applications of LAMBDA 4.0 are discussed, including both GNSS and non-GNSS problems. Hence, some illustrative examples are given for what concerns *estimation* and *evaluation*.

- In Section 5, we summarize the Terms and Conditions relative to the proper use of this new LAMBDA 4.0 toolbox.

2. The LAMBDA method

The LAMBDA method (Teunissen, 1995) is an efficient approach to ambiguity resolution, important step in the estimation of parameters in mixed-integer models. Following a 3-step orthogonal decomposition of the problem, as introduced in Teunissen (1993), LAMBDA makes use of an admissible transformation, or Z-transformation, in order to decorrelate the ambiguities. This decorrelation process increases the success rate for several estimators, while improving the computational efficiency of many others. Three different classes of estimators are defined (Teunissen, 2003c), and will be described later in this section.

2.1. Parameter estimation in mixed-integer models

We start with a linear(ized) mixed-integer model, given $\boldsymbol{y} \in \mathbb{R}^m$ as vector of observables, and $Q_{yy} \in \mathbb{R}^{m\times m}$ its variance-covariance (vc-)matrix, such that

$$\mathrm{E}\{\boldsymbol{y}\} = A\boldsymbol{a} + B\boldsymbol{b}, \qquad \mathrm{D}\{\boldsymbol{y}\} = Q_{yy} \tag{1}$$

where $\mathrm{E}\{\cdot\}$ and $\mathrm{D}\{\cdot\}$ are the expectation and dispersion operators, respectively. The vectors $\boldsymbol{a} \in \mathbb{Z}^n$ and $\boldsymbol{b} \in \mathbb{R}^p$ refer respectively to integer ambiguities and real-valued parameters, while the full-rank design matrix is given by $[A, B] \in \mathbb{R}^{m\times(n+p)}$.

In general a three-step procedure is employed to solve such mixed-integer models, based on a weighted least-squares criterion, where we consider the quadratic objective function

$$\|\boldsymbol{y} - A\boldsymbol{a} - B\boldsymbol{b}\|^2_{Q_{yy}} = \|\hat{\boldsymbol{e}}\|^2_{Q_{yy}} + \|\hat{\boldsymbol{a}} - \boldsymbol{a}\|^2_{Q_{\hat{a}\hat{a}}} + \left\|\hat{\boldsymbol{b}}(\boldsymbol{a}) - \boldsymbol{b}\right\|^2_{Q_{\hat{b}(a)\hat{b}(a)}} \tag{2}$$

where $\|\mathbf{x}\|^2_{\mathrm{Q}} = \mathbf{x}^{\mathbf{T}}\mathrm{Q}^{-1}\mathbf{x}$. By this orthogonal decomposition (Teunissen, 1993), we compute

Step 1 | Float solution

In this first step, the integer constraints on the ambiguities are neglected, assuming $\boldsymbol{a} \in \mathbb{R}^n$ and $\boldsymbol{b} \in \mathbb{R}^p$, and a standard least-squares solution is initially found as

$$\begin{pmatrix}\hat{\boldsymbol{a}}\\ \hat{\boldsymbol{b}}\end{pmatrix}, \qquad \begin{bmatrix} Q_{\hat{a}\hat{a}} & Q^T_{\hat{b}\hat{a}} \\ Q_{\hat{b}\hat{a}} & Q_{\hat{b}\hat{b}} \end{bmatrix} \tag{3}$$

where $\hat{\boldsymbol{e}} = \boldsymbol{y} - A\hat{\boldsymbol{a}} - B\hat{\boldsymbol{b}}$ is the least-squares residual vector, meanwhile $Q_{\hat{a}\hat{a}} \in \mathbb{R}^{n\times n}$ and $Q_{\hat{b}\hat{b}} \in \mathbb{R}^{p\times p}$ are the vc-matrices of the float-valued ambiguity and real-valued parameters' estimator, respectively. Lastly, $Q_{\hat{b}\hat{a}} \in \mathbb{R}^{p\times n}$ refers to their covariance matrix.

Step 2 | Ambiguity resolution

A second step follows, where starting with the ambiguity *float* estimator $\hat{\boldsymbol{a}} \in \mathbb{R}^n$, and its vc-matrix $Q_{\hat{a}\hat{a}}$, we can obtain a *fixed* ambiguity estimator as

$$\check{\boldsymbol{a}} = \mathcal{I}_{\boldsymbol{a}}(\hat{\boldsymbol{a}}) \tag{4}$$

given a certain mapping function $\mathcal{I}_{\boldsymbol{a}}$ that is defined for a selected estimator. Three different classes of estimators are introduced in Teunissen (2003c), as summarized in Section 2.3.

Step 3 | Fixed solution

In the third step, the float solution $\hat{\boldsymbol{b}}$ of all remaining real-valued parameters is then conditionally updated using the fixed ambiguities $\check{\boldsymbol{a}}$, such that

$$\check{\boldsymbol{b}} = \hat{\boldsymbol{b}}(\check{\boldsymbol{a}}) \overset{\text{def}}{=} \hat{\boldsymbol{b}} - Q_{\hat{b}\hat{a}} Q_{\hat{a}\hat{a}}^{-1}(\hat{\boldsymbol{a}} - \check{\boldsymbol{a}}) \tag{5}$$

with its vc-matrix given – in the normally distributed case – as

$$Q_{\check{b}\check{b}} = Q_{\hat{b}\hat{b}} - Q_{\hat{b}\hat{a}} Q_{\hat{a}\hat{a}}^{-1} Q_{\hat{a}\hat{b}} + Q_{\hat{b}\hat{a}} Q_{\hat{a}\hat{a}}^{-1} Q_{\check{a}\check{a}} Q_{\hat{a}\hat{a}}^{-1} Q_{\hat{a}\hat{b}} \tag{6}$$

where for $Q_{\check{a}\check{a}} \cong 0$, i.e. the fixed ambiguity is assumed to be deterministic, we obtain

$$Q_{\check{b}\check{b}} \cong Q_{\hat{b}\hat{b}} - Q_{\hat{b}\hat{a}} Q_{\hat{a}\hat{a}}^{-1} Q_{\hat{a}\hat{b}} \tag{7}$$

2.2. Ambiguity decorrelation

The ambiguity resolution step, shown in Eq.(4), refers still to the original parametrization of ambiguities. However, those ones are generally highly correlated and in some cases their float estimator has a poor precision. By reparametrizing the ambiguities, the precision of ambiguity components can be improved and their correlation can be largely reduced.

This ambiguity decorrelation process is also referred to as "Z-transformation", where the new set of float ambiguities $\hat{\boldsymbol{z}} \in \mathbb{R}^n$ is obtained as follows

$$\hat{\boldsymbol{z}} = Z^T \hat{\boldsymbol{a}} \tag{8}$$

with $Z \in \mathbb{Z}^{n \times n}$ being a unimodular matrix, and a reparametrized float solution is given by

$$\begin{pmatrix} \hat{\boldsymbol{z}} \\ \hat{\boldsymbol{b}} \end{pmatrix}, \quad \begin{bmatrix} Q_{\hat{z}\hat{z}} & Q_{\hat{b}\hat{z}}^T \\ Q_{\hat{b}\hat{z}} & Q_{\hat{b}\hat{b}} \end{bmatrix} \tag{9}$$

similarly as previously shown in Eq.(3).

The transformed vc-matrices are now defined as

$$Q_{\hat{z}\hat{z}} = Z^T Q_{\hat{a}\hat{a}} Z, \qquad Q_{\hat{b}\hat{z}} = Q_{\hat{b}\hat{a}} Z \tag{10}$$

and an ambiguity fixed solution, following this decorrelation, is obtained with

$$\check{\boldsymbol{z}} = \mathcal{I}_{\boldsymbol{z}}(\hat{\boldsymbol{z}}) \tag{11}$$

where $\mathcal{I}_{\boldsymbol{z}}$ is a certain mapping function. However, for several estimators we have

$$\check{\boldsymbol{z}} = \mathcal{I}_{\boldsymbol{z}}(Z^T \hat{\boldsymbol{a}}) \neq Z^T \mathcal{I}_{\boldsymbol{a}}(\hat{\boldsymbol{a}}) = Z^T \check{\boldsymbol{a}} \tag{12}$$

and the fixed solution might be dependent upon the parametrization. For a few estimators, the relation $\mathcal{I}_{\boldsymbol{z}}(Z^T \hat{\boldsymbol{a}}) = Z^T \mathcal{I}_{\boldsymbol{a}}(\hat{\boldsymbol{a}})$ holds, but the decorrelation remains an important way to enhance their computational performances.

After an ambiguity fixed solution is obtained, the $\boldsymbol{b}$-parameters are conditionally updated following Step 3 from Section 2.1. However, the back-transformation is not per se required since we can directly work with the new $\boldsymbol{z}$-parametrization, i.e.

$$\check{\boldsymbol{b}} = \hat{\boldsymbol{b}}(\check{\boldsymbol{z}}) \overset{\text{def}}{=} \hat{\boldsymbol{b}} - Q_{\hat{b}\hat{z}} Q_{\hat{z}\hat{z}}^{-1} (\hat{\boldsymbol{z}} - \check{\boldsymbol{z}}) \tag{13}$$

thus saving unnecessary additional computations, and assuming $Q_{\check{z}\check{z}} \cong 0$, we have

$$Q_{\check{b}\check{b}} \cong Q_{\hat{b}\hat{b}} - Q_{\hat{b}\hat{z}} Q_{\hat{z}\hat{z}}^{-1} Q_{\hat{z}\hat{b}} \tag{14}$$

with the transformed matrices given in Eq.(10).

2.3. Different classes of estimators

Three classes of estimators are introduced in Teunissen (2003c):

I. **Integer estimators**, i.e. I-class (Section 2.3.1)
II. **Integer Aperture estimators**, i.e. IA-class (Section 2.3.2)
III. **Integer Equivariant estimators**, i.e. IE-class (Section 2.3.3)

where a set relationship exists among such classes, as we have I $\subset$ IA $\subset$ IE. Each class is characterized by different properties, based on the family of maps $\mathcal{I}$ being defined. A small illustration with an example of a single member per class is given in Figure 2, where this aforementioned set-relationship is also visible.

In LAMBDA 4.0 toolbox, only a few members of each different class are implemented, and we refer the reader to Verhagen (2005) for more mathematical details on several other existing estimators that are currently available in the body of knowledge.

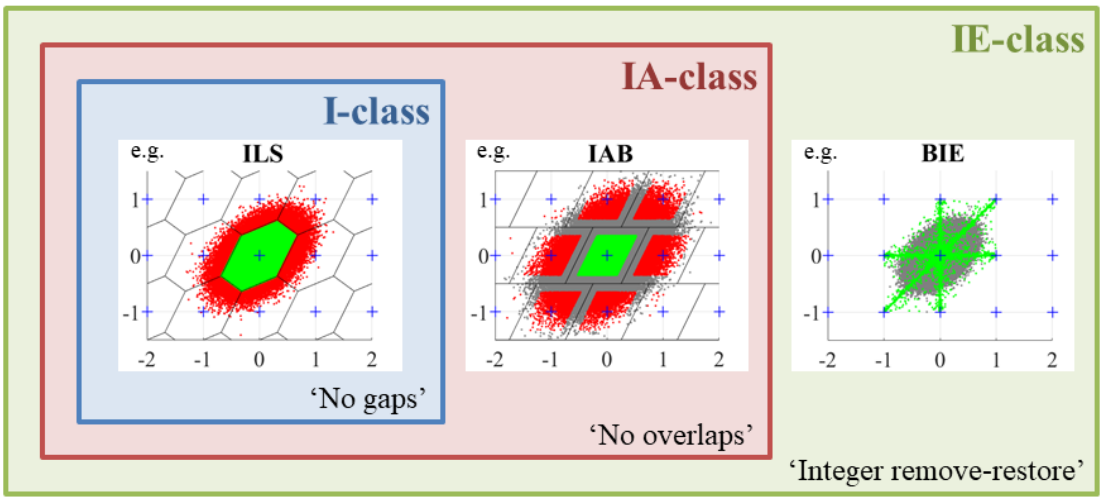


Figure 2: A graphical overview of the three main classes and their set relationship

2.3.1. Integer (I) estimator class

The I-class has been introduced in Teunissen (1999a) and it is the most restrictive class. In fact, an integer solution $\breve{\boldsymbol{a}}_{\boldsymbol{I}} \in \mathbb{Z}^n$ is required, and I-estimators are given by many-to-one mapping functions $\boldsymbol{\mathcal{I}}_{\boldsymbol{I}}: \mathbb{R}^n \to \mathbb{Z}^n$, defining so-called pull-in regions. These regions leave no gaps, have no overlaps and are equivariant to integer translations, such that

$$\boldsymbol{\mathcal{I}}_{\boldsymbol{I}}(\hat{\boldsymbol{a}} + \boldsymbol{z}) = \boldsymbol{\mathcal{I}}_{\boldsymbol{I}}(\hat{\boldsymbol{a}}) + \boldsymbol{z}, \qquad \forall \boldsymbol{z} \in \mathbb{Z}^n \tag{15}$$

where the integer remove-restore property holds for any integer shift to the float input.

An expression for these **I-estimators** can be given as

$$\breve{\boldsymbol{a}}_{\boldsymbol{I}} \overset{\text{def}}{=} \boldsymbol{\mathcal{I}}_{\boldsymbol{I}}(\hat{\boldsymbol{a}}) = \sum_{\boldsymbol{z} \in \mathbb{Z}^n} \boldsymbol{z}\, s_z(\hat{\boldsymbol{a}}) \equiv \hat{\boldsymbol{a}} + \sum_{\boldsymbol{z} \in \mathbb{Z}^n} (\boldsymbol{z} - \hat{\boldsymbol{a}})\, s_z(\hat{\boldsymbol{a}}) \tag{16}$$

where $s_z(\boldsymbol{x})$ is an indicator function, equal to 1 if $\boldsymbol{x} \in S_z$ or equal to 0 otherwise. A pull-in region S_z concerns all vectors in $\mathbb{R}^n$ that are pulled to the same integer $\boldsymbol{z}$-vector, so that

$$S_z = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{z} = \boldsymbol{\mathcal{I}}_{\boldsymbol{I}}(\boldsymbol{x})\}, \qquad \boldsymbol{z} \in \mathbb{Z}^n \tag{17}$$

with the volume of each S_z being equal to one.

Three conditions shall be satisfied:

$$\bigcup_{z\in\mathbb{Z}^n} S_z = \mathbb{R}^n \tag{18}$$

$$\mathrm{Int}(S_{z_1}) \cap \mathrm{Int}(S_{z_2}) = \emptyset, \qquad \forall \boldsymbol{z_1}, \boldsymbol{z_2} \in \mathbb{Z}^n, \qquad \boldsymbol{z_1} \neq \boldsymbol{z_2} \tag{19}$$

$$S_z = S_0 + \boldsymbol{z}, \qquad \forall \boldsymbol{z} \in \mathbb{Z}^n \tag{20}$$

being respectively "no gaps", "no overlaps" and "integer remove-restore". All these three properties hold for any arbitrary integer estimator $\breve{\boldsymbol{a}}_{\boldsymbol{I}} \in \mathbb{Z}^n$, where the first condition makes sure that an integer value is *always* returned given a certain float $\hat{\boldsymbol{a}} \in \mathbb{R}^n$.

2.3.2. Integer Aperture (IA) estimator class

The IA-class has been introduced in Teunissen (2003b) and is larger than the I-class. In fact, it is defined by dropping the 'no gaps' condition to the pull-in regions. An integer solution $\breve{\boldsymbol{a}}_{\mathbf{IA}} \in \mathbb{Z}^n$ follows only if $\hat{\boldsymbol{a}}$ belongs to the IA pull-in region, otherwise a float solution will be returned. Such IA-estimators reduce the probability of wrongly fixing the ambiguities, and for $\breve{\boldsymbol{a}}_{\mathbf{IA}} = \hat{\boldsymbol{a}}$ no improvements are expected since $\hat{\boldsymbol{b}}(\hat{\boldsymbol{a}}) \equiv \hat{\boldsymbol{b}}$.

An expression for these IA-estimators can be given as follows

$$\breve{\boldsymbol{a}}_{\boldsymbol{IA}} \overset{\text{def}}{=} \boldsymbol{\mathcal{I}}_{\boldsymbol{IA}}(\hat{\boldsymbol{a}}) = \hat{\boldsymbol{a}} + \sum_{\boldsymbol{z}\in\mathbb{Z}^n} (\boldsymbol{z} - \hat{\boldsymbol{a}})\, v_z(\hat{\boldsymbol{a}}) \tag{21}$$

where $v_z(\boldsymbol{x})$ is an indicator function of $\Omega_z = \Omega \cap S_z$, where $\Omega \subset \mathbb{R}^n$ is here translational equivariant. By comparing this expression with the one given in Eq.(16) for I-estimators, we observe a difference in the binary weights $s_z(\boldsymbol{x})$ and $v_z(\boldsymbol{x})$, respectively adopted for the I- and IA-class. In fact, in this second case, we have

$$\sum_{\boldsymbol{z}\in\mathbb{Z}^n} v_z(\boldsymbol{x}) \leq 1 \tag{22}$$

and this means that the outcome is not always an integer value for IA-estimators.

2.3.3. Integer Equivariant (IE) estimator class

The IE-class has been introduced in Teunissen (2002) and it is the largest class, so both the I-estimators and IA-estimators are special cases of IE-estimators. In this larger class, the integer remove-restore principle still applies, and the fixed ambiguity solution can be real valued, i.e. $\breve{\boldsymbol{a}}_{\boldsymbol{IE}} \in \mathbb{R}^n$.

Many different IE-estimators can be designed, and for some specific choices, see (ibid), we can write

$$\breve{\boldsymbol{a}}_{\mathbf{IE}} \stackrel{\text{def}}{=} \boldsymbol{\mathcal{I}}_{\mathbf{IE}}(\hat{\boldsymbol{a}}) = \sum_{\boldsymbol{z}\in\mathbb{Z}^n} \boldsymbol{z}\, \omega_z(\hat{\boldsymbol{a}}) = \hat{\boldsymbol{a}} + \sum_{\boldsymbol{z}\in\mathbb{Z}^n} (\boldsymbol{z}-\hat{\boldsymbol{a}})\, \omega_z(\hat{\boldsymbol{a}}) \tag{23}$$

where $\omega_z(\boldsymbol{x})$ is no more a binary indicator, but a weighting function that can vary between 0 and 1. In this particular case, the output solution is indeed a weighted sum of all integer candidates over the entire set $\mathbb{Z}^n$, where $\sum_{\boldsymbol{z}\in\mathbb{Z}^n} \omega_z(\hat{\boldsymbol{a}}) = 1$.

Given that it is not very practical to work with an infinite n-dimensional set of integers, the latter can generally be approximated by a finite set $\Omega_\alpha \subset \mathbb{Z}^n$, where $\alpha \ll 1$ refers to the significance level (Teunissen, 2005b). Once the size of the integer set has been defined, all integer vectors can be collected, e.g. making use of LAMBDA 4.0, and an approximated solution can be computed by making use of proper weights (Teunissen, 2020a).

We refer the reader to Verhagen (2005) for additional mathematical details on these three classes of estimators, meanwhile we continue now by introducing LAMBDA 4.0 toolbox, which currently implements a few members from each class.

3. LAMBDA 4.0 toolbox

The LAMBDA 4.0 toolbox mainly focuses on "Step 2" described in Section 2.1. It merges all main LAMBDA and Ps-LAMBDA functionalities, and it provides now more advanced algorithms and additional (classes of) estimators. A flowchart illustration for this toolbox is given in Figure 3, where inputs and outputs are illustrated, along with the two main scripts relative to LAMBDA (estimation) and Ps-LAMBDA (evaluation) functionalities.

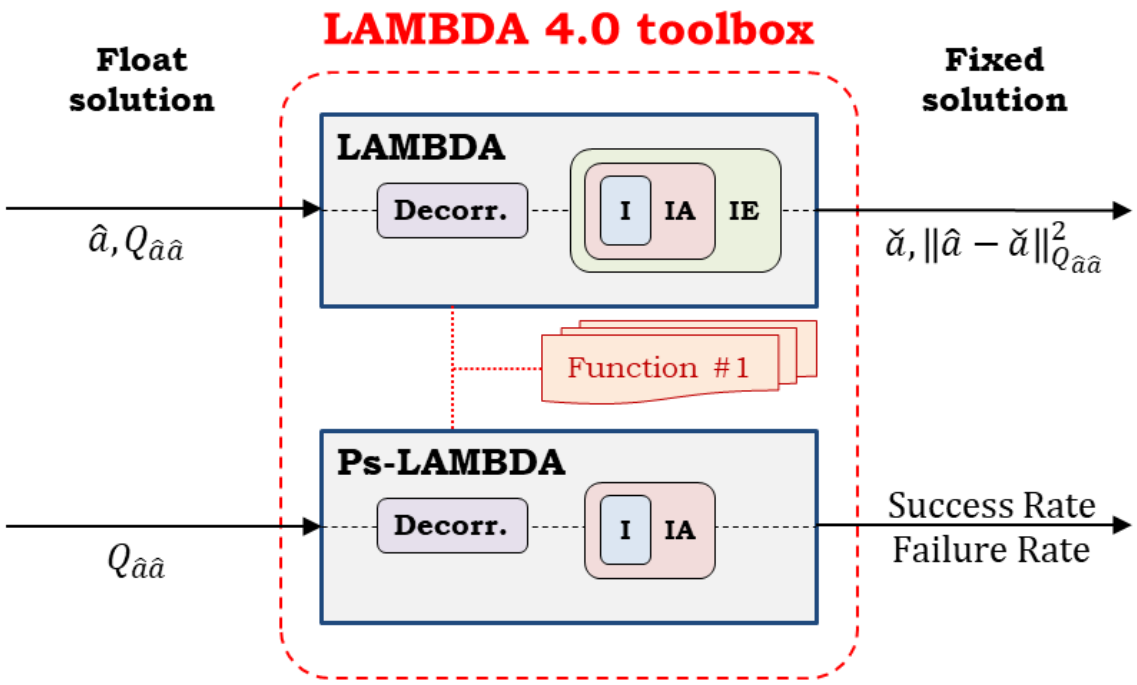


Figure 3: Flowchart for the LAMBDA 4.0 toolbox, including inputs and outputs, along with the two main scripts: LAMBDA and Ps-LAMBDA.

It should be noted that all toolbox functionalities are 'open source' and can be used by the LAMBDA users for more advanced computations. Hence, they can be seen as standalone functions in this new toolbox, which might more easily be integrated in external software programs.

Before describing LAMBDA and Ps-LAMBDA functionalities, including estimators and methods been implemented, we briefly discuss the Z-transformation adopted to decorrelate the ambiguity parameters. Note that in Ps-LAMBDA, the decorrelation is optional, and a statistical evaluation using the original parametrization is therefore also possible.

3.1. Admissible Z-transformation

In LAMBDA, an admissible transformation is obtained by means of a unimodular matrix $Z \in \mathbb{Z}^n$, which re-parametrizes the ambiguities into $\boldsymbol{z} = Z^T \boldsymbol{a}$. Due to the integerness of $\boldsymbol{a}$, it is fundamental that a back-transformation from $\boldsymbol{z} \in \mathbb{Z}^n$ also leads to integer ambiguities, such that $\boldsymbol{a} = Z^{-T}\boldsymbol{z}$ with $\boldsymbol{a} \in \mathbb{Z}^n$. This so-called Z-matrix is computed here starting with a $\mathrm{L^TDL}$-decomposition (Section 3.1.1), then alternating permuting operations with Integer Gauss Transformations (IGTs). We refer the reader to de Jonge and Tiberius (1996) for more mathematical details on this ambiguity decorrelation process.

3.1.1. $\mathbf{L^TDL}$-decomposition

Given $Q_{\hat{a}\hat{a}} \in \mathbb{R}^n$ as vc-matrix of the float ambiguities, we have

$$Q_{\hat{a}\hat{a}} = L_{\hat{a}}^T D_{\hat{a}} L_{\hat{a}}, \qquad L_{\hat{a}} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & 1 \end{bmatrix}, \qquad D_{\hat{a}} = \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix} \tag{24}$$

where $L_{\hat{a}} \in \mathbb{R}^n$ and $D_{\hat{a}} \in \mathbb{R}^n$ are respectively a lower unit-triangular and diagonal matrix, with the latter containing conditional variances $\sigma^2_{\hat{a}_{i|i+1,\ldots,n}} = d_i$. Note that the last-to-first convention has been adopted in previous implementations, and this justifies the consistent use here of a $\mathrm{L^TDL}$ decomposition rather than a $\mathrm{LDL^T}$ decomposition.

3.1.2. Integer Gauss Transformations (IGTs)

The Integer Gauss Transformations (IGT), see Teunissen (1994), is a particular type of Z-transformation, which represents an admissible ambiguity transformation given by

$$Z' = \begin{bmatrix} I_{i-1} & & \\ & \mathrm{Z_{IGT}} & \\ & & I_{n-i-1} \end{bmatrix}, \qquad \mathrm{Z_{IGT}} = \begin{bmatrix} 1 & 0 \\ \alpha_{\mathrm{IGT}} & 1 \end{bmatrix} \tag{25}$$

with $\alpha_{\mathrm{IGT}} \in \mathbb{Z}$ and $\det(Z_{\mathrm{IGT}}) = 1$, such that

$$L' = LZ' = \begin{bmatrix} L'_{11} & 0 & 0 \\ L'_{21} & L'_{22} & 0 \\ L'_{31} & L'_{32} & L'_{33} \end{bmatrix}, \qquad L'_{22} = \begin{bmatrix} 1 & 0 \\ l_{i+1,i} + \alpha_{\mathrm{IGT}} & 1 \end{bmatrix} \tag{26}$$

so the conditional variances are not directly affected by such a Z-transformation. Only the blocks in red color are modified, and unconditional variances will also change. It is trivial to observe that in $L'_{32} = L_{32}Z_{\mathrm{IGT}}$ we are adding the second column of L_{32}, being scaled by the α_{IGT} coefficient, to the first one.

A full decorrelation in L'_{22} could be achieved by choosing $\alpha_{\text{IGT}} = -l_{i+1,i}$, however this is generally not possible given the integer condition on α_{IGT}, such that Z_{IGT} and Z' remain still admissible transformations. An approximation is then adopted, i.e. $\alpha_{\text{IGT}} = -\lceil l_{i+1,i} \rfloor$, with $\lceil \cdot \rfloor$ referring to a nearest integer operator, thus leaving the new off-diagonal term $l'_{i+1,i}$ in L'_{22} bounded within 0.5 in modulo.

Such procedure could be re-iterated for all off-diagonal entries of L, see Teunissen (1994) and de Jonge and Tiberius (1996), such that for $Z'(i,j) = \alpha_{\text{IGT}}$, with $i > j$, we have

$$\begin{aligned} l'_{i,j} &= l_{i,j} + \alpha_{\text{IGT}} \\ l'_{k,j} &= l_{k,j} + \alpha_{\text{IGT}} l_{k,i}, \qquad k = i+1, \dots, n \end{aligned} \tag{27}$$

and we could proceed iterating by columns or rows. In LAMBDA 4.0 toolbox, a dedicated standalone functions for computing such IGTs is provided, based on iterations over rows from top to bottom, i.e. $i = 2, \dots, n$, and columns from left to right, i.e. $j = 1, \dots, i-1$.

3.1.3. Permuting operations

The permuting operations are fundamental for the ambiguity re-parametrization, as they allow modifying the spectrum of conditional variances. Permutations in LAMBDA refer only to adjacent swapping, where a local transformation is adopted here based on a 2-by-2 permutation matrix $Z_{\text{PERM}} \in \mathbb{Z}^{2\times 2}$. Other non-local transformations are possible, while not discussed here, but a key aspect concerns the efficiency of such operations. In fact, after a permutation, the $\text{L}^{\text{T}}\text{DL}$-decomposition needs to be recomputed, but this is not necessary for some particular trivial permutations (as adjacent swapping). We refer to the Section 3.6 in de Jonge and Tiberius (1996) for more mathematical details.

In LAMBDA 4.0 toolbox, a so-called "lazy transformation" is employed, as introduced in Section 3.1.3 of MLAMBDA by Chang et al. (2005). As permutations modify also entries of the L-matrix, the previously bounded values might be no more bounded and therefore often the IGTs are wasted due to a successive permutation. Such a "lazy" strategy consists in avoiding certain IGTs and ultimately leading to a flattened spectrum of conditional variances, but without fully bounding the L matrix. This is justified by the fact that the bootstrapped success rate is only dependent on the D matrix, nevertheless in LAMBDA 4.0 we still compute a full decorrelation by means of IGTs at the end of the iterative process, so following what been discussed in Section 3.1.2. Hence, differently from MLAMBDA, we always assure that $|l'_{i,j}| \leq 0.5, \ \forall i > j$.

3.2. LAMBDA functionalities

The overview of LAMBDA functionalities is given in Figure 4.

LAMBDA	Class	Description	Options
estimatorIR	I	**Integer Rounding**	-
estimatorIB	I	**Integer Bootstrapping**	-
estimatorILS	I	**Integer Least Squares** (search-and-shrink)	• # of candidates
estimatorILS_enum	I	**Integer Least Squares** (enumeration)	• # of candidates
estimatorPAR	I	**Partial Ambiguity Resolution** (with ILS estimator)	• # of candidates • Minimum SR
estimatorVIB	I	**Vectorial Integer Bootstrapping** (with IR or ILS estimator)	• Use IR or ILS • Partitioning of blocks
estimatorIA_RT	IA	**Integer Aperture w/ Ratio Test**	• Aperture parameter μ_{RT} • Maximum FR
estimatorIAB	IA	**Integer Aperture Bootstrapping**	• Aperture parameter β_{IAB} • Maximum FR
estimatorBIE	IE	**Best Integer Equivariant** (finite approximation)	• Significance level α_{BIE}

Figure 4: Overview of all LAMBDA functionalities available in this new v4.0 toolbox.

The LAMBDA 4.0 toolbox implements several methods, and includes all the three classes of estimators (see Section 2.3). A few members from each class are implemented, while for some methods a few different configurations are also possible.

In the next sections we will briefly review the implemented methods, such as

- Integer Rounding | Section 3.2.1
- Integer Bootstrapping | Section 3.2.2
- Integer Least Squares | Section 3.2.3
  - by Enumeration | Section 3.2.3.1
  - by Search-and-Shrink | Section 3.2.3.2
- Partial Ambiguity Resolution | Section 3.2.4
- Vectorial Integer Bootstrapping | Section 3.2.5
- Integer Aperture with Ratio Test | Section 3.2.6
- Integer Aperture Bootstrapping | Section 3.2.7
- Best Integer Equivariant | Section 3.2.8

3.2.1. **I-class |** Integer Rounding (IR)

In the class of Integer estimators, the Integer Rounding (IR) is perhaps the simplest way to obtain an integer vector from the real-valued float solution. This IR estimator reads as

$$\breve{\boldsymbol{a}}_{\mathbf{IR}} = \begin{pmatrix} \lceil \hat{a}_1 \rfloor \\ \vdots \\ \lceil \hat{a}_n \rfloor \end{pmatrix} \tag{28}$$

where $\lceil \cdot \rfloor$ stands for scalar rounding to the nearest integer.

In general, the IR pull-in regions are n-dimensional unit cubes centered at each grid point, where we can write

$$\mathcal{P}_{z,IR} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \mid \left| \boldsymbol{c}_i^T (\boldsymbol{x} - \boldsymbol{z}) \right| \leq \frac{1}{2}, i = 1, \ldots, n \right\}, \qquad \boldsymbol{z} \in \mathbb{Z}^n \tag{29}$$

with $\boldsymbol{c_i}$ being the canonical unit vector with '1' as its i-th entry, and '0' otherwise.

PARAMETRIZATION:

The IR estimator is dependent upon the ambiguity parametrization, so $\breve{\mathbf{z}}_{\mathbf{IR}} \neq Z^T \breve{\boldsymbol{a}}_{\mathbf{IR}}$, where $\hat{\boldsymbol{z}} = Z^T \hat{\boldsymbol{a}}$, and this affects the probability of correct fixing, so $P(\breve{\boldsymbol{a}}_{\mathbf{IR}} = \boldsymbol{a}) \neq P(\breve{\mathbf{z}}_{\mathbf{IR}} = \boldsymbol{z})$.

An example of 2D pull-in region is given in Figure 5, where all float solutions in the pull-in region of the true integer vector are marked in green color, otherwise in red color.

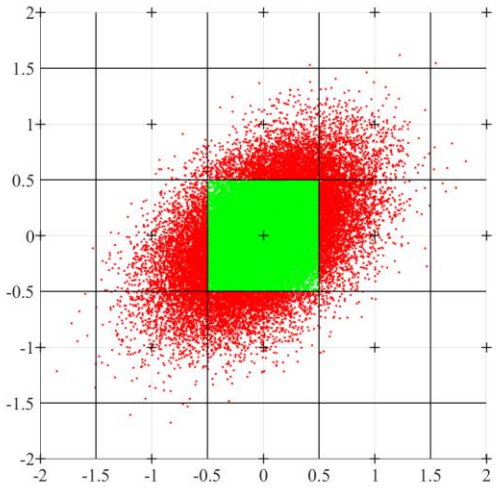


Figure 5: Illustration of the IR pull-in regions, along with 40'000 samples being generated based on $\hat{\boldsymbol{a}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}_{\hat{a}\hat{a}})$. In green, we show the float ambiguity samples that are pulled to the true integer vector, i.e. $\boldsymbol{a} = \mathbf{0}$.

3.2.2. **I-class |** Integer Bootstrapping (IB)

The Integer Bootstrapping (IB) is somehow based on IR, whereas it takes into account the correlation between ambiguities. This IB estimator follows from a sequential least-squares adjustment and can be computed as follow

$$\check{a}_{n,\mathrm{IB}} = \lceil \hat{a}_n \rfloor$$
$$\check{a}_{j,\mathrm{IB}} = \lceil \hat{a}_{j|J} \rfloor = \left\lceil \hat{a}_j - \sum_{i=j+1}^{n} l_{ij}\left(\hat{a}_{i|I} - \check{a}_{i,\mathrm{IB}}\right) \right\rfloor, \qquad \forall j = n-1, \dots, 1 \tag{30}$$

for $\lceil \cdot \rfloor$ as rounding operator, and $\hat{a}_{j|J}$ refers to the j-th (float) ambiguity obtained through a conditioning on previous $J = \{j+1, \dots, n\}$ sequentially rounded ambiguities. Moreover

$$l_{ij} = \frac{\sigma_{\hat{a}_j \hat{a}_{i|I}}}{\sigma^2_{\hat{a}_{i|I}}} \tag{31}$$

where $\sigma^2_{\hat{a}_{i|I}}$ are conditional variances, while $\sigma_{\hat{a}_j \hat{a}_{i|I}}$ is the covariance between $\hat{a}_j$ and $\hat{a}_{i|I}$.

In general, the pull-in regions are n-dimensional parallelotopes (or simply n-parallelotope) centered at each grid point, where we can write

$$\mathcal{P}_{z,\mathrm{IB}} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \mid \left| \boldsymbol{c_i^T} L^{-T}(\boldsymbol{x} - \boldsymbol{z}) \right| \leq \frac{1}{2}, i = 1, \dots, n \right\}, \qquad \boldsymbol{z} \in \mathbb{Z}^n \tag{32}$$

with $\boldsymbol{c_i}$ being the canonical unit vector with '1' as its i-th entry, '0' otherwise. The matrix $L \in \mathbb{R}^n$ is again the lower unit-triangular matrix obtained in the $\mathrm{L^TDL}$-decomposition, see Eq.(24) in Section 3.1.1, whose entries are given by the l_{ij} terms in Eq.(30).

PARAMETRIZATION:

The IB estimator is dependent upon the ambiguity parametrization, so $\check{\boldsymbol{z}}_{\mathbf{IB}} \neq Z^T \check{\boldsymbol{a}}_{\mathbf{IB}}$, where $\hat{\boldsymbol{z}} = Z^T \hat{\boldsymbol{a}}$, and this affects the probability of correct fixing, so $P(\check{\boldsymbol{a}}_{\mathbf{IB}} = \boldsymbol{a}) \neq P(\check{\boldsymbol{z}}_{\mathbf{IB}} = \boldsymbol{z})$.

Note that even a simple permutation of the original ambiguity components might lead to a different solution. Moreover, for this IB estimator, the exact success rate can analytically be computed based on the formulation later described in Section 3.3.1.

An example of 2D pull-in region is given in Figure 6, where parallelograms are shown for the IB estimator, similarly as illustrated for IR in Figure 5. The float solutions in green color are pulled to the *true* integer vector $\boldsymbol{a} = \boldsymbol{0}$, adopted for these numerical simulations.

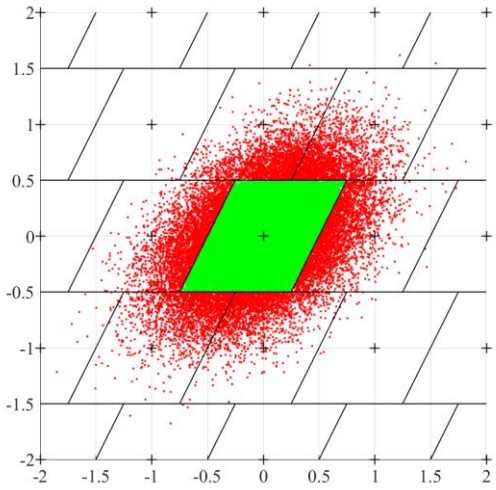


Figure 6: Illustration of the IB pull-in regions, along with 40'000 samples being generated based on $\hat{a} \sim \mathcal{N}(0, Q_{\hat{a}\hat{a}})$. In green, we show the float ambiguity samples that are pulled to the true integer vector, i.e. $a = 0$.

3.2.3. **I-class |** Integer Least Squares (ILS)

The Integer Least-Squares (ILS) estimator is optimal within the class of Integer estimators, see Teunissen (1999b), where the optimality refers to the fact that it maximizes the success rate (i.e. probability of correct fixing). The ILS estimator is given as

$$\breve{\boldsymbol{a}}_{\mathbf{ILS}} = \arg\min_{\boldsymbol{z}\in\mathbb{Z}^n} \|\hat{\boldsymbol{a}} - \boldsymbol{z}\|^2_{Q_{\hat{a}\hat{a}}} \tag{33}$$

with $\|\mathbf{x}\|^2_{\mathrm{Q}} \stackrel{\text{def}}{=} \mathbf{x}^{\mathbf{T}}\mathrm{Q}^{-1}\mathbf{x}$, being the vector norm weighted by matrix $Q \in \mathbb{R}^{n\times n}$.

In general, these ILS pull-in regions are n-dimensional polytopes (or simply n-polytopes) centered at each grid point, where we can write

$$\mathcal{P}_{z,\mathrm{ILS}} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \mid |w_u(\boldsymbol{x}-\boldsymbol{z})| \le \frac{1}{2}\|\boldsymbol{u}\|_{Q_{\hat{a}\hat{a}}}, \qquad \forall \boldsymbol{u} \in \mathbb{Z}^n \right\}, \qquad \boldsymbol{z} \in \mathbb{Z}^n \tag{34}$$

with

$$w_u(\boldsymbol{x}-\boldsymbol{z}) = \frac{\boldsymbol{u}^T Q^{-1}_{\hat{a}\hat{a}}(\boldsymbol{x}-\boldsymbol{z})}{\|\boldsymbol{u}\|_{Q_{\hat{a}\hat{a}}}} \tag{35}$$

being the orthogonal projection of $(\boldsymbol{x}-\boldsymbol{z})$ onto the direction $\boldsymbol{u}$. Hence, this ILS pull-in region is the intersection of banded subsets, i.e. centered in $\boldsymbol{z}$ and having width $\|\boldsymbol{u}\|_{Q_{\hat{a}\hat{a}}}$.

The ILS estimator is however invariant to Z-transformations, so differently from what seen for IR and IB estimators, and its success rate will not change. Still, the parametrization can affect the computational complexity during the search of suitable integer candidates.

PARAMETRIZATION:

The ILS estimator is invariant upon parametrization, so $\check{\boldsymbol{z}}_{\mathbf{ILS}} = Z^T \check{\boldsymbol{a}}_{\mathbf{ILS}}$, and therefore it has $P(\check{\boldsymbol{a}}_{\mathbf{ILS}} = \boldsymbol{a}) = P(\check{\boldsymbol{z}}_{\mathbf{ILS}} = \boldsymbol{z})$ for any $\hat{\boldsymbol{z}} = Z^T \hat{\boldsymbol{a}}$, given an admissible matrix $Z \in \mathbb{Z}^{n\times n}$.

An example of 2D pull-in region is given in Figure 7, with the ILS regions being hexagons that seem to better fit the distribution of the float ambiguity estimates. As stated, the probability of correctly fix is here optimal, i.e. the largest possible within the I-class.

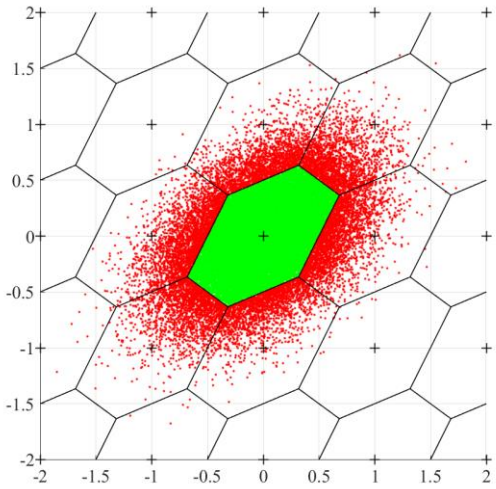


Figure 7: Illustration of the ILS pull-in regions, along with 40'000 samples being generated based on $\hat{\boldsymbol{a}} \sim \mathcal{N}(\mathbf{0}, Q_{\hat{a}\hat{a}})$. In green, we show the float ambiguity samples that are pulled to the true integer vector, i.e. $\boldsymbol{a} = \mathbf{0}$.

INTEGER SEARCH

Still, the Eq.(33) requires a search for the Integer Least-Squares solution, and efficiency of the search process might be affected by re-parametrizations. We consider the decorrelated ambiguity given in Section 2.2, and we can write

$$F(\boldsymbol{z}) = (\hat{\boldsymbol{z}} - \boldsymbol{z})^T Q_{\hat{z}\hat{z}}^{-1} (\hat{\boldsymbol{z}} - \boldsymbol{z}) \leq \chi^2, \qquad \boldsymbol{z} \in \mathbb{Z}^n \tag{36}$$

where $\chi > 0$ determines the size of the search ellipsoid, which can be initially defined by the user based on a previous guess solution, e.g. using IB.

Therefore, we have

$$F(\check{\mathbf{z}}_{\mathrm{ILS}}) \leq F(\boldsymbol{u}), \qquad \forall \boldsymbol{u} \in \mathbb{Z}^n \tag{37}$$

with $\check{\mathbf{z}}_{\mathrm{ILS}} \in \mathbb{Z}^n$ being the (optimal) solution, i.e. the grid point inside the hyper-ellipsoid that gives the minimum value of the function $F(\boldsymbol{z})$.

The Eq.(36) can be re-written making use of $Q_{\hat{z}\hat{z}} = L_{\hat{z}}^T D_{\hat{z}} L_{\hat{z}}$ (Section 3.1.1), and by defining $\tilde{\boldsymbol{z}} = \boldsymbol{z} + L_{\hat{z}}^{-T}(\hat{\boldsymbol{z}} - \boldsymbol{z})$, we have

$$L_{\hat{z}}^T(\tilde{\boldsymbol{z}} - \boldsymbol{z}) = (\hat{\boldsymbol{z}} - \boldsymbol{z}) \tag{38}$$

and for $j \in [1, n]$ this leads to

$$\tilde{z}_j = \hat{z}_{j|J} = \hat{z}_j - \sum_{i=j+1}^{i=n} l_{ji}\left(\hat{z}_{i|I} - z_i\right) \tag{39}$$

At this point, we can write the hyper-ellipsoid as

$$(\tilde{\boldsymbol{z}} - \boldsymbol{z})^{\boldsymbol{T}} D_{\hat{z}}^{-1}(\tilde{\boldsymbol{z}} - \boldsymbol{z}) \leq \chi^2, \qquad \boldsymbol{z} \in \mathbb{Z}^n \tag{40}$$

or equivalently

$$\sum_{j=1}^{j=n} \frac{\left(\tilde{z}_j - z_j\right)^2}{\sigma_{\hat{z}_{j|J}}^2} \leq \chi^2, \qquad z_j \in \mathbb{Z} \tag{41}$$

that is satisfied by looking at each individual j-th bound, so having

$$\tilde{z}_j - \sigma_{\hat{z}_{j|J}} \sqrt{\chi^2 - \sum_{i=j+1}^{i=n} \frac{(\tilde{z}_i - z_i)^2}{\sigma_{\hat{z}_{i|I}}^2}} \leq z_j \leq \tilde{z}_j + \sigma_{\hat{z}_{j|J}} \sqrt{\chi^2 - \sum_{i=j+1}^{i=n} \frac{(\tilde{z}_i - z_i)^2}{\sigma_{\hat{z}_{i|I}}^2}} \tag{42}$$

where we start at $j = n$ with $(z_n - \hat{z}_n) \in \left[-\sigma_{\hat{z}_n}\chi, +\sigma_{\hat{z}_n}\chi\right]$.

An illustration of the 2D search ellipse before and after decorrelation is given in Figure 8, where the ILS pull-in regions are also shown. The blue dot refers to the float solution, which is the center of the search ellipse, whereas the green region corresponds to the ILS pull-in region for that float solution. As visible, multiple integer candidates are found in this search ellipse (blue line) that is centered on a certain float solution (blue dot), while and the integer vector that minimizes the objective function in Eq.(36) is found in the ILS pull-in region (green hexagon).

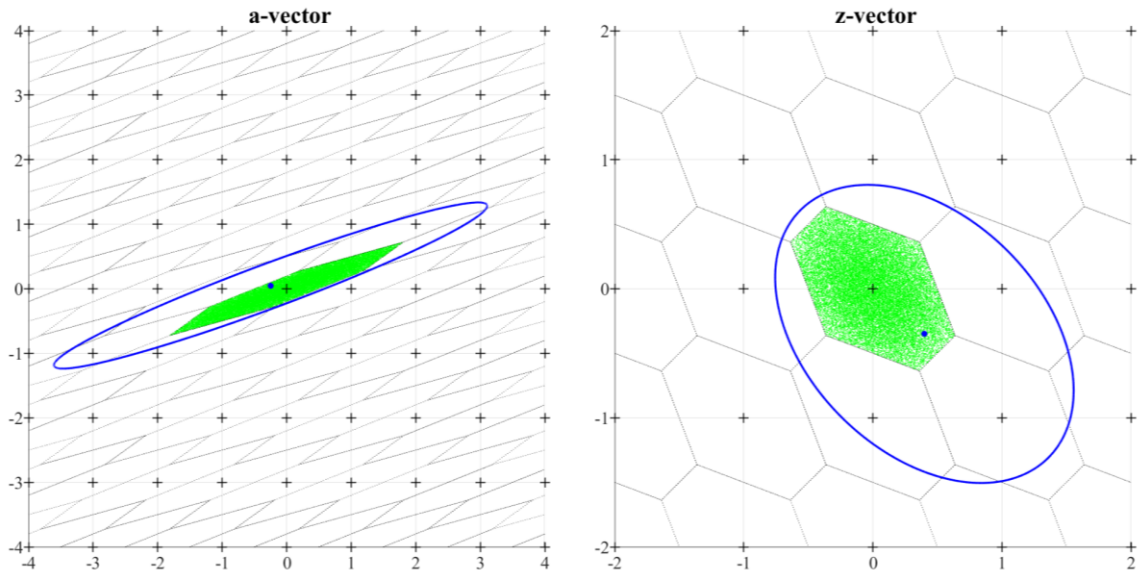


Figure 8: The 2D search ellipse is shown before (LEFT) and after (RIGHT) the decorrelation process, with an illustration of the float solution (blue dot) and ILS pull-in region (in green).

In the following sub-sections, we describe two search strategies, i.e.

- Enumeration (see Section 3.2.3.1)
- Search-and-shrink (see Section 3.2.3.2)

where the latter has been enhanced by a new algorithm introduced in Ghasemmehdi and Agrell (2011), thus reducing the number of conditioning operations during the search and improving the efficiency (i.e. computational time) by more than one order of magnitude.

3.2.3.1. ILS by Enumeration

The search by enumeration has been implemented in all older versions of LAMBDA, and it is extensively described in de Jonge and Tiberius (1996). As discussed, by choosing a χ^2 value, we define a search ellipsoid in $\mathbb{R}^n$ where our integer candidates could be found. We will distinguish three cases for defining χ^2, therefore

- If only one candidate is requested, we can directly use $\chi^2 = F(\check{\mathbf{z}}_{\mathbf{IB}})$, i.e. using the integer bootstrapped estimate.
- If at most $n+1$ candidates are requested, in each step of the conditional rounding procedure two integer candidates are taken: the nearest and second-nearest integer, where a conditional rounding is proceeded for both cases. In this way we can get $n+1$ values for $F(\boldsymbol{z})$. If $p \leq n+1$ candidates are requested, the values of $F(\boldsymbol{z})$ are ordered in ascending order and $\chi^2 = F\left(\boldsymbol{z}^{(p)}\right)$.

- If more than $n+1$ candidates are requested, the volume of the search ellipsoid can be used. In (ibid), a relation[1] between the volume and the number of integer grid points inside the search ellipsoid is given for large n dimensionalities.

Once the size of the search ellipsoid is set to χ^2, all the integer candidates in the search ellipsoid are collected by scanning the interval (from left to right) around the conditional estimates at each ambiguity level using Eq.(42). The intervals' bounds can be computed recursively during the sequential conditional adjustment.

3.2.3.2. ILS by Search-and-Shrink (NEW ALGORITHM)

The alternative search strategy is based on shrinking the search ellipsoid during the search and it was proposed in Teunissen (1993). Its old implementation in LAMBDA 3.0 was based on Chang et al. (2005), where the shrinking strategy was extended to the case with more than one optimal estimate being required. A new algorithm is nevertheless adopted here, as introduced in Ghasemmehdi and Agrell (2011).

- **OLD ALGORITHM (Verhagen et al., 2012)**

The value χ^2 can initially be set to infinity, since in the very first step we can consider a bootstrapped solution, so $\chi^2 = F(\check{\mathbf{z}}_{\mathbf{IB}})$. Still, if p integer candidates are requested then we set χ^2 as the maximum of the squared norms $F(\boldsymbol{z}^{(1)}), F(\boldsymbol{z}^{(2)}), \dots, F(\boldsymbol{z}^{(p)})$, where we have $\boldsymbol{z}^{(1)} = \check{\mathbf{z}}_{\mathbf{IB}}$. The remaining $p-1$ candidates $\boldsymbol{z}^{(j)}, \forall j > 1$, can be generated by varying the first ambiguity component such that $z_1^{(j)}$ is then equal to the second-nearest, third-nearest, etc…, integer with respect to the conditional estimate $\hat{z}_{1|2\dots n}$.

Hence, we have

$$\chi^2 = F(\boldsymbol{z}^{(p)}) \geq \cdots \geq F(\boldsymbol{z}^{(2)}) \geq F(\boldsymbol{z}^{(1)}) \tag{43}$$

and the ellipsoidal region is now shrunk. Still, we are able to guarantee that at least p candidates are inside this ellipsoid. The search is then continued at 2nd level, where the second-nearest integer of $\hat{z}_{2|3\dots n}$ is taken.

[1] The volume within a n-dimensional ellipsoid is given by

$$E_n = \chi^2 V_n \sqrt{|Q_{\hat{a}\hat{a}}|}, \qquad V_n = \frac{\pi^{\frac{n}{2}}}{\frac{\pi}{2}\Gamma\left(\frac{n}{2}\right)}$$

with $\Gamma(\cdot)$ as the Gamma function. This approximation is known to work well if the number of requested candidate is large and χ^2 is therefore computed by setting $E_n = 1.5p$.

If this integer value lies inside the bounds, the corresponding conditional estimate $\hat{z}_{1|2\ldots n}$ is determined and rounded to the nearest integer, while all other ambiguities, i.e. level 3 to n, remain unchanged. If the vector $z^{(p+1)}$ satisfies $F\left(z^{(p+1)}\right) < \chi^2$, then we replace χ^2 with this new value, and the corresponding integer candidates is also replaced by $z^{(p+1)}$.

As a result of the previous operation, the ellipsoidal region is again shrunk and one can continue the search at 1st level until no new candidates can be found there. In that case, the search is continued as before in the second level, until no more new candidates can be found there either. Eventually, one continues the search process in all levels until no new candidate (besides the current p candidates) can be found, whose squared norm is smaller than the current χ^2. The p integer candidates are finally the sought-for best solutions and the method is exhaustive since it *always*[2] assures that p best solutions are found.

- **NEW ALGORITHM (Ghasemmehdi and Agrell, 2011)**

In Ghasemmehdi and Agrell (2011), a modification of the aforementioned implementation is presented[3], where focus goes on the removal of non-necessary conditioning operations when spanning (down) the search tree. The idea is that we generally operate a conditioning once moving to a different integer component in the search tree, therefore updating all level below. Still, if we vary components in higher levels we will need to re-compute the conditioning of lower levels and in some cases this might be useless.

We refer the reader to (ibid) for more details, whereas a quite different notation has been employed. Still both algorithms can be found in Fig. 2 (ibid) denoted as method #3 and method #8, respectively. In this original work, only the best integer solution is provided, whereas multiple solutions are allowed in LAMBDA 4.0, as described by Eq.(43).

3.2.3.3. Comparison between OLD and NEW algorithms

At this point, in Figure 9 we numerically compare the old and new algorithms, respectively available in LAMBDA 3.0 (2012) and LAMBDA 4.0 (2024) implementations.

[2] The cases where multiple candidates lie on the surface of the same hyper-ellipsoid are in general very rare and taken care in LAMBDA software implementation.
[3] This new algorithm has been integrated also MLAMBDA (Cheng et al., 2005), but starting from its 2016 version and unfortunately with very little acknowledgment of the new method.

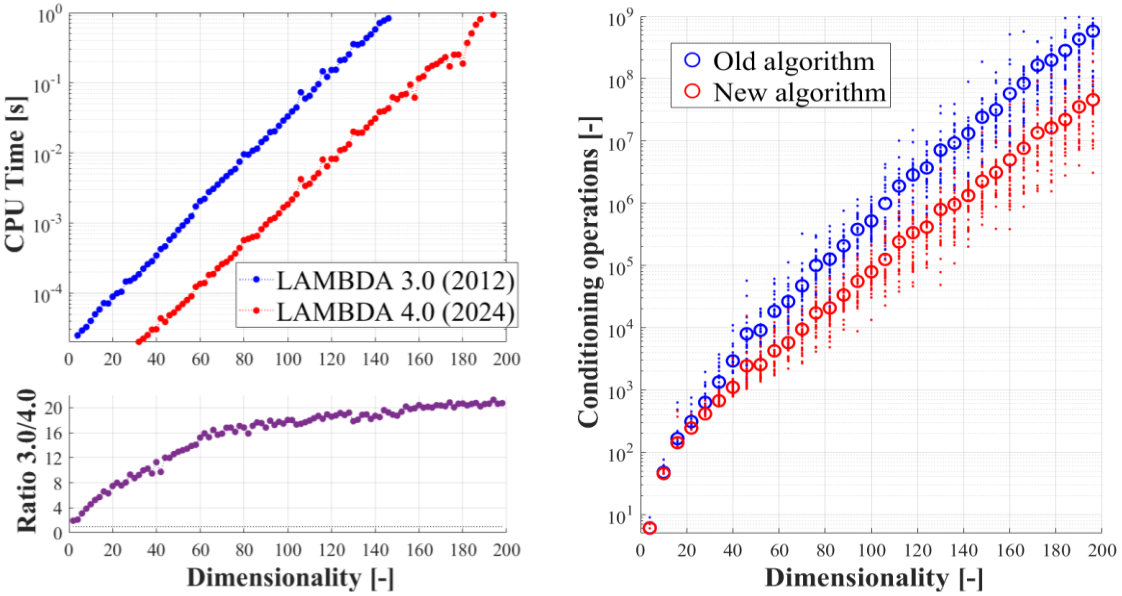


Figure 9: Numerical comparison between LAMBDA 3.0 (OLD) and LAMBDA 4.0 (NEW) in terms of search algorithm computations for finding an optimal ILS solution (see text).

The test is based here on a single-baseline geometry-free ionosphere-fixed dual-frequency model using GPS L1+L2 signals, with a precision of 20 cm and 2 mm for code and phase, respectively. The number of satellites has been varied between 2 and 100 to increase the problem dimensionality, where larger improvements are expected for larger n. In Figure 9, we show the computational time being averaged over 1000 runs (top-left plot) and its ratio between LAMBDA 3.0 and 4.0 (bottom-left plot). Moreover, the number of conditioning operations is also illustrated (right plot), again showing results for each run (dot) and the mean value (circle) based on different float estimates generated.

The averaging over different float values is important, given that computational time (and number of conditioning operations) can also be dependent on the specific float estimates, which defines the center of the search ellipsoid. In the previous results, it is visible that the improvement in computation time is one order of magnitude larger for the new algorithm, especially at higher problem dimensionalities. This is consistent with the larger differences found in terms of conditioning operations employed by the old and new algorithms, with the latter one outperforming the legacy implementation.

Additional numerical comparisons can be found in Jazaeri et al. (2014), where these two search algorithms have been extensively compared via numerical simulations.

3.2.4. **I-class |** Partial Ambiguity Resolution (PAR)

So far, we have discussed integer estimators that aim to fix the entire ambiguity set, but in some cases it is not possible to reliably fix all ambiguity components. Still, one could have sufficient confidence to fix a subset of these ambiguities. This is referred here to as Partial Ambiguity Resolution, or PAR firstly introduced in Teunissen et al. (1999). The selection of a subset to fix is certainly a key issue, and different approaches might be considered, e.g. see Verhagen et al. (2011). In this new LAMBDA 4.0 implementation we still consider a minimum SR criterion, as described in the next sub-section. This well-known criterion might be actually suitable for I-estimators and IA-estimators, whereas for the IE-estimator provided in this toolbox we will introduce later a different PAR approach.

3.2.4.1. Minimum SR criterion

The minimum SR criterion (Teunissen, 2001a) for PAR considers a subset of decorrelated ambiguities, such that its corresponding bootstrapped SR, see Section 3.3.1, is larger than or at least equal to a minimum selected[4] value $P_0 \in [0,1]$.

Ultimately, the goal is to select the largest possible subset such that

$$P_k \overset{\text{def}}{=} \prod_{i=k}^{i=n} \left[2\Phi\left(\frac{0.5}{\sigma_{\hat{z}_{i|I}}}\right) - 1 \right] \geq P_0 \tag{44}$$

where Φ refers to the cumulative normal distribution function (Section 3.3.1). In Eq.(44), note that only the last $n-k+1$ ambiguities will be fixed.

By default the ILS estimator is adopted here for fixing this subset, where the use of IB success rate is justified by its simple analytical expression that allows efficiently defining a suitable subset. Note that by considering decorrelated ambiguities, we are able to enhance their bootstrapped SR, while at the same time the components will be sorted by 'most to least' precise based on their conditional variances.

As a matter of example, one can start with the n-th decorrelated ambiguity and then check if its success rate is larger than or at least equal to P_0. If this is the case, we can continue by checking the conditional ambiguity $\hat{z}_{n-1|n}$, and therefore $\hat{z}_{k-1|K}$ for $K = \{k, \dots, n\}$. This sequential procedure continues until the success rate becomes too small, e.g. $P_{k-1} < P_0$, and all remaining $k-1$ ambiguity components are conditioned onto K fixed ambiguities.

[4] Two extreme cases exist:
- for $P_0 = 0$ we consider the entire ambiguity set (i.e. Full Ambiguity Resolution, or FAR),
- for $P_0 = 1$ we will **not** perform any ambiguity fixing (i.e. Float solution).

The PAR ambiguity vector can be written as

$$\check{\boldsymbol{z}}_{\mathbf{PAR}} \overset{\text{def}}{=} \begin{pmatrix} \hat{\boldsymbol{z}}_{\mathbf{1|2}} \\ \check{\boldsymbol{z}}_{\mathbf{2}} \end{pmatrix} \tag{45}$$

with

$$\hat{\boldsymbol{z}}_{\mathbf{1|2}} = \begin{pmatrix} \hat{z}_{1|K} \\ \vdots \\ \hat{z}_{k-1|K} \end{pmatrix} \in \mathbb{R}^{k-1}, \qquad \check{\boldsymbol{z}}_{\mathbf{2}} = \mathcal{I}_{z_2}(\hat{\boldsymbol{z}}_{\mathbf{2}}) \in \mathbb{Z}^{n-k+1} \tag{46}$$

where $\check{\boldsymbol{z}}_{\mathbf{2}}$ refers to the subset of ambiguities fixed to integers for an integer mapping given by $\mathcal{I}_{z_2}: \mathbb{R}^{n-k+1} \to \mathbb{Z}^{n-k+1}$. As mentioned, we make use of the ILS estimator for fixing this most precise subset, whereas the remaining $k-1$ components are then conditioned and defined in $\hat{\boldsymbol{z}}_{\mathbf{1|2}}$, computed as

$$\hat{\boldsymbol{z}}_{\mathbf{1|2}} = \hat{\boldsymbol{z}}_{\mathbf{1}}(\check{\boldsymbol{z}}_{\mathbf{2}}) \overset{\text{def}}{=} \hat{\boldsymbol{z}}_{\mathbf{1}} - Q_{\hat{z}_1\hat{z}_2} Q_{\hat{z}_2\hat{z}_2}^{-1} (\hat{\boldsymbol{z}}_{\mathbf{2}} - \check{\boldsymbol{z}}_{\mathbf{2}}) \tag{47}$$

with $Q_{\hat{z}_2\hat{z}_2}$ being the variance matrix of $\hat{\boldsymbol{z}}_{\mathbf{2}}$ and $Q_{\hat{z}_1\hat{z}_2}$ being the covariance matrix between the two subsets $\hat{\boldsymbol{z}}_{\mathbf{1}}$ and $\hat{\boldsymbol{z}}_{\mathbf{2}}$.

BACK Z-TRANSFORMATION

The reader should be careful in observing that a (fixed) solution via PAR can be obtained in terms of the original ambiguities by applying a back transformation such that

$$\check{\boldsymbol{a}}_{\mathbf{PAR}} = Z^{-T} \check{\boldsymbol{z}}_{\mathbf{PAR}} \tag{48}$$

where $\check{\boldsymbol{a}}_{\mathbf{PAR}}$ will generally not contain integer elements when only a subset of ambiguities has been fixed. Hence, the entries in $\check{\boldsymbol{a}}_{\mathbf{PAR}}$ are a linear function of previously decorrelated ambiguities in $\check{\boldsymbol{z}}_{\mathbf{PAR}}$, which nonetheless are not all integer-valued.

In practice, this back Z-transformation is not required for the calculation, given that the fixed real-valued parameters can be obtained directly with

$$\check{\boldsymbol{b}}_{\mathbf{PAR}} = \hat{\boldsymbol{b}} - Q_{\hat{b}\hat{z}_2} Q_{\hat{z}_2\hat{z}_2}^{-1} (\hat{\boldsymbol{z}}_{\mathbf{2}} - \check{\boldsymbol{z}}_{\mathbf{2}}) \tag{49}$$

and by assuming a sufficiently high minimum success rate is selected, the approximation adopted in Eq.(7) can be considered, and it follows that

$$Q_{\check{b}\check{b}}^{\text{PAR}} \cong Q_{\hat{b}\hat{b}} - Q_{\hat{b}\hat{z}_2} Q_{\hat{z}_2\hat{z}_2}^{-1} Q_{\hat{z}_2\hat{b}} \tag{50}$$

where $Q_{\hat{b}\hat{z}_2}$ and $Q_{\hat{z}_2\hat{z}_2}$ are submatrices of $Q_{\hat{b}\hat{z}} = Q_{\hat{b}\hat{a}} Z$ and $Q_{\hat{z}\hat{z}}$, respectively.

3.2.5. **I-class |** Vectorial Integer Bootstrapping (VIB)

The Vectorial Integer Bootstrapping (VIB) estimator has been introduced by Teunissen et al. (2021), where the principle of integer bootstrapping is extended to a vectorial form. The VIB estimator can be computed such as

$$\breve{\boldsymbol{a}}_i = \left\lceil \hat{\boldsymbol{a}}_{i|I} \right\rfloor_i = \left\lceil \breve{\boldsymbol{a}}_i - \sum_{j=i+1}^{j=n} Q_{ij|J} Q_{jj|J}^{-1} \left(\hat{\boldsymbol{a}}_{j|J} - \breve{\boldsymbol{a}}_j \right) \right\rfloor_i, \qquad i = 1, \dots, \nu \tag{51}$$

where $\lceil \cdot \rfloor_i : \mathbb{R}^{n_i} \to \mathbb{Z}^{n_i}$ is an admissible integer mapping, meanwhile $\hat{\boldsymbol{a}}_{i|I} \in \mathbb{R}^{n_i}$ refers to the least-squares estimator of $\boldsymbol{a}_i$ conditioned on the values of the previous $I = \{i+1, \dots, n\}$ integer estimated vectors. The covariance matrix of $\hat{\boldsymbol{a}}_i$ and $\hat{\boldsymbol{a}}_{j|J}$ is $Q_{ij|J}$, while the $Q_{jj|J}$ is the variance matrix of $\hat{\boldsymbol{a}}_{j|J}$, while starting with $\hat{\boldsymbol{a}}_{i|I} = \hat{\boldsymbol{a}}_n$ for $i = n$.

The admissible integer mapping is still to be chosen and can arbitrarily be defined using the previous estimators such as IR, IB or ILS. Still, if for a certain n_i-dimensional block we consider IB as the estimator, this is equivalent to a further partitioning into n_i (scalar) blocks and therefore the VIB_{IB} estimator will lead to the exact same (scalar) formulation for the Integer Bootstrapping (Section 3.2.2).

Moreover, in LAMBDA 4.0 toolbox we only consider the adoption of the same integer estimator for all blocks being defined by the partitioning, thus having

- **VIB_{IR} estimator**, where each mapping makes use of IR (Section 3.2.1);
- **VIB_{ILS} estimator**, where each mapping makes use of ILS (Section 3.2.3).

Different properties and success rate bounds can be found in (ibid), where the expression for the VIB pull-in region is given as follows

$$\mathcal{P}_{z,VIB} = \{ \boldsymbol{x} \in \mathbb{R}^n \mid \mathcal{I}_{\text{VIB}}(\boldsymbol{x}) = \boldsymbol{z} \}, \qquad \boldsymbol{z} \in \mathbb{Z}^n \tag{52}$$

with

$$\mathcal{I}_{\text{VIB}}(\boldsymbol{x}) = \mathcal{I}(\boldsymbol{x} + [\boldsymbol{\mathcal{L}} - I_n][\boldsymbol{x} - \mathcal{I}_{\text{VIB}}(\boldsymbol{x})]) \tag{53}$$

for $\boldsymbol{\mathcal{L}} \in \mathbb{R}^{n \times n}$ given as a lower block-triangular matrix, see (ibid), and $\mathcal{I}: \mathbb{R}^n \to \mathbb{Z}^n$ being

$$\mathcal{I}(\boldsymbol{x}) = \begin{pmatrix} \lceil \boldsymbol{x_1} \rfloor_1 \\ \vdots \\ \lceil \boldsymbol{x_\nu} \rfloor_\nu \end{pmatrix}, \qquad \boldsymbol{x} = \begin{pmatrix} \boldsymbol{x_1} \\ \vdots \\ \boldsymbol{x_\nu} \end{pmatrix} \tag{54}$$

where $\nu \in [1, n]$ is the number of partitioning blocks adopted for the full set.

The success rate for two previous VIB estimators follows a performance ordering, which is summarized in Figure 10. Given a selected parametrization, and an arbitrary partitioning with $\nu > 0$ and $n = n_1 + \cdots + n_\nu$, we always have that

$$\text{IR} \leq \text{VIB}_{\text{IR}} \leq \text{IB} \leq \text{VIB}_{\text{ILS}} \leq \text{ILS} \tag{55}$$

where some limit cases can be identified for $\nu \to 1$ and $\nu \to n$.

As previously mentioned, the adoption of $\nu = n$ scalar blocks is equivalent to a (scalar) IB solution and well-known bounds from Ps-LAMBDA (Section 3.3) might therefore be considered. On the other hand, for $\nu = 1$, we are considering the full ambiguity set and VIB simply becomes an IR or ILS depending upon the mapping being defined.

A fixed ν partitioning case, with arbitrary estimators adopted in each block, is also possible and its success rate would be bounded as $\text{VIB}_{\text{IR}} \leq \text{VIB}(\nu) \leq \text{VIB}_{\text{ILS}}$. Still, it has **not** been considered in the LAMBDA 4.0 toolbox, but it could easily be constructed based on its standalone toolbox functionalities.

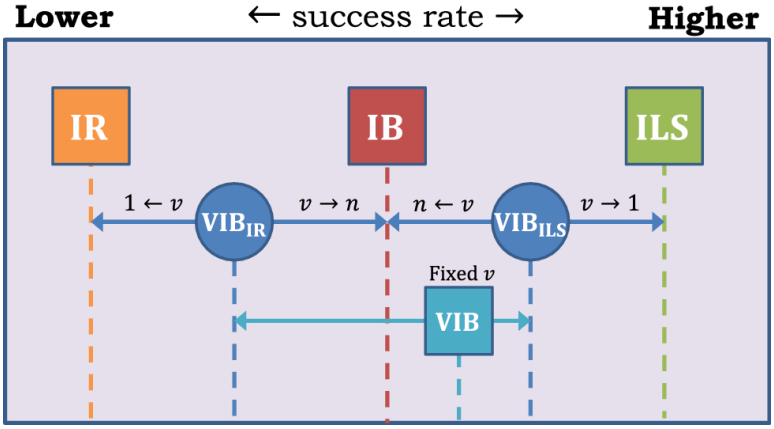


Figure 10: Performance ordering in terms of success rate for legacy integer estimators, i.e. {IR, IB, ILS}, along with certain VIB estimators discussed in the text.

Similarly to the IB estimator, also the VIB estimator is dependent upon the parametrization as the latter affects the conditioning among different blocks. Therefore, for $\hat{\boldsymbol{a}} \sim \mathcal{N}(\boldsymbol{a}, Q_{\hat{a}\hat{a}})$ being normally distributed and $\breve{\boldsymbol{a}}_{\mathbf{VIB}}$ being the VIB estimator of $\boldsymbol{a} \in \mathbb{Z}^n$, we have

$$\text{P}(\breve{\boldsymbol{a}}_{\mathbf{VIB}} = \boldsymbol{a}) = \prod_{i=1}^{i=\nu} \text{P}\left(\hat{\boldsymbol{a}}_{i|\boldsymbol{I}} \in \mathcal{P}_{i,\boldsymbol{a}}\right) \tag{56}$$

with $\text{P}\left(\hat{\boldsymbol{a}}_{i|\boldsymbol{I}} \in \mathcal{P}_i + a\right)$ as the probability that the i-th conditioned float ambiguity block, for $I = \{i+1, \ldots, \nu\}$, belongs to the pull-in region $\mathcal{P}_{i,\boldsymbol{a}} = \left\{\boldsymbol{x} \in \mathbb{R}^{n_i} \,|\, [\boldsymbol{x}]_{\boldsymbol{i}} = \boldsymbol{a}\right\}$ for $i = 1, \ldots, \nu$.

Each probability term depends upon the chosen integer mapping $\lceil\cdot\rfloor_i : \mathbb{R}^{n_i} \to \mathbb{Z}^{n_i}$, whereas the IB formulation (in Section 3.3.1), could still be adopted as an upper bound or a lower bound for the success rate of the VIB_{IR} or VIB_{ILS} estimators, respectively.

PARAMETRIZATION:

The VIB estimator is dependent upon the ambiguity parametrization, so $\check{\boldsymbol{z}}_{\mathbf{VIB}} \neq Z^T \check{\boldsymbol{a}}_{\mathbf{VIB}}$, where $\hat{\boldsymbol{z}} = Z^T \hat{\boldsymbol{a}}$, and this affects its success rate, so $P(\check{\boldsymbol{a}}_{\mathbf{VIB}} = \boldsymbol{a}) \neq P(\check{\boldsymbol{z}}_{\mathbf{VIB}} = \boldsymbol{z})$. Note that, given the same parametrization, these estimators are also affected by the partitioning being selected, as well as for the estimator being used in each subset.

The VIB estimator is therefore dependent upon the partitioning being adopted, as the latter will define the shape of the VIB pull-in region. As a reference, the 3D pull-in regions of some legacy I-estimators are illustrated in Figure 11, so considering Integer Bootstrapping (IB, see Section 3.2.2) and Integer Least-Squares (ILS, see Section 3.2.3). We omit the Integer Rounding (IR, see Section 3.2.1) estimator since its pull-in region is simply a unit cube and projections are unit squares.

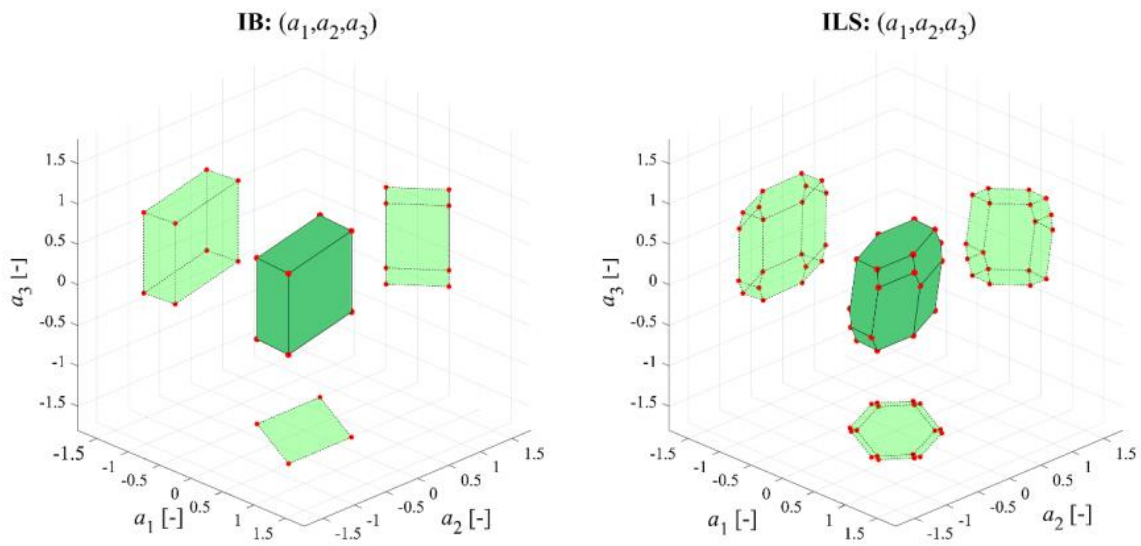


Figure 11: An illustration of the 3D pull-in regions and their 2D projections is given for some legacy integer estimators, i.e. Integer Bootstrapping (IB, left) and Integer Least-Squares (ILS, right).

The 3D pull-in regions for both VIB_{IR} (left) and VIB_{ILS} (right) estimators are illustrated in the next Figure 12, taken from (ibid), while assuming a different partitioning. For the sake of clarity, the projections on the three axes are shown, along with the vertices (red dots).

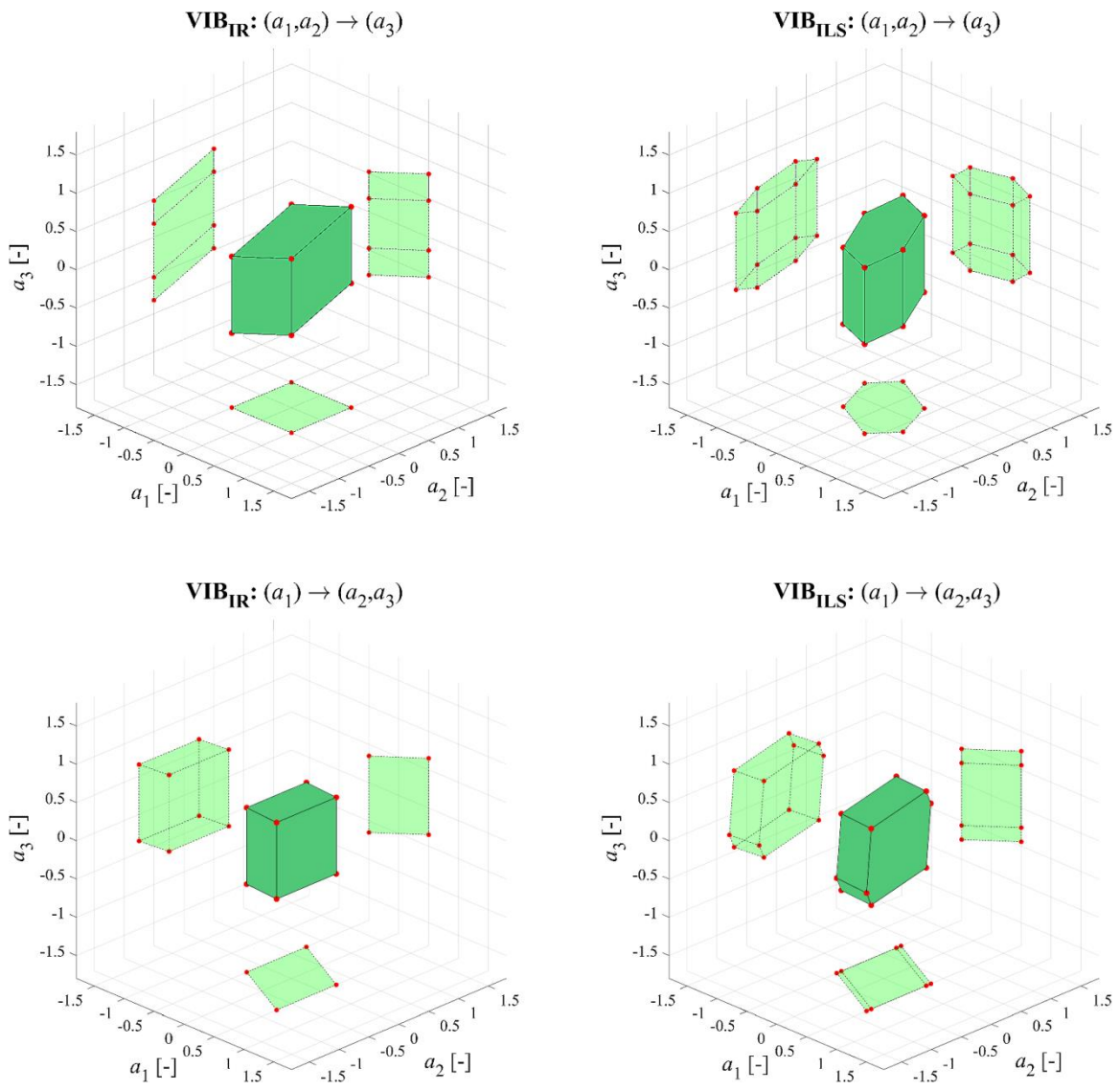


Figure 12: An illustration of 3D pull-in region and their 2D projections for both VIB_{IR} (left) and VIB_{ILS} (right) estimators is given, while assuming two different types of partitioning, i.e. {1,2} in the top row and {2,1} in the bottom row.

3.2.6. **IA-class |** Integer Aperture with Ratio Test (IA-RT)

So far, we have discussed some members of the I-class, including the ILS estimator that is optimal as it has the largest success rate possible. At this point we consider another class of estimators, being defined by Integer Aperture (IA) as discussed in Section 2.3.2. This class does not only concern integer estimator, but it generally also involves an acceptance test on the integer solution. Hence, it unified the integer ambiguity estimation and validation, with the 'fixed' solution being an integer only if the acceptance criterion is met.

The Ratio Test (RT) is a member of the IA-class and it is probably one of the most popular acceptance tests, see Teunissen and Verhagen (2009). It is de facto a discrimination test since it tests the closeness of the float solution to the optimal integer one compared to other integer candidates. Hence, we can define

$$\text{Accept } \breve{\boldsymbol{a}} \in \mathbb{Z}^n \text{ iff } \frac{F(\breve{\boldsymbol{a}})}{F(\breve{\boldsymbol{a}}')} \leq \mu_{\text{RT}} \tag{57}$$

where $\mu_{\text{RT}} \in [0,1]$ is the threshold value, whereas $\breve{\boldsymbol{a}}$ and $\breve{\boldsymbol{a}}'$ are the best and 2nd-best ILS estimator of $\boldsymbol{a}$,. The objective function $F: \mathbb{Z}^n \to \mathbb{R}$ has been defined in Eq.(36), so we have that $F(\breve{\boldsymbol{a}}) \leq F(\breve{\boldsymbol{a}}')$. Two limit cases are found: if $\mu_{\text{RT}} = 0$, then the test is always rejected and the aperture region is null, while if $\mu_{\text{RT}} = 1$, then the test is always accepted and an integer value is returned. Note that the value $c = 1/\mu_{\text{RT}}$ is often used in literature.

Differently from I-estimators, an undecided region exists here where we have

$$P_s + P_f + P_u = 1, \qquad \begin{cases} P_s \overset{\text{def}}{=} \text{P}(\breve{\boldsymbol{a}}_{\mathbf{IA}} = \boldsymbol{a}) \\ P_f \overset{\text{def}}{=} \text{P}(\breve{\boldsymbol{a}}_{\mathbf{IA}} = \boldsymbol{z} \neq \boldsymbol{a}) \\ P_u \overset{\text{def}}{=} \text{P}(\breve{\boldsymbol{a}}_{\mathbf{IA}} = \hat{\boldsymbol{a}}) \end{cases} \tag{58}$$

and therefore a relation between the failure rate and aperture parameter μ_{RT} exists, where the integer aperture estimator $\breve{\boldsymbol{a}}_{\mathbf{IA}}$ follows the definition already given in Eq.(20). Note that an Optimal Integer Aperture (OIA) estimator (Teunissen, 2005c), can be found by solving

$$\max_{\Omega_0} P_s \quad \text{subject to given} \quad P_f \tag{59}$$

where, given S_0 as ILS pull-in region, the aperture pull-in region Ω_0 is defined as

$$\Omega_0 = \left\{ \boldsymbol{x} \in S_0 \mid \sum_{z \in \mathbb{Z}^n} f_{\hat{a}}(\boldsymbol{x} + \boldsymbol{z}) \leq \lambda_{\text{FF}} f_{\hat{a}}(\boldsymbol{x} + \boldsymbol{a}) \right\} \tag{60}$$

with λ_{FF} chosen so as to satisfy the a priori fixed failure-rate $P_f > 0$. The term $f_{\hat{a}}(\boldsymbol{x})$ refers to the Probability Density Function (PDF) of the float solution $\hat{\boldsymbol{a}} \in \mathbb{R}^n$.

In general, given $\boldsymbol{a} \in \mathbb{Z}^n$ as the true integer vector, four cases can be distinguished:

- Correct acceptance (success) | Accept $\breve{\boldsymbol{a}}_{\mathbf{IA}} = \boldsymbol{a}$ ☺
- Correct rejection | Reject $\breve{\boldsymbol{a}}_{\mathbf{IA}} \neq \boldsymbol{a}$ ☺
- Unnecessary rejection (false alarm) | Reject $\breve{\boldsymbol{a}}_{\mathbf{IA}} = \boldsymbol{a}$ ☹
- Wrong acceptance (failure) | Accept $\breve{\boldsymbol{a}}_{\mathbf{IA}} \neq \boldsymbol{a}$ ☹

that have been illustrated in the following Figure 13.

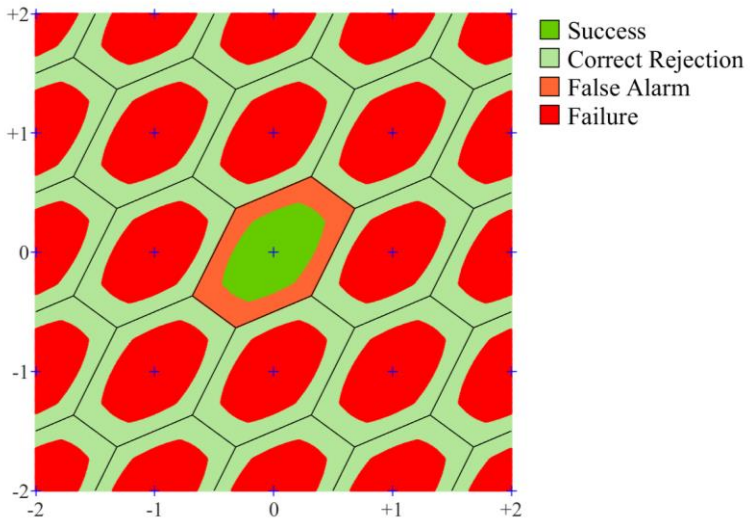


Figure 13: Illustration of acceptance regions with Ratio Test (in bright green and red colors).

A few illustrations of a 2D aperture pull-in region are given in Figure 14, similarly to what shown for the I-class. However, in this case we can observe that an undecided region in grey color. The latter represents the region where the ratio test is not accepted, and a float solution is returned, and so avoiding to incorrectly fixing the ambiguities. In this way, it becomes possible to reduce the failure rate.

As discussed in Step 3 of Section 2.1, a wrong fix of ambiguities can actually decrease the accuracy of the real-valued parameters given that they are statistically conditioned onto the ambiguities. Figure 14 compares different threshold values, being $\mu_{\mathrm{RT}} = [0.2,0.4,0.6,0.8]$ and it is visible that the size of the acceptance region (in green) is driven by the μ_{RT} value while both SR and FR are being affected. For instance, a larger μ_{RT} value will result in a larger acceptance region, and higher probabilities of failure and success.

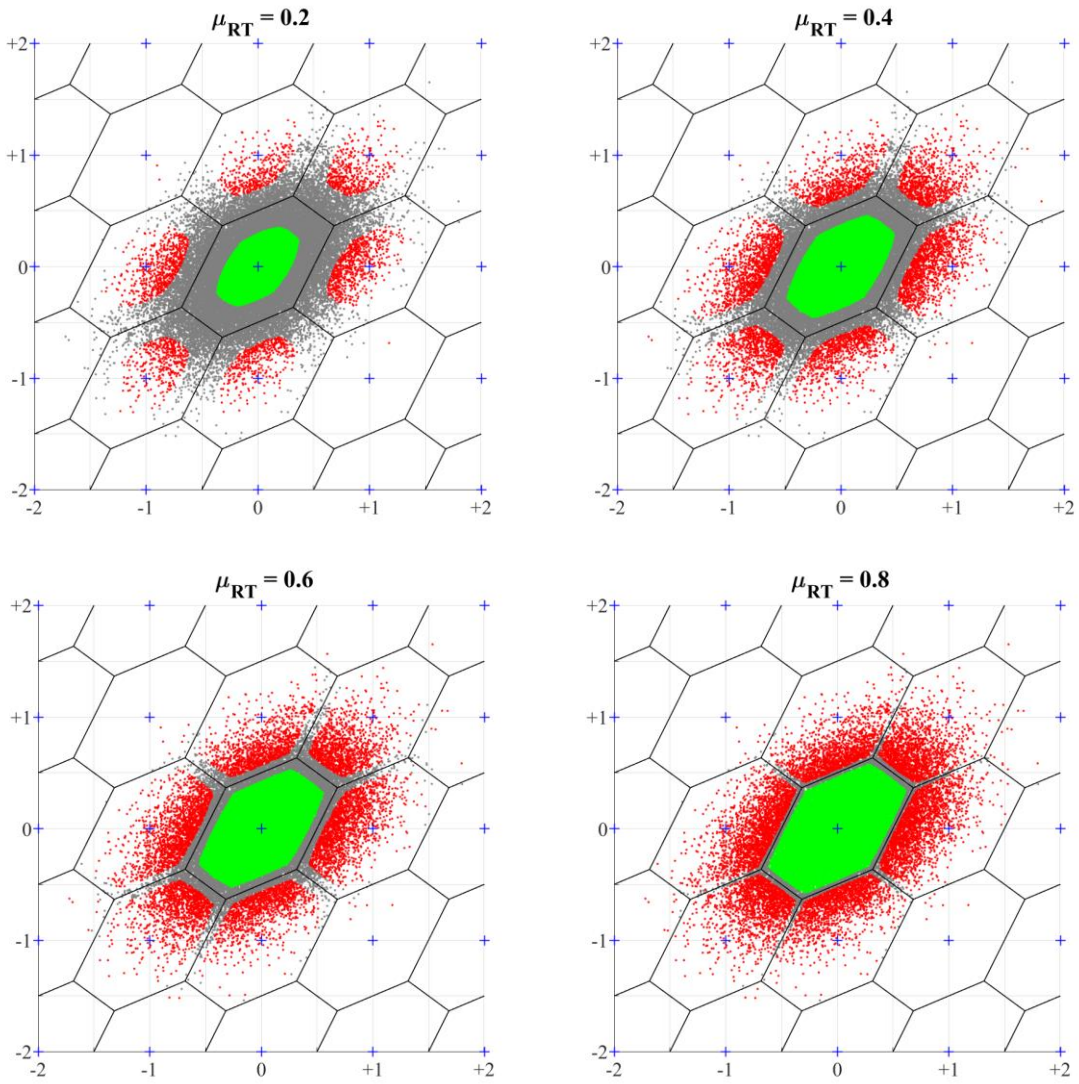


Figure 14: Illustration of the IA pull-in regions, along with 40'000 samples being generated based on $\hat{a} \sim \mathcal{N}(0, Q_{\hat{a}\hat{a}})$. In green, we show the float ambiguity samples that are pulled to the true integer vector, i.e. $a = 0$, while in grey color the samples that have not been fixed to an integer value. Different μ_{RT} values are adopted in the plots, being 0.2, 0.4, 0.6 and 0.8.

In the limit case $\mu_{RT} = 1$, we get $P_u = 0$ and the ILS pull-in region from Figure 7 is found, whereas for the case $\mu_{RT} = 0$ we get $P_u = 1$ and a float estimator $\hat{\boldsymbol{a}} \in \mathbb{R}^n$ is returned.

For the sake of clarity, a numerical example is shown in Figure 15, where the probabilities P_s, P_f and $P_u$are numerically computed over the same 40'000 samples, i.e. $\hat{\boldsymbol{a}} \sim \boldsymbol{\mathcal{N}}(\boldsymbol{0}, Q_{\hat{a}\hat{a}})$, and are then plotted based on a range of values $\mu_{RT} \in [0,1]$.

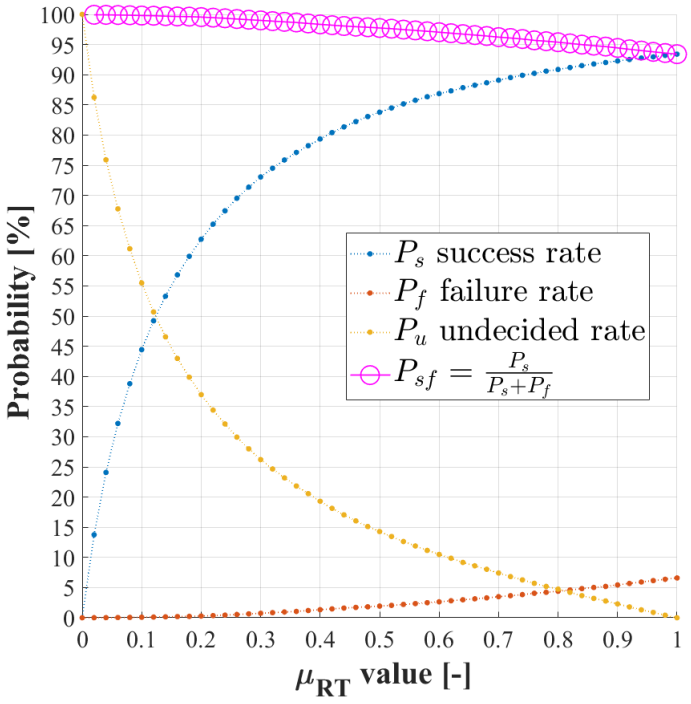


Figure 15: Example of different probabilities computed for a 2D problem using 40'000 samples being generated based on $\hat{a} \sim \mathcal{N}(0, Q_{\hat{a}\hat{a}})$, and using a different aperture parameter.

In this last numerical example, another useful performance indicator has been considered, i.e. the successful-fix rate P_{sf}, which is defined as follows

$$P_{sf} = \frac{P_s}{P_s + P_f} \tag{61}$$

being the probability of successfully fixing the float ambiguity. Note that $P_s + P_f = P_{fix}$ is the probability that a fix is made, which is equal to $1 - P_u$, where P_u gets larger for smaller μ_{RT} values. In general, the user would like to have $P_{sf} \approx 1$, and this could be achieved by limiting the failure-rate P_f at a sufficiently small value.

As mentioned, the LAMBDA user has the possibility of specifying a certain μ_{RT} value and performing a ratio test on the fixed ambiguity solution. Still, it is also possible to define a maximum failure-rate value that should not be exceeded, thus having a Fixed Failure-rate Ratio Test (FFRT), as being introduced in Teunissen and Verhagen (2009). In this model-driven Ratio Test, the aperture parameter μ_{FFRT} is found based on lookup-tables that more conveniently relates it to an expected failure rate, as discussed in the next section.

3.2.6.1. Fixed Failure-rate Ratio Test (Hou et al. 2016)

The new μ_{FFRT} values adopted in LAMBDA 4.0 are based on the lookup-tables provided in Hou et al. (2016). These are generated by fitting functions, whose coefficients are made available as supplementary material. These coefficients have been integrated in this new LAMBDA release, and might further improve the fixing results by avoiding unnecessary false alarms (ibid). The fitting function μ_{FFRT} is given at each dimensionality n as

$$\mu_{\text{FFRT}} = \begin{cases} 0, & P_f \geq 0.2 \\ f_\mu(P_f), & P_f^{\text{tol}} \leq P_f \leq 0.2 \\ 1, & \text{otherwise} \end{cases} \tag{62}$$

where P_f^{tol} is a tolerable failure rate, and we have

$$f_\mu(x) = p_1 x^{p_2} + p_3 \tag{63}$$

where $\{p_1, p_2, p_3\}$ is a set of coefficients[5] provided in (ibid).

3.2.7. **IA-class |** Integer Aperture Bootstrapping (IAB)

In Teunissen (2005a), a new estimator in the IA-class has been introduced, named Integer Aperture Bootstrapping (IAB) method. The IB estimator from the I-class (Section 3.2.2), is one of the simplest methods for integer estimation with a close-to-optimal performance, especially following the decorrelation of ambiguities.

Moreover, IB has an analytical exact expression for both success and failure rates, but user has no freedom in controlling such rates for a given mathematical model. Hence, the IAB estimator is introduced here to allow users having a controlled failure rate (with an exact expression) by defining again the aperture of the IAB pull-in region, i.e. $\beta_{\text{IAB}} \in [0,1]$.

Following the definitions in Eq.(58), along with the expression of pull-in regions for the IB estimator, e.g. see Section 3.2.2, we can define the IAB pull-in region as

$$\mathcal{P}_{\text{z,IAB}} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \,|\, \frac{\boldsymbol{x} - \boldsymbol{z}}{\beta_{\text{IAB}}} \in \mathcal{P}_{\text{0,IB}} \right\}, \qquad \boldsymbol{z} \in \mathbb{Z}^n \tag{64}$$

where $\mathcal{P}_{\text{0,IB}}$ is the IB pull-in region centered on $\boldsymbol{z} = \boldsymbol{0}$, and this last result is made possible given the translation equivariant property of IA-estimators (see Section 2.3.2).

[5] Note that the look-up tables span $P_f^{\text{tol}} = [0.05, \ldots, 0.09, 0.1, 0.2, \ldots, 0.9, 1]/100$, while supporting a dimensionality $n \in [1,66]$. We refer to Hou et al. (2016) for additional information.

The IAB computations follow as

1. **Decorrelation:** $\hat{\boldsymbol{z}} = Z^T \hat{\boldsymbol{a}}$ for $Z \in \mathbb{Z}^{n\times n}$ (unimodular)
2. **Compute IB solution:** $\check{\boldsymbol{z}}_{\mathbf{IB}} = \mathcal{I}_{\mathrm{IB}}(\hat{\boldsymbol{z}})$ for $\mathcal{I}_{\mathrm{IB}}: \mathbb{R}^n \to \mathbb{Z}^n$
3. **Compute IB residual:** $\check{\boldsymbol{\epsilon}}(\hat{\boldsymbol{z}}) = \hat{\boldsymbol{z}} - \check{\boldsymbol{z}}_{\mathbf{IB}}$
4. **Accept $\check{\boldsymbol{z}}_{\mathbf{IB}}$ if:** $\check{\boldsymbol{\epsilon}}(\hat{\boldsymbol{z}})/\beta_{\mathrm{IAB}} \in \mathcal{P}_{0,\mathrm{IB}}$

where in the last point we need to make use of the IB estimator on the scaled ambiguity residual. If the outcome is the zero vector, then the condition in Eq.(64) is satisfied for this specific float ambiguity $\hat{\boldsymbol{a}} \in \mathbb{R}^n$. Hence, the IAB requires only applying the IB procedure twice, which is generally more efficient than using ILS (as needed for the IA-RT case).

The probabilities of success, failure and undecided (later further discussed in Section 3.3) can be found analytically and are given from (ibid) as follow

$$
\begin{aligned}
P_{s,\mathrm{IAB}} &= \prod_{i=1}^{i=n}\left[2\Phi\left(\frac{0.5\beta_{\mathrm{IAB}}}{\sigma_{i|I}}\right)-1\right] \\
P_{f,\mathrm{IAB}} &= \sum_{\boldsymbol{z}\in\mathbb{Z}^n\setminus\{\mathbf{0}\}}\prod_{i=1}^{i=n}\left[\Phi\left(\frac{0.5\beta_{\mathrm{IAB}}-\boldsymbol{c}_i^T L^{-1}\boldsymbol{z}}{\sigma_{i|I}}\right)+\Phi\left(\frac{0.5\beta_{\mathrm{IAB}}+\boldsymbol{c}_i^T L^{-1}\boldsymbol{z}}{\sigma_{i|I}}\right)-1\right] \\
P_{u,\mathrm{IAB}} &= 1-P_s-P_f
\end{aligned}
\tag{65}
$$

where the IAB aperture parameter β_{IAB} has been highlighted. In this way, there is a direct analytical relationship between the aperture parameter and the success/failure rates. The user could select a certain value for β_{IAB}, where two limit cases can be identified:

- $\beta_{\mathrm{IAB}} \to 0$, then $P_{s,\mathrm{IAB}} \to 0, P_{f,\mathrm{IAB}} \to 0$ and $P_{u,\mathrm{IAB}} \to 1$, so no fixing occurs;
- $\beta_{\mathrm{IAB}} \to 1$, then $P_{s,\mathrm{IAB}} \to P_{s,\mathrm{IB}}, P_{f,\mathrm{IAB}} \to P_{f,\mathrm{IB}}$ and $P_{u,\mathrm{IAB}} \to 0$.

and therefore in the latter case we will always return an IB fixed solution.

Similarly to Figure 14, we can briefly illustrate a similar 2D example while using different aperture parameters, i.e. $\beta_{\mathrm{IAB}} = [0.2,0.4,0.6,0.8]$. The shape of IB pull-in region is one key difference with respect to the IA-RT discussed in Section 3.2.6. However, in this case we have a known relationship between failure rate $P_{f,\mathrm{IAB}}$ and aperture parameter β_{IAB}. Hence, we can further investigate a "controlled-FR approach" based on this IAB estimator.

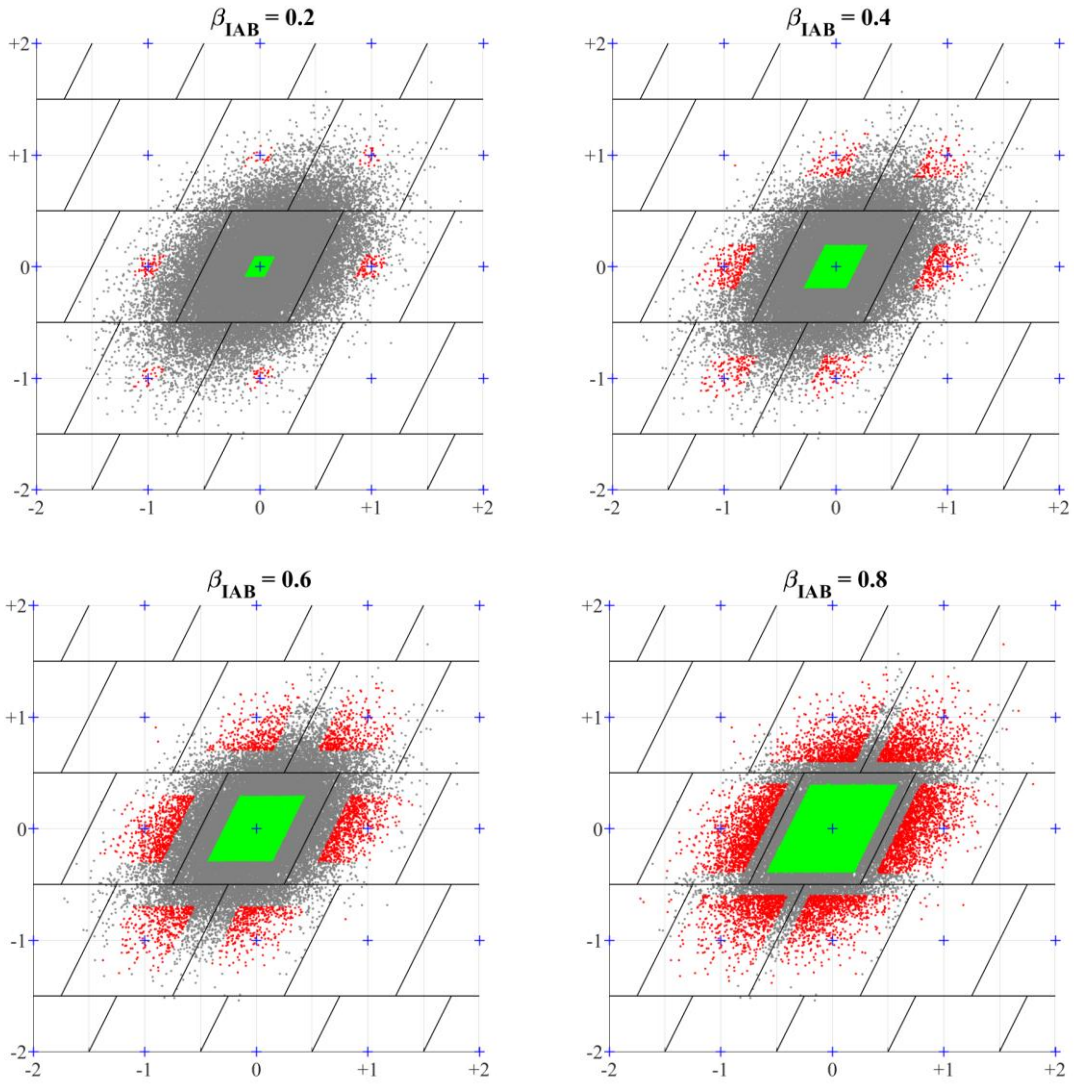


Figure 16: Illustration of the IAB pull-in regions, along with 40'000 samples being generated based on $\hat{a} \sim \mathcal{N}(\mathbf{0}, Q_{\hat{a}\hat{a}})$. In green, we show the float ambiguity samples that are pulled to the true integer vector, i.e. $a = \mathbf{0}$, while in grey color the samples that have not been fixed to an integer value. Different β_{IAB} values are adopted in the plots, being 0.2, 0.4, 0.6 and 0.8.

3.2.7.1. IAB with Controlled Failure-rate

The Eq.(65) gives an analytical relation $P_f = P_f(\beta_{\text{IAB}})$, nonetheless involving an infinite summation over $\mathbb{Z}^n$. A finite approximation is possible, but it requires an enumeration of integer candidates *sufficiently* close to $\boldsymbol{z} = \mathbf{0}$ in order to well approximate the IAB failure rate. Hence, LAMBDA 4.0 makes use of efficient numerical simulations and a bisection method to compute this aperture coefficient β_{IAB} for a user-selected maximum FR.

Given the user-selected maximum failure rate $P_f^{\mathrm{MAX}} \in (0,1)$, we seek β_{IAB} so that

$$P_f^{\mathrm{MAX}} \geq P_{f,\mathrm{IAB}}(\beta_{\mathrm{IAB}}) \tag{66}$$

where we make use of the expression in Eq.(65) for the IAB failure rate. In this way, we can search for an aperture parameter β_{IAB} that satisfies the inequality while approximating the FR by a finite summation over a finite set $\Omega_z^{\mathrm{IAB}} \subset \mathbb{Z}^n \backslash \{\mathbf{0}\}$, such that

$$P_f^{\mathrm{MAX}} \geq \sum_{\boldsymbol{z} \in \Omega_z^{\mathrm{IAB}}} \prod_{i=1}^{i=n} \left[\Phi\left(\frac{0.5\beta_{\mathrm{IAB}} - \boldsymbol{c_i^T} L^{-1} \boldsymbol{z}}{\sigma_{i|I}} \right) + \Phi\left(\frac{0.5\beta_{\mathrm{IAB}} + \boldsymbol{c_i^T} L^{-1} \boldsymbol{z}}{\sigma_{i|I}} \right) - 1 \right] \tag{67}$$

with the equality solved via bisection method[6], thus considering the objective function

$$f_{\mathrm{IAB}}(\beta) = P_f^{\mathrm{MAX}} - P_{f,\mathrm{IAB}}(\beta) = 0 \tag{68}$$

where we start with an interval $\beta \in [\beta^-, \beta^+]$, for $\beta^- = 0$ and $\beta^+ = 1$. At each iteration the middle point in the interval is evaluated, i.e. $\beta^{\mathrm{MP}} = (\beta^- + \beta^+)/2$ and this replaces the extremum with an objective function having the same sign, i.e.

$$\begin{aligned} &\text{if } f_{\mathrm{IAB}}(\beta^{\mathrm{MP}}) f_{\mathrm{IAB}}(\beta^+) < 0, \qquad \text{then } \beta^- = \beta^{\mathrm{MP}} \\ &\text{if } f_{\mathrm{IAB}}(\beta^-) f_{\mathrm{IAB}}(\beta^{\mathrm{MP}}) < 0, \qquad \text{then } \beta^+ = \beta^{\mathrm{MP}} \end{aligned} \tag{69}$$

so with k-th iterations the interval for β is reduced to 2^{-k}. The final solution will provide the aperture parameter value that satisfy the equality in Eq.(67). It is important to realize that, given $P_f^{\mathrm{MAX}} \in (0,1)$, at the very first iteration we have

$$\begin{aligned} f_{\mathrm{IAB}}(\beta^- = 0) &= P_f^{\mathrm{MAX}} \geq 0 \\ f_{\mathrm{IAB}}(\beta^+ = 1) &= P_f^{\mathrm{MAX}} - P_{f,\mathrm{IB}} \end{aligned} \tag{70}$$

and if $P_f^{\mathrm{MAX}} \geq P_{f,\mathrm{IB}}$, then the IB estimator will already satisfy our controlled-FR criterion, since its failure rate will be smaller than the maximum value selected by the user. On the other hand, if $P_f^{\mathrm{MAX}} < P_{f,\mathrm{IB}}$, we need to reduce β in order to reduce also IAB failure rate $P_{f,\mathrm{IAB}}(\beta)$ till it gets to the threshold P_f^{MAX}.

[6] We omit the computational steps here, and we refer to Appendix A.3.1 in Verhagen (2005) for additional information. The tolerance value for the root finding iterations is set to a small value, e.g. $|f_{\mathrm{IAB}}(\beta_k)| < 10^{-5}$ for any $k > 0$, which provides a sufficiently accurate result.

We remark that the proposed approach is still based on an approximation of IAB failure rate, but it allows to efficiently compute the aperture parameter without the bottleneck of an infinite summation in Eq.(65). The finite set Ω_z^{IAB} considers "nIntegers" as the integers closer to $\boldsymbol{z} = \mathbf{0}$ (i.e., excluded from this set) by making use of ILS estimator.

3.2.8. **IE-class |** Best Integer Equivariant (BIE)

The Best Integer Equivariant (BIE) estimator, introduced in Teunissen (2003a), belongs to the IE-class and is actually optimal within this class. We should be careful since optimality implies that a certain criterion is being considered. For the ILS estimator, i.e. optimal in the I-class, this criterion was the success rate. In fact, the ILS provides the highest success rate possible among admissible integer estimators. If SR ≈ 1, the fixed real-valued parameters' vector $\breve{\boldsymbol{b}} = \widehat{\boldsymbol{b}}(\breve{\boldsymbol{a}})$, being conditioned onto $\breve{\boldsymbol{a}} \in \mathbb{Z}^n$, have a higher probability of being close to the true $\boldsymbol{b} \in \mathbb{R}^n$ than based on the float estimator $\widehat{\boldsymbol{a}} \in \mathbb{R}^n$.

The optimality criterion for BIE is a minimum Mean Squared Error (MSE), since the MSE is generally a well-known probabilistic measure for the closeness of certain estimator to its target value and often used as a measure for the quality of the 'ambiguity-float' solution itself. In LAMBDA 4.0, only a BIE formulation for normally distributed data is adopted, whereas the user shall consider that in Teunissen (2020a) a more general formulation for the class of Elliptically Contour Distributions (ECDs) has been provided. In (ibid), the attention is given also to contaminated normal distribution and multivariate t-distribution.

The BIE expression for the generic ECDs case follows as

$$\breve{\boldsymbol{a}}_{\mathbf{BIE}} = \sum_{\boldsymbol{z} \in \mathbb{Z}^n} \boldsymbol{z}\, w_z(\widehat{\boldsymbol{a}}) \tag{71}$$

with weights $w_z \colon \mathbb{R}^n \to \mathbb{R}$ given as

$$w_z(\widehat{\boldsymbol{a}}) = \frac{h_z(\widehat{\boldsymbol{a}})}{\sum_{u \in \mathbb{Z}^n} h_u(\widehat{\boldsymbol{a}})}, \qquad \rightarrow \qquad \sum_{\boldsymbol{z} \in \mathbb{Z}^n} w_z(\widehat{\boldsymbol{a}}) = 1 \tag{72}$$

and the auxiliary function h_z is defined – in the normally distributed case – as

$$h_z(\widehat{\boldsymbol{a}}) \propto \exp\left(-\frac{\|\widehat{\boldsymbol{a}} - \boldsymbol{z}\|^2_{Q_{\hat{a}\hat{a}}}}{2}\right), \qquad \forall \boldsymbol{z} \in \mathbb{Z}^n \tag{73}$$

where $\propto$ reads "proportional to".

Ultimately, we can obtain

$$\breve{\boldsymbol{a}}_{\mathbf{BIE}} = \frac{\sum_{\boldsymbol{z}\in\mathbb{Z}^n} \boldsymbol{z} \exp\left(-\frac{\|\hat{\boldsymbol{a}}-\boldsymbol{z}\|^2_{Q_{\hat{a}\hat{a}}}}{2}\right)}{\sum_{\boldsymbol{u}\in\mathbb{Z}^n} \exp\left(-\frac{\|\hat{\boldsymbol{a}}-\boldsymbol{u}\|^2_{Q_{\hat{a}\hat{a}}}}{2}\right)} \tag{74}$$

while the fixed real-valued parameter vector follows from

$$\breve{\boldsymbol{b}}_{\mathbf{BIE}} = \hat{\boldsymbol{b}}(\breve{\boldsymbol{a}}_{\boldsymbol{BIE}}) \overset{\text{def}}{=} \hat{\boldsymbol{b}} - Q_{\hat{b}\hat{a}} Q^{-1}_{\hat{a}\hat{a}} (\hat{\boldsymbol{a}} - \breve{\boldsymbol{a}}_{\mathbf{BIE}}) \tag{75}$$

so having a similar expression as seen in previous estimators. However, in this case, the outcome of the ambiguity fixing process is a real-valued vector $\breve{\boldsymbol{a}}_{\mathbf{BIE}} \in \mathbb{R}^n$, such that

$$\mathrm{E}\{\breve{\boldsymbol{b}}_{\mathbf{BIE}}\} = \begin{Bmatrix}\mathrm{E}\{\hat{\boldsymbol{b}}\}\\ \mathrm{E}\{\breve{\boldsymbol{b}}\}\end{Bmatrix}, \qquad \mathrm{D}\{\breve{\boldsymbol{b}}_{\mathbf{BIE}}\} \leq \begin{Bmatrix}\mathrm{D}\{\hat{\boldsymbol{b}}\}\\ \mathrm{D}\{\breve{\boldsymbol{b}}\}\end{Bmatrix} \tag{76}$$

so it is unbiased as the float and fixed solutions, whereas it will always outperform both of them based on a minimum MSE criterion. Note that previous results can be generalized for any arbitrary linear combination $\theta = \boldsymbol{l}_{\boldsymbol{a}}^{\boldsymbol{T}}\,\boldsymbol{a} + \boldsymbol{l}_{\boldsymbol{b}}^{\boldsymbol{T}}\,\boldsymbol{b}$ with $\boldsymbol{l}_{\boldsymbol{a}} \in \mathbb{R}^n$ and $\boldsymbol{l}_{\boldsymbol{b}} \in \mathbb{R}^p$, see (ibid).

An illustration of BIE solutions is given in Figure 17, where the ILS pull-in regions have been drawn as reference. Both the 'no gaps' and 'no overlap' conditions are **not** satisfied in the IE-class, while a star-like pattern emerges due to the pulling effect of each individual integer candidates, proportional to their weights.

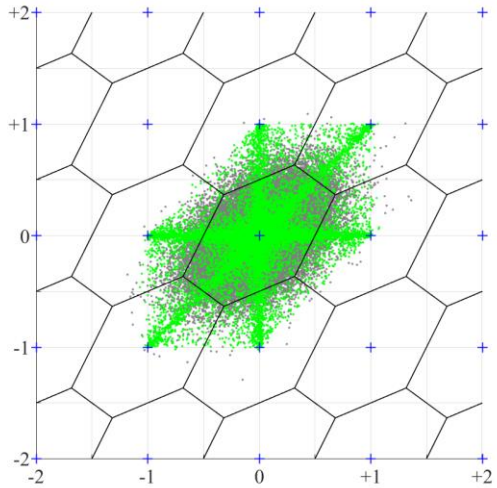


Figure 17: Illustration of the BIE solution (in green) and float solution (in grey) for 40'000 samples generated based on a bivariate normal distribution centered at $a = 0$.

Two illustrations are given in Figure 18, for a different scaling of the variance-covariance matrix of the ambiguities, thus considering a more and less precise $Q_{\hat{a}\hat{a}}$ matrix respectively in the left and right plot. Assuming a factorization $Q_{\hat{a}\hat{a}} = \sigma^2 \tilde{Q}_{\hat{a}\hat{a}}$, given $\sigma \in (0, +\infty)$, then

$$\lim_{\sigma \to 0} \breve{\boldsymbol{a}}_{\mathbf{BIE}} = \breve{\boldsymbol{a}}_{\mathrm{ILS}} \in \mathbb{Z}^n, \qquad \lim_{\sigma \to \infty} \breve{\boldsymbol{a}}_{\mathbf{BIE}} = \hat{\boldsymbol{a}} \in \mathbb{R}^n \tag{77}$$

as somehow also visible in the following two examples.

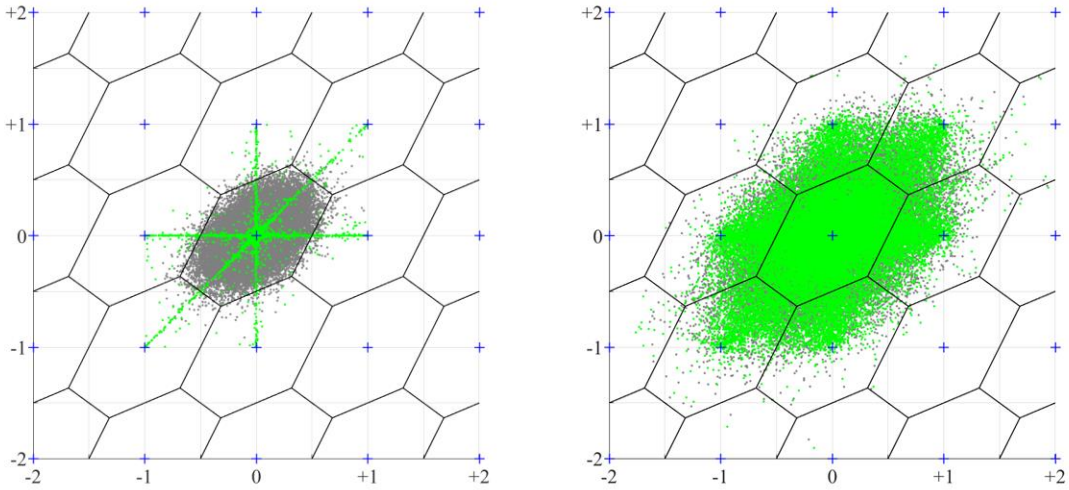


Figure 18: An illustration of the BIE solutions, considering a scaling factor on the variance-covariance matrix for the float ambiguities. See text for more details.

The last result follows from the simple fact that a very high precision of ambiguities will give a larger weight to the ILS integer candidate, which is closer (under the metric $Q_{\hat{a}\hat{a}}$) to the float solution. On the other hand, the weights of different integer candidates will not differ much in the case of very poor precision, and the net effect will be close to zero.

An additional example for a scalar ambiguity component is presented in Figure 19, where three different precision values are considered. Numerical simulations are adopted based on 100'000 samples normally distributed around $a = 0$ and the relative frequency is shown for the three cases. For the ILS the probability mass function is illustrated (blue bars), along with the success rate value, i.e. $P_{s,\mathrm{ILS}} = \mathrm{P}(\breve{a}_{\mathrm{ILS}} = 0)$ with $\breve{a}_{\mathrm{ILS}} \in \mathbb{Z}$ from Eq.(33).

At this point, we proceed in Section 3.2.8.1 by describing the BIE computation strategy adopted in this LAMBDA 4.0 implementation, thus involving a finite set approximation as introduced in (Teunissen, 2005b).

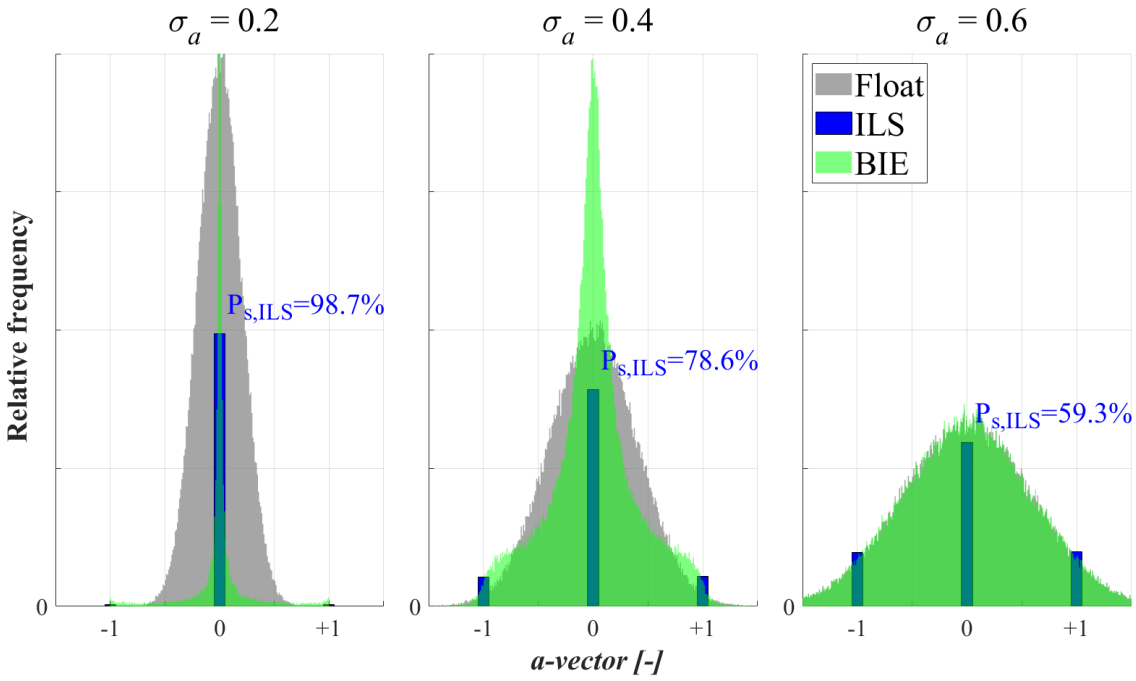


Figure 19: Illustration of distribution for Float, ILS and BIE is provided based on a 1D example and numerical simulations using three different precision values $\sigma_{\hat{a}}$ for generating the 100'000 ambiguity samples $\hat{a} \sim \mathcal{N}(\mathbf{0}, \sigma_{\hat{a}})$.

3.2.8.1. Computation of the BIE estimator

The BIE solution requires a summation over an infinite set of integer candidates, which is certainly not feasible. Hence, an approximation is provided in (ibid) where the summation is performed on the integer set Θ_x^λ being defined as

$$\Theta_x^\lambda = \{\boldsymbol{z} \in \mathbb{Z}^n \mid \|\boldsymbol{x} - \boldsymbol{z}\|^2_{Q_{\hat{a}\hat{a}}} < \lambda^2\}, \qquad \boldsymbol{x} \in \mathbb{R}^n \tag{78}$$

so considering all integer vectors $\boldsymbol{z}$ within a certain distance from $\boldsymbol{x}$, which refers here to a float ambiguity vector. The parameter λ can be found as

$$\mathrm{P}\big(\hat{\boldsymbol{a}} \in E_a^\lambda\big) = \mathrm{P}\big(\|\hat{\boldsymbol{a}} - \boldsymbol{a}\|^2_{Q_{\hat{a}\hat{a}}} < \lambda^2\big) = 1 - \alpha \tag{79}$$

with $\alpha \ll 1$ being a small value, and the ellipsoidal region given by

$$E_z^\lambda = \{\boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{x} - \boldsymbol{z}\|^2_{Q_{\hat{a}\hat{a}}} < \lambda^2\}, \qquad \boldsymbol{z} \in \mathbb{Z}^n \tag{80}$$

with its shape determined by $Q_{\hat{a}\hat{a}}$, and its size governed by λ. In this way, it is possible to find $\lambda = \lambda(\alpha)$ following a Chi-squared (central) distribution with n degrees of freedom.

Given $\alpha \ll 1$, it might happen that no integer candidates are found and the set $\Theta_{\hat{a}}^{\lambda}$ is empty, which generally occurs when $Q_{\hat{a}\hat{a}}$ is extremely precise. In such rare cases, LAMBDA 4.0 will still compute a BIE solution based on multiple integer candidates surrounding the ILS pull-in region. Hence, the ILS estimator is adopted with $p = 1 + 2(2^n - 1)$ best solutions[7] and all integer candidates are then weighted accordingly. This alternative procedure for the empty set scenario $\Theta_{\hat{a}}^{\lambda} = \emptyset$, assures that a BIE solution is always made available.

3.2.8.2. PAR (BIE) estimator based on ADOP criterion

The BIE estimator is optimal in the MSE sense and outperforms both float and ILS-fixed solutions. However, it is also relatively computational expensive, especially when working with a large dimensionality n and/or when involving some ambiguity components that are not sufficiently precise. As discussed in Section 3.2.8, given a low ambiguity precision, the BIE estimator is expected to perform similarly to the float solution; thence this might not always justify the large computation burden involved.

In LAMBDA 4.0, a new partial AR strategy is introduced for BIE, where the objective is however still to work with the largest subset possible. In fact, we know that BIE will always outperform the float solution and does not require a minimum SR for improving the precision of real valued parameters. This PAR approach is therefore no more based on a minimum SR criterion, but on a maximum (expected) computational effort. Hence, this is based on decorrelated ambiguities, which are already sorted by precision and will certainly increase the efficiency of the search over candidate integer vectors.

WARNING: this PAR approach shall **not** be used with estimators other than BIE, as the latter is the only one proved to *always* outperform the float solution in the MSE sense.

The PAR criterion for BIE considers incremental subsets of the decorrelate ambiguity, and it computes the Ambiguity Dilution of Precision (ADOP, see Section 3.3.4) for a subset

$$\mathrm{ADOP}(k) = \left(\prod_{i=k}^{i=n} \sigma_{\hat{z}_{i|I}} \right)^{\frac{1}{n+1-k}}, \qquad I = \{i+1, .., n\} \tag{81}$$

where $\sigma_{\hat{z}_{i|I}}$ is the conditional standard deviation, which is found via a $\mathrm{L^T DL}$-decomposition, e.g. see Eq.(24), of the variance-covariance matrix for the decorrelated ambiguities.

[7] This value is defined based on Minkowski's theorem on the upper bound for the number of facets of a n-parallelohedron. This number does not exceed $2(2^n - 1)$, where the facet can represent the interface with nearby pull-in regions in $\mathbb{R}^n$. See Lehrsatz VI (Theorem VI) by Minkowski (1897).

Hence, from 'last-to-first' we compute

$$\begin{aligned}
&\mathrm{ADOP}(n) = \sigma_{\hat{z}_n},\\
&\mathrm{ADOP}(n-1) = \left(\sigma_{\hat{z}_{n-1|n}}\sigma_{\hat{z}_n}\right)^{\frac{1}{2}},\\
&\mathrm{ADOP}(n-2) = \left(\sigma_{\hat{z}_{n-2|n-1,n}}\sigma_{\hat{z}_{n-1|n}}\sigma_{\hat{z}_n}\right)^{\frac{1}{3}}\\
&\vdots\\
&\mathrm{ADOP}(1) = \left(\prod_{i=1}^{i=n}\sigma_{\hat{z}_{i|I}}\right)^{\frac{1}{n}}, \qquad I = \{i+1,..,n\}
\end{aligned} \tag{82}$$

and, following Teunissen et al. (1997), we can relate this ADOP quantity[8] to the volume V_q of the hyper-ellipsoid in $\mathbb{R}^q$ with $q = n + 1 - k$. This relation is given as follows

$$\mathrm{V}_q = \chi_\alpha^q U_q \cdot [\mathrm{ADOP}(n+1-q)]^q \tag{83}$$

where U_q is the volume of the unit ball in $\mathbb{R}^q$, i.e.

$$\mathrm{U}_q = \frac{\pi^{\frac{q}{2}}}{\Gamma\left(\frac{q}{2}+1\right)} \tag{84}$$

while $\Gamma(x)$ is the gamma function, which can also be computed using a recursive relation $\Gamma(x+1) = x\Gamma(x)$, given the initial values $\Gamma(1) = 1$ and $\Gamma(1/2) = \sqrt{\pi}$.

The computation of ADOP in a subset of the ambiguity was already proposed in Section 3.4 from Teunissen and Odijk (1997), however in this context it is adopted to approximate the computational effort foreseen for a BIE solution in different most precise subsets of the decorrelated ambiguities. Similarly to Section 3.2.4.1, the PAR (BIE) solution becomes

$$\check{\mathbf{z}}_{\mathbf{PAR(BIE)}} \overset{\text{def}}{=} \begin{pmatrix} \hat{\mathbf{z}}_{\mathbf{1|2}} \\ \check{\mathbf{z}}_{\mathbf{2,BIE}} \end{pmatrix} \tag{85}$$

with $\hat{\mathbf{z}}_{\mathbf{1|2}} = \hat{\mathbf{z}}_{\mathbf{1}}\left(\check{\mathbf{z}}_{\mathbf{2,BIE}}\right) \in \mathbb{R}^{n-q}$ and $\check{\mathbf{z}}_{\mathbf{2,BIE}} \in \mathbb{R}^q$ that is also a real-valued vector, i.e. output of the BIE estimator in that q-dimensional subset (i.e. of most precise ambiguities). At this point we can briefly summarize the computation steps for the PAR (BIE) solution.

[8] Note that for the full set we have $\mathrm{ADOP}(1) \equiv \mathrm{ADOP}$, as being originally defined in (ibid).

COMPUTATION

The PAR (BIE) solution starts considering $q = 1$ for $k = n$, so we compute the volume V_1 in Eq.(83) based on ADOP(n). As the volume is taken as an approximation of the number of integer candidates, it also provides an estimation of the expected computational effort for a BIE solution $\check{\mathbf{z}}_{\mathbf{2,BIE}} \in \mathbb{R}^q$, thus requiring an enumeration of all the possible integer vectors within that q-dimensional subset. This expected effort can be checked without any actual BIE computation, and we iterate $q = 2, q = 3, ...$, till the point where the expected number of integers is above a certain user-selected threshold.

Given that decorrelated ambiguities are sorted from most to least precise, we expect the ADOP$(n + 1 - q)$ to increase for larger q values. For $q = n$, we have ADOP$(1) =$ ADOP, i.e. the Ambiguity Dilution of Precision over the entire set, and a nominal BIE solution is being computed. At each q-th iteration, the dimensionality changes as also $\chi^2_\alpha(q)$ given a certain $\alpha \ll 1$. This is the Chi-squared (central) distribution with q degrees of freedom for defining the size of the q-dimensional ellipsoid that we are considering.

In Figure 20, an illustrative example is given on the relation between 'ADOP' and 'number of integer candidates' from BIE computations using $\alpha = 10^{-6}$. Hence, we adopt different values of ADOP at different dimensionality $n = 1, \dots, 8$. When comparing the volume of the ellipsoid with the number of integers used in the BIE (left plot), it becomes visible that this approximation becomes accurate for larger ellipsoids, e.g. above 100 candidates. This result is consistent with (ibid), and can been used for this PAR (BIE) implementation.

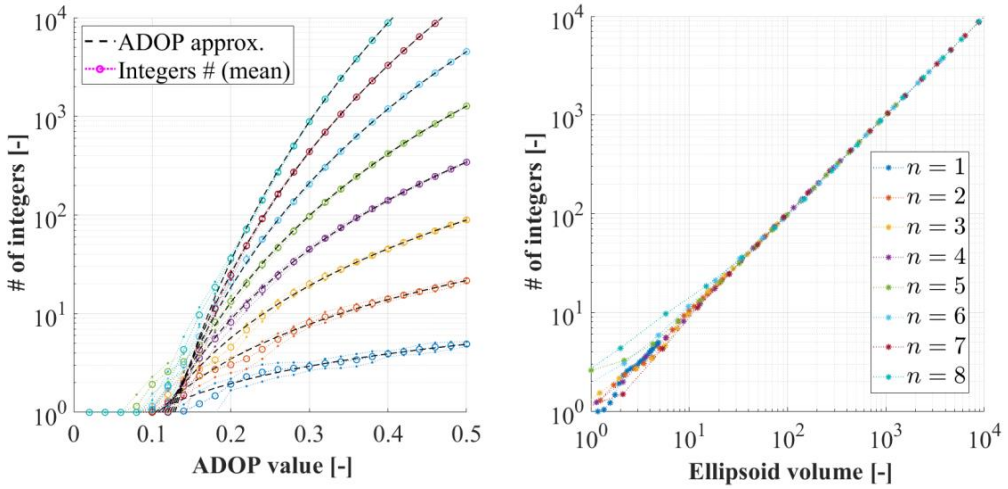


Figure 20: the # of integer used in BIE ($\alpha = 10^{-6}$) is shown on the y-axis, based on different ADOP values (left plot) and with respect to the associated ellipsoid volume (right plot). Note that the mean number of integers has been computed here based on 500 simulations.

3.3. Ps-LAMBDA functionalities

The overview of Ps-LAMBDA functionalities is given in Figure 21.

Ps-LAMBDA	Description	IR	VIB-IR	IB	VIB-ILS	ILS
IBexact	**Integer Bootstrapping** (exact SR, analytical)	UB	UB	exact	LB	LB
ADOPapprox	**ADOP method** (approximation)	UB	UB	UB	approx.	approx.
LB_Variance	**Variance method** (Lower Bound)	LB	LB	LB	LB	LB
UB_ADOP	**ADOP method** (Upper Bound)	UB	UB	UB	UB	UB
LB_Eigenvalue	**Eigenvalue method** (Lower Bound)	-	-	-	-	LB
UB_Eigenvalue	**Eigenvalue method** (Upper Bound)	UB	UB	UB	UB	UB
LB_Pullin	**Pull-in region method** (Lower Bound)	-	-	-	-	LB
UB_Pullin	**Pull-in region method** (Upper Bound)	UB	UB	UB	UB	UB
Numerical	**Numerical** (simulation-based)	approx.	approx.	approx.	approx.	approx.

Figure 21: Overview of all Ps-LAMBDA functionalities available in this new v4.0 toolbox.

The LAMBDA 4.0 toolbox implements several methods for statistical evaluation, based on analytical expressions, lower/upped bounds and approximations for the success rate. For what concerns numerical simulations, those are available for the I-estimators, as well as for IA-estimators, where we compute both success rate and failure rate.

In the next sections we will briefly review the implemented methods, i.e.

- Analytical IB formulation | Section 3.2.1
- ADOP approximation | Section 3.3.2
- Lower Bound by Variance method | Section 3.3.3
- Upper Bound by ADOP method | Section 3.3.4
- Lower Bound by Eigenvalue method | Section 3.3.5
- Upper Bound by Eigenvalue method | Section 3.3.6
- Lower Bound by Pull-in region method | Section 3.3.7
- Upper Bound by Pull-in region method | Section 3.3.8
- Numerical simulations | Section 3.3.9

while we start with a short review of some definitions adopted in the next sections.

We assume the float ambiguity vectors $\hat{\boldsymbol{a}} \in \mathbb{R}^n$ to be normally distributed around $\boldsymbol{a} \in \mathbb{Z}^n$, which is the true but unknown integer vector. The Probability Density Function (PDF) is

$$f_{\hat{a}}(\boldsymbol{x}|\boldsymbol{a}) = \frac{1}{\sqrt{\det(2\pi Q_{\hat{a}\hat{a}})}} \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{a}\|^2_{Q_{\hat{a}\hat{a}}}}{2}\right) \tag{86}$$

where $Q_{\hat{a}\hat{a}} \in \mathbb{R}^{n\times n}$ is the vc-matrix of the float ambiguities.

At this point, we define:

- **Success Rate (SR)**

The success rate P_s is the probability of correctly fixing the ambiguity, i.e. it is equal to the probability that $\hat{\boldsymbol{a}}$ resides in the aperture pull-in region Ω_a (Verhagen, 2005), which refers to the integer vector $\boldsymbol{a}$**,** and this SR probability can be computed as

$$P_s = \int_{\Omega_a} f_{\hat{a}}(\boldsymbol{x}|\boldsymbol{a})\boldsymbol{dx} \tag{87}$$

where Ω_a depends on the estimator being selected.

- **Failure Rate (FR)**

The failure rate P_f is the probability of wrongly fixing the ambiguity, so as the probability that the float solution is fixed to a wrong integer. In this case, $\hat{\boldsymbol{a}}$ resides in aperture pull-in regions Ω_z for $\boldsymbol{z} \in \mathbb{Z}^n$ with $\boldsymbol{z} \neq \boldsymbol{a}$. This FR probability can be computed as

$$P_f = \sum_{\boldsymbol{z}\in\mathbb{Z}^n\setminus\{\boldsymbol{a}\}} \int_{\Omega_z} f_{\hat{a}}(\boldsymbol{x}|\boldsymbol{a})\boldsymbol{dx} \tag{88}$$

with the PDF still centered on the true, but unknown, integer ambiguity vector $\boldsymbol{a}$.

- **Undecided Rate (UR)**

The undecided rate P_u is the probability that the ambiguity is not fixed to an integer vector, and therefore a float solution is returned. This UR probability can be computed as

$$P_u = 1 - (P_s + P_f) \equiv 1 - \sum_{\boldsymbol{z}\in\mathbb{Z}^n} \int_{\Omega_z} f_{\hat{a}}(\boldsymbol{x}|\boldsymbol{a})\boldsymbol{dx} \tag{89}$$

where for I-estimators this probability is exactly zero, i.e. $P_u = 0$, as a consequence of the so-called 'no gaps' condition introduced in Eq.(18). For IA-estimators, $P_u \geq 0$ and this is the complement of the fix-probability, i.e. probability that the outcome is an integer, which is not always guarantee in the IA-class since $P_s + P_f = 1 - P_u \leq 1$.

An illustration is given in Figure 22, for both I-class and IA-class, based on a 2D example, where the ILS pull-in regions have been shown for reference in blue color.

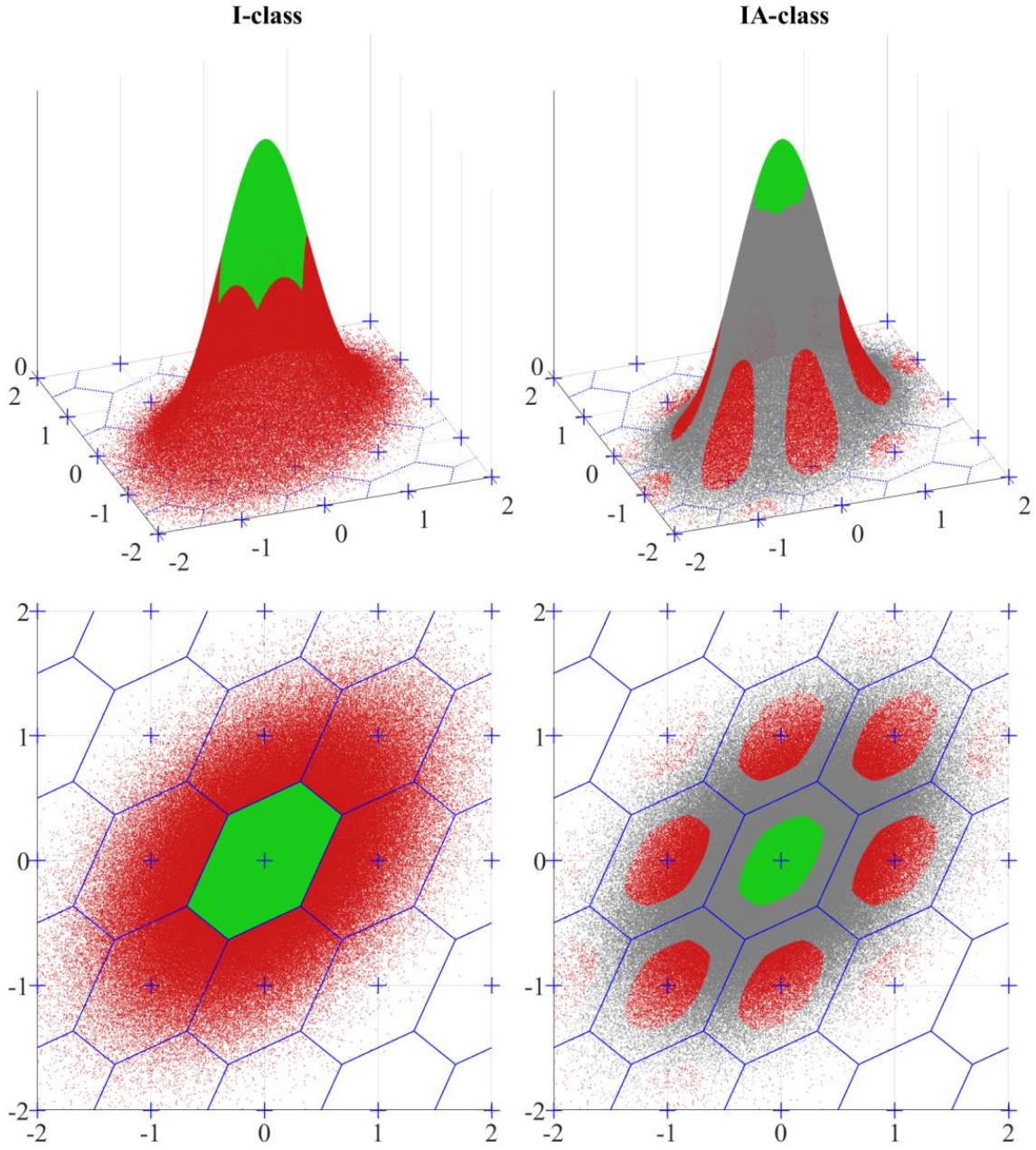


Figure 22: An illustration of PDFs with colored regions (in 2D and 3D) for successful fixes (in green), wrong fixes (in red) and float solutions (in grey). On the left and right sides, an example for Integer estimators and Integer Aperture estimators is shown, respectively.

Lastly, it is also convenient to introduce an additional diagnostic, i.e.

- **Successful-fix Rate (SfR)**

The successful-fix rate is the probability of a successful fixing, given as

$$P_{sf} \overset{\text{def}}{=} \frac{P_s}{P_s + P_f} = \frac{P_s}{1 - P_u} \tag{90}$$

where for I-estimators $P_{sf} = P_s$ since $P_s + P_f = 1$. We refer the reader to the example in Figure 15 relative to IA-estimators where this successful-fix rate has been shown based on a numerical simulation in 2D, while varying the aperture parameter.

3.3.1. Analytical IB formulation

The integer bootstrapping success rate can be evaluated exactly (Teunissen 1998):

$$P_{s,\text{IB}} = \text{P}(\breve{\boldsymbol{a}}_{\mathbf{IB}} = \boldsymbol{a}) = \prod_{i=1}^{i=n} \left[2\Phi\left(\frac{0.5}{\sigma_{\hat{z}_{i|I}}}\right) - 1 \right] \tag{91}$$

with $\boldsymbol{a} \in \mathbb{Z}^n$ and the cumulative normal distribution $\Phi: \mathbb{R} \to [0,1]$ is given as

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{t^2}{2}\right) dt \tag{92}$$

while $\sigma_{\hat{z}_{i|I}}$ is the conditional standard deviation of each i-th ambiguity component being conditioned onto previous ones, where $I = \{i+1, \dots, n\}$. This analytical formulation also shows, as already mentioned in Section 3.2.2, that a re-parametrization of ambiguities can lead to different results and it is generally related to different SR values.

Note that in LAMBDA 4.0 the triangular decomposition is considered with a convention 'last-to-first', where the last components are the most precise ones. This analytical result is also exploited in PAR estimator (see Section 3.2.4), as ordering of conditional variances allows defining a larger subset that satisfy the minimum SR criterion given in Eq.(44).

An example for a 1D case is shown in Figure 23, where the IB success rate is shown for different values $\sigma \in \mathbb{R}$, i.e. the unconditional standard deviation since $n = 1$. In general, it is visible how the success rate drops below 95% for $\sigma > 0.25$, and in the multidimensional case the product in Eq.(91) will quickly become even smaller.

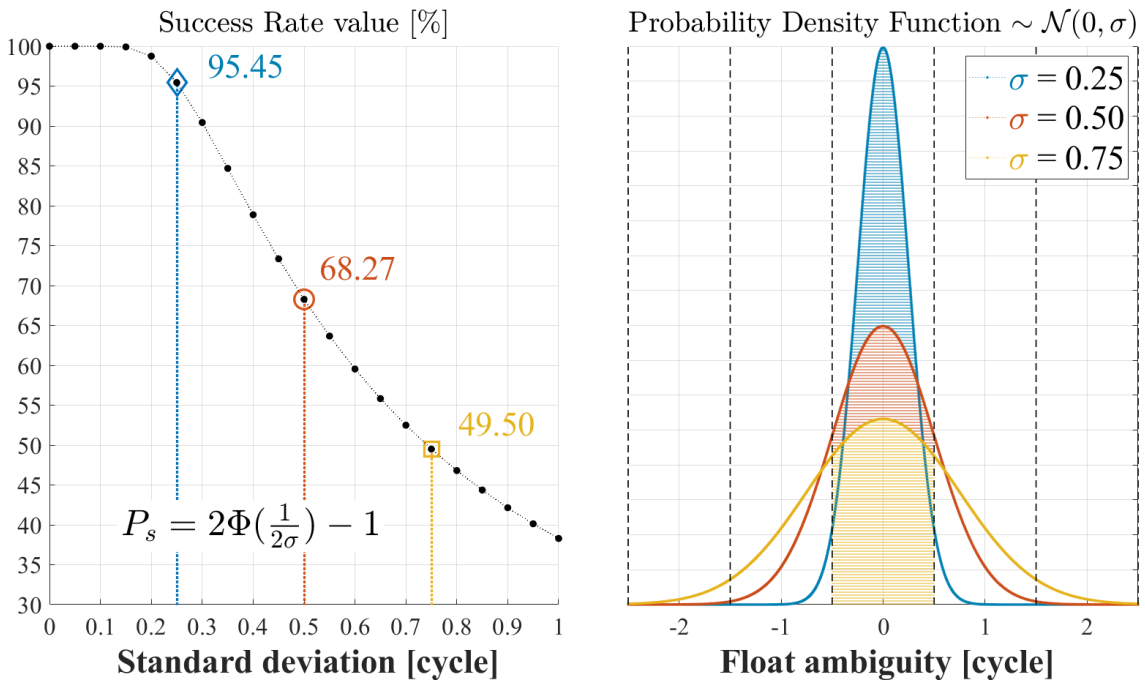


Figure 23: Illustration of SR for Integer Bootstrapping estimator given different precision $\sigma \in [0, 1]$ for the (scalar) ambiguity normally distributed around the *true* integer $a = 0$.

3.3.2. ADOP approximation

Given that the Z-transformation is volume preserving, for a re-parametrization $\hat{\boldsymbol{z}} = Z^T \hat{\boldsymbol{a}}$ and an admissible matrix $Z \in \mathbb{Z}^{n \times n}$ it follows that $|\det(Q_{\hat{a}\hat{a}})| = |\det(Q_{\hat{z}\hat{z}})|$. On the other hand, by exploiting the triangular decomposition in Section 3.1.1, we can write

$$|\det(Q_{\hat{a}\hat{a}})| = |\det(D_{\hat{a}})| \triangleq |\det(D_{\hat{z}})| = |\det(Q_{\hat{z}\hat{z}})| \tag{93}$$

with $D_{\hat{a}}$ and $D_{\hat{z}}$ being diagonal matrices, such that

$$\prod_{i=1}^{i=n} \sigma^2_{\hat{a}_{i|I}} \triangleq \prod_{i=1}^{i=n} \sigma^2_{\hat{z}_{i|I}}, \qquad I = \{i+1, .., n\} \tag{94}$$

that are the conditional variances for the original and decorrelated ambiguities. In this way, it is possible to define an upper bound (Teunissen, 2000) for the IB success rate described in the previous section.

Hence, starting from Eq.(91), we can define

$$P_{s,\mathrm{IB}} \leq P_{s,\mathrm{ADOP}} \stackrel{\mathrm{def}}{=} \left[2\Phi\left(\frac{0.5}{\mathrm{ADOP}}\right) - 1\right]^n \tag{95}$$

with the Ambiguity Dilution of Precision, or ADOP (Teunissen, 1997), given as

$$\mathrm{ADOP} = |\det(Q_{\hat{a}\hat{a}})|^{\frac{1}{2n}} \tag{96}$$

that captures the main characteristics of the ambiguity precision.

This quantity has been described in Teunissen and Odijk (1997) or Odijk and Teunissen (2008), and it is Z-invariant. The latter follows from the volume preserving properties of the Z-transformation, and allows very easily computing an upper bound for the IB success rate. We should be careful however, that this ADOP formulation is **not** an upper bound to ILS, but only for IB. A similar formulation for VIB estimators is provided in Teunissen et al. (2021), but is not considered here since dependent upon the specific partitioning being selected by the user. We refer to Lemma 7 in (ibid) for more mathematical details.

One main advantage in the adoption of the ADOP, is that the user might evaluate (before any decorrelation) if the SR is too small. In this case, it can be immediately concluded that IB will not be successful for *any* parametrization of the ambiguities. When the latter ones are perfectly decorrelated, the ADOP equals the geometric mean of the standard deviations of the ambiguities, therefore it can be considered as a measure of the average precision over the entire ambiguity subsets.

3.3.3. Lower Bound with Variance method

So far, we have seen that an exact SR formulation exists for the IB estimator, and an upper bound can also be found based on the ADOP formulation. Both these solutions represent an upper bound to IR estimators, as well as for $\mathrm{VIB}_{\mathrm{IR}}$, and therefore it seems convenient to define also possible lower bounds. In general, the n-fold integral over the IR pull-in region is difficult to evaluate, except for the case where $Q_{\hat{a}\hat{a}}$ is diagonal. In Teunissen (1998a), a SR lower bound for $Q_{\hat{a}\hat{a}}$ being non-diagonal is provided, such that

$$P_{s,\mathrm{IR}} = \mathrm{P}(\breve{\boldsymbol{a}}_{\mathbf{IR}} = \boldsymbol{a}) \geq \prod_{i=1}^{i=n}\left[2\Phi\left(\frac{0.5}{\sigma_{\hat{a}_i}}\right) - 1\right] \tag{97}$$

where we are now considering the unconditioned standard deviations $\sigma_{\hat{a}_i}, \forall i = 1, \ldots, n$ in the cumulative normal distribution function $\Phi\colon \mathbb{R} \to [0,1]$.

This previous lower bound also varies based on the ambiguity parametrization, while we know that it will always be smaller than the IB success rate computed in Eq.(91). When considering the SR performance ordering described in Figure 10, we can conclude that this variance method provides a lower bound for all I-estimators in LAMBDA 4.0, and in the limit $\sigma_{\hat{a}_i} \to 0, \forall i = 1, \ldots, n$, we have $P_{s,\mathrm{IR}} \to 1$, so the simplest IR solution might have a sufficient success rate value without the need of other more complex I-estimators.

3.3.4. Upper Bound with ADOP method

In addition to a lower bound for all I-estimators implemented in LAMBDA 4.0, we also have an upper bound for ILS, which is optimal in the I-class. It follows that such an upper bound holds for any estimator in this class and it has been introduced in Hassibi and Boyd (1998), whereas the proof has been given later in Teunissen (2000). It follows that

$$P_{s,\mathrm{ILS}} \leq \mathrm{P}\left(\chi_n^2 \leq \frac{c_n}{\mathrm{ADOP}^2}\right), \qquad c_n = \frac{1}{\pi}\left[\frac{n}{2}\Gamma\left(\frac{n}{2}\right)\right]^{\frac{2}{n}} \tag{98}$$

with $\Gamma(\cdot)$ as the gamma function and χ_n^2 being the Chi-squared (central) distribution.

3.3.5. Lower Bound with Eigenvalue method

For what concerns the ILS estimator, is it possible to define other lower bounds other than using the exact IB solution. This is based on bounding the actual vc-matrix by a diagonal matrix, where the ILS success rate for such bounding matrix becomes straightforward, see Teunissen (1998a). The simplest way is to make use of the maximum eigenvalue $\lambda_{\max}$, i.e.

$$Q_{\hat{a}\hat{a}} \leq \lambda_{\max} I_n \tag{99}$$

where I_n is the identity matrix of order n. A lower bound for ILS follows as

$$P_{s,\mathrm{ILS}} \geq P_{\mathrm{LB}}^{\mathrm{eig}} \overset{\mathrm{def}}{=} \left[2\Phi\left(\frac{0.5}{\sqrt{\lambda_{\max}}}\right) - 1\right]^n \tag{100}$$

In general, this bound might be too loose if the hyper-ellipsoid is highly elongated, and for instance when the ambiguities are highly correlated. Still, following a decorrelation step, the elongation of the ambiguity search space is considerably reduced and this bound might become much sharper when using the eigenvalues of the transformed vc-matrix.

3.3.6. Upper Bound with Eigenvalue method

The previous same procedure might be employed to find an upper bound based on the use of the minimum eigenvalue $\lambda_{\min}$, so we consider

$$Q_{\hat{a}\hat{a}} \geq \lambda_{\min} I_n \tag{101}$$

and an upper bound for ILS follows as

$$P_{s,\mathrm{ILS}} \leq P_{\mathrm{UB}}^{\mathrm{eig}} \overset{\mathrm{def}}{=} \left[2\Phi\left(\frac{0.5}{\sqrt{\lambda_{\min}}} \right) - 1 \right]^n \tag{102}$$

Even if $Q_{\hat{a}\hat{a}}$ is diagonal, both lower and upper bounds being computed with the eigenvalue method might be too loose. In fact, the two bounds coincide only when the vc-matrix is a scaled unit matrix, i.e. $Q_{\hat{a}\hat{a}} = \lambda I_n$ for an arbitrary $\lambda > 0$. Note that in this particular case also the ADOP is trivial to be found, as it will directly be given by $\sqrt{\lambda}$.

In general, following the definition of ADOP, we can also show that

$$P_{\mathrm{LB}}^{\mathrm{eig}} \leq P_{s,\mathrm{ADOP}} \leq P_{\mathrm{UB}}^{\mathrm{eig}} \tag{103}$$

where we have seen that $P_{s,\mathrm{ILS}} \approx P_{s,\mathrm{ADOP}} \geq P_{s,\mathrm{IB}}$ for any parametrization. In Figure 24, this last relationship is illustrated with bounding ellipses for bounds based on the eigenvalue methods, along with an ADOP approximation (as given in Section 3.3.2).

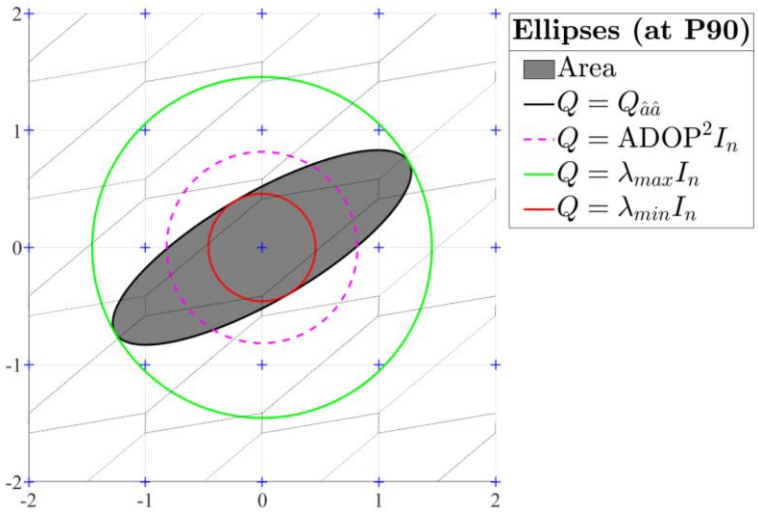


Figure 24: Illustration of the lower/upper bounding ellipses for the vc-matrix of ambiguities based on its minimum and maximum eigenvalue, along with the ADOP approximation. The ellipses are plotted at P90, i.e. containing samples with a 90% probability.

We should observe that while the upper bound based on the eigenvalues' approach holds for any estimator in the I-class, the lower bound refers *only* to the ILS estimator. After the decorrelation, the ratio between two extreme eigenvalues is pushed towards its minimum of one, therefore the two bounds will move closer to each other.

3.3.7. Lower Bound with Pull-in region method

Another approach for finding lower and upper bounds for the ILS success rate is presented in Teunissen (1998a), so bounding the integration (pull-in) region. For the lower bound, such region L_a is selected such that is completely contained by the original pull-in region, i.e. $\mathrm{L}_a \subset \mathcal{P}_{a,\mathrm{ILS}}$, and therefore it follows that

$$P_{s,\mathrm{ILS}} = \mathrm{P}\left(\hat{\boldsymbol{a}} \in \mathcal{P}_{a,\mathrm{ILS}}\right) \geq \mathrm{P}(\hat{\boldsymbol{a}} \in \mathrm{L}_a) \tag{104}$$

where this region should be chosen such that the corresponding probabilities are easily evaluated in practice. For instance, we can define an ellipsoidal region L_a to expand this pull-in region from the inside. The new region can be defined as

$$\mathrm{L}_a = \left\{ \boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{x} - \boldsymbol{a}\|^2_{Q_{\hat{a}\hat{a}}} \leq \frac{1}{4} \min_{\boldsymbol{u} \in \mathbb{Z}^n \setminus \{\boldsymbol{0}\}} \|\boldsymbol{u}\|^2_{Q_{\hat{a}\hat{a}}} \right\} \tag{105}$$

where this ellipsoid will depend on the vc-matrix $Q_{\hat{a}\hat{a}} \in \mathbb{R}^{n \times n}$. A graphical representation of this region for two different examples is given in Figure 25.

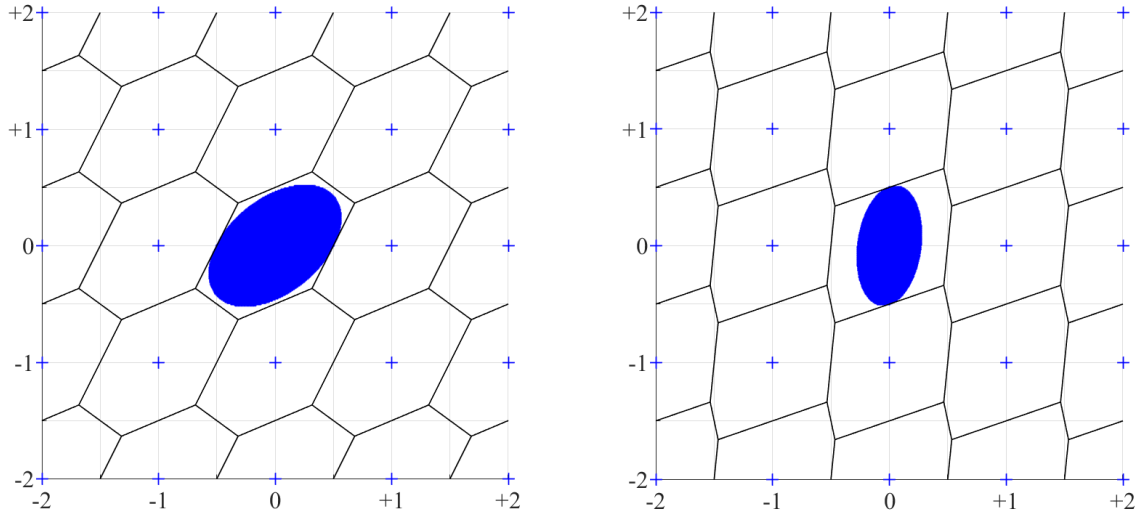


Figure 25: Illustration of two examples for the ellipsoidal region (blue) contained by the pull-in region $\mathcal{P}_{0,\mathrm{ILS}}$ based on two different variance-covariance matrices $Q_{\hat{a}\hat{a}} \in \mathbb{R}^{2\times 2}$.

The lower bound can therefore be evaluated based on

$$P_{\text{LB,region}} \overset{\text{def}}{=} \text{P}(\hat{\boldsymbol{a}} \in \text{L}_a) = \text{P}\left(\chi_n^2 \leq \frac{1}{4} \min_{\boldsymbol{u} \in \mathbb{Z}^n \setminus \{\boldsymbol{0}\}} \|\boldsymbol{u}\|_{Q_{\hat{a}\hat{a}}}^2\right) \tag{106}$$

where χ_n^2 refers to the chi-squared (central) distribution with n degrees of freedom.

3.3.8. Upper Bound with Pull-in region method

A class of upper bounds for ILS estimators has been introduced in Teunissen (1998a), i.e. based on a region U_a completely containing its pull-in region. Therefore, we consider

$$\text{P}_{\text{s,ILS}} = \text{P}\left(\hat{\boldsymbol{a}} \in \mathcal{P}_{a,\text{ILS}}\right) \leq \text{P}(\hat{\boldsymbol{a}} \in \text{U}_a) \tag{107}$$

where $\mathcal{P}_{a,\text{ILS}} \subset \text{U}_a$. From the expression of the ILS pull-in region in Eq.(34), we can define *any* finite intersection of $p < n$ banded subsets that will enclose $\mathcal{P}_{a,\text{ILS}}$, so having

$$\text{U}_a = \left\{\boldsymbol{x} \in \mathbb{R}^n \mid \left|\frac{\boldsymbol{c_i^T} Q_{\hat{a}\hat{a}}^{-1}(\boldsymbol{x} - \boldsymbol{a})}{\|\boldsymbol{c_i}\|_{Q_{\hat{a}\hat{a}}}}\right| \leq \frac{1}{2}\|\boldsymbol{c_i}\|_{Q_{\hat{a}\hat{a}}}, \qquad i = 1, \ldots, p\right\} \tag{108}$$

with an illustration in 2D given in the following Figure 26. The latter shows two different banded subsets (in yellow and cyan), which intersect in a convex region (in green). This new region completely contains the pull-in region for the ILS estimator (in light green), but other choices for these banded subsets are still possible for different $\boldsymbol{c_i}$-vectors.

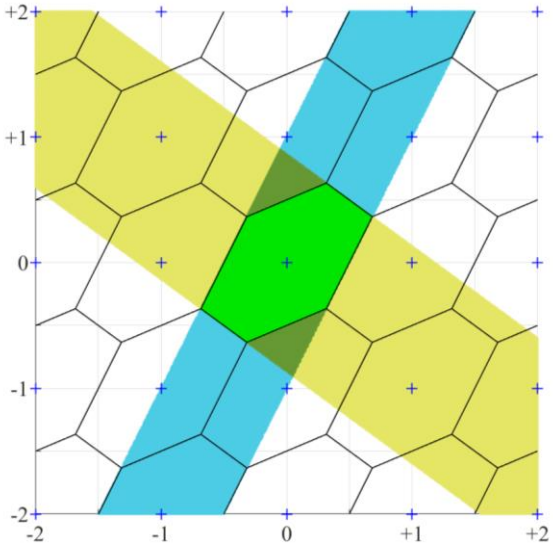


Figure 26: Illustration of the integration region (green), which completely contains the ILS pull-in region (light green), being defined here by the intersection of two banded subsets.

Given $p > 1$, by defining the p-vector $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_p)^T$ for

$$\nu_i = \frac{\boldsymbol{c_i}^T Q_{\hat{a}\hat{a}}^{-1}(\boldsymbol{x} - \boldsymbol{a})}{\|\boldsymbol{c_i}\|^2_{Q_{\hat{a}\hat{a}}}}, \qquad i = 1 \ldots p \tag{109}$$

then we can write

$$\mathrm{U}_a = \left\{ \boldsymbol{x} \in \mathbb{R}^n \mid \bigcap\nolimits_{i=1}^{i=p} |\nu_i| \leq \frac{1}{2} \right\} \tag{110}$$

and therefore $\mathrm{P}(\hat{\boldsymbol{a}} \in \mathrm{U}_a)$ equals the probability that the 'component-wise rounding' of this vector $\boldsymbol{\nu} \in \mathbb{R}^p$ produces the zero vector.

Given that this probability cannot be evaluated exactly, the latter can still be bounded from above using an IB upper bound of integer rounding (Teunissen, 1998b), such that

$$P_{\mathrm{UB,region}} \overset{\mathrm{def}}{=} \prod_{i=1}^{i=p} \left[2\Phi\left(\frac{0.5}{\sigma_{\nu_{i|I}}} \right) - 1 \right] \geq \mathrm{P}(\hat{\boldsymbol{a}} \in \mathrm{U}_a) \tag{111}$$

with the conditional standard deviations $\sigma_{\nu_{i|I}}$ of vector $\boldsymbol{\nu}$. These are equal to the square root of the diagonal elements of D_ν from a $\mathrm{L^TDL}$-decomposition of $Q_{\nu\nu}$, whose entries are

$$\sigma_{\nu_i \nu_j} = \frac{\boldsymbol{c_i}^T Q_{\hat{a}\hat{a}}^{-1} \boldsymbol{c_j}}{\|\boldsymbol{c_i}\|_{Q_{\hat{a}\hat{a}}} \|\boldsymbol{c_j}\|_{Q_{\hat{a}\hat{a}}}}, \qquad \boldsymbol{c_i}, \boldsymbol{c_j} \in \mathbb{Z}^n \tag{112}$$

where $\boldsymbol{c_i}, i = 1, \ldots, p$, need to be linearly independent to guarantee the full-rank of $Q_{\nu\nu}$.

Hence, Eq.(111) and Eq.(112) provide a class of upper bounds as the user still has freedom in choosing the independent set of $p \leq n$ integer vectors $\boldsymbol{c_i} \in \mathbb{Z}^n$, see (Teunissen, 1998a) or (Verhagen, 2005, p.41). The simplest choice would be to choose them as the standard (canonical) basis, i.e. as the n columns of the unit matrix I_n.

In that particular case, the required conditional variance will simplify to

$$\sigma^2_{\nu_{i|I}} = \frac{1}{\|\boldsymbol{c_i}\|^2_{Q_{\hat{a}\hat{a}}} \sigma^2_{\hat{a}_{i|I}}}, \qquad i = 1, \ldots, p \tag{113}$$

where we recall the conditional standard deviations $\sigma_{\nu_{i|I}}$ already used in Eq.(111).

3.3.9. Numerical simulations

The computation of success and failure rates can be approximated by means of numerical simulations. This Monte Carlo method assumes the float solution $\widehat{\boldsymbol{a}}$ as normally distributed around the true (but unknown) integer vector $\boldsymbol{a}$, i.e. $\widehat{\boldsymbol{a}} \sim \boldsymbol{\mathcal{N}}(\boldsymbol{a}, Q_{\hat{a}\hat{a}})$**.** Given the symmetry about the mean $\boldsymbol{a}$, we can make use of the 'integer remove-restore' property and directly draw samples from $\boldsymbol{\mathcal{N}}(\mathbf{0}, Q_{\hat{a}\hat{a}})$. This procedure can be adopted either before or after the decorrelation of ambiguities, whereas for many estimators this leads to different results.

In LAMBDA 4.0, we generate n independent random samples with a univariate standard normal distribution $\mathcal{N}(0,1)$, then collected in a vector $\boldsymbol{s} \in \mathbb{R}^n$. By computing a Cholesky factor $G \in \mathbb{R}^{n\times n}$ as $Q_{\hat{a}\hat{a}} = GG^T$, it becomes then possible to transform $\widehat{\boldsymbol{a}} = G\boldsymbol{s}$, where we have $\widehat{\boldsymbol{a}} \sim \mathcal{N}(\mathbf{0}, Q_{\hat{a}\hat{a}})$. Note that a certain number of samples $N > 0$ could be specified by the user, nevertheless can also be computed following Section 5.1 in Teunissen (1998a) or Section 3.4 in Teunissen (2001b). A minimum number of samples $N_{\min}$ is defined for the approximation precision, as briefly discussed here.

The number of samples required to have an accurate simulation of success rate is based on a Chebyshev inequality (or Bienaymé–Chebyshev inequality), i.e.

$$\mathrm{P}\left(\left|\frac{N_0}{N} - P_0\right| > \epsilon_{\mathrm{SR}}\right) < \frac{P_0(1-P_0)}{N\epsilon_{\mathrm{SR}}^2} \tag{114}$$

where the term on the left side is the probability that the relative frequency N_0/N differs more than ϵ_{SR} from $P_0 = \mathrm{P}(\breve{\boldsymbol{a}} = \mathbf{0})$, the latter being the probability of correctly fixing or success rate. After setting this probability to a small value, e.g. $\mathrm{P}_{\max} = 0.01$, it is possible to use the previous inequality to find the minimum number of samples being required as

$$N_{\min} = \left\lceil \frac{P_0(1-P_0)}{\mathrm{P}_{\max} \cdot \epsilon_{\mathrm{SR}}^2} \right\rceil \tag{115}$$

where P_0 is approximated by the IB success rate[9]. At this point we can provide a numerical example that shows the minimum number of samples requires for each success rate value.

In Table 1, assuming $\mathrm{P}_{\max} = 0.01$ and $\epsilon_{\mathrm{SR}} = 0.001$, we note that for a SR equal to 0.999 and 0.99 we would need $100'000$ and $990'000$ samples, respectively. When considering a success rate $P_0 = 95\%$, this value further increases to almost five million samples, which is around 50 times larger than for $P_0 = 99.9\%$.

[9] The maximum of this function is found for $P_0 = 0.5$, whereas $N_{\min} \to 0$ for $P_0 \to 0$ or $P_0 \to 1$.

Table 1: Minimum number of samples needed for different success rates' approximations

$P_0 =$	50 %	90 %	95 %	99 %	99.9 %
$N_{\min} =$	$25 \cdot 10^6$	$9 \cdot 10^6$	$\sim 4.8 \cdot 10^6$	$990 \cdot 10^3$	$\sim 100 \cdot 10^3$

A further numerical example is given in Figure 27, where we consider an IB success rate of 99.331%, and we compare the exact IB success rate value with the one obtained via numerical simulations. For thirty independent instances, a number N of samples is used to approximate the SR, showing how a better approximation is possible when number of samples increases. From Eq.(115), while using $\mathrm{P}_{\max} = 0.01$ and $\epsilon_{\mathrm{SR}} = 0.001$, it follows that $N_{\min} \cong 665'000$, which is consistent with the numerical results found.

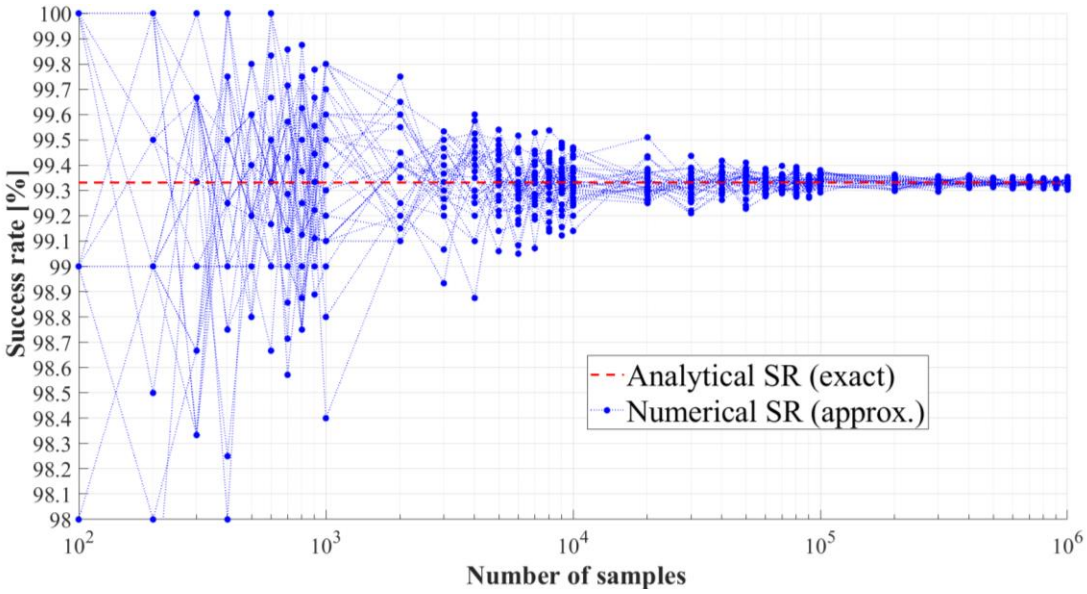


Figure 27: Graphical illustration of IB success rate approximation using different numbers of samples, and independently repeated over thirty multiple instances.

3.3.9.1. Monte Carlo approximation for SR and FR

In LAMBDA 4.0, both success and failure rates can be approximated. For I-estimators we know that $P_s + P_f = 1$, therefore FR is obtained using the SR numerical approximation, but this is not possible for IA-estimators, since we also have an undecided region, such that $P_s + P_f < 1$ and both SR and FR need to be found. In this new LAMBDA 4.0 release, the numerical simulations are made available for all estimators except for BIE, where there are no pull-in regions and its 'fixed' solution is de facto real valued.

In the case of I-estimators, we directly consider

$$P_s \approx \frac{N_s}{N}, \qquad P_f \approx \frac{N_f}{N} = \frac{N - N_s}{N} \tag{116}$$

where N_s is the number of float samples that are fixed by a selected I-estimator to the zero vector, which is assumed to be the true integer $\boldsymbol{a} \in \mathbb{Z}^n$.

In the case of IA-estimators, the previous expressions can still be used, but this time we have a different definition for N_s and N_f. In fact, we count the occurrences for which the output is an integer, and then whether it is fixed to $\boldsymbol{a} \in \mathbb{Z}^n$ or $\boldsymbol{z} \in \mathbb{Z}^n$ with $\boldsymbol{z} \neq \boldsymbol{a}$. Hence, we generally have $N_s + N_f < N$, while the user can also retrieve P_u or eventually compute the Successful-fix Rate (SfR), as being defined in Section 3.3.

All these routines have been optimized for being efficient in higher dimensional problems, where the enhanced search performances discussed in Section 3.2.3.2 would further benefit the numerical computations involved for these approximations. In all cases, the adoption of the ambiguity decorrelation before the numerical simulation is highly suggested.

4. Practical applications

The LAMBDA 4.0 toolbox provides an efficient implementation of different methods in the framework of 'LAMBDA Theory', and it aims to make available an extensive set of mathematical tools in order to tackle generic mixed-integer estimation models (see Section 2.1). For these models, the successful resolution of the unknown ambiguity parameters, which are subject to integer constraints, is fundamental.

Each specific problem might require an *ad hoc* solution, therefore LAMBDA 4.0 toolbox offers several standalone solutions that can be configured by the users, so easily adapting the current functionalities to existing software. Problems that can make use of LAMBDA are not limited to GNSS applications, but several non-GNSS applications have also shown the potentiality of LAMBDA method for ambiguity resolution.

Some ambiguity resolution problems arise in **GNSS applications**, e.g.

- precise point positioning (Teunissen, 2020b);
- integer cycle-slips resolution (Teunissen and Bakker, 2015);
- time and frequency transfer (Mi et al., 2023);
- atmosphere remote sensing (Lu et al., 2018);
- carrier-phase attitude determination (Giorgi, 2011);
- relative navigation for S/C formation flying (D'Amico and Montenbruck, 2010);
- …;

meanwhile many others can be found in **non-GNSS applications**, e.g.

- detection in MIMO communication systems (Damen et al., 2003).
- phase observations for InSAR deformation monitoring (Teunissen, 2006);
- use of acoustic waves for underwater navigation (Viegas and Cunha, 2007);
- fringe phase observations from VLBI (Hobiger et al., 2009);
- …;

and these are only some of the potential applications for the new LAMBDA 4.0 toolbox, while its rigorous mathematical framework is generalized to *any* mixed-integer model, as previously described in Section 2.

At this point, in the next sections, we can consider a few illustrative examples in order to demonstrate LAMBDA 4.0 capabilities, thus focusing on both "estimation and validation" (with LAMBDA), along with the "statistical evaluation" (with Ps-LAMBDA).

4.1. LAMBDA examples: estimation and validation

The LAMBDA 4.0 provides a versatile toolbox with many estimators, which are suitable for different problems. With LAMBDA main script (see Figure 28), the user has access to three different classes of estimators, which enable both estimation and validation in the integer ambiguity resolution process.

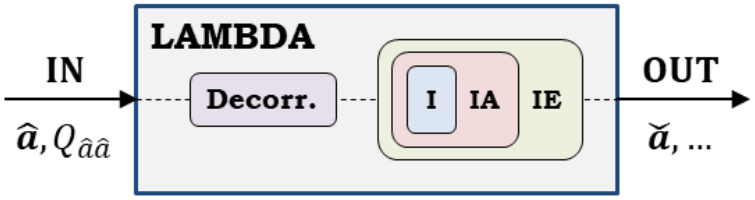


Figure 28: Overview of LAMBDA main script supporting estimators in the I-/IA-/IE-class.

Still, different methods might have different characteristics, and therefore could be suitable for specific types of problems, based on mixed-integer models. A few examples will be given in the following sub-sections. These illustrative examples are taken from the body of knowledge, and they demonstrate the potentiality of the LAMBDA toolbox.

4.1.1. Example #1: high dimensional ambiguity resolution

The advent of denser network processing, along with the growth of multi-frequency/multi-GNSS constellations, has contributed to the increase in dimensionality for many ambiguity resolution problems. In Li and Teunissen (2011), an analysis with a network of 19 stations has been presented based on LAMBDA 3.0, showing its efficiency even when the number of ambiguity components is more than 100.

However, nowadays, much larger networks are being processed and even thousands of ambiguity components might be found. In this case, even the adoption of the new integer search-and-shrink strategy (Section 3.2.3.2) might result in a computational bottleneck. It follows that more flexible solutions should be considered, as discussed next in relation to Vectorial Integer Bootstrapping estimators (see Section 3.2.5).

The VIB estimators, introduced in Teunissen et al. (2021), are currently integrated in this LAMBDA 4.0 toolbox. For integer ambiguity resolution in network processing, common widelane/narrowlane techniques can be shown (ibid) to be just a particular case of the VIB estimators. This vectorial approach has been adopted in the context of global network processing, more precisely for Orbit Determination and Time Synchronization (ODTS). In fact, the VIB estimators are also in use by TU Graz in their GROOPS[10] software adopted for the third IGS reprocessing campaign, see Strasser (2022).

In Massarweh et al. (2021), the VIB_{ILS} estimator is compared with its scalar counterpart in a small global network processing (14 stations) for GPS+Galileo constellation. In Figure 29, an example of satellite orbit midnight discontinuity over a 91-day period is shown for what concerns the scalar IB (left) and the VIB (right) solution.

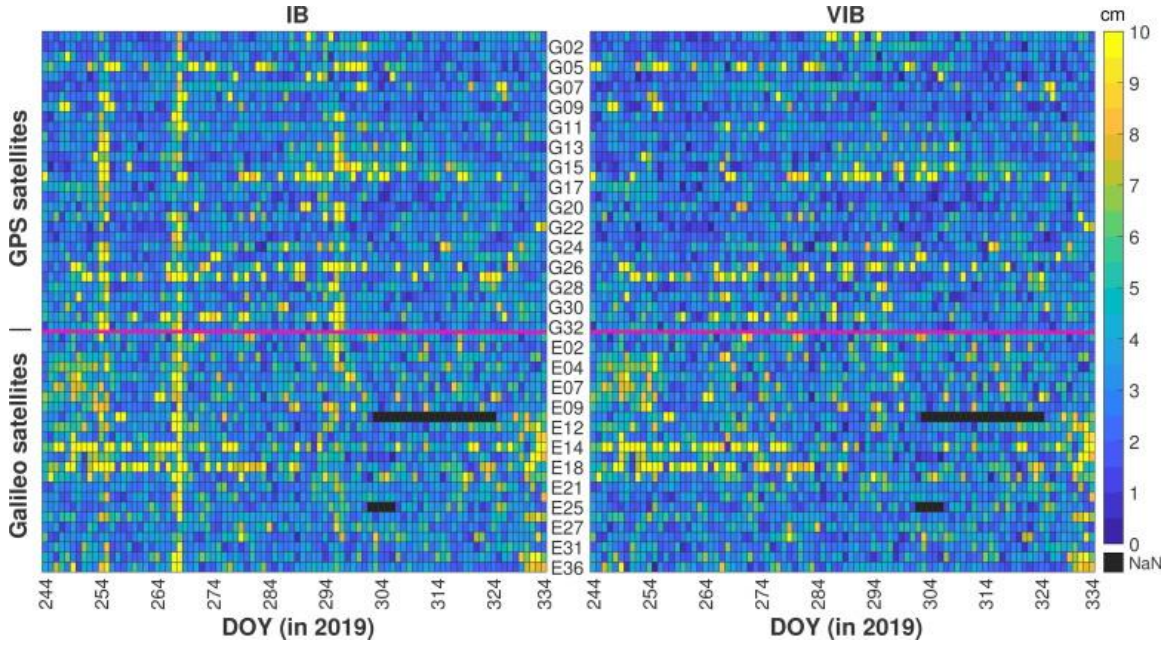


Figure 29: Example of GPS & Galileo satellite midnight discontinuity results over a 3-month campaign in 2019 based on a small global network of 14 stations. (ibid)

The VIB processing not only proved to be highly efficient, with thousands of ambiguities grouped in subsets of 200 components, fixed in fractions of a second, but it also provided more robust ODTS results validated over a 3-month period. Even larger problems could be tackled, given the flexibility in the VIB partitioning, i.e. selected by the user. We refer to Massarweh et al. (2021) for additional information.

[10] GROOPS stands for "Gravity Recovery Object Oriented Programming System software", which is available at https://github.com/groops-devs/groops. (Mayer-Guerr et al., 2021)

4.1.2. Example #2: controlled failure-rate

In Hou et al. (2016), new coefficients for the Fixed Failure-rate Ratio Test (FFRT) have been introduced. In this way it is possible to generate critical values according to an user-defined tolerable failure rate. This approach outperforms traditional methods where the critical value is directly defined by the user. An example from (ibid) is shown in Figure 30, where GPS dual-frequency data in a long baseline (182.7 km) is collected over one week, and is processed with a modified implementation of RTKLib (https://www.rtklib.com).

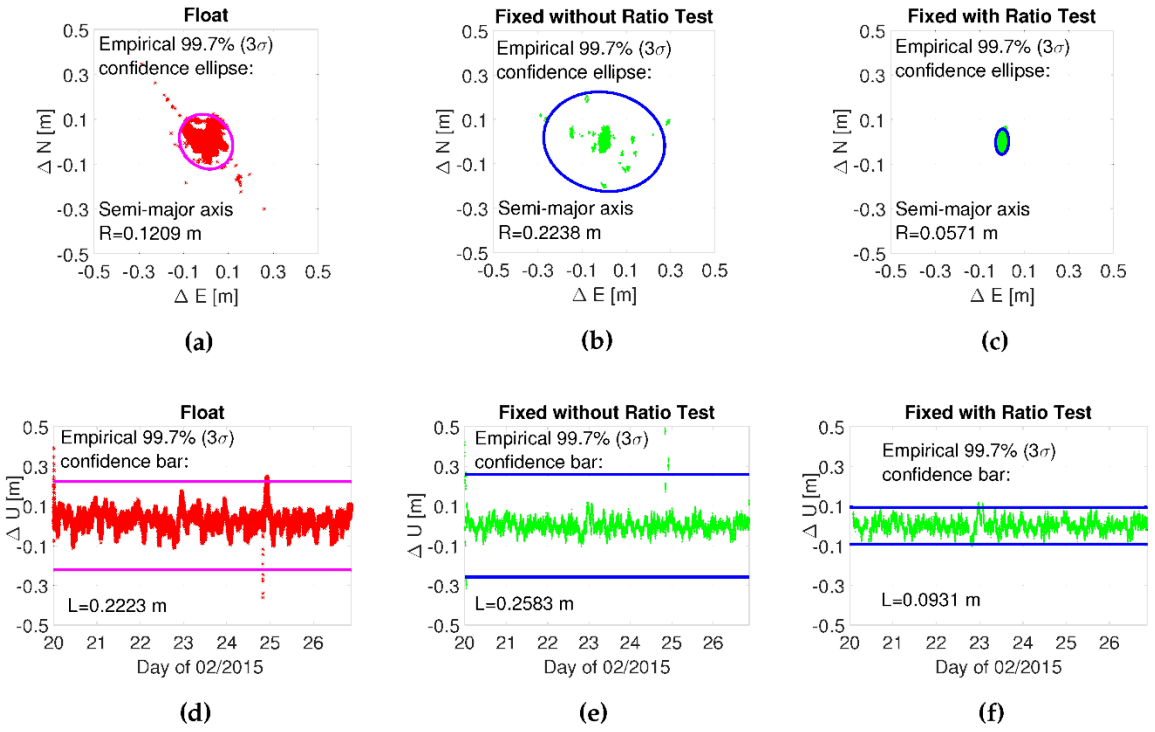


Figure 30: Positioning errors for the horizontal (top) and vertical (bottom) components, while considering a float solution, along with an ILS and IA-FFRT fixed solutions. (ibid)

The upper and lower plots show the horizontal and vertical errors, respectively, while the float solution (in red) is compared against two estimators (in green). The ILS fixed results without and with Ratio Test show a visible difference, where the test effectively prevents many incorrectly-fixed ambiguities.

In the context of short-baseline GPS performances, i.e. 11 km baseline data, the controlled FR of 0.1 % is adopted by Odijk et al. (2012) for instantaneous ambiguity resolution. The FFRT enables the user to have control of the rate of wrong fixes in contrast to traditional fixed ratio test values. Still, IA-FFRT estimator performances depend on the correctness of underlying functional and stochastic models.

Additional results are also presented in Odolinski et al. (2014), where LAMBDA FFRT is adopted to validate the resolved ambiguities in Real-Time Kinematics (RTK) positioning based on BDS+GPS multi-frequency data.

4.1.3. Example #3: on MSE-optimal solutions

The success and failure rates are not the only metrics in use for IAR, since estimators from the IE-class are actually not characterized by pull-in regions. In fact, for user positioning, a MSE criterion might be a very good choice for defining optimal solutions. The BIE, now available in LAMBDA 4.0 toolbox, is MSE-optimal in the IE-class, and it has been studied during the last two decades.

A more recent numerical example is given by Brack et al. (2023), where 2-epoch cm-level positioning is achieved using BIE (Figure 31) without external atmospheric corrections. These analyses by Brack et al. (2023) demonstrate the optimality of BIE, in particular with the objective to reduce the convergence time from tens of minutes to few epochs. This is possible especially for the horizontal component in absence of external corrections, and when combining multi-GNSS constellations.

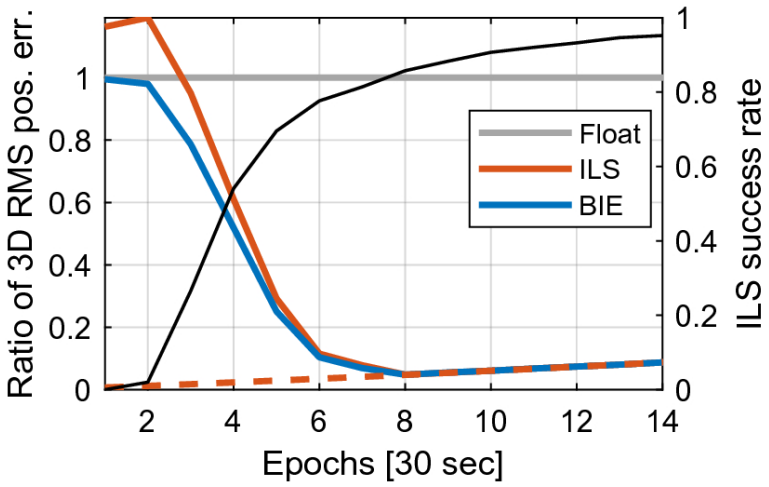
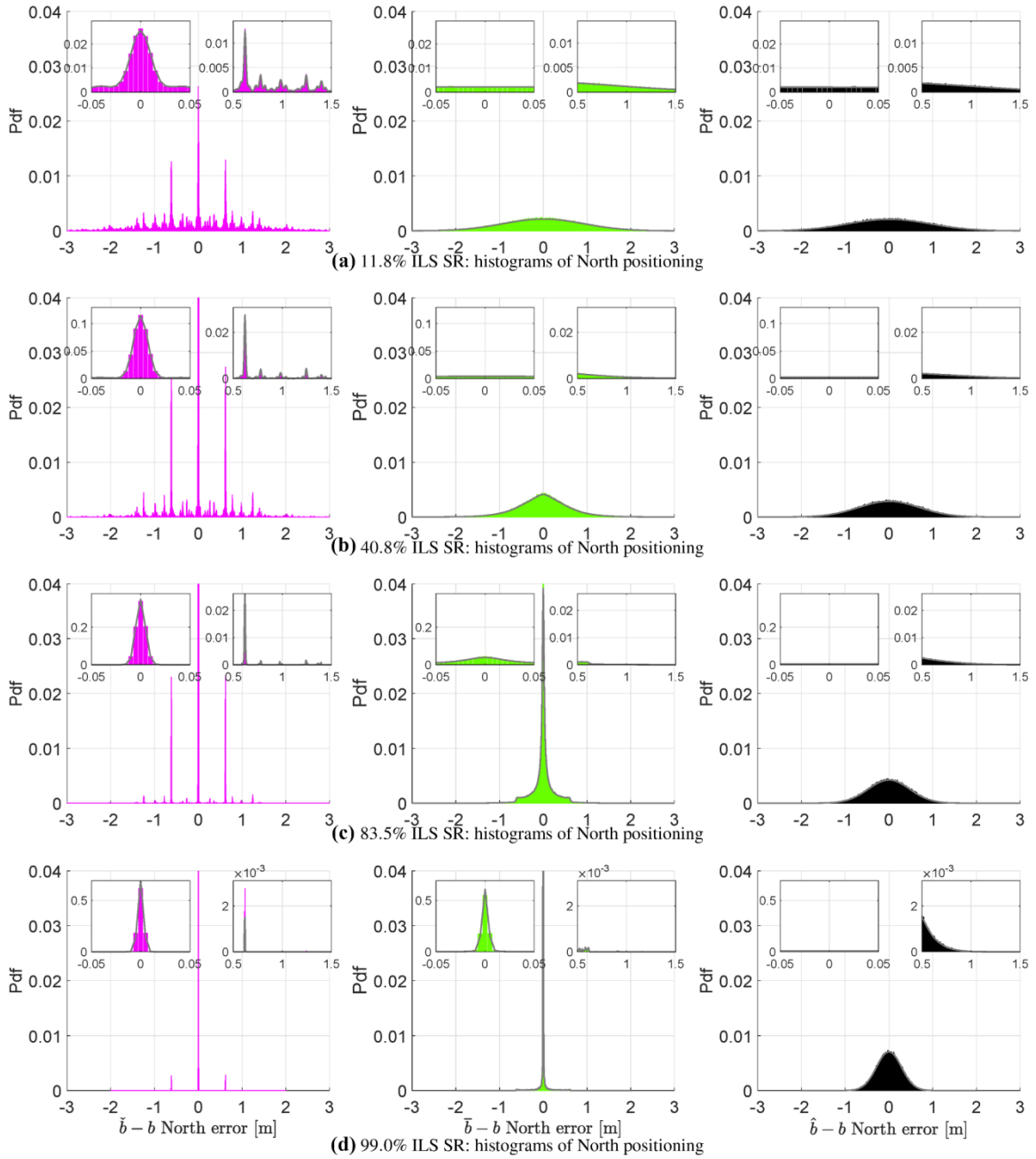


Figure 31: An example of simulated 3D RMS positioning errors relative to the float solution, where dashed line are the theoretically minimal values assuming the ambiguity known. (ibid)

Additional results are also presented in Odolinski and Teunissen (2020), where different baselines for RTK positioning are examined when adopting BIE. In Figure 32, an example of north positioning errors is given for different model strength, while comparing BIE (in green), ILS (in magenta) and Float (in back) solutions. As shown in Figure 19, for a very low or high success rate the BIE precision resembles the one of the Float or ILS solutions, respectively.



(a) 11.8% ILS SR: histograms of North positioning

(b) 40.8% ILS SR: histograms of North positioning

(c) 83.5% ILS SR: histograms of North positioning

(d) 99.0% ILS SR: histograms of North positioning

Figure 32: An illustration of histogram errors in RTK for the North positioning component is given for different model strengths, comparing ILS (magenta), BIE (green) and Float (black) solutions. (ibid)

Additional BIE-based implementations have been also proposed. For instance, Miao et al. (2024) presents a VIB-BIE for efficient and reliable IAR, thus combining the flexibility of vectorial methods, while using a BIE estimator in sequential ambiguity subsets.

4.2. Ps-LAMBDA examples: statistical evaluation

The LAMBDA 4.0 offers many relevant statistical measures for experimental and design purposed. With Ps-LAMBDA main script (in Figure 33), the user has access to different methods, which includes bounds and approximations for several estimators in I and IA classes. In addition, numerical approximations are possible, thus enabling the computation of SR and FR for the different estimators described in the previous section.

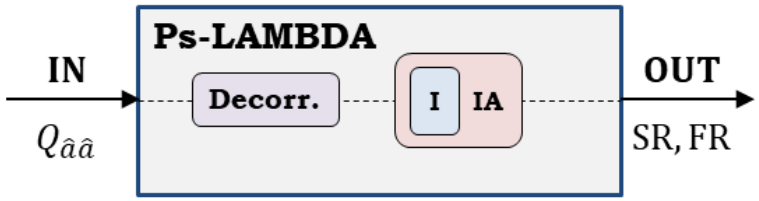


Figure 33: Overview of Ps-LAMBDA script supporting estimators in the I-class and IA-class.

For more information about the impact of decorrelation on SR bounds/approximations, we refer to Wang et al. (2016), where simulations are used to compare performances. At this point, we discuss here some possible applications of Ps-LAMBDA functionalities.

4.2.1. Example #1: assessment SR with biased models

An application of Ps-LAMBDA simulations is given in Li et al. (2014), where ambiguity resolution performances for biased model have been investigated in the presence of both tropospheric and ionospheric biases. An illustrative example is given in Figure 34.

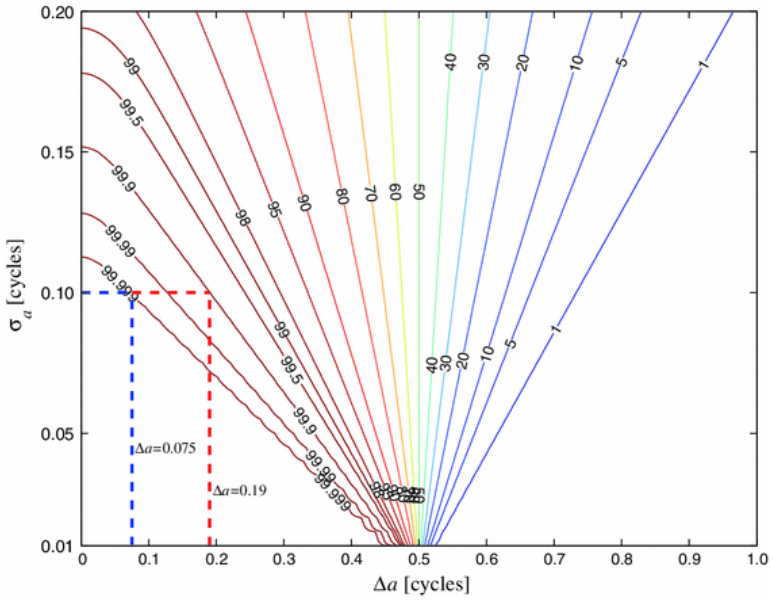


Figure 34: Success rate contours of bias-affected AR in 1D, given $\hat{a}^{b} \sim \mathcal{N}(\Delta a, \sigma_{\hat{a}}^{2})$, see (ibid).

A minimum success rate is set to $P_0 = 99.5$ %, and time-to-first-fix (TTFF) is the number of epochs needed to meet this requirement. An example for tropospheric biases from 0 to 3 cm is then provided in Figure 35 from (ibid), while considering a CORS baseline model (parameters are not estimated but held fixed), along with a static and a kinematic one. A few different ionospheric constraints are adopted, while the mean TTFF is illustrated with respect to the τ (tropospheric bias), where "u" refers to the unbiased model. We refer the reader to (ibid) for more details.

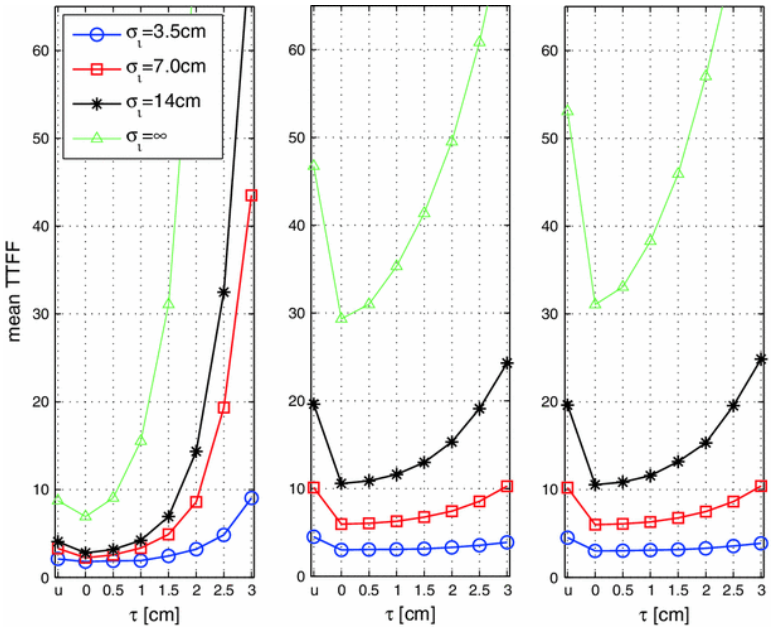


Figure 35: The mean Time-To-First-Fix (TTFF) is shown as function of the tropospheric bias τ between 0 and 3 centimeter, while considering different ionospheric constraints. From left to right, the iono-weighted CORS, static and kinematic baseline models, see (ibid).

4.2.2. Example #2: computation SR bounds and approximations

The Ps-LAMBDA functionalities enable to promptly compute several SR bounds and/or approximations based on a certain variance-covariance matrix $Q_{\hat{a}\hat{a}}$ given in input. In Wang et al. (2023), an example based on a dual-frequency GPS model is shown, starting with a standard deviation of 30 cm and 1 cm for code and phase observables, respectively. Then, a scaling factor F is adopted to analyze the ambiguity SR and boundary performance under different precisions, i.e. $Q^F_{\hat{a}\hat{a}} = FQ_{\hat{a}\hat{a}}$. In Figure 36, taken from (ibid), the different bounds and approximations are visible as function of $F \in [1,10]$, while the success rate for IR, IB and ILS estimators has been computed via numerical simulations.

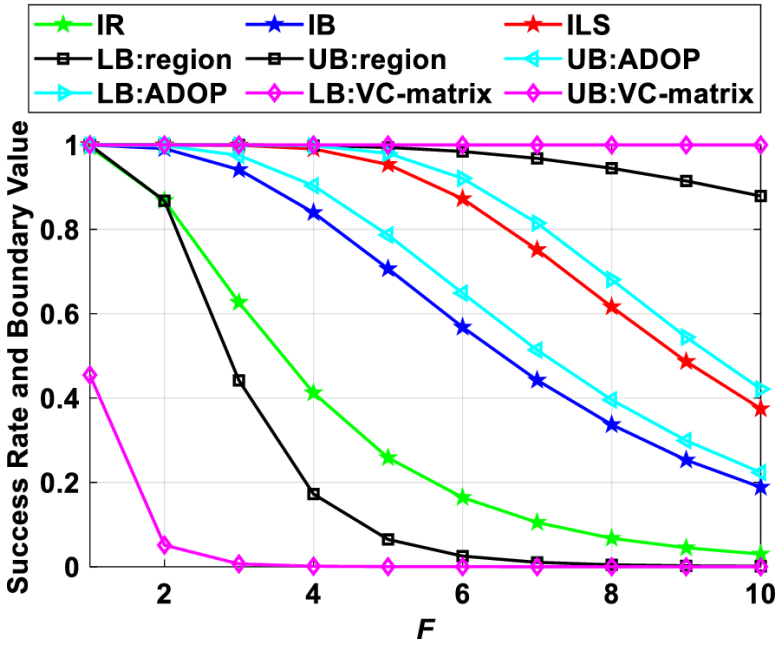


Figure 36: Example of success rate bounds and approximations based on static simulation for GPS dual-frequency data, where *F* is a scale factor applied to the ambiguity vc-matrix. More details can be found in (ibid).

4.2.3. Example #3: design analysis for future GNSS systems

The availability of analytical formulation for SR, FR and SfR, enable the user to evaluate expected performance of GNSS systems while varying key parameters such as the signal frequency. This is fundamental when designing new systems, for instance also in view of future navigation satellite constellations in Low Earth Orbit (LEO). Psychas et al. (2021) present a convergence analysis with emphasis on the role of frequency spacing, starting with Galileo E1+E5a signals and numerically evaluating PPP-RTK ambiguity resolution and positioning convergence capabilities as function of the spacing of frequencies.

In Figure 37 from (ibid), a numerical example is provided varying Galileo third frequency between 1 GHz and 2 GHz. The multi-epoch, ionosphere-float, triple-frequency user AR performances are provided for six observed satellites on DOY 166 in 2020. The formal SR is shown on the left plot, while accounting for a different number of epochs, whereas the horizontal positioning precision gain for Partial AR (>99.9%) is illustrated on the right plot based on a 30 cm and 3 mm standard deviation for code and phase observations. We refer to (ibid) for a complete discussion on the key role of multiple frequencies in PPP-RTK ambiguity resolution.

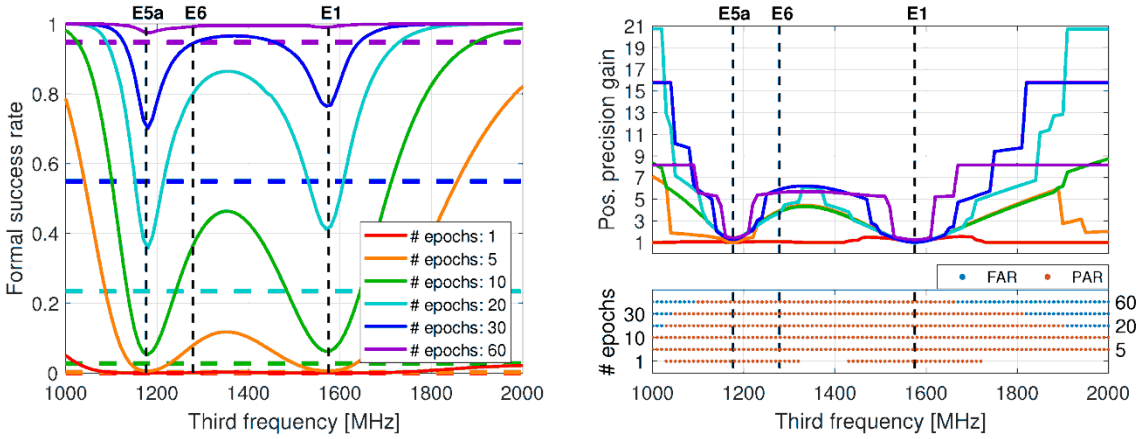


Figure 37: Numerical assessment based on the Galileo multi-epoch, ionosphere-float, triple-frequency user ambiguity resolution as function of a varying third frequency and number of used epochs for six satellites on DOY 166 in 2020. On the left and right plots, we respectively illustrate the SR and horizontal precision gain after successful PAR (99.9%), see (ibid).

The aforementioned analyses are made possible in this LAMBDA 4.0 toolbox, and another notable example concerns the investigation over a standalone GLONASS FDMA+CDMA RTK performance, as presented by Zaminpardaz et al. (2021). The worldwide ambiguity resolution performance is examined during DOY 222-229 in 2020, while using a cutoff elevation angle of 10°. In Figure 38 from (ibid), the ADOP value and bootstrapped success rate are shown on the top and bottom plots, respectively. Hence, GLONASS FDMA-only (L1+L2) and FDMA+CDMA (L1+L2+L3) scenarios are numerically compared.

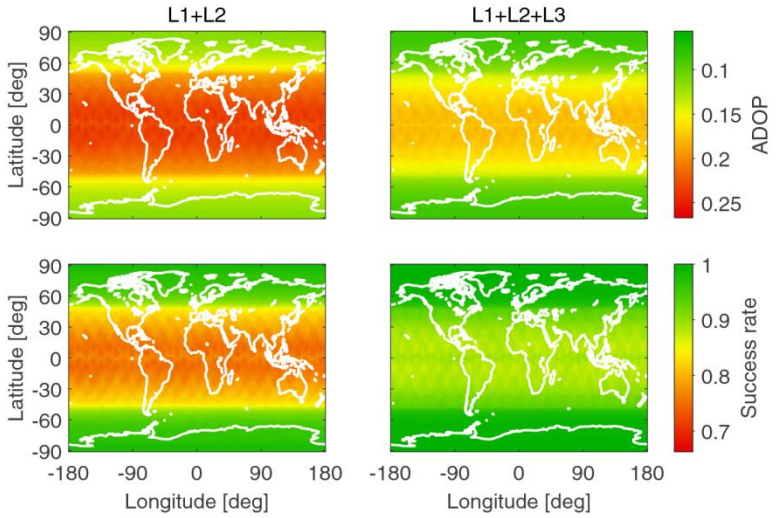


Figure 38: Example of color maps for the current average ADOP (top) and bootstrapped SR (bottom) for GLONASS FDMA-only (left) and FDMA+CDMA (right) performance. See text for more information, while we refer to (ibid) for more complete details.

Similar GNSS applications are also presented by Brack et al. (2021), where a five-system analysis is carried out for RTK performances, so using GPS (G), Galileo (E), BeiDou (C), QZSS (J) and GLONASS (R) constellations. An example for long baseline between PERT and NNOR stations (around 88.5 km) is shown in Figure 39, where GLONASS is based on a FDMA-only model, thus showing no benefits in the five-system configuration.

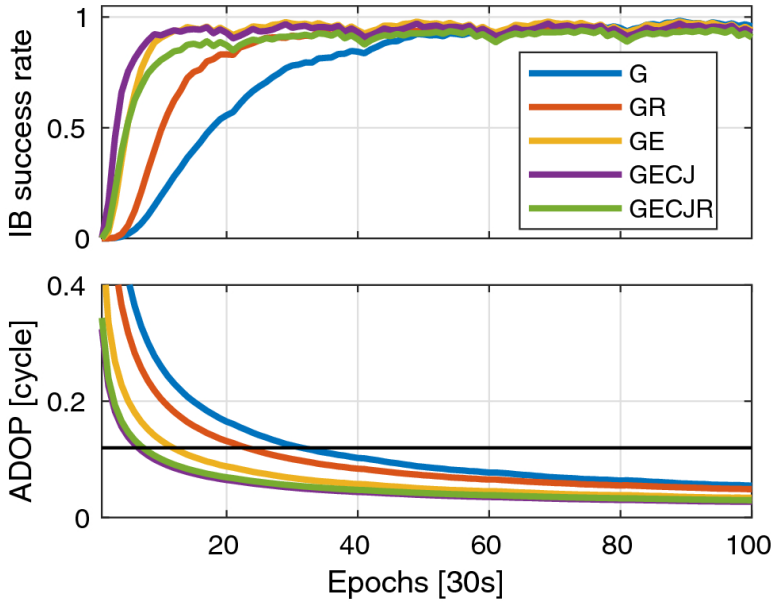


Figure 39: Example of average success rate (top) and ADOP (bottom) values with respect to the number of epochs for a dual-frequency scenario. The black line in the bottom plot marks 0.12 cycles, see (ibid).

A few additional examples can be found in (ibid), e.g. based on a short (CUT0-CUTA, 8.4 meters) and a medium (CUT0-PERT, 22.4 kilometers) baseline. However, such statistical evaluations are not limited to GNSS constellations, and several examples can be found in literature where the Ps-LAMBDA functionalities are exploited also in frequency-varying carrier phase terrestrial system. We refer the reader to Khodabandeh and Teunissen (2023) for a simulated example of the ambiguity-fixed performances based on cellular Long-Term Evolution (LTE) transmitters.

5. Terms and Conditions

In this section, the following definitions shall be considered:

- **LAMBDA 4.0**, being the toolbox software described in this documentation, and being developed by LAMBDA Team at the Delft University of Technology (TU Delft);

- **LAMBDA Team**, or "We", being the official providers and management team at TU Delft for this LAMBDA 4.0 toolbox;

- **LAMBDA User**, or "User", being the person, institution and/or program using the LAMBDA 4.0 toolbox.

By making use of this LAMBDA toolbox, the **User** agrees that

1. **LAMBDA 4.0** is available free of charge, it can be used by any individual and for any purpose except where specific restrictions are indicated.

2. We do ***not*** guarantee the accuracy of any data, software or services connected with LAMBDA 4.0, and accept *no* responsibility for any results of their use.

3. We accept ***no*** responsibility for the consequences of any temporary or permanent discontinuity to **LAMBDA 4.0**, while we will make all reasonable effort in order to maintain continuity and to provide an adequate warning in the case of any changes or discontinuities.

4. We do ***not*** authorize the redistribution of this software package, which much be obtained exclusively from the **Team** (Section 5.1).

5. We place ***no*** restrictions on the usage of **LAMBDA 4.0**, both for research and/or commercial uses, but we require the **User** to cite the toolbox (see Section 5.4).

5.1. Availability

The LAMBDA 4.0 toolbox is available free and open source from the **PNTLab website**:

http://pntlab.tudelft.nl/LAMBDA

5.2. Updates

We welcome any suggestion for improving the LAMBDA 4.0 code, the documentation and user manual. We also would like to encourage you to communicate to us about results obtained with the LAMBDA method, and comparisons made with other methods.

We would also be much obliged if you inform us in the case you decide to use the method commercially. As mentioned before, there are no restrictions on that, other than properly acknowledging the designers of the method and their employer, i.e. LAMBDA Team.

NOTE: 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 (Section 5.5) for coordinating the efforts.

5.3. Liability

Use of the accompanying LAMBDA 4.0 software is allowed, but no liability for the use of the software will be accepted by the LAMBDA Team. We ask you to refrain from passing the software to third parties. Instead the software can be requested directly from us.

NOTE: giving proper credits to the authors (see Section 5.4) is the only condition posed upon the free use of this LAMBDA 4.0 toolbox.

5.4. How to cite

We request users to cite the toolbox as follows:

Massarweh, L., Verhagen, S., and Teunissen, P.J.G. New LAMBDA toolbox for mixed-integer models: estimation and evaluation. *GPS Solut* 29,14 (2025). https://doi.org/10.1007/s10291-024-01738-z

5.5. Support contact

For any inquiry about this LAMBDA 4.0 toolbox, or other general questions on LAMBDA methods, we suggest users to contact us at

LAMBDAtoolbox-CITG-GRS@tudelft.nl

Appendix A: list of abbreviations

The list of abbreviations (acronyms & initialisms) is given below.

Abbreviation	Description
1D	1-dimensional
2D	2-dimensional
3D	3-dimensional
ADOP	Ambiguity Dilution of Precision
AR	Ambiguity Resolution
BIE	Best Integer Equivariant (estimator)
CDMA	Code-division multiple access
CDF	Cumulative Density Function
CiTG	Civiele Techniek en Geowetenschappen (faculty)
CORS	Continuously Operating Reference Station
DOI	Digital Object Identifier
ECD	Elliptical Contour Distribution
FAR	Full Ambiguity Resolution
FDMA	Frequency-division multiple access
FFRT	Fixed Failure-rate Ratio Test
FR	Failure Rate
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GRS	Geoscience and Remote Sensing (department)
IAB	Integer Aperture Bootstrapping (estimator)
IA-class	Integer Aperture class (of estimator)
IAR	Integer Ambiguity Resolution
IB	Integer Bootstrapping (estimator)
I-class	Integer class (of estimator)
IE-class	Integer Equivariant class (of estimator)
IGT	Integer Gauss Transformation
ILS	Integer Least-Squares (estimator)
InSAR	Interferometric Synthetic-Aperture Radar
IR	Integer Rounding (estimator)
LAMBDA	Least-Squares AMBiguity Decorrelation Adjustment
LTE	Long-Term Evolution
MATLAB	MATrix LABoratory
MIMO	Multiple-Input, Multiple-Output
MLAMBDA	Modified LAMBDA
MSE	Mean Squared Error
ODTS	Orbit Determination and Time Synchronization

OIA	Optimal Integer Aperture (estimator)
PAR	Partial Ambiguity Resolution
PDF	Probability Density Function
PPP	Precise Point Positioning
PPP-AR	Precise Point Positioning with Ambiguity Resolution
RT	Ratio Test
RTK	Real-Time Kinematics
SfR	Successful-fix Rate
SPP	Single Point Positioning
SR	Success Rate
TTFF	Time-To-First-Fix
UR	Undecided Rate
VIB	Vectorial Integer Bootstrapping (estimator)
VIB-ILS	VIB with ILS (estimator)
VIB-IR	VIB with IR (estimator)
VLBI	Very Long Baseline Interferometry

References

Brack, A., Männel, B., & Schuh, H. (2021). GLONASS FDMA data for RTK positioning: a five-system analysis. GPS Solut 25, 9. https://doi.org/10.1007/s10291-020-01043-5

Brack, A., Männel, B., & Schuh, H. (2023). Two-epoch centimeter-level PPP-RTK without external atmospheric corrections using best integer-equivariant estimation. GPS Solut 27, 12. https://doi.org/10.1007/s10291-022-01341-0

Chang, X., Yang, X., & Zhou, T. (2005). MLAMBDA: a modified LAMBDA method for integer least-squares estimation. Journal of Geodesy, 79: 552–565.

D'Amico, S., & Montenbruck, O. (2010). Differential GPS: An enabling technology for formation flying satellites. In Small Satellite Missions for Earth Observation: New Developments and Trends (pp. 457-465). Springer Berlin Heidelberg.

Damen, M.O., El Gamal, H., & Caire, G. (2003). On maximum-likelihood detection and the search for the closest lattice point. In IEEE Transactions on Information Theory, vol. 49, no. 10, pp. 2389-2402, Oct. 2003. https://doi.org/10.1109/TIT.2003.817444

De Jonge, P.J., & Tiberius, C. (1996). The LAMBDA method for integer ambiguity estimation: implementation aspects, LGR-Series, No 12. Technical report, Delft University of Technology.

Ghasemmehdi, A., & Agrell, E. (2011). Faster Recursions in Sphere Decoding. In IEEE Transactions on Information Theory, vol. 57, no. 6, pp. 3530-3536, June 2011. https://doi.org/10.1109/TIT.2011.2143830

Giorgi, G. (2011). GNSS Carrier Phase-Based Attitude Determination: Estimation and Applications. Ph.D. dissertation, Delft University of Technology, Delft, The Netherlands.

Hassibi, A., & Boyd, S. (1998). Integer parameter estimation in linear models with applications to GPS. IEEE Transactions on signal processing, 46(11), 2938-2952.

Hobiger, T., Sekido, M., Koyama, Y., & Kondo, T. (2009). Integer phase ambiguity estimation in next-generation geodetic very long baseline interferometry. Advances in Space Research, 43(1), 187-192. https://doi.org/10.1016/j.asr.2008.06.004

Hou, Y., Verhagen, S., & Wu, J. (2016). An Efficient Implementation of Fixed Failure-Rate Ratio Test for GNSS Ambiguity Resolution. Sensors. 2016; 16(7):945. https://doi.org/10.3390/s16070945

Jazaeri, S., Amiri-Simkooei, A., & Sharifi, M. (2014). Modified weighted integer least squares estimations for GNSS IAR, Survey Review, 46(335), 112-121.

Joosten, P. (2001). The LAMBDA-Method: MATLAB Implementation. Technical report, Mathematical Geodesy and Positioning, Delft University of Technology.

Khodabandeh, A., & Teunissen, P.J.G. (2023). Ambiguity-Fixing in Frequency-Varying Carrier Phase Measurements: Global Navigation Satellite System and Terrestrial Examples. NAVIGATION: Journal of the Institute of Navigation, 70(2).

Li, B., & Teunissen, P.J.G. (2011). High Dimensional Integer Ambiguity Resolution: A First Comparison between LAMBDA and Bernese. Journal of Navigation; 64(S1):S192-S210. https://doi.org/10.1017/S037346331100035X

Li, B., Verhagen, S., & Teunissen, P.J.G. (2014). Robustness of GNSS integer ambiguity resolution in the presence of atmospheric biases. GPS Solut 18, 283–296. https://doi.org/10.1007/s10291-013-0329-5

Lu, C., Li, X., Cheng, J., Dick, G., Ge, M., Wickert, J., & Schuh, H. (2018). Real-Time Tropospheric Delay Retrieval from Multi-GNSS PPP Ambiguity Resolution: Validation with Final Troposphere Products and a Numerical Weather Model. Remote Sensing. 2018; 10(3):481. https://doi.org/10.3390/rs10030481

Massarweh, L., Strasser, S., & Mayer-Gürr, T. (2021). On vectorial integer bootstrapping implementations in the estimation of satellite orbits and clocks based on small global networks. Advances in Space Research, 68(11), 4303-4320. https://doi.org/10.1016/j.asr.2021.09.023

Mayer-Gürr, T., Behzadpour, S., Eicker, A., Ellmer, M., Koch, B., Krauss, S., Pock, C., Rieser, D., Strasser, S., Süsser-Rechberger, B., et al. (2021). GROOPS: a software toolkit for gravity field recovery and GNSS processing. Comput Geosci 155(104):864. https://doi.org/10.1016/j.cageo.2021.104864

Mi, X., Zhang, B., El-Mowafy, A., et al. (2023). Undifferenced and uncombined GNSS time and frequency transfer with integer ambiguity resolution. J Geod 97, 13. https://doi.org/10.1007/s00190-022-01689-8

Miao, W., Li, B., Gao, Y., & Chen, G. (2024). Vectorial integer bootstrapping of best integer equivariant estimation (VIB-BIE) for efficient and reliable GNSS ambiguity resolution. J Geod 98, 30. https://doi.org/10.1007/s00190-024-01836-3

Minkowski, H. (1897). Allgemeine Lehrsätze über die convexen Polyeder. Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse 1897: 198-220

Odijk, D., & Teunissen, P.J.G. (2008). ADOP in closed form for a hierarchy of multi-frequency single-baseline GNSS models. J Geod 82, 473–492. https://doi.org/10.1007/s00190-007-0197-2

Odijk, D., Verhagen, S., & Teunissen, P.J.G. (2012). Medium-Distance GPS Ambiguity Resolution with Controlled Failure Rate. In: Kenyon, S., Pacino, M., Marti, U. (eds) Geodesy for Planet Earth. International Association of Geodesy Symposia, vol 136. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-20338-1_93

Odolinski, R., Odijk, D., & Teunissen, P.J.G. (2014). Combined GPS and BeiDou instantaneous RTK positioning. NAVIGATION: Journal of The Institute of Navigation, 61(2), 135-148.

Odolinski, R., & Teunissen, P.J.G. (2020). Best integer equivariant estimation: performance analysis using real data collected by low-cost, single- and dual-frequency, multi-GNSS receivers for short- to long-baseline RTK positioning. J Geod 94, 91. https://doi.org/10.1007/s00190-020-01423-2

Psychas, D., Verhagen, S., & Teunissen, P.J.G. (2019). LAMBDA-Python implementation, version 1.0. Delft University of Technology.

Psychas, D., Teunissen, P.J.G., & Verhagen, S., (2021). A Multi-Frequency Galileo PPP-RTK Convergence Analysis with an Emphasis on the Role of Frequency Spacing. Remote Sens., 13, 3077. https://doi.org/10.3390/rs13163077

Strang, G., & Borre, K. (1997). Linear Algebra, Geodesy, and GPS. Wellesley-Cambridge Press, Wellesley, MA.

Strasser, S. (2022). Reprocessing Multiple GNSS Constellations and a Global Station Network from 1994 to 2020 with the Raw Observation Approach. Ph.D. thesis, TU Graz, Verlag der Technischen Universität Graz. https://doi.org/10.3217/978-3-85125-885-1

Teunissen, P.J.G. (1993). Least-squares estimation of the integer GPS ambiguities. In: In IAG General Meeting. Invited Lecture. Section IV Theory and Methodology.

Teunissen, P.J.G. (1994). A new method for fast carrier phase ambiguity estimation. Proceedings of 1994 IEEE Position, Location and Navigation Symposium - PLANS'94, Las Vegas, NV, USA, pp. 562-573. https://doi.org/10.1109/PLANS.1994.303362

Teunissen, P.J.G. (1995). The least-squares ambiguity decorrelation adjustment: a method for fast GPS integer ambiguity estimation. J Geod 70(1):65–82. https://doi.org/10.1007/BF00863419

Teunissen, P.J.G. (1997). A canonical theory for short GPS baselines. Part IV: Precision versus reliability. Journal of Geodesy 71, 513–525. https://doi.org/10.1007/s001900050119

Teunissen, P.J.G. (1998a). On the integer normal distribution of the GPS ambiguities. Artificial Satellites. 33(2): pp. 49-64.

Teunissen, P.J.G. (1998b). Success probability of integer GPS ambiguity rounding and bootstrapping. Journal of Geodesy, 72: 606–612.

Teunissen, P.J.G. (1999a). The probability distribution of the GPS baseline for a class of integer ambiguity estimators. Journal of Geodesy 73, 275–284. https://doi.org/10.1007/s001900050244

Teunissen, P.J.G. (1999b). An optimality property of the integer least-squares estimator. Journal of Geodesy 73, 587–593. https://doi.org/10.1007/s001900050269

Teunissen, P.J.G. (2000). ADOP based upper bounds for the bootstrapped and the least-squares ambiguity success rates. Artificial Satellites, 35(4): 171–179.

Teunissen, P.J.G. (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, P.J.G. (2001b). Integer estimation in the presence of biases. Journal of Geodesy 75, 399–407. https://doi.org/10.1007/s001900100191

Teunissen, P.J.G. (2002). A new class of GNSS ambiguity estimators. Artificial Satellites, Vol. 37, No. 4, pp. 111-120.

Teunissen, P.J.G. (2003a). Theory of integer equivariant estimation with application to GNSS. Journal of Geodesy, 77, 402-410. https://doi.org/10.1007/s00190-003-0344-3

Teunissen, P.J.G. (2003b). Integer aperture GNSS ambiguity resolution. Artificial Satellites, 38(3): 79-88.

Teunissen, P.J.G. (2003c). Towards a unified theory of GNSS ambiguity resolution. J Global Position. Syst. 2(1), 1–12.

Teunissen, P.J.G. (2005a). Integer aperture bootstrapping: a new GNSS ambiguity estimator with controllable fail-rate. Journal of Geodesy 79, 389–397. https://doi.org/10.1007/s00190-005-0481-y

Teunissen, P.J.G. (2005b). On the computation of the best integer equivariant estimator. Artificial Satellites. 40 (3): pp. 161-171.

Teunissen, P.J.G. (2005c). GNSS ambiguity resolution with optimally controlled failure-rate. Artificial Satellites. 40 (4): pp. 219-227.

Teunissen, P.J.G. (2006). On InSAR ambiguity resolution for deformation monitoring. Artificial Satellites, 41(1), 19-22.

Teunissen, P.J.G. (2020a). Best integer equivariant estimation for elliptically contoured distributions. J Geod 94, 82. https://doi.org/10.1007/s00190-020-01407-2

Teunissen, P.J.G. (2020b). GNSS precise point positioning. In: Position, Navigation, and Timing Technologies in the 21st Century: integrated satellite navigation, sensor systems, and civil applications, vol 1, pp 503–528. https://doi.org/10.1002/9781119458449.ch20

Teunissen, P.J.G., & de Bakker, P.F. (2015). Multivariate Integer Cycle-Slip Resolution: A Single-Channel Analysis. In: Sneeuw, N., Novák, P., Crespi, M., Sansò, F. (eds) VIII Hotine-Marussi Symposium on Mathematical Geodesy. International Association of Geodesy Symposia, vol 142. Springer, Cham. https://doi.org/10.1007/1345_2015_69

Teunissen, P.J.G., Joosten, P., & Tiberius, C. (1999). Geometry-free Ambiguity Success Rates in Case of Partial Fixing. Proceedings of the 1999 National Technical Meeting of The Institute of Navigation, San Diego, CA, January 1999, pp. 201-207.

Teunissen, P.J.G., & Khodabandeh, A. (2015). Review and principles of PPP-RTK methods. J Geod 89, 217–240. https://doi.org/10.1007/s00190-014-0771-3

Teunissen, P.J.G., Massarweh, L., & Verhagen, S. (2021). Vectorial integer bootstrapping: flexible integer estimation with application to GNSS. J Geod 95, 99 (2021). https://doi.org/10.1007/s00190-021-01552-2

Teunissen, P.J.G., & Odijk, D. (1997). Ambiguity Dilution of Precision: Definition, Properties and Application. Proceedings of the 10th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1997), Kansas City, MO, pp. 891-899.

Teunissen, P.J.G., & Verhagen, S. (2009). The GNSS ambiguity ratio-test revisited: a better way of using it. Survey Rev 41(312):138–151

Verhagen, S. (2005). The GNSS integer ambiguities: Estimation and validation. Ph.D. dissertation, Delft University of Technology, Delft, The Netherlands. Accessible online at https://www.ncgeo.nl/downloads/58Verhagen.pdf

Verhagen, S., Li, B., & Teunissen, P.J.G. (2012). LAMBDA–Matlab implementation, version 3.0. Delft University of Technology and Curtin University.

Verhagen, S., Li, B., & Teunissen, P.J.G. (2013). Ps-LAMBDA: Ambiguity Success rate Evaluation Software for Interferometric Applications. Computers & Geosciences, 54:361-376. https://doi.org/10.1016/j.cageo.2013.01.014

Verhagen, S., Teunissen, P.J.G., van der Marel, H., & Li, B. (2011). GNSS ambiguity resolution: which subset to fix. In IGNSS symposium (pp. 15-17).

Viegas, D.C.d.N., & Cunha, S.R. (2007). Precise Positioning by Phase Processing of Sound Waves. In IEEE Transactions on Signal Processing, vol. 55, no. 12, pp. 5731-5738, Dec. 2007. https://doi.org/10.1109/TSP.2007.900166

Yang, Y., Zhou, F., & Song, S. (2024). Improving precise point positioning (PPP) performance with best integer equivariant (BIE) estimator. GPS Solut 28, 50. https://doi.org/10.1007/s10291-023-01585-4

Yong, C.Z., Harima, K., Rubinov, E., McClusky, S., & Odolinski, R. (2022). Instantaneous Best Integer Equivariant Position Estimation Using Google Pixel 4 Smartphones for Single- and Dual-Frequency, Multi-GNSS Short-Baseline RTK. Sensors, 22, 3772. https://doi.org/10.3390/s22103772

Wang, L., Feng, Y., Guo, J., & Wang, C. (2016). Impact of Decorrelation on Success Rate Bounds of Ambiguity Estimation. Journal of Navigation, 69(5):1061-1081. https://doi.org/10.1017/S0373463316000047

Wang, Z., Hou, X., Dan, Z., & Fang, K. (2023). Adaptive Kalman filter based on integer ambiguity validation in moving base RTK. GPS Solut 27, 34. https://doi.org/10.1007/s10291-022-01367-4

Zaminpardaz, S., Teunissen, P.J.G., & Khodabandeh, A. (2021). GLONASS–only FDMA+CDMA RTK: Performance and outlook. GPS Solut 25, 96. https://doi.org/10.1007/s10291-021-01132-z