J Geod (2005) 79: 389–397
DOI 10.1007/s00190-005-0481-y

ORIGINAL ARTICLE

P.J.G. Teunissen

Integer aperture bootstrapping: a new GNSS ambiguity estimator with controllable fail-rate

Received: 13 January 2004 / Accepted: 21 June 2005 / Published online: 25 July 2005

Abstract In this contribution, we introduce a new bootstrap-based method for Global Navigation Satellite System (GNSS) carrier-phase ambiguity resolution. Integer bootstrapping is known to be one of the simplest methods for integer ambiguity estimation with close-to-optimal performance. Its outcome is easy to compute due to the absence of an integer search, and its performance is close to optimal if the decorrelating Z-transformation of the LAMBDA method is used. Moreover, the bootstrapped estimator is presently the only integer estimator for which an exact and easy-to-compute expression of its fail-rate can be given. A possible disadvantage is, however, that the user has only a limited control over the fail-rate. Once the underlying mathematical model is given, the user has no freedom left in changing the value of the fail-rate. Here, we present an ambiguity estimator for which the user is given additional freedom. For this purpose, use is made of the class of integer aperture estimators as introduced in Teunissen (2003). This class is larger than the class of integer estimators. Integer aperture estimators are of a hybrid nature and can have integer outcomes as well as non-integer outcomes. The new estimator is referred to as integer aperture bootstrapping. This new estimator has all the advantages known from integer bootstrapping with the additional advantage that its fail-rate can be controlled by the user. This is made possible by giving the user the freedom over the aperture of the pull-in region. We also give an exact and easy-to-compute expression for its controllable fail-rate.

Keywords GNSS ambiguity resolution · Integer estimation · Integer aperture estimation · Integer aperture bootstrapping · Controllable fail-rate

P.J.G. Teunissen
Delft Institute of Earth Observation and Space systems,
Delft University of Technology,
Kluyverweg 1, 2629 HS Delft, The Netherlands
Fax: +31-15-2783711,
E-mail: P.J.G.Teunissen@LR.TUDelft.NL

1 Introduction

Global Navigation Satellite System (GNSS) carrier-phase ambiguity resolution is the process of resolving the carrier-phase ambiguities as integers. It is the key to fast and high-precision GNSS positioning and it applies to a great variety of GNSS models which are currently in use in navigation, surveying, geodesy and geophysics; see e.g. Leick (1995), Parkinson and Spilker (1996), Strang and Borre (1997), Teunissen and Kleusberg (1998), Farrell and Barth (1999), Misra and Enge (2001), Hofmann-Wellenhof et al. (2002), Seeber (2003).

All the linear(ized) GNSS models can, in principle, be cast in the following frame of observation equations:

$$E\{y\} = Aa + Bb, \quad a \in Z^n, b \in R^q \tag{1}$$

with $E\{.\}$ the mathematical expectation operator, y the m-vector of observables, a the n-vector of unknown integer parameters and b the q-vector of unknown real-valued parameters. The data vector y will then usually consist of the "observed minus computed" single-, dual- or multi-frequency double-difference (DD) phase and/or pseudorange (code) observations accumulated over all observation epochs. The entries of vector a are then the DD carrier-phase ambiguities, expressed in units of cycles rather than range, while the entries of the vector b will consist of the remaining unknown parameters, such as baseline components (coordinates) and possibly atmospheric delay parameters (troposphere and ionosphere).

The procedure for solving the above GNSS model (Eq. 1) can be divided conceptually into three steps. In the first step, one simply discards the integer constraints $a \in Z^n$ and performs a standard least-squares adjustment. As a result, one obtains the so-called float solution $\hat{a}$ and $\hat{b}$. This solution is real-valued. Then in the second step, the float solution $\hat{a}$ is further adjusted so as to take, in some pre-defined way, the "integerness" of the ambiguities into account. This gives

$$\hat{a}_S = S(\hat{a}) \tag{2}$$

in which S is an n-dimensional mapping that transforms the float solution into a corresponding integer solution. This

estimator is then used in the final step to adjust the float estimator $\hat{b}$. As a result, one obtains the so-called ambiguity fixed estimator of b as

$$\hat{b}_S = \hat{b} - Q_{\hat{b}\hat{a}} Q_{\hat{a}}^{-1} (\hat{a} - \hat{a}_S) \tag{3}$$

in which $Q_{\hat{a}}$ denotes the variance-covariance (vc-) matrix of $\hat{a}$ and $Q_{\hat{b}\hat{a}}$ denotes the covariance matrix of $\hat{b}$ and $\hat{a}$.

The above three-step procedure is still ambiguous in the sense that it leaves room for choosing the n-dimensional map S. Different choices for S will lead to different ambiguity estimators and thus also to different baseline estimators $\hat{b}_S$. One can therefore now think of constructing family of maps S with certain desirable properties. Three such classes of ambiguity estimators are the class of integer estimators, the class of integer equivariant estimators, and the class of integer aperture estimators. These classes were introduced, respectively, in Teunissen (1999, 2002, 2003). These three classes of estimators are subsets of one another. The first class is the most restrictive class. This is due to the fact that the outcomes of any estimator within this class are required to be integers. Well-known examples of estimators from this class are integer rounding, integer bootstrapping and integer least-squares. The most relaxed class is the class of integer equivariant estimators. These estimators are real-valued, and they only obey the integer remove-restore principle. An important estimator in this class is the best integer equivariant estimator since it has the smallest variance, even smaller than the variance of the float solution. The class of integer aperture estimators is a subset of the integer equivariant estimators but it encompasses the class of integer estimators. The integer aperture estimators are of a hybrid nature in the sense that their outcomes are either integers or non-integers. In this contribution, we use this class to introduce a new bootstrap-based method for GNSS carrier-phase ambiguity resolution.

Integer bootstrapping is known to be one of the simplest methods for integer ambiguity estimation with close to optimal performance. Its outcome is easy to compute due to the absence of an integer search, and its performance is close to optimal if the decorrelating Z-transformation of the LAMBDA method (Teunissen 1993) is used. Moreover, the bootstrapped estimator is the only integer estimator for which an exact and easy-to-compute expression of its fail-rate can be given. A possible disadvantage is that the user has only a limited control over the fail-rate. Once the underlying mathematical model is given, the user has no freedom left in changing the value of the fail-rate. Instead, we will present an ambiguity estimator for which the user is given this additional freedom. The new estimator is referred to as integer aperture bootstrapping. This new estimator has all the advantages of integer bootstrapping, but with the additional advantage that its fail-rate can be controlled by the user. This is made possible by giving the user the freedom over the aperture of the pull-in region. We also give an exact and easy-to-compute expression for its controllable fail-rate.

The contribution is organized as follows: Since the new estimator is bootstrap-based, we first give a brief review of the principle of integer bootstrapping in Sect. 2. This includes the conditions for integer estimation, the triangular-factorization-based computation of the bootstrapped solution, the bootstrapped pull-in regions, and the easy-to-compute bootstrapped fail-rate. In Sect. 3 we describe the class of integer aperture estimators and show how their probabilistic performance can be measured. Based on the results of Sects. 2 and 3, we introduce the principle of integer aperture bootstrapping in Sect. 4. It includes a definition of the aperture pull-in region of the new estimator, a description of how its output needs to be computed, and exact expressions for evaluating its probabilistic performance as function of a user-defined aperture parameter.

2 Integer bootstrapping

2.1 Integer estimation

The mapping S will have to be a mapping from the n-dimensional space of real numbers to the n-dimensional space of integers if we want to estimate the unknown ambiguity vector $a \in Z^n$ as an integer vector. But since the mapping $S : R^n \mapsto Z^n$ will then not be one-to-one, different real-valued vectors will be mapped by S to one and the same integer vector. One can therefore assign a subset $S_z \subset R^n$ to each integer vector $z \in Z^n$ such that it contains all real-valued vectors that will be mapped by S to z:

$$S_z = \{x \in R^n \mid z = S(x)\}, \quad z \in Z^n \tag{4}$$

This subset is referred to as the pull-in region of z. It is the region in which all float solutions are pulled to the same integer vector z. By using the concept of pull-in regions, one can now define classes of estimators by imposing various conditions on the pull-in regions. In Teunissen (1999) the following three conditions were imposed:

$$\begin{cases} (1)\ \bigcup_{z \in Z^n} S_z = R^n \\ (2)\ \mathrm{Int}(S_{z_1}) \bigcap \mathrm{Int}(S_{z_2}) = \emptyset, \quad \forall z_1, z_2 \in Z^n, z_1 \neq z_2 \\ (3)\ S_z = z + S_0, \quad \forall z \in Z^n \end{cases} \tag{5}$$

Each one of these three conditions describes a property of which it seems reasonable that it is possessed by a sensible integer estimator. The first condition states that the pull-in regions should not leave any gaps and the second that they should not overlap. The absence of gaps is needed in order to be able to map any float solution $\hat{a} \in R^n$ to Z^n, while the absence of overlaps is needed to guarantee that the float solution is mapped to just one integer vector. Note that we allow the pull-in regions to have common boundaries. This is permitted if we assume to have zero probability that $\hat{a}$ lies on one of the boundaries. This will be the case when the probability density function (pdf) of $\hat{a}$ is continuous.

The third and last condition of the definition follows from the requirement that $S(x + z) = S(x) + z, \quad \forall x \in R^n, z \in Z^n$. Also, this condition is a reasonable one to ask for. It states that when the float solution $\hat{a}$ is perturbed by an arbitrary integer vector, say $z \in Z^n$, then the corresponding integer solution is perturbed by the same amount. This property

allows one to apply the integer remove–restore technique: $S(\hat{a}-z)+z = S(\hat{a})$. It therefore allows one to work with the fractional parts of the entries of $\hat{a}$, instead of with its complete entries. With the above conditions one is now in a position to give a definition of the class of integer estimators.

Definition 1 (Integer estimators) *Let $\hat{a} \in R^n$ be the float solution and let the subsets $S_z \subset R^n$, $z \in Z^n$, satisfy all three conditions of (5). Then*

$$\breve{a} = \sum_{z \in Z^n} z s_z(\hat{a}), \text{ with } s_z(\hat{a}) = \begin{cases} 1 & \text{if } \hat{a} \in S_z \\ 0 & \text{if } \hat{a} \notin S_z \end{cases} \tag{6}$$

is said to be an integer estimator.

Note, since $\sum_{z \in Z^n} s_z(x) = 1$ for all $x \in R^n$, the $s_z(\hat{a})$ can be interpreted as weights. The integer estimator $\breve{a}$ is therefore equal to a weighted sum of integer vectors with binary weights. With the above definition one can now design one's own integer ambiguity estimator by a suitable choice of the pull-in regions in accordance with (5). Once one has come up with a division of the ambiguity space R^n into translational-invariant subsets, which fill the space without gaps and overlaps, one has defined one's own integer estimator. The three best-known integer estimators are integer rounding, integer bootstrapping and integer least-squares. For the present contribution it is the bootstrapped estimator which is of particular interest.

2.2 Bootstrapping

The bootstrapped estimator can be seen as a generalization of the process of integer rounding. It makes use of rounding, but it also takes some of the correlation between the ambiguities into account. The bootstrapped estimator follows from a sequential conditional least-squares adjustment, and it is computed as follows: if n ambiguities are available, one starts with the first ambiguity $\hat{a}_1$, and rounds its value to the nearest integer. Having obtained the integer value of this first ambiguity, the real-valued estimates of all remaining ambiguities are then corrected by virtue of their correlation with the first ambiguity. Then the second, but now corrected, real-valued ambiguity estimate is rounded to its nearest integer. Having obtained the integer value of the second ambiguity, the real-valued estimates of all remaining $n-2$ ambiguities are then again corrected, but now by virtue of their correlation with the second ambiguity. This process is continued until all ambiguities are considered. The entries of the bootstrapped estimator $\breve{a}_B = (\breve{a}_{B,1}, \ldots, \breve{a}_{B,n})^{\mathrm{T}} \in Z^n$ are thus given as

$$\begin{cases} \breve{a}_{B,1} = [\hat{a}_1] \\ \breve{a}_{B,2} = [\hat{a}_{2|1}] \ = [\hat{a}_2 - \sigma_{21}\sigma_1^{-2}(\hat{a}_1 - \breve{a}_{B,1})] \\ \qquad \vdots \\ \breve{a}_{B,n} = [\hat{a}_{n|N}] = [\hat{a}_n - \sum_{j=1}^{n-1} \sigma_{n,j|J}\sigma_{j|J}^{-2}(\hat{a}_{j|J} - \breve{a}_{B,j})] \end{cases} \tag{7}$$

where $'[.]'$ denotes rounding to the nearest integer, $\sigma_{i,j|J}$ denotes the covariance between $\hat{a}_i$ and $\hat{a}_{j|J}$, and $\sigma^2_{j|J}$ is the variance of $\hat{a}_{j|J}$. The shorthand notation $\hat{a}_{i|I}$ stands for the ith least-squares ambiguity obtained through a conditioning on the previous $I = \{1, \ldots, (i-1)\}$ sequentially rounded ambiguities.

When computing the bootstrapped solution it is very useful to make use of the triangular factorization of the ambiguity vc-matrix. Due to the close relationship that exists between sequential conditional least-squares estimation and the unique lower triangular factorization of the ambiguity vc-matrix, $Q_{\hat{a}} = LDL^{\mathrm{T}}$, we have the following statistical interpretation of the entries of L and D (Teunissen 2001):

$$(L)_{ij} = \begin{cases} 0 & \text{if } 1 \le i < j \le n \\ 1 & \text{if } i = j \\ \sigma_{i,j|J}\sigma_{j|J}^{-2} & \text{if } 1 \le j < i \le n \end{cases} \text{ and } D = \mathrm{diag}(\ldots, \sigma^2_{j|J}, \ldots) \tag{8}$$

This shows that the coefficients needed in (7) are given by the lower triangular entries of L. The unit lower triangular matrix L can also be used to describe the bootstrapped pull-in regions. They are given as

$$S_{B,z} = \{x \in R^n \mid\mid c_i^{\mathrm{T}} L^{-1}(x-z) \mid \le \frac{1}{2}, \ i = 1, \ldots, n\}, \quad \forall z \in Z^n \tag{9}$$

in which c_i denotes the canonical unit vector having a 1 as its ith entry and zeros otherwise. The pull-in regions of integer bootstrapping are multivariate versions of parallellograms. To see this, consider the two-dimensional case first. Let the lower triangular matrix L be given as

$$L = \begin{bmatrix} 1 & 0 \\ l & 1 \end{bmatrix}$$

Then

$$\begin{aligned} S_{B,0} &= \{x \in R^2 \mid\mid c_i^{\mathrm{T}} L^{-1} x \mid \le \tfrac{1}{2}, i = 1, 2\} \\ &= \{x \in R^2 \mid\mid x_1 \mid \le \tfrac{1}{2}, \mid x_2 - l x_1 \mid \le \tfrac{1}{2}\} \end{aligned}$$

which shows that the two-dimensional pull-in region equals a parallellogram. Its region is bounded by the two vertical lines $x_1 = \frac{1}{2}$ and $x_1 = -\frac{1}{2}$, and the two parallel slopes $x_2 = l x_1 + \frac{1}{2}$ and $x_2 = l x_1 - \frac{1}{2}$. The direction of the slope is governed by $l = \sigma_{21}\sigma_1^{-2}$. Hence, in the absence of correlation between the two ambiguities, the parallellogram reduces to the unit-square. In higher dimensions, the above construction of the pull-in region can be continued. In three dimensions for instance, the intersection of the pull-in region with the x_1x_2-plane remains a parallellogram, while along the third axis the pull-in region becomes bounded by two parallel planes. The bootstrapped pull-in regions reduce to the multivariate versions of the unit-square in case the ambiguity vc-matrix happens to be diagonal, $L = I_n$. This corresponds with the fact that bootstrapping reduces to rounding in the absence of any correlation between the ambiguities. Figure 1 shows an example of two sets of two-dimensional bootstrapped pull-in regions. Their difference in shape is due to the difference in ambiguity correlation between the two cases.

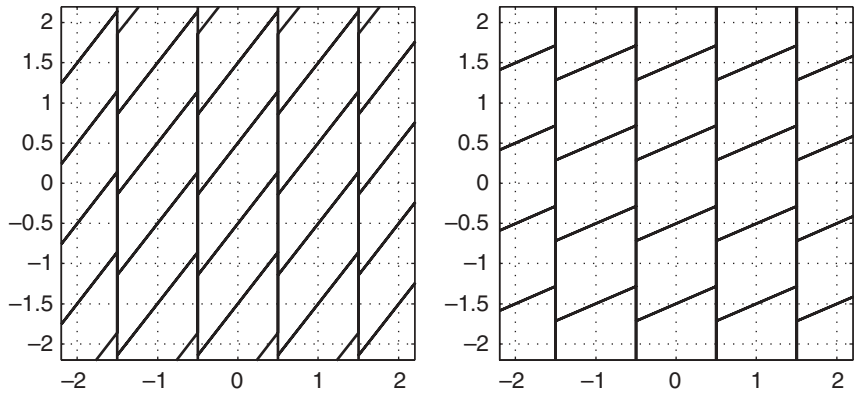

Fig. 1 Two sets of two-dimensional bootstrapped pull-in regions. *Left* high correlation, *right* low correlation

2.3 The bootstrap success-rate

For the evaluation of the integer estimator we need the distribution of $\breve{a}$. This distribution is of the discrete type and it will be denoted as $P(\breve{a} = z)$. It is a probability mass function (pmf) having zero masses at nongrid points and nonzero masses at some or all grid points. In order to obtain this pmf we need the probability density function (pdf) of the float solution $\hat{a}$. This pdf will be denoted as $f_{\hat{a}}(x)$. The pmf $P(\breve{a} = z)$ follows then from integrating $f_{\hat{a}}(x)$ over the pull-in regions S_z:

$$P(\breve{a} = z) = \int_{S_z} f_{\hat{a}}(x)\mathrm{d}x \ , \ z \in Z^n \tag{10}$$

This distribution depends on the pull-in regions S_z and thus on the chosen integer estimator. Having the problem of GNSS ambiguity resolution in mind, one is particularly interested in the probability of correct integer estimation, also referred to as the ambiguity success-rate. Also, this probability, $P(\breve{a} = a)$, depends on the chosen integer estimator. In general, however, it is rather difficult to obtain an exact and easy-to-compute expression for the success-rate. There is one exception, namely the success-rate of the bootstrapped estimator. As the following theorem shows, the bootstrapped success-rate can be computed relatively straightforward in an exact manner.

Theorem 1 (bootstrapped success-rate and upperbound) *Let the float solution be distributed as* $\hat{a} \sim \mathrm{N}(a, Q_{\hat{a}})$ *and let* $\breve{a}_B$ *be the integer bootstrapped estimator of* $a \in Z^n$*, in accordance with (7). Then*

$$\begin{aligned} P(\breve{a}_B = a) &= \prod_{i=1}^{n} \left(2\Phi\left(\tfrac{1}{2\sigma_{i|I}}\right) - 1\right) \\ &\leq \left(2\Phi\left(\tfrac{1}{2\mathrm{ADOP}}\right) - 1\right)^n \end{aligned} \tag{11}$$

with $\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp\{-\frac{1}{2}v^2\}\mathrm{d}v$ *and* $\mathrm{ADOP} = \sqrt{\det Q_{\hat{a}}}^{\frac{1}{n}}$ *(cycle).*

Proof The equality was first given in Teunissen (1998a) and the inequality in Teunissen (2000).

This result shows that the bootstrapped success-rate is determined by the conditional variances $\sigma^2_{i|I}$, $i = 1, \ldots, n$. They are the entries of the diagonal matrix D in the factorization of $Q_{\hat{a}} = LDL^{\mathrm{T}}$. □

The outcome of integer bootstrapping and its success-rate both depend on the chosen ambiguity parametrization. For instance, a simple reordering of the ambiguities will already affect the success-rate. The fact that the bootstrapped success-rate will not remain invariant when an arbitrary ambiguity transformation is applied is a consequence of the uniqueness of the triangular decomposition of $Q_{\hat{a}}$. The bootstrapped success-rate will change since the diagonal matrix D of the triangular decomposition changes when an arbitrary ambiguity transformation is applied. This lack of invariance of the bootstrapped success-rate implies that one can try to improve the performance of bootstrapping by choosing an appropriate ambiguity parametrization.

In Teunissen (1999) the integer least-squares estimator was shown to be the optimal estimator. Since bootstrapping becomes identical to integer least-squares when the vc-matrix is diagonal, one should aim at reducing the correlations between the ambiguities in order to improve the performance of the bootstrapped estimator. This is possible by applying the decorrelating Z-transformation of the LAMBDA method (Teunissen 1993). When this transformation is applied, one works with the more precise and decorrelated ambiguity vector $\hat{z} = Z\hat{a}$, instead of with the original ambiguity vector $\hat{a}$. Since the conditional variances of $\hat{z}$ are usually orders of magnitude smaller than the large conditional variances of $\hat{a}$, the bootstrapped success-rate of the transformed ambiguities will generally be much larger than that of the original ambiguities. In fact, practical experiments have shown that the bootstrapped success-rate of the Z-transformed ambiguities comes very close to the optimal success-rate of integer least-squares; see e.g. Thomsen (2000). For more information on the LAMBDA method, we refer to Teunissen (1995) and de Jonge and Tiberius (1996a,b), or to the textbooks by Hofmann-Wellenhof et al. (2002), Strang and Borre (1997), Teunissen and Kleusberg (1998), Misra and Enge (2001), Seeber (2003). Examples of its applications can be found in, e.g., Han and Rizos (1996), de Jonge et al. (1996), Boon

and Ambrosius (1997), Boon et al. (1997), Cox and Brading (1999), Peng et al. (1999), Verhagen (2001), Lee et al. (2003) and Wu et al. (2003).

Theorem 1 also gives an upper bound on the bootstrapped success-rate. One may question its usefulness since, after all, we already have an easy way to compute the success-rate exactly. The advantage of the upper bound is, however, that it is invariant for the class of admissible ambiguity transformations. An ambiguity transformation $\hat{z} = Z\hat{a}$ is said to be admissible if and only if all the entries of the matrix Z and its inverse are integer. These two conditions are needed to retain the integer nature of the ambiguities. It can be shown that the determinant of admissible ambiguity transformations always equal ± 1. We therefore have $\det(Q_{\hat{z}}) = \det(Z Q_{\hat{a}} Z^{\mathrm{T}})$, which shows the invariance of the ambiguity dilution of precision (ADOP). Thus, the same ADOP-value is obtained whatever admissible choice is made for the ambiguity parametrization. The decorrelating transformation of the LAMBDA method is an example of such an admissible ambiguity transformation.

Also the ADOP is easily computed from the triangular factorization of the ambiguity vc-matrix. Since the determinant of the lower triangular factor L equals one, we have $\det Q_{\hat{a}} = \det D$ and thus

$$\mathrm{ADOP} = \left(\prod_{i=1}^{n} \sigma_{i|I} \right)^{\frac{1}{n}} \tag{12}$$

This shows that the ADOP equals the geometric mean of the ambiguity standard deviations.

The advantage of the upper bound being invariant is that it only needs to be computed once to cover all possible ambiguity parametrizations. It can, therefore, be used to quickly decide on the potential usefulness of bootstrapping for ambiguity resolution. If in any application the upper bound turns out to be too small the conclusion must be that one cannot expect carrier-phase ambiguity resolution to be successful when it is based on the principle of bootstrapping, whatever ambiguity parametrization is chosen.

3 Integer aperture (IA) estimation

3.1 IA-estimators

So far we considered the principle of integer estimation with bootstrapping as an easy-to-compute and important example of integer estimation. The outcome of an integer estimator is, however, always an integer. This implies that a user has limited control over the fail-rate of an integer estimator. The fail-rate equals the probability of incorrect integer estimation. In the case of integer estimation, the fail-rate equals one minus the success-rate. The fail-rate depends, therefore, through the type of pull-in region on the choice of integer estimator and through the pdf of the float solution on the strength of the underlying mathematical model. Thus, once the choice of an integer estimator has been made, the fail-rate can only be improved by strengthening the mathematical model, for instance by using more data or more precise data. For a given mathematical model, however, the user has no means of changing the intrinsic fail-rate of the chosen integer estimator. Now, in order to accommodate this limitation, the class of integer aperture (IA)-estimators was introduced in Teunissen (2003). The class of IA-estimators is larger than the class of integer estimators. That is, all integer estimators are IA-estimators, but not all IA-estimators are integer estimators. The class of IA-estimators is defined by dropping one of the three conditions of (5), namely the condition that the pull-in regions should cover the ambiguity space R^n completely. We will therefore allow the pull-in regions of the IA-estimators to have gaps.

In order to introduce the new class of ambiguity estimators from first principles, let $\Omega \subset R^n$ be the region of R^n for which $\hat{a}$ is mapped to an integer if $\hat{a} \in \Omega$. It seems reasonable to ask of the region Ω that it has the property that if $\hat{a} \in \Omega$ then also $\hat{a} + z \in \Omega$, for all $z \in Z^n$. If this property would not hold, then float solutions could be mapped to integers whereas their fractional parts would not. We thus require Ω to be translational-invariant with respect to an arbitrary integer vector: $\Omega + z = \Omega$, for all $z \in Z^n$. However, Ω is not sufficient for defining our estimator. Ω only determines whether or not the float solution is mapped to an integer, but it does not tell us yet to which integer the float solution is mapped. We therefore define

$$\Omega_z = \Omega \cap S_z, \quad \forall z \in Z^n \tag{13}$$

where S_z is a pull-in region satisfying the conditions of (5). Then

$$\begin{cases} (1)\ \bigcup_z \Omega_z = \bigcup_z (\Omega \bigcap S_z) = \Omega \bigcap (\bigcup_z S_z) \\ \quad = \Omega \bigcap R^n = \Omega \\ (2)\ \Omega_{z_1} \bigcap \Omega_{z_2} = (\Omega \bigcap S_{z_1}) \bigcap (\Omega \bigcap S_{z_2}) \\ \quad = \Omega \bigcap (S_{z_1} \bigcap S_{z_2}) = \emptyset, \quad \forall z_1, z_2 \in Z^n, z_1 \neq z_2 \\ (3)\ \Omega_0 + z = (\Omega \bigcap S_0) + z = (\Omega + z) \bigcap (S_0 + z) \\ \quad = \Omega \bigcap S_z = \Omega_z, \quad \forall z \in Z^n \end{cases} \tag{14}$$

This shows that the subsets $\Omega_z \subset S_z$ satisfy the same conditions as those of (5), be it that R^n has now been replaced by $\Omega \subset R^n$. Hence, the mapping of the IA-estimator can now be defined as follows: the IA-estimator maps the float solution $\hat{a}$ to the integer vector z when $\hat{a} \in \Omega_z$, and it maps the float solution to itself when $\hat{a} \notin \Omega$. The class of IA-estimators can therefore be defined as follows.

Definition 2 (IA-estimators) *Let $\hat{a} \in R^n$ be the float solution and let the subsets $\Omega_z \subset R^n$, $z \in Z^n$, satisfy the conditions of (14). Then*

$$\hat{a}_{IA} = \hat{a} + \sum_{z \in Z^n} (z - \hat{a}) \omega_z(\hat{a}),$$

$$\text{with } \omega_z(\hat{a}) = \begin{cases} 1 & \text{if } \hat{a} \in \Omega_z \\ 0 & \text{if } \hat{a} \notin \Omega_z \end{cases} \tag{15}$$

is said to be an IA-estimator.

Note that every integer estimator is indeed an IA-estimator. Since the indicator functions $s_z(x)$ of the pull-in regions S_z

sum up to unity, $\sum_{z\in Z^n} s_z(x) = 1$, the integer estimator (6) may also be written as

$$\check{a} = \hat{a} + \sum_{z\in Z^n} (z - \hat{a}) s_z(\hat{a}) \tag{16}$$

Comparing this expression with that of (15) shows that the difference between the two estimators lies in their binary weights, $s_z(x)$ versus $\omega_z(x)$. Since the $s_z(x)$ sum up to unity for all $x \in R^n$, the outcome of an integer estimator will always be an integer. This is not true for an IA-estimator, since the binary weights $\omega_z(x)$ do not sum up to unity for all $x \in R^n$, but only for $x \in \Omega$. The IA-estimator is, therefore, a hybrid estimator having as outcome either the real-valued float solution $\hat{a}$ or an integer solution. The IA-estimator returns the float solution if $\hat{a} \notin \Omega$, and it will be equal to z when $\hat{a} \in \Omega_z$. Note, since Ω is the collection of all $\Omega_z = \Omega_0 + z$, the IA-estimator is completely determined once Ω_0 is known. Thus $\Omega_0 \subset S_0$ plays the same role for the IA-estimators as S_0 does for the integer estimators. By changing the size and shape of Ω_0 one changes the outcome of the IA-estimator. The subset Ω_0 can, therefore, be seen as an adjustable pull-in region with two limiting cases: the limiting case in which Ω_0 is empty, and the limiting case when Ω_0 equals S_0. In the first case, the IA-estimator becomes identical to the float solution $\hat{a}$, and in the second case, the IA-estimator becomes identical to an integer estimator. The subset Ω_0, therefore, determines the aperture of the pull-in region.

3.2 Performance of IA-estimators

In order to evaluate the performance of an IA-estimator as to whether it produces the correct integer outcome $a \in Z^n$, it is helpful to first classify its possible outcomes. An IA-estimator can produce one of the following three outcomes

$$\hat{a}_{IA} = \begin{cases} a \in Z^n & \text{(correct integer)} \\ z \in Z^n \setminus \{a\} & \text{(incorrect integer)} \\ \hat{a} \in R^n \setminus Z^n & \text{(no integer)} \end{cases} \tag{17}$$

A correct integer outcome may be considered a success, an incorrect integer outcome a failure, and an outcome where no correction at all is given to the float solution as indeterminate or undecided. The probability of success, the success-rate P_S, equals the integral of the pdf of the float solution, $f_{\hat{a}}(x)$, over Ω_a, whereas the probability of failure, the fail-rate P_F, equals the integral of $f_{\hat{a}}(x)$ over $\Omega \setminus \Omega_a$. The probability that the IA-estimator reproduces the float solution equals one minus the sum of the success-rate and the fail-rate. The respective probabilities are therefore given as

$$\begin{cases} P_S = P(\hat{a}_{IA} = a) = \int_{\Omega_a} f_{\hat{a}}(x)dx & \text{(success)} \\ P_F = \sum_{z\neq a} P(\hat{a}_{IA} = z) = \sum_{z\neq a} \int_{\Omega_z} f_{\hat{a}}(x)dx & \text{(failure)} \\ P_U = P(\hat{a}_{IA} = \hat{a}) = 1 - P_S - P_F & \text{(undecided)} \end{cases} \tag{18}$$

Note that these three probabilities are completely governed by $f_{\hat{a}}(x)$, the pdf of the float solution, and by Ω_0, the aperture pull-in region which uniquely defines the IA-estimator. By setting the size and shape of the aperture pull-in region $\Omega_0 \subset S_0$ for a given pdf $f_{\hat{a}}(x)$, the user has now gained some control on the level of the fail-rate which he or she is willing to accept. The result will be that of an integer estimator if Ω_0 is set equal to S_0. Then $\Omega = R^n$ and thus $P_U = 0$, from which it follows that $P_F = 1 - P_S$. In this case, the fail-rate is dictated by the choice of integer estimator and the strength of the underlying mathematical model. However, if Ω_0 is chosen as a true subset of S_0, then the user could set its size such that the fail-rate would still be acceptable even if he or she is confronted with a mathematical model that lacks enough strength.

Depending on the type of IA-estimator one is considering, the above integrals (Eq. 18) for computing the success-rate and the fail-rate may be difficult to evaluate exactly. Whether or not an exact evaluation is possible depends to a large extent on the complexity of the geometry of the aperture pull-in region Ω_0. If the geometry is too complex, one is usually forced to make use of the method of simulation. If one may assume that the float solution is Gaussian-distributed as $\hat{a} \sim \mathrm{N}(a, Q_{\hat{a}})$, the simulation of the fail-rate and the success-rate goes as follows: Since the shape of the Gaussian distribution is independent of the mean a, also P_S and P_F are independent of a. Hence, one may restrict attention to $\mathrm{N}(0, Q_{\hat{a}})$, draw samples from it and use these samples to obtain good approximations to both P_S and P_F.

As a first step one generates, using a random-number generator, n independent samples from the univariate standard normal distribution N(0, 1), say $s_1, \ldots, s_n$. These samples are then collected in the vector $s = (s_1, \ldots, s_n)^{\mathrm{T}}$ and transformed by means of $\hat{a} = Gs$, where the matrix G equals the Cholesky factor of $Q_{\hat{a}}$, i.e. $Q_{\hat{a}} = GG^{\mathrm{T}}$. Hence, $\hat{a}$ is now a sample from $\mathrm{N}(0, Q_{\hat{a}})$. This sample is then used as input for the IA-estimator. The outcome of the IA-estimator is then correct if the output equals the zero vector; it is incorrect if the output equals a nonzero integer vector and the outcome is undecided if the output equals the input. The first case corresponds with $\hat{a} \in \Omega_0$, the second case with $\hat{a} \in \Omega_z$ for some $z \in Z^n \setminus \{0\}$, and the third case with $\hat{a} \notin \Omega_z, \forall z \in Z^n$.

By repeating this process N times, one can count how often the zero vector is given as solution, say N_0-times, and how often a particular nonzero integer vector is given as solution, say N_z-times. An approximation to the required success-rate and the required fail-rate follows then from the relative frequencies as

$$P_S \approx \frac{N_0}{N}, \; P_F \approx \frac{\sum_{z\neq 0} N_z}{N} \tag{19}$$

Note that this procedure requires the evaluation whether or not the generated sample $\hat{a}$ resides in one of the aperture pull-in regions. Since $\Omega_z = \Omega \cap S_z$, with S_z the pull-in region of the chosen integer estimator, this evaluation is done in two steps. First the integer vector corresponding to $\hat{a}$ is computed, say $\check{a}$. Depending on the choice of S_z, this could be based on integer rounding, integer bootstrapping, integer least-squares or any other admissible integer estimator. Then in the second step, the residual $\check{\epsilon} = \hat{a} - \check{a}$ is used to verify whether or not $\check{\epsilon} \in \Omega_0$. Since this procedure has to be repeated N-times, it is

of importance that the integer solution $\breve{a}$ can be computed as efficiently as possible. For the case of GNSS this implies that one should not use the original ambiguities, but instead the transformed and decorrelated ambiguities as obtained with the LAMBDA method. As to the choice of N, we refer to Teunissen (1998b). For more advanced methods of approximating the integrals of P_S and P_F using Monte-Carlo or other methods, we refer to Evans and Schwartz (2000).

4 Integer aperture bootstrapping

4.1 The IA-bootstrapped estimator

In the previous section, we introduced the concept of IA-estimation. As we have seen, the prime difference between integer estimators and IA-estimators is that the latter have pull-in regions which are subsets of the space-filling pull-in regions of the integer estimators. Hence, one can design one's own IA-estimator by simply defining the size and shape of these aperture pull-in regions. Therefore, this also gives one the possibility to choose the integer aperture pull-in region Ω_0 such that the probabilistic evaluation is made easier. Since an exact probabilistic evaluation is possible for integer bootstrapping, one may wonder whether or not it is possible to generalize the principle of bootstrapping to the case of IA-estimation such that an exact evaluation of its probabilistic properties can still be given. This turns out to be the case if one defines the aperture pull-in region as a down-scaled version of the bootstrapped pull-in region.

Definition 3 (Bootstrapped aperture pull-in region) *The pull-in region of the IA-bootstrapped (IAB) estimator is defined as*

$$\Omega_{B,z} = \beta S_{B,z} \tag{20}$$

with

$$\begin{cases} \beta S_{B,z} = \{x \in R^n \mid \frac{1}{\beta}(x - z) \in S_{B,0}\} \\ S_{B,0} \;\; = \{x \in R^n \mid \mid c_i^T L^{-1} x \mid \leq \frac{1}{2}\,, i = 1, \ldots, n\} \end{cases}$$

where β $(0 \leq \beta \leq 1)$ is the aperture parameter, and L is the unique unit lower triangular matrix of the triangular decomposition $Q_{\hat{a}} = LDL^{\mathrm{T}}$.

The shape of the pull-in region $\Omega_{B,z}$ is thus identical to that of the bootstrapped pull-in region $S_{B,z}$; only their sizes may differ. By varying the aperture parameter β, one varies the size of $\Omega_{B,z} \subset S_{B,z}$ (see Fig. 2). Since both $\Omega_{B,z}$ and $S_{B,z}$ have the same shape, the computation of the IA-bootstrapped estimator is almost as simple as that of the original bootstrapped estimator. The computational steps are as follows: As before, one starts with the float solution $\hat{a}$ and computes the bootstrapped solution $\breve{a}_B$. This result identifies the aperture pull-in region $\Omega_{B,\breve{a}_B}$ for which it needs to be verified whether or not the float solution resides in it. Note that this verification is equivalent to the verification whether or not $\frac{1}{\beta}(\hat{a} - \breve{a}_B) \in S_{B,0}$. Thus, from $\hat{a}$ and $\breve{a}_B$, one forms the bootstrapped ambiguity residual $\breve{\epsilon}_B = \hat{a} - \breve{a}_B$, up-scales it to $\frac{1}{\beta}\breve{\epsilon}_B$, and verifies whether this up-scaled version still resides in $S_{B,0}$. This is done by using the same bootstrapping procedure as before, but now applied to the input $\frac{1}{\beta}\breve{\epsilon}_B$. If the outcome is the zero vector, then $\hat{a}_{IAB} = \breve{a}_B$, otherwise $\hat{a}_{IAB} = \hat{a}$.

The conclusion therefore is that the computation of the IAB-estimator is thus very simple indeed. It essentially consists of applying the bootstrapped procedure twice, once to the float solution $\hat{a}$ and once to the upscaled residual $\frac{1}{\beta}\breve{\epsilon}_B$.

4.2 Fail-rate and success-rate of IA-bootstrapping

Apart from the ease with which the IAB-estimator can be computed, it also has the advantage that analytical closed form expressions can be given for its fail-rate P_F and its success-rate P_S. They are given in the following theorem.

Theorem 2 (IAB-probabilities of success, failure and undecided) *Let the float solution be distributed as $\hat{a} \sim \mathrm{N}(a, Q_{\hat{a}})$ and let its vc-matrix have the unique triangular factorization $Q_{\hat{a}} = LDL^{\mathrm{T}}$, with L a unit lower triangular matrix. The IAB-probabilities of failure, P_F, success, P_S, and undecided, P_U, are then given as*

$$\begin{cases} P_F = \sum_{z \in Z^n \setminus \{0\}} \prod_{i=1}^{n} \left(\Phi\left(\frac{\beta - 2c_i^{\mathrm{T}} L^{-1} z}{2\sigma_{i|I}}\right) + \Phi\left(\frac{\beta + 2c_i^{\mathrm{T}} L^{-1} z}{2\sigma_{i|I}}\right) - 1 \right) \\ P_S = \prod_{i=1}^{n} \left(2\Phi\left(\frac{\beta}{2\sigma_{i|I}}\right) - 1 \right) \\ P_U = 1 - P_F - P_S \end{cases} \tag{21}$$

with β being the aperture parameter $(0 \leq \beta \leq 1)$, $\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp\{-\frac{1}{2}v^2\}\mathrm{d}v$, and where c_i denotes the canonical unit vector having as its ith entry a 1 and zeros otherwise.

Proof We will only prove P_F as the proof of P_S goes along similar lines. The fail-rate P_F equals the integral of the pdf of the float solution over all aperture pull-in regions $\Omega_z = \beta S_{B,z}$ except the one for which $z = a$. We therefore have

$$\begin{aligned} P_F &= \sum_{z \in Z^n \setminus \{a\}} \int_{\beta S_{B,z}} f_{\hat{a}}(x)\mathrm{d}x \\ &= \sum_{z \in Z^n \setminus \{0\}} \int_{S_{B,0}} \frac{1}{(2\pi)^{\frac{1}{2}n} \sqrt{\det(\frac{1}{\beta^2} Q_{\hat{a}})}} \\ &\quad \times \exp\left\{ -\frac{1}{2} \parallel x + \frac{1}{\beta} z \parallel^2_{\frac{1}{\beta^2} Q_{\hat{a}}} \right\} \mathrm{d}x \end{aligned}$$

We will now transform the integral into a simpler form. As the transformation we choose $F : x = Ly$, with L as the unit lower triangular matrix of $Q_{\hat{a}} = LDL^{\mathrm{T}}$. Then

$$\begin{aligned} P_F = \sum_{z \in Z^n \setminus \{0\}} \int_{F^{-1}(S_{B,0})} & \frac{1}{(2\pi)^{\frac{1}{2}n} \sqrt{\det(\frac{1}{\beta^2} D)}} \\ & \exp\left\{ -\frac{1}{2} \parallel y + \frac{1}{\beta} L^{-1} z \parallel^2_{\frac{1}{\beta^2} D} \right\} \mathrm{d}y \end{aligned}$$

with the transformed pull-in region

$$F^{-1}(S_{B,0}) = \{y \in R^n \mid \mid c_i^{\mathrm{T}} y \mid \leq \frac{1}{2}\,, i = 1, \ldots, n\}$$

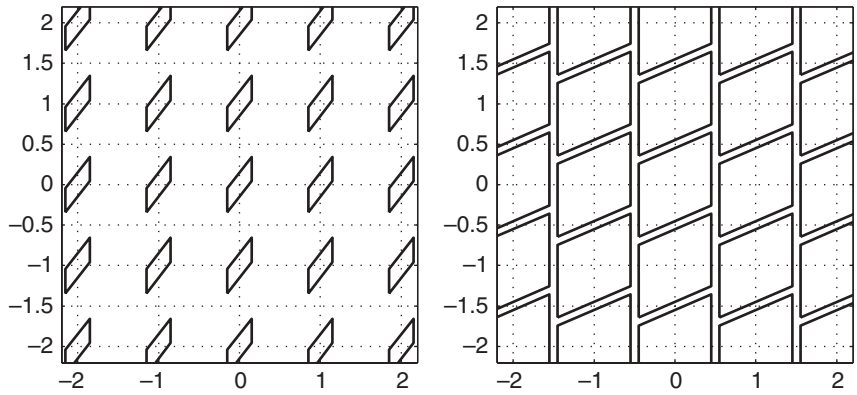

Fig. 2 Two sets of two-dimensional IA-bootstrapped pull-in regions. *left* high correlation, small β; *right* low correlation, large β

Recognizing that $\frac{1}{\beta^2}D$ is a diagonal matrix having the scaled conditional variances $\frac{1}{\beta^2}\sigma^2_{i|I}$ as its entries, and that the transformed pull-in region has become an origin-centred n-dimensional hyper-cube with all sides equal to 1, we may write the multivariate integral as a product of one-dimensional integrals. This gives

$$P_F = \sum_{z\in Z^n\setminus\{0\}} \prod_{i=1}^{n} \int_{|y_i|\le\frac{1}{2}} \frac{1}{\frac{1}{\beta}\sigma_{i|I}\sqrt{2\pi}} \times \exp\left\{-\frac{1}{2}\left(\frac{y_i + \frac{1}{\beta}c_i^T L^{-1} z}{\frac{1}{\beta}\sigma_{i|I}}\right)^2\right\} \mathrm{d}y$$

$$= \sum_{z\in Z^n\setminus\{0\}} \prod_{i=1}^{n} \int_{-\frac{\beta-2c_i^T L^{-1}z}{2\sigma_{i|I}}}^{\frac{\beta+2c_i^T L^{-1}z}{2\sigma_{i|I}}} \frac{1}{\sqrt{2\pi}} \exp\{-\frac{1}{2}v^2\}\mathrm{d}v$$

from which, when using the definition of $\Phi(x)$, the result follows. □

We recall that the performance of integer bootstrapping is dependent on the chosen ambiguity parametrization. The same holds true for IA-bootstrapping. Also this estimator should therefore only be applied in combination with the decorrelating Z-transformation of the LAMBDA method.

Note that the above result reduces to that of the bootstrapped estimator when the aperture parameter is set equal to 1. In that case, P_S becomes identical to the success-rate of bootstrapping and $P_F = 1 - P_S$, since $P_U = 0$. This maximum value of β is acceptable if the corresponding fail-rate P_F is at an acceptably low level. This will be the case when the underlying mathematical model has enough strength, and when the proper ambiguity parametrization has been chosen. In this case, one will accept $\breve{a}_B$ as the integer outcome to work with. However, in case that the fail-rate turns out to be too large when $\beta = 1$ is chosen, one will not automatically accept $\breve{a}_B$ as the integer outcome. In this case, one has now two ways to proceed: either one ignores the ambiguity resolution process altogether and simply uses the real-valued float solution to work with, or one uses the principle of IA estimation and sets the value of the aperture parameter at an acceptably smaller value. With the first approach, one will always be working with the real-valued float solution, even though it is known that the parameters to be estimated are integers, while with the second approach one uses the actual data themselves to decide whether to use $\hat{a}$ or $\breve{a}_B$. The first approach is model-driven and somewhat conservative, while the second IA approach is data-driven. With the IA approach one can still have an integer outcome with a fail-rate controlled by the user.

5 Conclusions

In this contribution we introduced a new GNSS ambiguity estimator with controllable fail-rate. This new estimator, referred to as the integer aperture bootstrap (IAB) estimator, combines the advantages of integer bootstrapping and IA estimation. Integer bootstrapping is known to be one of the simplest methods for integer ambiguity estimation with close to optimal performance. Its outcome is easy to compute due to the absence of an integer search, and its performance is close to optimal if the decorrelating Z-transformation of the LAMBDA method is used. Moreover, the bootstrapped estimator is the only integer estimator for which an exact and easy-to-compute expression of its fail-rate can be given. A disadvantage of the method is, however, that the user has only a limited control over its fail-rate. Once the underlying mathematical model is given, the user has no freedom left in changing the value of the fail-rate. This disadvantage is eliminated, however, with integer aperture estimation, where the user is given the freedom to set the aperture of the pull-in region so as to achieve a pre-specified failure-rate.

The IAB-estimator of the unknown integer ambiguity vector a is defined as

$$\hat{a}_{IAB} = \hat{a} + \sum_{z\in Z^n} (z - \hat{a})\omega_z(\hat{a}) \ , \ \text{with } \omega_z(\hat{a})$$

$$= \begin{cases} 1 & \text{if } \hat{a} \in \beta S_{B,z} \\ 0 & \text{if } \hat{a} \notin \beta S_{B,z} \end{cases}$$

where $S_{B,z}$ is the bootstrap pull-in region and β ($0 \leq \beta \leq 1$) is the aperture parameter. The IAB-estimator returns the float solution $\hat{a}$, if $\hat{a} \neq \beta S_{B,z}$ for all $z \in Z^n$; otherwise it returns the integer bootstrapped solution $\breve{a}_B$. By setting the size of β, the user can control the fail-rate, i.e. the probability that $\breve{a}_B$ fails to be equal to the correct, but unknown integer ambiguity vector a. Exact and analytical closed form expressions of the fail-rate and the success-rate have been given for evaluating these probabilities.

As with integer bootstrapping, the computation of the IAB-estimator is straightforward, and one starts with the float solution $\hat{a}$ and then computes the bootstrapped solution $\breve{a}_B$. This result identifies the aperture pull-in region $\beta S_{B,\breve{a}_B}$ for which it needs to be verified whether or not the float solution resides in it. This verification is equivalent to the verification whether or not $\frac{1}{\beta}(\hat{a} - \breve{a}_B) \in S_{B,0}$. Thus from $\hat{a}$ and $\breve{a}_B$, one forms the bootstrapped ambiguity residual $\breve{\epsilon}_B = \hat{a} - \breve{a}_B$, up-scales it to $\frac{1}{\beta}\breve{\epsilon}_B$ and verifies whether this up-scaled version still resides in $S_{B,0}$. This is done by using the same bootstrapping procedure as before, but now applied to the input $\frac{1}{\beta}\breve{\epsilon}_B$. If the outcome is the zero vector, then $\hat{a}_{IAB} = \breve{a}_B$, otherwise $\hat{a}_{IAB} = \hat{a}$. The computation of the IAB-estimator thus essentially consists of applying the bootstrapping procedure twice, once to the float solution $\hat{a}$ and once to the upscaled residual $\frac{1}{\beta}\breve{\epsilon}_B$.

References

Boon F, Ambrosius B (1997) Results of real-time applications of the LAMBDA method in GPS based aircraft landings. In: Proceedings KIS97, pp 339–345

Boon F, de Jonge PJ, Tiberius CCJM (1997) Precise aircraft positioning by fast ambiguity resolution using improved troposphere modelling. In: Proceedings ION GPS-97, vol 2, pp 1877–1884

Cox DB, Brading JDW (1999) Integration of LAMBDA ambiguity resolution with Kalman filter for relative navigation of spacecraft. In: Proceedings ION NTM 99, pp 739–745

de Jonge PJ, Tiberius CCJM (1996a) The LAMBDA method for integer ambiguity estimation: implementation aspects. Publications of the Delft Computing Centre, LGR-Series No. 12

de Jonge PJ, Tiberius CCJM (1996b) Integer estimation with the LAMBDA method. In: Beutler et al (eds) Proceedings IAG symposium no. 115, GPS trends in terrestrial, airborne and spaceborne applications. Springer, Berlin Heidelberg New York, pp 280–284

de Jonge PJ, Tiberius CCJM, Teunissen PJG (1996) Computational aspects of the LAMBDA method for GPS ambiguity resolution. In: Proceedings ION GPS-96, pp 935–944

Evans M, Schwartz T (2000) Approximating integrals via Monte Carlo and deterministic methods, Oxford University Press, New York

Farrell JA, Barth M (1999) The global positioning system and inertial navigation. McGraw-Hill, New York

Han S, Rizos C (1996) Integrated method for instantaneous ambiguity resolution using new generation GPS receivers. In: Proceedings IEEE PLANS'96, Atlanta, pp 254–261

Hofmann-Wellenhof B, Lichtenegger H, Collins J (2002) Global positioning system: theory and practice, 5th edn. Springer, Berlin Heidelberg New York

Lee HK, Wang J, Rizos C (2003) An integer ambiguity resolution procedure for GPS/Pseudolite/INS integration. J Geod (in press)

Leick A (1995) GPS satellite surveying. 2nd edn. John Wiley, New York

Misra P, Enge P (2001) Global positioning system: signals, measurements, and performance. Ganga-Jamuna Press

Parkinson B, Spilker JJ (eds) (1996) GPS: theory and applications, vols. 1 and 2. AIAA, Washington

Peng HM, Chang FR, Wang LS (1999) Attitude determination using GPS carrier phase and compass data. In: Proceedings ION NTM-1999. San Diego, pp 727–732

Seeber G (2003) Satellite geodesy, 2nd edn. Walter de Gruyter, Berlin

Strang G, Borre K (1997) Linear algebra, geodesy, and GPS. Wellesley-Cambridge Press, Wellesley

Teunissen PJG, Kleusberg A (eds) (1998) GPS for geodesy, 2nd enlarged edn. Springer, Berlin Heidelberg New York

Teunissen PJG (1993) Least-squares estimation of the integer GPS ambiguities. In: Invited lecture, section IV theory and methodology, IAG general meeting, Beijing, China, August 1993. LGR Series, No. 6, Delft Geodetic Computing Centre

Teunissen PJG (1995) The least-squares ambiguity decorrelation adjustment: a method for fast GPS integer ambiguity estimation. J Geod 70:65–82

Teunissen PJG (1998a) The success probability of integer GPS rounding and bootstrapping. J Geod 72:606–612

Teunissen PJG (1998b) On the integer normal distribution of the GPS ambiguities. Artif Satellites 33(2):49–64

Teunissen PJG (1999) An optimality property of the integer least-squares estimator. J Geod 73:587–593

Teunissen PJG (2000) ADOP based upperbounds for the bootstrapped and the least-squares ambiguity success rates. Artif Satellites 35(4):171–179

Teunissen PJG (2001) GNSS ambiguity bootstrapping: theory and applications. In: Proceedings KIS2001, Banff, Canada, pp 246–254

Teunissen PJG (2002) A new class of GNSS ambiguity estimators. Artif Satellites 37(4):111–120

Teunissen PJG (2003) Integer aperture GNSS ambiguity resolution. Artif Satellites 38(3):79–88

Thomsen HF (2000) Evaluation of upper and lower bounds on the success probability. In: Proceedings ION GPS-2000, Salt Lake City, pp 183–188

Tiberius CCJM, de Jonge PJ (1995) Fast positioning with the LAMBDA method. In: Proceedings DSNS 95, No. 30, Bergen, Norway

Tiberius CCJM, Teunissen PJG, de Jonge PJ (1997) Kinematic GPS: performance and quality control. In: Proceedings KIS97, pp 289–299

Uratani C, Sone K, Muto Y, Maruo S, Sugimoto S (2003) Dynamical models for carrier-phase kinematic GPS positioning. In: Proceedings ION GPS/GNSS-2003, Portland, pp 809–818

Verhagen S (2001) Ambiguity resolution and success rates with an integrated GNSS – pseudolite positioning system. In: Proceedings ION GPS-2001, Salt Lake City, pp 3036–3043

Wei M, Schwarz K-P (1995) Fast ambiguity resolution using an integer nonlinear programming method. In: Proceedings ION GPS-95, Palm Spring, pp 1101–1110

Wu F, Kubo N, Yasuda A (2003) A study of hybrid modernized GPS and Galileo positioning in Japan. J JPN Inst Navigation 109:171–178