



HAL
open science

Calibration of the clock-phase biases of GNSS networks: the closure-ambiguity approach

André Lannes, Jean-Louis Prieur

► **To cite this version:**

André Lannes, Jean-Louis Prieur. Calibration of the clock-phase biases of GNSS networks: the closure-ambiguity approach. *Journal of Geodesy*, 2013, pp.1-23. 10.1007/s00190-013-0641-4 . hal-00830353

HAL Id: hal-00830353

<https://centralesupelec.hal.science/hal-00830353>

Submitted on 5 Jun 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Calibration of the clock-phase biases of GNSS networks: the closure-ambiguity approach

A Lannes¹ and JL Prieur²

¹CNRS/SUPELEC/Univ Paris-Sud, France

Address: Supélec, 2 rue E. Belin, 57070 Metz, France

e-mail: andre.lannes@cerfacs.fr

Fax: 33 3 87 76 47 00

²Université de Toulouse – UPS-OMP – IRAP, Toulouse, France

Address: 14 avenue Edouard Belin, 31400 Toulouse, France

Abstract. In global navigation satellite systems (GNSS), the problem of retrieving clock-phase biases from network data has a basic rank defect. We analyse the different ways of removing this rank defect, and define a particular strategy for obtaining these phase biases in a standard form. The minimum-constrained problem to be solved in the least-squares (LS) sense depends on some integer vector which can be fixed in an arbitrary manner. We propose to solve the problem via an undifferenced approach based on the notion of closure ambiguity. We present a theoretical justification of this closure-ambiguity approach (CAA), and the main elements for a practical implementation. The links with other methods are also established. We analyse all those methods in a unified interpretative framework, and derive functional relations between the corresponding solutions and our CAA solution. This could be interesting for many GNSS applications like real-time kinematic precise point positioning for instance. To compare the methods providing LS estimates of clock-phase biases, we define a particular solution playing the role of reference solution. For this solution, when a phase bias is estimated for the first time, its fractional part is confined to the one-cycle width interval centred on zero; the integer-ambiguity set is modified accordingly. Our theoretical study is illustrated with some simple and generic examples; it could have applications in data processing of most GNSS networks, and particularly global networks using GPS, GLONASS, Galileo, or BeiDou/Compass satellites.

Keywords. Global and regional networks · Clock biases · Uncalibrated phase delays (UPD) · Fractional clock biases (FCB) · Network real-time kinematics (NRTK) · Real-time kinematic precise point positioning (RTK-PPP) · Closure difference (CD) · Nearest lattice point (NLP) · Integer least squares (ILS)

1 Introduction

In global navigation satellite systems (GNSS), the calibration of the clock-phase biases of global networks is a challenging problem. In particular, the knowledge of the satellite clock-phase biases is needed for precise point positioning (PPP); see, e.g., Zumberge et al. 1997; Ge et al. 2008; Bertiger et al. 2010; Geng et al. 2010; Li et al. 2013. In the general context defined below, the equations governing this GNSS calibration problem have a basic rank defect. In this paper, we analyse the different ways of removing this rank defect, and define a particular strategy for obtaining the clock-phase biases in a standard form. The link with other related approaches, such as those proposed by Blewitt (1989), de Jonge (1998), Collins et al. (2010), and Loyer et al. (2012), is established in that framework.

When modelling the multi-frequency (code and phase) observations of GNSS networks, the system to be considered include phase structures of the form

$$\begin{cases} [\beta_{r\kappa}(i) - \beta_{s\kappa}(j)] + N(i, j) = b_{\kappa}(i, j) \\ \text{for } \kappa = 1, \dots, k \end{cases} \quad (1)$$

Here, κ is the epoch index; k is the index of the current epoch; $\beta_{r\kappa}(i)$ and $\beta_{s\kappa}(j)$ are clock-phase biases. These terms are also called ‘uncalibrated phase delays’ (UPD). They are expressed in cycles, and depend on the frequency of the transmitted carrier wave; subscripts r and s stand for receiver and satellite,¹ respectively; i is the index of the receiver, and j that of the satellite; $N(i, j)$ is the integer ambiguity of the corresponding carrier-phase measurement. The terms $b_{\kappa}(i, j)$ include the corresponding phase data and all the other contributions of such equations; see, e.g., Eqs. (1) and (10) of Lannes and Teunissen 2011, and Eqs. (1) and (4) of Loyer et al. 2012. The set of receiver-satellite pairs (i, j) involved in Eq. (1) forms the observational graph \mathcal{H}_{κ} of the GNSS scenario

¹In this paper, satellite should be understood as satellite transmitter.

of epoch κ . This graph is assumed to be connected; see Appendix A. Note that the wide-lane equation of the ionosphere-free mode is typically of form (1); N is then a wide-lane integer ambiguity; see, e.g., Eq. (4) of Loyer et al. 2012.

As explicitly clarified further on, whenever phase structures such as (1) appear in GNSS-network problems, a related rank defect is to be removed. In this paper, we restrict ourselves to the rank defects induced those phase structures. This does not mean of course that those basic rank defects are the only ones to be handled in practice; see, in particular, Teunissen and Odijk (2003). A standard approach for tackling the rank defects is known as the S-system approach of Baarda 1973, Teunissen 1984, de Jonge 1998. Examples of such S-system solutions are to be found in de Jonge 1998; Teunissen et al. 2010; Zhang et al. 2011; Odijk et al. 2012.

In the geodetic and GNSS literature, there exist several ways of removing this basic rank defect. The most general approach is based on the S-system theory already mentioned. Other strategies derive from the pioneering contribution of Blewitt (1989): the relationship between the undifferenced (UD) ambiguities and the double differenced (DD) ambiguities is completed so that the operator D thus defined is invertible. Let us also mention the approach of Collins et al. (2010) which is based on the concept of ‘ambiguity datum fixing.’ The important developments of those approaches, both at a conceptual and technical level, were often conducted with different physical objectives. They have thus progressively and insidiously masked the fundamental links between the related methods.

Briefly, the Blewitt procedure can be divided in three steps. In the first step, with regard to Eq. (1) for example, the UD data are processed by considering the term on the left-hand side of that equation as a ‘constant functional variable;’ a float estimate of this ‘biased-ambiguity variable’ is thus obtained. In the second step, the corresponding DD ambiguities are computed, and then fixed at integer values. In the third step, the clock-phase biases β_κ are estimated by using as data the UD ambiguities provided by the action of D^{-1} on the column matrix formed by those fixed ambiguities. The theoretical analysis developed in the present paper provides in particular an answer to the following question: what is the link between the UD ambiguities thus fixed and the fixed ‘closure-delay’ or ‘closure-difference’ (CD) ambiguities of the UD approach of Lannes and Teunissen (2011)? A similar question arises for the UD approach of Collins et al. (2010); an answer is also provided.

In this general GNSS context, the main objective of the paper is to present a unified interpretative framework in which the various contributions in the related fields of research can be understood and compared more easily. This can lead to improvements of some related methods. For example, we show that removing the rank defect via the D -matrix of Blewitt (1989) can be analysed

in a theoretical framework tightly linked to the S-system approach of Teunissen (1984). We thus show that the intermediate differencing stage of the Blewitt approach can be avoided, without any counterpart, via the approach of Teunissen (1984) as it is formulated for example in Lannes and Teunissen (2011): the ‘closure ambiguities’ to be fixed then appear, from the outset, in the very formulation of the UD problem to be solved; compare with what is done in Sect. 4 of Ge et al. (2005) for instance.

The theoretical guidelines of this paper are presented in Sect. 2. We first identify the rank defect in question. The minimum-constrained problem to be solved in the least-squares (LS) sense depends on some integer vector which can be fixed in an arbitrary manner. To compare the methods providing LS estimates of the clock-phase biases, we then introduce a particular solution playing the role of reference solution. For this solution, when a clock-phase bias is estimated for the first time, its fractional part is confined to the one-cycle width interval centred on zero; the integer-ambiguity set is modified accordingly. Section 3 is devoted to the algebraic framework of our analysis. This framework mainly derives from the original contributions of Lannes and Gratton (2009), and Lannes and Teunissen (2011). As a similar problem arises in phase-closure imaging in astronomy, we also took profit of the analysis presented in Lannes and Prieur (2011). A natural way for finding the reference solution is to adopt an approach based on the notion of closure ambiguity. The principle of the corresponding ‘closure-ambiguity approach’ (CAA) is defined in that framework (Sect. 4). The bulk of our contribution follows the main theoretical guidelines presented in Sect. 2. In a related option which is presented in Sect. 5, the CAA principle is directly introduced via the S-system approach of Baarda (1973), Teunissen (1984) and de Jonge (1998). The corresponding development is performed in the S-system framework defined in Appendix B. The study developed in Sects. 3.3, 3.4 and 3.15 of de Jonge (1998) is thus extended to the cases where the union of the graphs \mathcal{H}_κ is taken into account progressively. Section 6 is devoted to the QR implementation of the CAA principle; related information is to be found in Appendix C. In many methods, the rank defect in question is removed in an implicit manner or intuitively. In Sect. 7, on the grounds of some results established in Sects. 3.5 and 3.6, we identify the related constraints explicitly, and thus establish the link between the solutions provided by those methods and the CAA-(S-System) solutions; see Fig. 6 in particular.

Our analysis is illustrated with some simple and generic examples. It could have applications in data processing of most GNSS networks, and particularly global networks using GPS, GLONASS, Galileo, or BeiDou/Compass satellites. The main results provided by this study are commented upon in Sect. 8; some conclusions are also presented with possible applications to software packages used for processing GNSS networks.

2 Theoretical guidelines

The problem is formulated in Sect. 2.1; the related rank defect is identified in Sect. 2.2. This rank defect can be removed by imposing some constraints without affecting the GNSS results such as the estimates of the station-position parameters, for example. The particular LS solutions thus obtained are defined in Sect. 2.3. We then define the family of those solutions (Sect. 2.4). To compare the particular solutions given by the various GNSS methods providing LS estimates of clock-phase biases, we then introduce a particular solution playing the role of reference solution (Sect. 2.5).

2.1 Formulation of the problem

In our formulation of the problem, the components of the ambiguity vector N are the integer ambiguities $N(i, j)$ involved in the phase measurements until the current epoch. We thereby assume that the time-invariant property of the ambiguities holds. Regarded as a function, N therefore takes its values on the edges of

$$\mathcal{G}_k \stackrel{\text{def}}{=} \bigcup_{\kappa=1}^k \mathcal{H}_\kappa \quad (2)$$

where \mathcal{H}_κ is the observational graph of epoch κ . In what follows, H_k denotes the ‘characteristic function’ of \mathcal{H}_κ with regard to \mathcal{G}_k :

$$\text{for all } (i, j) \in \mathcal{G}_k, \quad H_k(i, j) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } (i, j) \in \mathcal{H}_\kappa; \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

The number of edges (r_i, s_j) of \mathcal{H}_κ is denoted by $n_{e\kappa}$; $n_{e\kappa}$ is less than or equal to the number of edges of \mathcal{G}_k .

To illustrate our analysis, we consider a ‘simulated network’ including four receivers and five to eight satellites; see Fig. 1. The scenarios of the first three epochs are defined by the characteristic functions H_1 , H_2 and H_3 displayed in that figure. While looking simple at first sight, this example is rather elaborate. Indeed, it includes the case of the appearance of new satellites in the field of view of the network (s_6 and s_7 at epoch 2, s_8 at epoch 3), and also the case of the disappearance of one satellite (s_3 at epoch 3).

Remark 2.1. When a satellite comes back in the field of view of the network, it is dealt with as a new satellite. In the case of global networks, if need be, the successive passes are thus dealt with in a simple manner \square

At epoch $\kappa \leq k$, only some components of N may be active; see Fig. 1. To formalize this point, we introduce the operator \mathcal{R}_κ^e that restricts N (which is defined on the edges of \mathcal{G}_k) to the edges of \mathcal{H}_κ :

$$\text{for all } (i, j) \in \mathcal{H}_\kappa, \quad (\mathcal{R}_\kappa^e N)(i, j) \stackrel{\text{def}}{=} N(i, j) \quad (4)$$

	s ₁	s ₂	s ₃	s ₄	s ₅			
r ₁	1	1	.	1	.			
r ₂	.	1	1	.	.			
r ₃	1	.	1	1	1			
r ₄	.	.	1	1	.			
	s ₁	s ₂	s ₃	s ₄	s ₅	s ₆	s ₇	
r ₁	1	1	.	1	.	.	1	
r ₂	.	1	0	.	.	1	1	
r ₃	1	1	0	1	1	1	.	
r ₄	.	.	1	1	1	.	.	
	s ₁	s ₂	s ₃	s ₄	s ₅	s ₆	s ₇	s ₈
r ₁	1	1	.	1	.	.	1	.
r ₂	.	1	0	.	.	1	1	1
r ₃	0	1	0	1	1	1	.	.
r ₄	.	.	0	1	1	.	1	1

Figure 1: Characteristic functions of \mathcal{H}_k with regard to \mathcal{G}_k for $k = 1, 2, 3$ (example). From top to bottom, H_1 , H_2 , and H_3 . The dots define the edges (r_i, s_j) for which no data have been obtained until epoch k included. Here, $n_{e1} = 11$, $n_{e2} = 15$, and $n_{e3} = 16$. By definition, \mathcal{G}_k is the union of the observational graphs until epoch k included. The number of the edges of \mathcal{G}_k is 11 at epoch 1, 17 at epoch 2, and 20 at epoch 3. Six edges appear at epoch 2: (r_1, s_7) , (r_2, s_6) , (r_2, s_7) , (r_3, s_2) , (r_3, s_6) and (r_4, s_5) ; two edges disappear: (r_2, s_3) and (r_3, s_3) . Note that satellites s_6 and s_7 are then detected by the network. Three edges appear at epoch 3: (r_2, s_8) , (r_4, s_7) and (r_4, s_8) ; two edges disappear: (r_3, s_1) and (r_4, s_3) . Satellite s_3 then disappears. At each epoch, the large-sized numbers define the edges of $\mathcal{G}_{st,k}$, the selected spanning tree of \mathcal{G}_k ; see Fig. 5 further on.

Equation (1) can then be written in the form

$$\begin{cases} B_\kappa \beta_\kappa + \mathcal{R}_\kappa^e N = b_\kappa \\ \text{for } \kappa = 1, \dots, k \end{cases} \quad (5)$$

where B_κ is the following bias operator:

$$(B_\kappa \beta_\kappa)(i, j) \stackrel{\text{def}}{=} \beta_{r\kappa}(i) - \beta_{s\kappa}(j) \quad (\text{for all } (i, j) \in \mathcal{H}_\kappa) \quad (6)$$

In what follows, we will assume that Receiver 1 defines the reference for the receiver and satellite biases:

$$\beta_{r\kappa}(1) = 0 \quad (\kappa = 1, \dots, k) \quad (7)$$

This is commonly used by the GNSS investigators for removing the rank defect of operators such as B_κ . The number $n_{b\kappa}$ of phase biases of epoch κ to be estimated

is therefore equal to $n_{v\kappa} - 1$ where $n_{v\kappa}$ is the number of vertices of \mathcal{H}_κ :

$$n_{b\kappa} = n_{v\kappa} - 1 \quad (n_{v\kappa} = n_{r\kappa} + n_{s\kappa}) \quad (8)$$

With regard to its functional variables β_1, \dots, β_k and N , Eq. (5) proves to have a basic rank defect. We now specify this point.

2.2 Identification of the rank defect

For clarity, let us set

$$\mathcal{G} \stackrel{\text{def}}{=} \mathcal{G}_k \quad (9)$$

At epoch k , the number of ambiguities $N(i, j)$ involved in the problem is equal to the number of edges of \mathcal{G} (for example twenty in Fig. 1 for $k = 3$). Again, for clarity, this number is simply denoted by n_e . We then set

$$n_{st} = n_v - 1 \quad (n_v = n_r + n_s) \quad (10)$$

where n_v is the number of vertices of \mathcal{G} ; n_r and n_s are the number of receivers and satellites (respectively) involved in that graph (four and eight in Fig. 1 for $k = 3$). As specified in Sect. A2, n_{st} is the number of edges of any spanning tree \mathcal{G}_{st} of \mathcal{G} . The total number of phase biases to be estimated at epoch k , $\sum_{\kappa=1}^k n_{b\kappa}$, is generally much larger than n_{st} ; see Eqs. (8) and (10). The part played by the vertices of \mathcal{G} is not obvious. We now show that n_{st} defines the ‘size’ of the rank defect in question.

Let us denote by B the operator from $\mathbb{R}^{n_{st}}$ into \mathbb{R}^{n_e} defined by the relation

$$(B\alpha)(i, j) \stackrel{\text{def}}{=} \alpha_r(i) - \alpha_s(j) \quad (\text{for all } (i, j) \in \mathcal{G}) \quad (11)$$

Denoting by μ any integer-valued function taking its values on the vertices of \mathcal{G} other than the reference receiver, we have

$$\mathcal{R}_\kappa^e B\mu = B_\kappa \mathcal{R}_\kappa^v \mu \quad (12)$$

where $\mathcal{R}_\kappa^v \mu$ is the restriction of μ to the vertices of \mathcal{H}_κ (other than the reference receiver). Note that μ can be regarded as a vector of $\mathbb{Z}^{n_{st}}$. It then follows from Eq. (5) that for any μ in $\mathbb{Z}^{n_{st}}$,

$$\begin{cases} B_\kappa(\beta_\kappa + \mathcal{R}_\kappa^v \mu) + \mathcal{R}_\kappa^e(N - B\mu) = b_\kappa \\ \text{for } \kappa = 1, \dots, k \end{cases} \quad (13)$$

Via the operators B_κ , \mathcal{R}_κ^v , \mathcal{R}_κ^e and B , any variation of the ‘vertex-ambiguity’ vector μ can thus be compensated by a variation of the ‘edge-ambiguity’ vector N . As a result, with regard to the bias and ambiguity variables, Eq. (5) is not of full rank. The dimension of the rank defect is equal to that of vector μ , i.e., n_{st} .

2.3 Particular LS solutions

In GNSS, for the reasons specified in Remark 2.2 (at the end of this section), each clock-phase bias is to be estimated up to a constant integer. As a result, the choice of μ in $\mathbb{Z}^{n_{st}}$ does not affect the significant part of the values of the bias functions

$$w_\kappa \stackrel{\text{def}}{=} \beta_\kappa + \mathcal{R}_\kappa^v \mu \quad (\kappa = 1, \dots, k) \quad (14)$$

to be estimated; see Eq. (13). The ambiguity vector to be retrieved

$$v \stackrel{\text{def}}{=} N - B\mu \quad (15)$$

is of course affected by this choice, but this has no actual GNSS impact. As a result, the GNSS methods providing estimates of the clock-phase biases must remove the rank defect of Eq. (5) by choosing μ in $\mathbb{Z}^{n_{st}}$ somehow, implicitly or explicitly.

In practice, as clarified in the remainder of the paper, removing this rank defect amounts to imposing n_{st} constraints on some values of the biases or ambiguities to be retrieved. In other words μ is defined via these constraints. The minimum-constrained problem to be solved in the LS sense is therefore of the form

$$\begin{cases} B_\kappa w_\kappa + \mathcal{R}_\kappa^e v = b_\kappa & (\kappa = 1, \dots, k) \\ \text{subject to } n_{st} \text{ constraints on } w_\kappa \text{ or } v \end{cases} \quad (16)$$

With regard to a particular set of such constraints, where v is an integer-valued function from Eq. (15), the LS solution of Eq. (16),

$$(\check{w}_1, \dots, \check{w}_k; \check{v}) \quad (17)$$

is then unique. For example, the solution provided by the CAA method defined in Sect. 4 is the particular LS solution obtained by imposing the *a priori* constraint $v = 0$ on a spanning tree of \mathcal{G} (chosen arbitrarily). The particular LS solution introduced in Sect. 2.5 is defined by imposing, *a posteriori*, n_{st} constraints on some bias values. In our analysis, this particular solution plays the role of reference solution; it is denoted by $(\bar{w}_1, \dots, \bar{w}_k; \bar{v})$.

Remark 2.2. The satellite components of the biases thus obtained (for example those of the reference solution) can be broadcasted to the network users for PPP applications. The fact that $\check{w}_{s\kappa}(j)$ is an LS estimate of $\beta_{s\kappa}(j)$ up to some unknown constant integer does not raise any difficulty. One is then simply led to redefine the integer ambiguities involved in the PPP problem to be solved; see, e.g., Sect. 9 in Lannes and Teunissen 2011 \square

2.4 Equivalent LS solutions

Given some particular LS solution such as (17), we have

$$B_\kappa \check{w}_\kappa + \mathcal{R}_\kappa^e \check{v} \stackrel{\text{LS}}{=} b_\kappa$$

Like for Eq. (13), it then follows from Eq. (12) that

$$B_\kappa(\check{w}_\kappa + \mathcal{R}_\kappa^v \mu) + \mathcal{R}_\kappa^e(\check{v} - B\mu) \stackrel{\text{LS}}{=} b_\kappa$$

The LS solutions of Eq. (5) are therefore of the form

$$(\check{w}_1^{(\mu)}, \dots, \check{w}_k^{(\mu)}; \check{v}^{(\mu)}) \quad (18)$$

with

$$\check{w}_\kappa^{(\mu)} \stackrel{\text{def}}{=} \check{w}_\kappa + \mathcal{R}_\kappa^v \mu, \quad \check{v}^{(\mu)} \stackrel{\text{def}}{=} \check{v} - B\mu \quad (19)$$

where μ is any vector of $\mathbb{Z}^{n_{\text{st}}}$.

The methods providing LS estimates of the phase biases generally differ by the choice of the imposed constraints. To compare their results, it is convenient to represent the equivalent solutions (18)-(19) by a reference particular solution. This is done in Sect. 2.5.

Remark 2.3. For any fixed μ in $\mathbb{Z}^{n_{\text{st}}}$, the temporal variations of the estimated phase biases make sense. For example, if satellite s_j remains in the field of view of the network from epoch 1 to κ , we have

$$(\mathcal{R}_\kappa^v \mu)_s(j) = (\mathcal{R}_1^v \mu)_s(j) = \mu_s(j)$$

hence from Eqs. (19) and (14),

$$\begin{aligned} \check{w}_{s\kappa}^{(\mu)}(j) - \check{w}_{s1}^{(\mu)}(j) &= \check{w}_{s\kappa}(j) - \check{w}_{s1}(j) \\ &\simeq \beta_{s\kappa}(j) - \beta_{s1}(j) \end{aligned}$$

A similar result of course holds for the receiver clock-phase biases \square

2.5 Reference solution

We here concentrate on the family of equivalent LS solutions (18)-(19) generated by a particular solution such as (17): $(\check{w}_1, \dots, \check{w}_k; \check{v})$. In our analysis, the reference solution of this family is the particular solution

$$(\bar{w}_1, \dots, \bar{w}_k; \bar{v}) \quad (20)$$

defined as follows: \bar{w}_κ and \bar{v} are of the form (19)

$$\bar{w}_\kappa \stackrel{\text{def}}{=} \check{w}_\kappa + \mathcal{R}_\kappa^v \check{\mu}, \quad \bar{v} \stackrel{\text{def}}{=} \check{v} - B\check{\mu} \quad (21)$$

in which $\check{\mu}$ is defined by imposing specific constraints on n_{st} bias values; note that here, these constraints are imposed *a posteriori* on the solution $(\check{w}_1, \dots, \check{w}_k; \bar{v})$ provided by any method. We first require the phase bias \bar{w} to be small at epoch 1. More precisely, we impose the condition $|\bar{w}_1| \leq 1/2$, i.e. explicitly,

$$\begin{cases} |\bar{w}_{r1}(i)| \leq 1/2 & \text{for } i = 2, \dots, n_{r1} \\ |\bar{w}_{s1}(j)| \leq 1/2 & \text{for } j = 1, \dots, n_{s1} \end{cases} \quad (22)$$

The following values of $\check{\mu}$ are defined accordingly:

$$\begin{cases} \check{\mu}_{r1}(i) := -\lfloor \check{w}_{r1}(i) \rfloor & \text{for } i = 2, \dots, n_{r1} \\ \check{\mu}_{s1}(j) := -\lfloor \check{w}_{s1}(j) \rfloor & \text{for } j = 1, \dots, n_{s1} \end{cases} \quad (23)$$

Here, $\lfloor x \rfloor$ denotes the integer closest to x . Likewise, at each epoch κ when some satellite(s) s_j appear(s) in the field of view of the network (see Fig. 1), we then impose the condition(s)

$$|\bar{w}_{s\kappa}(j)| \leq 1/2 \quad (24)$$

by setting

$$\check{\mu}_{s\kappa}(j) := -\lfloor \check{w}_{s\kappa}(j) \rfloor \quad (25)$$

(In the case where new receivers would be activated, similar conditions would be imposed.) At epoch k , we have thus completely defined some vector $\check{\mu}$ of $\mathbb{Z}^{n_{\text{st}}}$; \bar{v} is then obtained via the relation $\bar{v} := \check{v} - B\check{\mu}$; see Eq. (21).

Remark 2.4. When some LS solution $(\check{w}_1, \dots, \check{w}_k; \check{v})$ has been found, for instance that provided by the CAA method defined in Sects. 4 to 6, the reference solution of its equivalent solutions is obtained as described in this section. Clearly, this can also be done for the LS solution of any method providing estimates of the phase biases; see Sect. 7 together with, e.g., Blewitt 1989; Ge et al. 2005; Laurichesse and Mercier 2007; Collins et al. 2010; and Loyer et al. 2012. To compare and validate the results provided by all these methods (and many others), one may inspect the ambiguity sets of their reference solutions. These reference ambiguity solutions should be identical on all the edges of \mathcal{G} for all methods; otherwise, this would be an indication that the methods are in disagreement, and that some of those results are wrong. The comparison of the reference solutions is therefore a good diagnosis for testing the compatibility of these methods \square

Remark 2.5. From a technical point of view, one might try to solve Eq. (16) in the LS sense by imposing the nonlinear bias constraints (22) and (24) on w_1 and some w_κ , from the outset. It is not easy at all to solve the problem that way. Moreover, the number of edge ambiguities to be fixed would then be equal to n_e , whereas the number of ambiguities to be fixed in the CAA approach (for example) is equal to $n_e - n_{\text{st}}$; see Sects. 4 and 6 \square

3 Algebraic framework

The preliminary analysis developed in Sect. 2 shows that graph \mathcal{G} , operator B , and $\mathbb{Z}^{n_{\text{st}}}$ play a key role in the formulation of the problem and the definition of its solutions; see, in particular, Eqs. (13) and (18)-(19). The aim of this section is to define the corresponding algebraic framework.

We first define related spaces of functions (Sect. 3.1). The key property on which our analysis is based is presented in Sect. 3.2. The related notions of closure difference, CD ambiguity (also called closure ambiguity), and closure matrix are specified in Sects. 3.3 and 3.4. Sections 3.5 and 3.6 are devoted to some generalized inverses

of the UD-CD and UD-DD relationships. The analysis concerning the operator D introduced in the appendix B of Blewitt (1989) is thus completed. We now draw freely from the elementary notions introduced in Appendix A.

3.1 Reference spaces

Given some graph $\mathcal{G} \equiv \mathcal{G}(\mathcal{V}, \mathcal{E})$, with vertex set \mathcal{V} and edge set \mathcal{E} (see Sect. A1), we introduce some functionals spaces which play a key role in the algebraic analysis of the problem. In what follows, the GNSS grid associated with \mathcal{G} is denoted by G ; see Fig. A1.

3.1.1 Vertex-bias space

Let V_b be the space of real-valued functions

$$\alpha \stackrel{\text{def}}{=} (\alpha_r, \alpha_s) \quad (26)$$

taking their values on the vertices of \mathcal{G} with $\alpha_r(1) = 0$. This space, which is referred to as the vertex-bias space, is associated with the definition of (virtual) phase biases α on the vertices of \mathcal{G} (other than the reference receiver). From Eq. (10),

$$V_b \cong \mathbb{R}^{n_{st}} \quad (27)$$

Here, the symbol \cong means ‘isomorphic to.’ Note that $\mathbb{Z}^{n_{st}}$ is the ‘integer lattice’ of V_b : $V_b(\mathbb{Z}) \cong \mathbb{Z}^{n_{st}}$. The integer vector $\mu \stackrel{\text{def}}{=} (\mu_r, \mu_s)$ is a point of this lattice.

3.1.2 Edge-delay space

A real-valued function ϑ taking its values on G , and thereby on \mathcal{E} , can be regarded as a vector of the edge-delay space

$$E \cong \mathbb{R}^{n_e} \quad (28)$$

The values of ϑ on G are then regarded as the components of ϑ in the standard basis of E ; \mathbb{Z}^{n_e} is the ‘integer lattice’ of E : $E(\mathbb{Z}) \cong \mathbb{Z}^{n_e}$. The integer-ambiguity vector N is a point of this lattice.

3.1.3 Spanning-tree delay space. Closure-delay space

Given some spanning tree \mathcal{G}_{st} of \mathcal{G} , grid G can be decomposed into two subgrids: G_{st} and G_c ; see Sect. A2. These grids include n_{st} and n_c points, respectively (see Fig. A2):

$$n_c = n_e - n_{st} \quad (29)$$

The functions of E that vanish on G_c form a subspace of E denoted by E_{st} : the spanning-tree delay space. Likewise, the functions of E that vanish on G_{st} form

a subspace of E denoted by E_c : the closure-delay space; this terminology is justified in Sect. 3.3. The corresponding integer lattices are denoted by $E_{st}(\mathbb{Z})$ and $E_c(\mathbb{Z})$, respectively. As illustrated in Fig. 2, the Euclidean space E is the orthogonal sum of E_{st} and E_c . Clearly,

$$\dim E_{st} = n_{st}, \quad \dim E_c = n_c \quad (30)$$

The orthogonal projections of ϑ on E_{st} and E_c are respectively denoted by $Q_{st}\vartheta$ and $Q_c\vartheta$.

3.1.4 Edge-bias space

By definition, the bias operator is the operator from V_b into E defined by Eq. (11). The range of B , which is denoted by E_b (see Fig. 2), can be referred to as the edge-bias space. Its functions are of the form $\alpha_r(i) - \alpha_s(j)$.

The operator from V_b into E_{st} induced by B is denoted by B_{st} . Likewise, the operator from V_b into E_c induced by B is denoted by B_c .

The matrix of B is generally expressed in the standard bases of V_b and E . For example, let us sort the edges of the graph shown in Fig. A1 in the order obtained via the application of the Kruskal algorithm; see Sect. A2. The points of G are then ordered as follows:

$$\begin{aligned} & (1, 1), (1, 3), (1, 4), (2, 1), (2, 2), (3, 2), \\ & (2, 4), (3, 3), (3, 4) \end{aligned}$$

We then have

$$[B][\alpha] = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \alpha_r(2) \\ \alpha_r(3) \\ \alpha_s(1) \\ \alpha_s(2) \\ \alpha_s(3) \\ \alpha_s(4) \end{bmatrix}$$

The columns of $[B]$ then define the standard basis of E_b . Clearly,

$$[B_{st}] = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \end{bmatrix} \quad (31)$$

and

$$[B_c] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \quad (32)$$

The condition $B_{st}\alpha = 0$, i.e., $B\alpha = 0$ on the edges of \mathcal{G}_{st} , implies that α is constant on \mathcal{V} ; as $\alpha_r(1) = 0$, this constant is zero. The null space of B_{st} is therefore reduced

to $\{0\}$. As $B\alpha = 0$ implies $B_{\text{st}}\alpha = 0$, the null space of B is also reduced to $\{0\}$. We thus have

$$\ker B = \ker B_{\text{st}} = \{0\} \quad (33)$$

As a result, B is of full rank, hence from Eq. (27),

$$\dim E_{\text{b}} = n_{\text{st}} \quad (34)$$

The edge-bias space E_{b} and its ambiguity lattice $E_{\text{b}}(\mathbb{Z}) = BV_{\text{b}}(\mathbb{Z})$ are isomorphic to the vertex-bias space V_{b} and its integer lattice $V_{\text{b}}(\mathbb{Z})$, respectively; see Sect. 3.1.1.

3.2 Key property

As $\ker B_{\text{st}} = \{0\}$ (Eq. (33)), and $\dim E_{\text{st}} = \dim V_{\text{b}}$ (see Eqs. (30) and (27)), B_{st} maps V_{b} onto E_{st} ; B_{st} is therefore invertible. As specified in this section, our analysis derives from this property.

Let us concentrate on the vertex-bias function

$$\alpha_{\text{st}}^{(\vartheta)} \stackrel{\text{def}}{=} B_{\text{st}}^{-1} Q_{\text{st}} \vartheta \quad (\alpha^{(\vartheta)} \equiv \alpha_{\text{st}}^{(\vartheta)}) \quad (35)$$

When no confusion may arise, subscript st is omitted. According to its definition (which is illustrated in Fig. 2), $Q_{\text{st}}\vartheta$ is the function of E_{st} whose values are those of ϑ on subgrid G_{st} .

The values of $\alpha^{(\vartheta)}$ can be obtained from those of $Q_{\text{st}}\vartheta$ in a very simple manner; the corresponding recursive process is described in Sect. 5 of Lannes and Teunissen (2011). The column vectors of $[B_{\text{st}}]^{-1}$ can thus be easily obtained. In fact, $[B_{\text{st}}]$ is a particular unimodular² matrix whose inverse can be obtained via another integer-programming technique; see Sect. A1.4 in Lannes and Teunissen (2011). For example, the inverse of matrix (31) is

$$[B_{\text{st}}]^{-1} = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & -1 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \quad (36)$$

Let us now consider the following edge-bias function:

$$\vartheta_{\text{b}} \stackrel{\text{def}}{=} B\alpha^{(\vartheta)} \quad (37)$$

According to Eq. (35), the values of ϑ_{b} and ϑ coincide on G_{st} . The function ϑ_{c} defined by the relation

$$\vartheta_{\text{c}} \stackrel{\text{def}}{=} \vartheta - \vartheta_{\text{b}} \quad (38)$$

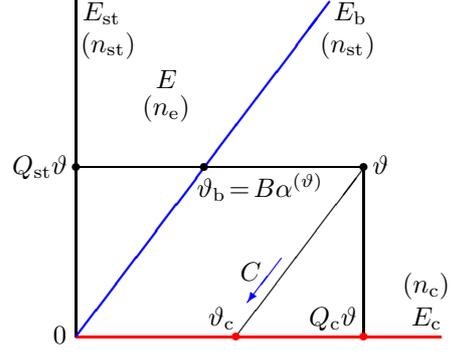


Figure 2: *Geometrical illustration of Property 1.* In this geometrical representation of the edge-delay space $E \cong \mathbb{R}^{n_e}$, E_{st} is the spanning-tree delay space. This space is isomorphic to the vertex-bias space $V_{\text{b}} \cong \mathbb{R}^{n_{\text{st}}}$. The orthogonal complement of E_{st} in the Euclidean space E is the closure-delay space E_{c} . The range of the bias operator B , the edge-bias space, is a subspace of E denoted by E_{b} . This space is isomorphic to E_{st} and thereby to V_{b} . (The dimensions of these spaces are written within parentheses.) As illustrated here, E is the oblique direct sum of E_{b} and E_{c} . The closure operator C is the oblique projection of E onto E_{c} along E_{b} ; for further details see Property 1.

therefore lies in E_{c} . We thus have the following property (see Fig. 2):

Property 1. *Any edge function ϑ of E can be decomposed in the form $\vartheta = \vartheta_{\text{b}} + \vartheta_{\text{c}}$ with $\vartheta_{\text{b}} \stackrel{\text{def}}{=} B\alpha^{(\vartheta)}$ and ϑ_{c} in E_{c} . For a given spanning tree, this decomposition is unique. As a corollary, E is the oblique direct sum of E_{b} and E_{c} : $E = E_{\text{b}} + E_{\text{c}}$ with $E_{\text{b}} \cap E_{\text{c}} = \{0\}$.*

As illustrated in Fig. 2, ϑ_{c} is the oblique projection of ϑ on E_{c} along E_{b} . The corresponding operator is the ‘closure operator’ C :

$$\vartheta_{\text{c}} = C\vartheta \quad (39)$$

Its null space (i.e., its kernel) is the range of B :

$$\ker C = E_{\text{b}} \quad (40)$$

with $\dim E_{\text{b}} = n_{\text{st}}$ (Eq. (34)).

According to Property 1, any function N of the ambiguity lattice $E(\mathbb{Z}) \cong \mathbb{Z}^{n_e}$ can be decomposed in the form

$$N = N_{\text{b}} + N_{\text{c}} \quad (41)$$

with $N_{\text{b}} \stackrel{\text{def}}{=} B\mu_{\text{st}}^{(N)}$ where (from Eq. (35))

$$\mu_{\text{st}}^{(N)} \stackrel{\text{def}}{=} B_{\text{st}}^{-1} Q_{\text{st}} N \quad (\mu^{(N)} \equiv \mu_{\text{st}}^{(N)}) \quad (42)$$

As B_{st} is unimodular, $\mu^{(N)}$ is an integer-valued function; $N_{\text{b}} \stackrel{\text{def}}{=} B\mu^{(N)}$ and $N_{\text{c}} \stackrel{\text{def}}{=} CN$ are therefore points of the integer lattices $E_{\text{b}}(\mathbb{Z}) \cong \mathbb{Z}^{n_{\text{st}}}$ and $E_{\text{c}}(\mathbb{Z}) \cong \mathbb{Z}^{n_c}$, respectively. As a result, the integer lattice $E(\mathbb{Z})$ is the oblique direct sum of the integer lattices $E_{\text{b}}(\mathbb{Z})$ and $E_{\text{c}}(\mathbb{Z})$:

$$E(\mathbb{Z}) = E_{\text{b}}(\mathbb{Z}) + E_{\text{c}}(\mathbb{Z}) \quad E_{\text{b}}(\mathbb{Z}) \cap E_{\text{c}}(\mathbb{Z}) = \{0\} \quad (43)$$

²By definition, a unimodular matrix is a square integer matrix with determinant ± 1 .

3.3 Closure delays (closure differences) and closure ambiguities

According to Eqs. (38) and (37), the quantities $\vartheta_c(i_\ell, j_\ell)$, for $\ell = 1, \dots, n_c$, can be computed via the formula

$$\vartheta_c(i_\ell, j_\ell) = \vartheta(i_\ell, j_\ell) - [\alpha_r^{(\vartheta)}(i_\ell) - \alpha_s^{(\vartheta)}(j_\ell)] \quad (44)$$

where $\alpha^{(\vartheta)}$ is determined via Eq. (35). As clarified in this section, these quantities can be referred to as the ‘closure delays’ or the ‘closure differences’ of ϑ ; the $N_c(i_\ell, j_\ell)$ ’s are therefore ‘CD ambiguities,’ also simply called ‘closure ambiguities.’

In the example of Fig. A2, let us consider the second loop, i.e., the loop associated with the closure point $(i_2, j_2) = (3, 3)$. In G , the successive points of this loop are the following: $(3, 3)$, $(3, 2)$, $(2, 2)$, $(2, 1)$, $(1, 1)$, and $(1, 3)$. Since $\vartheta_b(i, j) = \alpha_r^{(\vartheta)}(i) - \alpha_s^{(\vartheta)}(j)$, we then have, in a telescoping manner,

$$\begin{aligned} \vartheta_b(3, 3) - \vartheta_b(3, 2) + \vartheta_b(2, 2) - \vartheta_b(2, 1) \\ + \vartheta_b(1, 1) - \vartheta_b(1, 3) = 0. \end{aligned}$$

Furthermore, as ϑ_c vanishes on G_{st} ,

$$\begin{aligned} \vartheta_c(3, 3) - \vartheta_c(3, 2) + \vartheta_c(2, 2) - \vartheta_c(2, 1) \\ + \vartheta_c(1, 1) - \vartheta_c(1, 3) = \vartheta_c(3, 3) \end{aligned}$$

Since $\vartheta = \vartheta_b + \vartheta_c$ from Property 1, it follows that

$$\begin{aligned} \vartheta(3, 3) - \vartheta(3, 2) + \vartheta(2, 2) - \vartheta(2, 1) \\ + \vartheta(1, 1) - \vartheta(1, 3) = \vartheta_c(3, 3) \end{aligned}$$

This explicitly shows that $\vartheta_c(i_2, j_2)$ can be regarded as the closure difference of ϑ on the second loop. The generalization is straightforward. In the example of Fig. A2, we thus have

$$\left\{ \begin{aligned} \vartheta_c(2, 4) &= \vartheta(2, 4) - \vartheta(2, 1) + \vartheta(1, 1) - \vartheta(1, 4) \\ \vartheta_c(3, 3) &= \vartheta(3, 3) - \vartheta(3, 2) + \vartheta(2, 2) - \vartheta(2, 1) \\ &\quad + \vartheta(1, 1) - \vartheta(1, 3) \\ \vartheta_c(3, 4) &= \vartheta(3, 4) - \vartheta(3, 2) + \vartheta(2, 2) - \vartheta(2, 1) \\ &\quad + \vartheta(1, 1) - \vartheta(1, 4) \end{aligned} \right. \quad (45)$$

More generally, owing to the telescoping structure of their construction, the closure differences $\vartheta_c(i_\ell, j_\ell)$ are associated with loops whose order is even, and greater than or equal to 4. In this limit case, the notion of closure difference (CD) reduces to that of double difference (DD). According to Eq. (44), the $\vartheta_c(i_\ell, j_\ell)$ ’s can however be computed without knowing the edges of their loop. How to identify these edges, if need be, is specified in Sect. 3.4. Subject to some condition, these CD’s can be expressed as linear combinations of DD’s. The related matter is analysed in Sect. 10 of Lannes and Teunissen (2011).

3.4 Closure matrix

According to the definitions of B_{st} and B_c (introduced in Sect. 3.1.4), the vector $\vartheta_b \stackrel{\text{def}}{=} B\alpha^{(\vartheta)}$ can be orthogonally decomposed in the form

$$\vartheta_b = B_{st}\alpha^{(\vartheta)} + B_c\alpha^{(\vartheta)} = Q_{st}\vartheta + B_c\alpha^{(\vartheta)}$$

Likewise,

$$\vartheta = Q_{st}\vartheta + Q_c\vartheta$$

where $Q_c\vartheta$ is the orthogonal projection of ϑ on E_c ; see Fig. 2. It then follows from Eq. (35) that

$$\vartheta_c = \vartheta - \vartheta_b = Q_c\vartheta - B_c\alpha^{(\vartheta)} = Q_c\vartheta - B_cB_{st}^{-1}Q_{st}\vartheta$$

Denoting by $[C]$ the matrix of C expressed in the standard bases of E and E_c , we thus have, from Eq. (39),

$$[C][\vartheta] = -[B_c][B_{st}]^{-1}[Q_{st}\vartheta] + [Q_c\vartheta].$$

The column vectors of $[C]$ corresponding to the spanning-tree edges (on which $Q_c\vartheta$ vanishes) are therefore those of $-[B_c][B_{st}]^{-1}$. It is also clear that the column vectors of $[C]$ corresponding to the closure edges (on which $Q_{st}\vartheta$ vanishes) are those of the identity matrix on E_c . Consequently, with regard to the orthogonal direct sum $E_{st} \oplus E_c$,

$$[C] = \begin{bmatrix} -[B_c][B_{st}]^{-1} & [I_{c,c}] \end{bmatrix} \quad (46)$$

In the example of Fig. A2, we thus have, from Eqs. (31), (36), and (32), with the same edge ordering,

$$\begin{aligned} [B_c][B_{st}]^{-1} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \\ &\times \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & -1 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

As a result,

$$[C] = \begin{bmatrix} 1 & 0 & -1 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & -1 & 1 & -1 & 0 & 1 & 0 \\ 1 & 0 & -1 & -1 & 1 & -1 & 0 & 0 & 1 \end{bmatrix}$$

Applied to $[\vartheta]$, this matrix of course yields Eq. (45). More generally, the edges of a ‘closure loop’ are identified via the nonzero entries of the corresponding row of $[C]$. In fact, this is the most efficient way of identifying the loops in question. Note however that in the CAA method presented through Sects. 4 to 6, the action of this matrix is never explicitly performed.

3.5 On some generalized inverse of the UD-CD relationship

The closure operator C , which is an oblique projection, is not of full rank. The simplest way of removing its rank defect is to introduce the operator C_\diamond from E into $E_{st} \times E_c$

$$C_\diamond \vartheta \stackrel{\text{def}}{=} (Q_{st}\vartheta, C\vartheta) \quad (47)$$

According to Property 1, C_\diamond is invertible; this can be immediately understood from Fig. 2 for example; C_\diamond^{-1} can then be regarded as some generalized inverse of C . We now specify this point, explicitly, in matrix terms. The corresponding development is aimed at analysing the approaches of Blewitt (1989) and Collins et al. (2010) in an elementary manner; see Sects. 3.6 and 7 further on.

In the standard bases of $E = E_{st} \oplus E_c$ and $E_{st} \times E_c$, the matrix of C_\diamond can be written in the form (see Eq. (46))

$$[C_\diamond] \stackrel{\text{def}}{=} \begin{bmatrix} [Q_{st}] \\ [C] \end{bmatrix} = \begin{bmatrix} [I_{st,st}] & [0_{st,c}] \\ -[B_c][B_{st}]^{-1} & [I_{c,c}] \end{bmatrix} \quad (48)$$

It is readily verified that

$$[C_\diamond]^{-1} = \begin{bmatrix} [I_{st,st}] & [0_{st,c}] \\ [B_c][B_{st}]^{-1} & [I_{c,c}] \end{bmatrix} \quad (49)$$

Given some point \check{N}_{st} arbitrarily fixed in $E_{st}(\mathbb{Z})$, let us now consider the ambiguity point \check{N} of $E(\mathbb{Z})$ defined by the relation

$$[\check{N}] \stackrel{\text{def}}{=} [C_\diamond]^{-1} \begin{bmatrix} [\check{N}_{st}] \\ [N_c] \end{bmatrix} \quad (N_c \stackrel{\text{def}}{=} CN) \quad (50)$$

In the following property,

$$E_b^{(N)}(\mathbb{Z}) \stackrel{\text{def}}{=} N + E_b(\mathbb{Z}) \quad (51)$$

is the ‘affine lattice’ passing through N and parallel to the integer lattice $E_b(\mathbb{Z})$ of the edge-bias space E_b ; see Sect. 3.1.4 and Fig. 3.

Property 2. *The ambiguity point \check{N} is the point of the affine lattice $E_b^{(N)}(\mathbb{Z})$ whose projection on E_{st} is equal to \check{N}_{st} . More precisely, $\check{N} = N_c + B\mu^{(\check{N}_{st})}$. As a corollary, in the special case where \check{N}_{st} is set equal to 0, \check{N} is nothing else than N_c .*

For reasons of clarity and brevity, the proof is left to the reader. Note that this property can also be understood within the S-system framework; see for instance the table given in Sect. 1.6 of Teunissen (1984).

3.6 On the Blewitt generalized inverse of the UD-DD relationship

We now apply the results of the previous section to the UD-DD relationship, and thus make the link with the approach of Blewitt (1989).

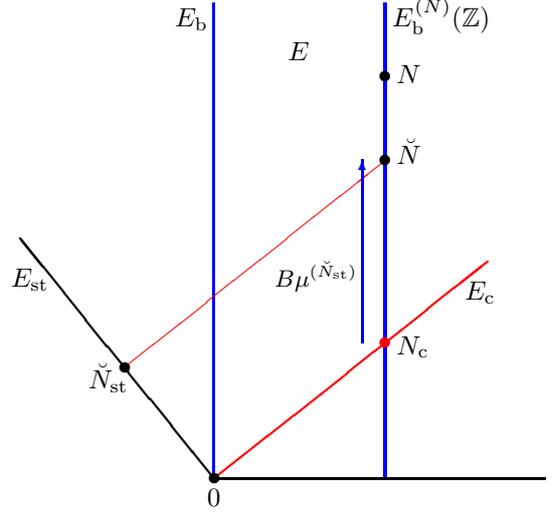


Figure 3: *Geometrical illustration of Property 2.* In this symbolic representation of the edge-delay space E , $E_b^{(N)}(\mathbb{Z})$ is the affine lattice passing through N and parallel to the integer lattice $E_b(\mathbb{Z})$ of the edge-bias space E_b (here, for clarity, the vertical axis); \check{N} is the UD ambiguity obtained via the relationship (50) in which \check{N}_{st} is arbitrarily fixed in $E_{st}(\mathbb{Z})$, and N_c is the CD ambiguity point of N (the closure ambiguity of N). In the important special case where \check{N}_{st} is set equal to 0, \check{N} reduces to N_c .

According to Eq. (68) of Lannes and Teunissen (2011), the maximum number of independent DD’s is less than or equal to n_c : $n_d^m \leq n_c$. For clarity, let us now set $n_d := n_d^m$. In the important special case where

$$n_c = n_d \quad (52)$$

the information contained in the DD data is equivalent to that contained in the closure data. Let us then denote by $D_{d,e}$ the operator providing a maximum set of n_d DD’s. By definition, $D_{d,e}$ is an operator from E into \mathbb{R}^{n_d} , i.e. then, \mathbb{R}^{n_c} . By sorting the edges of \mathcal{G} as specified in Sect. 3.1.4, the matrix of $D_{d,e}$ has then the following block structure:

$$[D_{d,e}] = \begin{bmatrix} [D_{d,st}] & [D_{d,c}] \end{bmatrix} \quad (53)$$

Here, matrix $[D_{d,e}]$ is expressed in the standard bases of $E = E_{st} \oplus E_c$ and \mathbb{R}^{n_d} . The columns of $[D_{d,st}]$ and $[D_{d,c}]$ therefore correspond to the edges of \mathcal{G}_{st} and to the closure edges, respectively. Provided that Condition (52) is satisfied, $[D_{d,c}]$ is invertible; moreover, the entries of $[D_{d,c}]^{-1}$ are then equal to ± 1 or 0; see Lannes and Teunissen 2011.

Like for C (see Eq. (48)), we then introduce the operator

$$[D_\diamond] \stackrel{\text{def}}{=} \begin{bmatrix} [Q_{st}] \\ [D_{d,e}] \end{bmatrix} \quad (54)$$

As $N = N_b + CN$ from Property 1, and $[D_{d,e}][N_b] = 0$,

we have, from Eq. (53),

$$[D_d] \stackrel{\text{def}}{=} [D_{d,e}][N] = [D_{d,c}][CN] \tag{55}$$

It then follows from Eq. (48) that

$$[D_\diamond] = \begin{bmatrix} [I_{\text{st,st}}] & [0_{\text{st,c}}] \\ [0_{\text{c,st}}] & [D_{d,c}] \end{bmatrix} [C_\diamond] \tag{56}$$

We then have

$$[D_\diamond]^{-1} \begin{bmatrix} [\check{N}_{\text{st}}] \\ [N_d] \end{bmatrix} = [C_\diamond]^{-1} \begin{bmatrix} [\check{N}_{\text{st}}] \\ [D_{d,c}]^{-1}[N_d] \end{bmatrix}$$

where $[D_{d,c}]^{-1}[N_d] = [CN]$ from Eq. (55). It then follows from Eq. (50) that

$$[D_\diamond]^{-1} \begin{bmatrix} [\check{N}_{\text{st}}] \\ [N_d] \end{bmatrix} = [\check{N}] \tag{57}$$

Given some DD ambiguity set (55), and some point \check{N}_{st} arbitrarily fixed in $E_{\text{st}}(\mathbb{Z})$, the UD ambiguity thus obtained is equal to $N_c + B\mu^{(\check{N}_{\text{st}})}$; see Property 2 and Fig. 3. Via the action of $[D_\diamond]^{-1}$, we thus retrieve the closure-ambiguity point N_c up to the vector $B\mu^{(\check{N}_{\text{st}})}$ of $E_b(\mathbb{Z})$.

Remark 3.1. In fact, $[D_\diamond]$ is a version of the D -matrix of Blewitt (1989). In the appendix B of that paper, the spanning tree \mathcal{G}_{st} is implicitly defined by arbitrarily selecting a set of “ $n - m$ undifferenced biases which pass the Gram-Schmidt test” of that appendix, i.e., in our algebraic framework, by arbitrarily selecting a set of ‘ n_{st} edges whose characteristic functions pass the Gram-Schmidt test.’ Property 2 therefore completes the analysis of the UD-DD relationship of Blewitt (1989) by specifying how the ambiguity solution \check{N} is related to N_c by the choice of \check{N}_{st} \square

Remark 3.2. Equation (57) corresponds to Eq. (8) of Ge et al. (2012) in which

$$\begin{bmatrix} \mathbf{G} \rightarrow [Q_{\text{st}}] \\ \mathbf{D} \rightarrow [D_{d,e}] \end{bmatrix} \rightarrow [D_\diamond], \quad \begin{bmatrix} \mathbf{b} \rightarrow [\check{N}_{\text{st}}] \\ \mathbf{N} \rightarrow [N_d] \end{bmatrix}, \quad \mathbf{B} \rightarrow \check{N}$$

Here, the notation ‘ $a \rightarrow b$ ’ means ‘ a stands for b .’ The arbitrary ambiguity set $\mathbf{b} \rightarrow [\check{N}_{\text{st}}]$, which is then referred to as the *reference ambiguity*, has of course nothing to do with the ambiguity set \check{v} of the *reference solution* defined in Sect. 2.5 \square

4 CAA principle

As already emphasized, with regard to the variables introduced in Eqs. (14) and (15), the choice of μ in $\mathbb{Z}^{n_{\text{st}}}$ is defined via appropriate constraints; see Sect. 2.3. As a matter of fact, the most natural way of removing the rank defect is to select μ via *a priori* constraints on the ambiguity variable v . Indeed, these constraints can then be integrated in the very definition of v . The CAA principle presented in this section results from the following preliminary analysis. (A possible introduction of this principle via the S-system approach is presented in Sect. 5; it should however be noted that the other classes of possible choices for μ do not then appear so easily.)

Let \mathcal{G}_{st} be a spanning tree of \mathcal{G} chosen arbitrarily; see Sect. A2 and Fig. 1. In the algebraic framework defined in Sect. 3, the ambiguity vector N can be decomposed in the form $N = N_b + N_c$ where

$$N_b \stackrel{\text{def}}{=} B\mu^{(N)}, \quad N_c = N - B\mu^{(N)} \tag{58}$$

in which $\mu^{(N)} \stackrel{\text{def}}{=} B_{\text{st}}^{-1}Q_{\text{st}}N$; see Eq. (42). In these equations, N_b and N_c have an implicit subscript: st for \mathcal{G}_{st} . Note that $\mu^{(N)}$ is defined from the values of N on the edges of \mathcal{G}_{st} in a unique manner. (As N is unknown, this definition is virtual.) By construction, $\mu^{(N)}$ lies in $V_b(\mathbb{Z}) \cong \mathbb{Z}^{n_{\text{st}}}$, and N_c vanishes on the edges of \mathcal{G}_{st} . As specified in Sect. 3.3, $N_c \stackrel{\text{def}}{=} CN$ is the closure ambiguity of N with regard to \mathcal{G}_{st} .

The spanning tree \mathcal{G}_{st} is built progressively from epoch 1 to k . More precisely, when new satellites are to be taken into account, it is completed as specified in Sect. 6.3 (see Figs. 1 and 5); n_{st} is thus equal to n_{b1} plus the number of satellites having appeared in the field of view of the network from epoch 2 until k included. The number of values of N_c that are not equal to zero by definition is equal to $n_c = n_e - n_{\text{st}}$ (Eq. (29)). For example, in Fig. 1, for $k = 3$, we have $n_e = 20$, $n_{\text{st}} = n_{b1} + 2 + 1$ with $n_{b1} = 8$, hence $n_c = 9$.

With regard to Eqs. (14) to (16), the CAA solution corresponds to the following choice of μ :

$$\mu := \mu^{(N)} \tag{59}$$

From Eq. (58), we then have $v = N - B\mu = N_c$: the ambiguity functional variable v vanishes on the edges of \mathcal{G}_{st} . According to Eqs. (14) and (15), the ‘*estimable functional variable*’ of the corresponding LS approach is then $(w_1, \dots, w_k; v)$ where

$$\begin{cases} w_\kappa = \beta_\kappa + \mathcal{R}_\kappa^v \mu, & v = N_c \\ \text{with } \mu := \mu^{(N)} \end{cases} \tag{60}$$

The choice of μ defined in Eq. (59) therefore defines the closure-ambiguity approach (CAA). Clearly, this choice depends on the selected spanning tree \mathcal{G}_{st} ; the definition of the integer-valued function v therefore also depends on this choice. Note that all these spanning trees have the same number of edges: $n_{\text{st}} = n_v - 1$; see Sect. A2. As a result, whatever the choice of \mathcal{G}_{st} , the number of ambiguities to be fixed is the same: $n_c = n_e - n_{\text{st}}$ (Eq. (29)).

Let us now denote by $(\check{w}_1, \dots, \check{w}_k; \check{v})$ the LS solution resulting from such a choice. The reference solution of its equivalent solutions, which can be obtained as specified in Sect. 2.5, does not depend on the choice of \mathcal{G}_{st} . Indeed, a change in \mathcal{G}_{st} amounts to selecting another μ in $V_b(\mathbb{Z})$; see Eq. (42) and Fig. 6 further on.

Remark 4.1. Although the ambiguity variables of the closure-ambiguity approach are closure ambiguities, it is an UD approach. Indeed, the action of the closure operator C (induced by the selected spanning tree \mathcal{G}_{st}) is never explicitly performed; see Sect. 6 \square

Remark 4.2. According to Property 1 and Eq. (12), Eq. (5) can always be written in the form

$$B_\kappa w_\kappa + \mathcal{R}_\kappa^e v = b_\kappa \quad (\text{for } \kappa = 1, \dots, k)$$

where w_κ and v are defined in Eq. (60). In the situations where the particular problem to be dealt with has still a rank defect (owing to other variables involved in b_κ for example), this rank defect can be removed by tackling the problem with w_κ and v substituted for β_κ and N , respectively. In that sense, the closure-ambiguity approach can be extended to all the situations where an equation (or several equations) of type (5) is (are) involved in the problem; see, for example, the case of small networks in Lannes and Teunissen (2011) \square

5 Derivation of the CAA principle via the S-system approach

The S-system approach (Baarda 1973) was used by some investigators to remove various GNSS rank defects; see, e.g., Teunissen 1984, de Jonge, 1998, Teunissen and Odijk 2003. In the previous section, the CAA principle was introduced in a concise manner in the theoretical framework defined through Sects. 2 and 3; the aim of the present section is to show that this principle can be introduced via the S-system approach directly.

In the general framework of the S-system approach presented in Appendix B, the Euclidean space \mathfrak{E} to be considered is then the space of the functional variable

$$\xi \stackrel{\text{def}}{=} (\beta_1, \dots, \beta_k; \vartheta) \quad (61)$$

of Eq. (5); ϑ is then the float version of the integer-valued functional variable N . Denoting by

$$V_\beta \stackrel{\text{def}}{=} V_{\beta_1} \oplus V_{\beta_2} \oplus \dots \oplus V_{\beta_k} \quad (62)$$

the direct sum of the vertex-bias spaces V_{β_κ} with generic vector β_κ , we have

$$\mathfrak{E} = V_\beta \oplus E \quad (63)$$

where E is the edge-delay space with generic vector ϑ ; see Sect. 3.1.2. As illustrated in Figs. 2 and 4, given some spanning tree \mathcal{G}_{st} of \mathcal{G} , E can be decomposed in the Euclidean orthogonal form

$$E = E_{\text{st}} \oplus E_c \quad (64)$$

From Eqs. (8) and (30), we have

$$\left| \begin{array}{l} \dim V_\beta = \sum_{\kappa=1}^k n_{b_\kappa} \\ \dim E_{\text{st}} = n_{\text{st}} \end{array} \right. \quad \dim E_c = n_c \quad (65)$$

The function b_κ of Eq. (5) lies in some space isomorphic to $\mathbb{R}^{n_{e\kappa}}$; see the context of Eq. (3), and Fig. 1. Setting

$$m := \sum_{\kappa=1}^k n_{e\kappa} \quad (66)$$

we then define the operator \mathcal{A} of the S-system approach as the operator from \mathfrak{E} into \mathbb{R}^m :

$$\mathcal{A}\xi \stackrel{\text{def}}{=} \begin{pmatrix} B_1\beta_1 + \mathcal{R}_1^e\vartheta \\ \vdots \\ B_k\beta_k + \mathcal{R}_k^e\vartheta \end{pmatrix} \quad (67)$$

Clearly, Eq. (5) can then be explicitly written in the form

$$\mathcal{A} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_k \\ N \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_k \end{pmatrix} \quad (68)$$

Let \mathcal{K} now be the operator from $V_b \cong \mathbb{R}^{n_{\text{st}}}$ into \mathfrak{E} (see Sect. 3.1.1):

$$\mathcal{K}\alpha \stackrel{\text{def}}{=} \begin{pmatrix} -\mathcal{R}_1^v\alpha \\ \vdots \\ -\mathcal{R}_k^v\alpha \\ B\alpha \end{pmatrix} \quad (69)$$

The float version of Eq. (12) yields

$$-B_\kappa\mathcal{R}_\kappa^v\alpha + \mathcal{R}_\kappa^e B\alpha = 0 \quad (\text{for } \kappa = 1, \dots, k) \quad (70)$$

The null space of \mathcal{A} is therefore the range of \mathcal{K} ; see Eqs. (67) and (69). We thus have

$$\ker \mathcal{A} = \text{Ran } \mathcal{K} \quad \text{with} \quad \dim(\ker \mathcal{A}) = n_{\text{st}} \quad (71)$$

The dimension of the null space of \mathcal{A} is therefore equal to that of E_{st} ; see Eq. (65) and Fig. 4.

In the framework of the S-system approach, we now remove the rank defect of Eq. (5) by imposing the following constraint: *the functional variable to be estimated lies in the orthogonal complement of E_{st} in \mathfrak{E}* ; see Appendix B. We thus define \mathfrak{F} via the relation $\mathfrak{F} := E_{\text{st}}^\perp$. Clearly, that constraint is the same as that introduced in Sect. 4; see Fig. 2. The corresponding estimable functional variable is then obtained as follows.

From Eqs. (63) and (64), \mathfrak{F} is the direct sum of V_β and E_c :

$$\mathfrak{F} = V_\beta \oplus E_c \quad (72)$$

We now show that \mathfrak{E} is the direct sum of \mathfrak{F} and $\ker \mathcal{A}$ (see Fig. 4):

$$\mathfrak{E} = \mathfrak{F} \oplus \ker \mathcal{A} \quad (73)$$

Proof. Let ξ be a function lying in $\mathfrak{F} \cap \ker \mathcal{A}$. As ξ lies in \mathfrak{F} , its component ϑ vanishes on \mathcal{G}_{st} . As ξ also lies in $\text{Ran } \mathcal{K}$ from Eq. (71), ϑ is in the range of B from Eq. (69). We then have $\vartheta = B_{\text{st}}\alpha = 0$, hence $\alpha = 0$ from Eq. (33), and therefore $\xi = 0$ from Eq. (69). As a result, $\mathfrak{F} \cap \ker \mathcal{A} = \{0\}$. As $\dim(\ker \mathcal{A}) = \dim E_{\text{st}}$ and $\dim \mathfrak{E} = \dim \mathfrak{F} + \dim(\ker \mathcal{A})$, Eq. (73) is thus established \square

6 CAA implementation

In the closure-ambiguity approach, the rank defect of Eq. (5) is removed via the choice of the variables w_κ and v defined in Eq. (60). The equation to be solved in the LS sense is then Eq. (16) with the constraint $v = 0$ on the edges of \mathcal{G}_{st} . The CAA problem is therefore: solve in the LS sense the equation

$$\begin{cases} B_\kappa w_\kappa + \mathcal{R}_\kappa^e v = b_\kappa & (\kappa = 1, \dots, k) \\ \text{subject to the constraint 'v = 0 on } \mathcal{G}_{\text{st}} \end{cases} \quad (79)$$

In what follows, the noise is taken into account by using the variance-covariance matrix \mathbf{V}_κ of b_κ in the definition of the ‘Euclidean forms’ \mathbf{B}_κ , \mathbf{A}_κ and \mathbf{b}_κ of B_κ , \mathcal{R}_κ^e and b_κ , respectively; see the context of Eq. (C6) in Sect. C3. The system (79) to be solved in the Euclidean LS sense is then the following:

$$\mathbf{B}_\kappa \mathbf{w}_\kappa + \mathbf{A}_\kappa \mathbf{v} = \mathbf{b}_\kappa \quad (\kappa = 1, \dots, k) \quad (80)$$

i.e.,

$$\begin{bmatrix} \mathbf{B}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_1 \\ \mathbf{0} & \mathbf{B}_2 & \mathbf{0} & \mathbf{0} & \mathbf{A}_2 \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_k & \mathbf{A}_k \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_k \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_k \end{bmatrix} \quad (81)$$

The angular block structure of matrix $[\mathbf{B} \ \mathbf{A}]$ is well suited to recursive QR factorization. (A complete implementation of the corresponding LS procedure is presented in Appendix C.) The advantage of this procedure is to provide some gain in numerical accuracy when dealing with large-scale problems; see Björck (1996). More interestingly, the corresponding techniques prove to be very efficient for GNSS quality control; see, e.g., Blewitt 1989; Tiberius 1998; Loehnert et al. 2000; Lannes and Gratton 2009. As specified in Sects. 9.5 and 7.3 of this last paper, the recursive identification of the outliers is made easier (with regard to the related variational calculations) by storing in memory the parameters of the Givens rotations involved in the two QR steps of epoch k .

In this section, we first give a survey of the QR implementation of the CAA principle (Sect. 6.1). We then specify the definitions of the column matrices \mathbf{w}_k and \mathbf{v} of Eq. (81); this is done in Sects. 6.2 and 6.3, respectively. More information about the technical aspects of this implementation, like for instance the construction of matrices \mathbf{A}_k , \mathbf{B}_k and \mathbf{b}_k , is to be found in Appendix C.

Remark 6.1. In most situations encountered in practice, the second member of equations such as (79) includes a large number of additional variables; see, e.g., Ge et al. 2005; Loyer et al. 2012. In the remainder of the paper, we assume that the related models have been well selected so that we can concentrate on the first members of these equations only \square

Remark 6.2. The method presented in this section can be applied as such for solving the wide-lane (WL) equation (4) of Loyer et al. (2012); N_{WL} and τ_{WL} are then to be substituted for N and β , respectively; b is then the term on the left-hand side of that equation. In our approach, the WL ambiguity point \check{v}_{WL} would then be the closure-ambiguity point of N_{WL} : $\check{v}_{\text{WL}} = CN_{\text{WL}}$; N_{WL} would thus be retrieved up to a vector of $E_b(\mathbb{Z})$ \square

6.1 Survey

In a first stage, at each epoch k , the ‘float solution’ $\hat{\mathbf{v}}$ is computed or updated. This is done by solving the float version of Eq. (81) in the Euclidean LS sense via recursive QR factorization; see Sect. C1 and Eq. (C1). The integer least-squares (ILS) solution $\check{\mathbf{v}}$, and thereby \check{v} , is then defined as the solution of the nearest-lattice-point (NLP) problem (C3). This solution is obtained in a second stage via appropriate integer-programming techniques; see Lannes and Prieur (2013), and Lannes (2013) for the parallelization of the related LLL³/LAMBDA reduction/decorrelation algorithms. The ambiguities are thus fixed. Once at some epoch k_{val} , these ambiguities can be validated (see Verhagen and Teunissen 2006), the bias matrices $\check{\mathbf{w}}_k$, and thereby the phase biases \check{w}_k , are obtained via the relations (C4) for $k \leq k_{\text{val}}$. The variance-covariance matrix of \check{w}_k , which is required for implementing the PPP mode properly (see Sect. 9 in Lannes and Teunissen 2011), is obtained in that framework; see the end of Sect. C1.

Remark 6.3. Once at some epoch k_{val} , all the closure ambiguities have been validated, the QR recursive process only needs to focus on the few closure ambiguities associated with the new closure edges. At the epochs $k > k_{\text{val}}$, the new closure ambiguities can therefore be determined very quickly. Indeed, the dimension of the matrix \mathbf{v} handled by the QR process is then much smaller than previously. This shows that this approach is well suited to integer-ambiguity resolution in real time. For instance, the satellite-clock biases could then be broadcasted to the network users in real time; see Remark 2.2 \square

6.2 Definition of \mathbf{w}_k

If we assume for instance that all the receivers of the network are active (see Fig. 1), the receiver phase bias $w_{rk}(i)$ is then the entry of \mathbf{w}_k with index $i - 1$. The indices are shifted by -1 since we used the convention $\beta_{rk}(1) = 0$ (Eq. (7)).

The satellite phase bias $w_{sk}(j)$ is then the entry of \mathbf{w}_k with index $(n_r - 1 + j)$; see for example the first two epochs of Fig. 1. To handle the cases of disappearance of one or more satellite(s), we introduce an index function ϖ_{sk} defined so that $w_{sk}(j)$ is the entry of \mathbf{w}_k with index $(n_r - 1 + \varpi_{sk}(j))$. For example, at epoch $k = 3$ of Fig. 1, for $j > 3$, we have $\varpi_{sk}(j) = j - 1$.

³Here LLL stands for Lenstra, Lenstra, Lovász, the authors of the famous LLL algorithm (Lenstra et al. 1982).

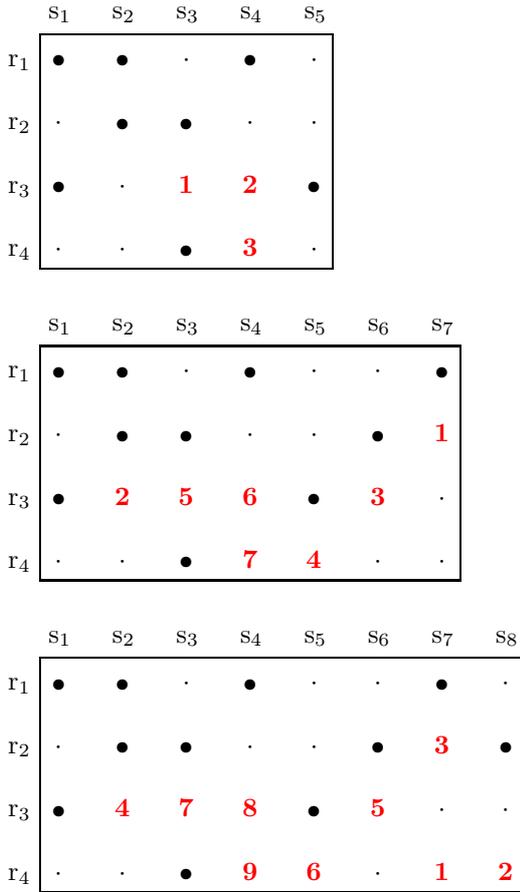


Figure 5: *Spanning trees $\mathcal{G}_{st,k}$ and closure-edge lists $\mathcal{L}_{c,k}$.* The example shown here corresponds to that introduced in Fig. 1. From top to bottom, epoch $k = 1$ ($n_{st,1} = 8$), epoch $k = 2$ ($n_{st,2} = 10$), and epoch $k = 3$ ($n_{st,3} = 11$). The large dots correspond to the edges of these spanning trees; the small dots define the edges (r_i, s_j) that do not appear in \mathcal{G}_k at those epochs. The spanning tree of \mathcal{G}_2 is obtained from that of \mathcal{G}_1 by adding the edges (r_1, s_7) and (r_2, s_6) . Likewise, the spanning tree of \mathcal{G}_3 is obtained from that of \mathcal{G}_2 by adding the edge (r_2, s_8) . The closure edges of $\mathcal{L}_{c,k}$ are ordered as numbered; see text.

Difficulties with some receivers would be handled in a similar manner by introducing index functions $\varpi_{rk}(i)$.

6.3 Definition of \mathbf{v}

The entries of \mathbf{v} in Eq. (81) are the closure ambiguities to be taken into account from epochs 1 to k included. The corresponding closure-edge list $\mathcal{L}_{c,k}$ is defined in this section; see for example the lower grid of Fig. 5. In the implementation of the QR method, those entries are to be put at the top of the column matrix \mathbf{v} at the epoch $\kappa \leq k$ where they appear; see Sect. C2. This means that for all $\kappa' < \kappa$, the matrices $\mathbf{A}_{\kappa'}$ have then (on their left-hand side) implicit additional columns whose entries are zero. We now concentrate on the closure-edge lists $\mathcal{L}_{c,\kappa}$ for $\kappa = 1, \dots, k$.

At epoch $k = 1$, the spanning tree $\mathcal{G}_{st,1}$ of $\mathcal{G}_1 = \mathcal{H}_1$ is built as specified in Sect. A2. For example, for the scenario defined in Fig. 1, the edges of $\mathcal{G}_{st,1}$ then correspond to the large dots of the upper grid of Fig. 5; $\mathcal{L}_{c,1}$ then includes three closure edges ordered as they are encountered when scanning that grid line by line:

$$\mathcal{L}_{c,1} = \{(r_3, s_3), (r_3, s_4), (r_4, s_4)\} \quad (k = 1)$$

At epoch $k = 2$, the spanning tree $\mathcal{G}_{st,2}$ of $\mathcal{G}_2 = \mathcal{H}_1 \cup \mathcal{H}_2$ is completed by adding the edges corresponding to the first active edges involving the new satellites, for example (r_1, s_7) , and (r_2, s_6) in Fig. 5. When implementing the QR method, the new closure edges must then be the first terms of $\mathcal{L}_{c,2}$; see Fig. C4 in Sect. C2. The previous terms of that list are then shifted rightwards. In the example of Fig. 1, we then have

$$\mathcal{L}_{c,2} = \{(r_2, s_7), (r_3, s_2), (r_3, s_6), (r_4, s_5), (r_3, s_3), (r_3, s_4), (r_4, s_4)\} \quad (k = 2)$$

We proceed similarly for the next epochs; see the lower grid of Fig. 5.

By construction, the number of closure edges n_c is a non-decreasing function of k ; see Eq. (29) and Fig. 5. In fact, this number defines the dimension of the NLP problem (C3) to be solved at epoch k .

7 Equivalent ambiguity solutions: related methodological aspects

In this section, we analyse some other methods which are used for solving the GNSS phase-calibration problem, and compare the corresponding solutions with the CAA solution. This done for Blewitt (1989) in Sect. 7.1, for Collins et al. (2010) in Sect. 7.2, and for Loyer et al. (2012) in Sect. 7.3.

7.1 The Blewitt (1989) approach

In his original contribution published in 1989, Blewitt proposes a resolution of the problem in three stages. In the first one, the float solution is obtained in UD mode. In the second one, double differencing of the float biased ambiguities thus obtained provides DD ambiguities. The latter are then fixed via some sequential adjustment algorithm. In the third and final stage, the UD ambiguities are derived from those fixed DD ambiguities. This is done via the inverse of the operator D defined in the appendix B of that paper. The estimates of station-receiver locations, orbital parameters, etc., are then updated. Briefly, one may therefore say that the Blewitt approach is hybrid: it is a ‘UD-DD-UD approach.’

The D -matrix of our analysis is the matrix $[D_\circ]$ introduced in Sect. 3.6: Eq. (54). Given some DD ambiguity set N_d (such as that defined in Eq. (55)), some spanning tree \mathcal{G}_{st} , and some ambiguity point N_{st} arbitrarily

fixed in $E_{\text{st}}(\mathbb{Z})$, it is possible to retrieve N up to a vector of $E_{\text{b}}(\mathbb{Z})$. More precisely, the UD ambiguity obtained via Eq. (57),

$$\check{N} := [D_{\diamond}]^{-1} \begin{bmatrix} [\check{N}_{\text{st}}] \\ [N_{\text{d}}] \end{bmatrix}$$

is equal to $N_{\text{c}} + B\mu^{(\check{N}_{\text{st}})}$; see Fig. 3. Removing the rank defect via the inverse of $[D_{\diamond}]$ therefore amounts to imposing the constraint $Q_{\text{st}}v = \check{N}_{\text{st}}$ where \check{N}_{st} can be chosen in an arbitrary manner in $E_{\text{st}}(\mathbb{Z})$.

7.2 The Collins et al. (2010) approach

In the last statement of Sect. 7.1, one recognizes the concept of *ambiguity datum fixing* of the UD approach of Collins et al. 2010; the latter is thus closely linked to that of Blewitt. However, with regard to Eq. (16), the ‘direct problem’ to be solved in the LS sense is then

$$\begin{cases} B_{\kappa}w_{\kappa} + \mathcal{R}_{\kappa}^{\text{e}}v = b_{\kappa} & (\kappa = 1, \dots, k) \\ \text{subject to the constraint } Q_{\text{st}}v = \check{N}_{\text{st}} \end{cases} \quad (82)$$

Let us denote by $(\check{w}_1, \dots, \check{w}_k; \check{v})$ the solution of this problem, i.e., the Blewitt/Collins solution. Let $(\check{w}_1, \dots, \check{w}_k; \check{v})$ now be the CAA solution obtained with the same spanning tree \mathcal{G}_{st} . As shown below, we then have (see Fig. 6)

$$\check{w}_{\kappa} = \check{\beta}_{\kappa} - \mathcal{R}_{\kappa}^{\text{v}}\mu^{(\check{N}_{\text{st}})}, \quad \check{v} = \check{v} + B\mu^{(\check{N}_{\text{st}})} \quad (83)$$

Proof. The LS solutions of Eq. (5) are of the form (18)-(19). Equation (83) then follows from the fact that by construction $\check{v} + B\mu^{(\check{N}_{\text{st}})}$ satisfies the constraint of Eq. (82). Indeed, as $Q_{\text{st}}\check{v} = 0$, we have

$$\begin{aligned} Q_{\text{st}}(\check{v} + B\mu^{(\check{N}_{\text{st}})}) &= Q_{\text{st}}B\mu^{(\check{N}_{\text{st}})} = B_{\text{st}}\mu^{(\check{N}_{\text{st}})} \\ &= B_{\text{st}}(B_{\text{st}}^{-1}Q_{\text{st}}\check{N}_{\text{st}}) = \check{N}_{\text{st}} \quad \square \end{aligned}$$

Remark 7.1. In the special case where \check{N}_{st} is set equal to zero on all the edges of \mathcal{G}_{st} , the Blewitt/Collins solution coincides with the CAA solution; see Eq. (83) and Fig. 6. This explicitly shows that the Blewitt and Collins approaches can lead to the same results as the CAA method. However, even in that case, Blewitt’s approach is not equivalent in terms of efficiency. The direct implementation of the Collins/CAA principle is *a priori* preferable: the results are then obtained without any differencing operation, and without any generalized inversion of the fixed ambiguity set; see Sects. 4 and 6. As illustrated in Fig. 6, the reference ambiguity solutions \bar{v} obtained with the Blewitt, Collins and CAA methods must of course be identical on every edge of \mathcal{G} , and this for any choice of \mathcal{G}_{st} and \check{N}_{st} ; see Remark 2.4 \square

Remark 7.2. The algebraic analysis of the PPP mode can also be illustrated by a figure such as Fig. 6. The dimension of E is then equal to the number of satellites seen by the PPP-user receiver r_i : $E \cong \mathbb{R}^{n_s}$; E_{st} is then

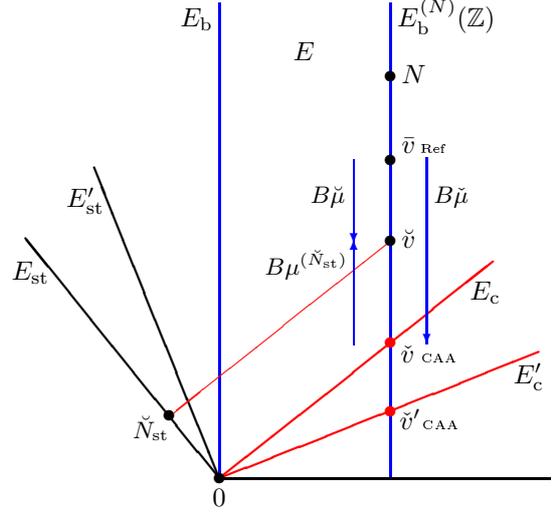


Figure 6: *Links between some equivalent ambiguity solutions.* In this geometrical representation of the edge-delay space E , the equivalent ambiguity solutions lie in the affine lattice $E_{\text{b}}^{(N)}(\mathbb{Z})$ passing through N and parallel to the integer lattice $E_{\text{b}}(\mathbb{Z})$ of the edge-bias space E_{b} (here for clarity the vertical axis). The Blewitt or Collins solution \check{v} obtained by imposing the constraint $Q_{\text{st}}v = \check{N}_{\text{st}}$ (for some \check{N}_{st} arbitrarily fixed in $E_{\text{st}}(\mathbb{Z})$) corresponds to the intersection of $E_{\text{b}}^{(N)}(\mathbb{Z})$ with the affine space passing through \check{N}_{st} and parallel to $E_{\text{c}}(\mathbb{Z})$ (the closure-ambiguity lattice induced by the choice of \mathcal{G}_{st}); see Fig. 3. The CAA solutions induced by the choice of the spanning trees \mathcal{G}_{st} and $\mathcal{G}_{\text{st}'}$, \check{v} and \check{v}' , correspond to the intersections of $E_{\text{b}}^{(N)}(\mathbb{Z})$ with $E_{\text{c}}(\mathbb{Z})$ and $E'_{\text{c}}(\mathbb{Z})$, respectively. Note that $\check{v} = N - B\mu^{(N)}$ from Property 1, and $\check{v} = \check{v} + B\mu^{(\check{N}_{\text{st}})}$ from Property 2. Here, \bar{v} represents the reference solution: $\bar{v} = \check{v} - B\check{\mu}$; see Eqs. (21), (23) and (25). Likewise, \bar{v} is obtained from the Blewitt or Collins solution \check{v} via the relation $\bar{v} = \check{v} - B\check{\mu}$.

the space $E_{\ell,1} \stackrel{\text{def}}{=} \{\vartheta \in E : \vartheta(\ell, j) = 0 \text{ if } j \neq 1\}$. Here, 1 is the index of the first satellite for example; $E_{\ell,1}$ is a one-dimensional space. Note that E_{b} then reduces to the one-dimensional space whose vectors are of the form $(1_1, 1_2, \dots, 1_{n_s}) \times \beta$ where β is the unknown receiver clock-phase bias. The choice $\check{N}_{\text{st}} := 0$ then amounts to setting $N(\ell, 1) := 0$. The reader is invited to complete that transposition \square

7.3 The Loyer et al. (2012) approach

With regard to the way the narrow-lane (NL) ambiguity N_1 is fixed, the zero-difference approach of Loyer et al. (2012) can be considered as a variant of the Collins et al. (2010) approach. We now clarify this point.

In that particular approach, the rank defect is removed after having obtained the ‘float ambiguity solution’ \hat{N}_1 via the introduction of some additional constraints; see Sect. 2.6 of that paper. These authors then implicitly choose some spanning tree \mathcal{G}_{st} , and set the ambiguities to $[\hat{N}_1(i, j)]$ on the edges (r_i, s_j) of \mathcal{G}_{st} . The remaining UD ambiguities (i.e., the n_{c} ambiguities associated with

the closure edges) are fixed via some ‘bootstrapping operation.’ This operation provides the ‘Babai point’ of the discrete-search algorithms presented in Lannes and Prieur (2013).

Setting $N := N_1$ and $\check{N}_{st} := [Q_{st}\hat{N}_1]$, we thus have $N - B\mu = \check{N}_{st}$ on \mathcal{G}_{st} , hence $\mu := B_{st}^{-1}Q_{st}(N - \check{N}_{st})$. It then follows from Eqs. (42) and (58) that the ambiguity solution of the Loyer et al. (2012) approach is

$$N - B\mu = (N - B\mu^{(N)}) + B\mu^{(\check{N}_{st})} = N_c + B\mu^{(\check{N}_{st})}$$

Provided that the solution provided by the bootstrapping process is the right NLP solution \check{v} , we then have (like for the Blewitt and Collins solutions in Fig. 6)

$$\check{v} = \tilde{v} + B\mu^{(\check{N}_{st})}$$

where \tilde{v} is the CAA solution obtained with the same spanning tree. The choice $\check{N}_{st} := 0$ would lead to a ‘mild version’ of the corresponding NL closure-ambiguity approach; see how the NLP problem is solved in Sect. 6.1.

8 Conclusion

In this paper, we have examined the problem of calibrating the clock-phase biases of GNSS networks. In the context specified in Sect. 1, the basic rank defect of this problem is related to the way these phase biases and the carrier-wave ambiguities are involved in the observational equations. We have analysed the different ways of removing this rank defect, and defined a particular strategy for obtaining clock-phase bias estimates in a standard form.

This rank defect is intrinsically related to the structure of Eq. (1), and cannot be resolved by additional experimental data. As a result, according to Eq. (13), any variation of the vertex ambiguity vector μ can be compensated by a variation of the edge-ambiguity vector N ; μ is a vector of $\mathbb{Z}^{n_{st}}$; see Sect. 2.2. For PPP applications, the satellite clock-phase biases can be estimated up to constant integers. The choice of μ in $\mathbb{Z}^{n_{st}}$ does not therefore affect the significant part of these bias values. The retrieved ambiguities are of course affected by this choice, but this has no actual impact on the GNSS results such as the estimates of the station-position parameters, for example.

In this paper, we propose a particular approach, the closure-ambiguity approach (CAA), which is a natural way of finding a solution; see Sects. 4 to 6. It is an undifferenced method based on some particular constraints. The related choice of μ , which is associated with the notion of closure ambiguity, is similar to that implicitly made by de Jonge (1998); it defines the very principle of the closure-ambiguity approach. Thanks to the introduction of graph \mathcal{G} , the union of the observational graphs until the current epoch, the closure ambiguities are dealt with in an optimal manner. In particular, compared to

the approaches presented in de Jonge (1998) and Lannes and Gratton (2009), no graph transition is to be performed. These new aspects have been illustrated with some simple and generic examples; see Figs. 1 and 5.

We have analysed the main classes of other methods used for calibrating GNSS networks, and established the link between those methods and our CAA method. More precisely, this paper presents a unified interpretative framework in which all those methods can be understood and compared more easily (see Sect. 7). We have thus been able to derive functional relations between the solutions provided by the methods of Blewitt (1989), Collins et al. (2010) and Loyer et al. (2012). Those solutions are displayed in Fig. 6 which gives a synthetic representation of the results provided by our approach (CAA) and all those methods.

We have also shown that the intermediate differencing stage of Blewitt’s approach can be avoided, without any counterpart, by removing the rank defect via our approach or that of Teunissen (1984): the closure ambiguities to be fixed then appear in the very formulation of the UD problem to be solved. The NLP techniques of ambiguity resolution can thus be directly applied to the float solution; see Sect. 6.1. Compared to the hybrid UD-DD-UD methods deriving from the basic contribution of Blewitt (1989), the technical implementation of the CAA method is simpler and more efficient; some CPU time can thus be saved.

It also appeared that the concept of ambiguity datum fixing of Collins et al. (2010) comes within our CAA-(S-system) framework. When the ambiguities are fixed at zero on the edges of some arbitrary spanning tree of \mathcal{G} , the remaining ambiguities to be fixed are nothing else than the closure ambiguities of the ambiguity set N under consideration. Compared to Blewitt’s approach, one may therefore say that the UD approaches of Collins et al. (2010) and Loyer et al. (2012) are closer to our CAA-(S-system) principle.

To compare the methods providing LS estimates of clock-phase biases, we have introduced a reference particular solution. For this solution, when a clock-phase bias is estimated for the first time, its fractional part is confined to the one-cycle width interval centred on zero; the integer-ambiguity set is modified accordingly. The notion of reference solution is very useful for testing the compatibility of all those GNSS methods: pertinent methods should lead to the same reference ambiguity solution; see Fig. 6. This test is independent of the selected spanning tree.

The QR implementation of the CAA method has been presented in Sect. 6 and Appendix C, in an exhaustive manner. Completed by the NLP algorithms developed for huge networks (Lannes 2013, Lannes and Prieur 2013), all the elements are now gathered for implementing very powerful techniques. The implications of our approach may concern the software packages used for processing most GNSS networks, and particularly global networks

using GPS, GLONASS, Galileo, or BeiDou/Compass satellites. The CAA principle is well suited for handling the integer-ambiguity problem of all those networks. As outlined in Remark 6.3, it could lead to applications in real-time kinematic precise point positioning (RTK-PPP).

In order to facilitate its integration in existing GNSS software packages, our approach has been described in a precise manner. The gain of such an implementation would be both in terms of performance and reliability. Indeed, our undifferenced approach proposes an appropriate procedure for ambiguity resolution and clock-phase bias calibration, and this with a rigorous handling of the rank defect to be removed. Another possible application of this work would be to use the method described in Sect. 2.5 for comparing the solutions obtained by different software packages.

Appendix A Elementary notions on GNSS graphs

In this appendix, we present some preliminary notions of algebraic graph theory; these elementary notions are used throughout the paper. Further details about the functional spaces and the operators involved in the GNSS problems can be found in Sect. 3. We first define the notions of GNSS grid and GNSS graph (Sect. A1). We then introduce the concepts of spanning tree and loop (Sect. A2).

A1 GNSS grid and graph

For our present purpose (see Sects. 2.1 and 2.2 in particular), we consider a typical situation in which the network has n_r receivers r_i and n_s satellites s_j . (We recall that ‘satellite’ should be here understood as ‘satellite transmitter;’ see Sect. 1.) The ‘network grid’ G_o then includes n_r rows, n_s columns, and $n_r n_s$ points; see Fig. A1. A function such as $N(i, j)$ for example takes its values on some points (i, j) of that grid. Those points form a subgrid denoted by G .

In the example presented in Fig. A1, the points (i, j) of G are displayed as large dots in the upper part of this figure. Those points correspond to the ‘edges’ (r_i, s_j) of the graph associated with the GNSS network; this graph is displayed in the lower part of Fig. A1; \mathcal{E} denotes the set of its edges; n_e is their number. The receivers and the satellites involved in the definition of these edges define the ‘vertices’ of this graph; \mathcal{V} denotes the set of its vertices, and n_v their number: $n_v = n_r + n_s$. A graph such as \mathcal{G} is therefore defined by the pair $(\mathcal{V}, \mathcal{E})$: $\mathcal{G} \equiv \mathcal{G}(\mathcal{V}, \mathcal{E})$. We now assume that \mathcal{G} is connected: given any two vertices of \mathcal{V} , there exists a path of edges of \mathcal{E} connecting these vertices; see, e.g., Biggs 1996.

A2 GNSS spanning tree and loops

As illustrated in Fig. A2, a spanning tree of a connected graph $\mathcal{G} \equiv \mathcal{G}(\mathcal{V}, \mathcal{E})$ is a subgraph $\mathcal{G}_{st} \equiv \mathcal{G}_{st}(\mathcal{V}, \mathcal{E}_{st})$ formed

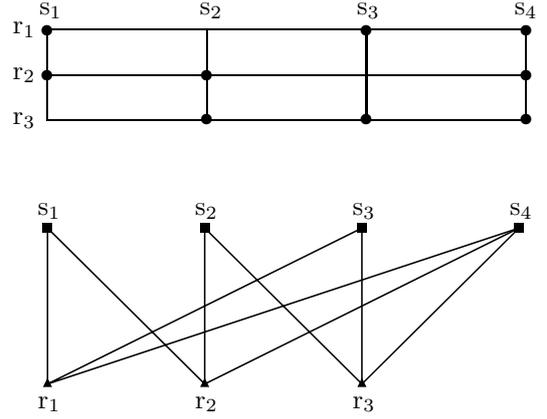


Figure A1: *Subgrid G and graph \mathcal{G}* . In the example described here, the network grid G_o includes twelve points ($n_r = 3, n_s = 4$), while its subgrid G includes nine points only; these points are shown as large dots. The corresponding graph \mathcal{G} includes seven vertices and nine edges: $n_v = n_r + n_s = 7, n_e = 9$; r_1 does not see s_2, r_2 does not see s_3 , and r_3 does not see s_1 .

by n_v vertices and $n_e - 1$ edges, with no ‘cycle’ in it. Here, ‘cycle’ is used in the sense defined in algebraic graph theory; see, e.g., Biggs (1996). In the GNSS community, to avoid any confusion with the usual notion of wave cycle, the term of ‘loop’ can be substituted for that of ‘cycle.’ In this context, the number of loops defined through a given fixed (but arbitrary) spanning tree is the number of edges of \mathcal{E} that do not lie in \mathcal{E}_{st} . These edges, $c(\ell) \stackrel{\text{def}}{=} (r_{i_\ell}, s_{j_\ell})$, are said to be ‘(loop-)closure edges’ (see Fig. A2). Their number is denoted by n_c :

$$n_c = n_e - n_{st} \quad (\text{A1})$$

where

$$n_{st} \stackrel{\text{def}}{=} n_v - 1 = n_r + n_s - 1 \quad (\text{A2})$$

Many spanning trees of the same graph can be constructed. Here, we are going to present the Kruskal algorithm which is often used in algebraic graph theory; see Biggs 1996. The first step of this algorithm consists in ordering the edges of \mathcal{E} , thus generating a sequence of the form $\{(r_{i_q}, s_{j_q}) : q = 1, \dots, n_e\}$. The spanning tree is then obtained as follows. Set $q = 0, n_{st} = 0$, and $\mathcal{E}_{st} = \emptyset$ (the empty set). Then,

- (i) if $n_{st} = n_v - 1$, terminate the process; otherwise, set $q := q + 1$;
- (ii) when the vertices of edge (r_{i_q}, s_{j_q}) are not connected via edges of \mathcal{E}_{st} , set $\mathcal{E}_{st} := \mathcal{E}_{st} \cup \{(r_{i_q}, s_{j_q})\}$, $n_{st} := n_{st} + 1$; then go to step (i).

By construction, the spanning tree thus found depends on how the edges are ordered in the first step. The subgrid of G corresponding to the edges of \mathcal{G}_{st} is denoted

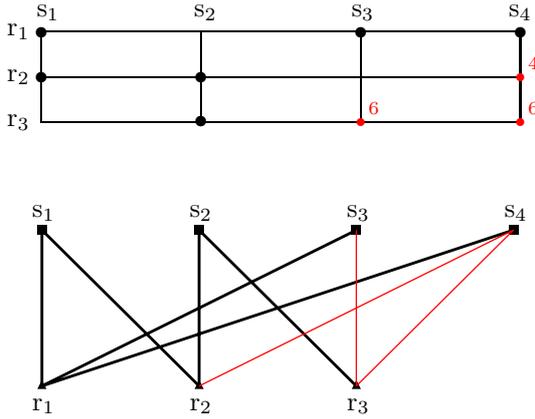


Figure A2: *GNSS spanning tree and loops.* Here, the edges of the selected spanning tree \mathcal{G}_{st} of the graph \mathcal{G} introduced in Fig. A1 are shown as thick lines. The points of the corresponding subgrid \mathcal{G}_{st} are shown as large dots. The remaining points of \mathcal{G} (the small dots of \mathcal{G}) correspond to the (loop-)closure edges (the thin edges of \mathcal{G}). We then have one loop of order four, and 2 loops of order six: (r_2, s_4, r_1, s_1) , $(r_3, s_3, r_1, s_1, r_2, s_2)$ and $(r_3, s_4, r_1, s_1, r_2, s_2)$. In \mathcal{G} , these orders are shown as small numbers.

by \mathcal{G}_{st} ; \mathcal{G}_c is that corresponding to the closure edges:

$$\mathcal{G}_c \stackrel{\text{def}}{=} \{(i, j) \in \mathcal{G} : (i, j) \notin \mathcal{G}_{st}\} \tag{A3}$$

Clearly, \mathcal{G}_c includes n_c loop-closure points; see Eq. (A1) and Fig. A2.

To illustrate the action of the Kruskal algorithm, let us consider the graph \mathcal{G} of Fig. A1. To build a spanning tree of \mathcal{G} from its grid \mathcal{G} , let us order the edges of \mathcal{G} by scanning \mathcal{G} from left to right and top to bottom. The algorithm examines the edges of \mathcal{G} in that order and adds them to the current version of \mathcal{E}_{st} when condition (ii) holds. In this example, this is the case for the first five edges; the vertices s_1, s_3, s_4, r_1, r_2 are thus connected. The sixth edge, (r_2, s_4) , therefore includes two vertices already connected. This edge is therefore the first closure edge: $c(1) = (r_2, s_4)$. The next edge, (r_3, s_2) , is added to \mathcal{E}_{st} since it corresponds to the first connection of s_2 with the edges of the current version of \mathcal{E}_{st} . All the vertices of \mathcal{G} are then connected. The remaining edges are therefore closure edges: $c(2) = (r_3, s_3)$, $c(3) = (r_3, s_4)$. The \mathcal{G}_{st} -edge set thus obtained is the following (see Fig. A2):

$$\mathcal{E}_{st} := \{(r_1, s_1), (r_1, s_3), (r_1, s_4), (r_2, s_1), (r_2, s_2), (r_3, s_2)\}$$

Note that this procedure does not provide the edge path of \mathcal{E}_{st} that links the vertices of the closure edge under consideration. Clearly, closure paths are not needed to be known for the construction of \mathcal{G}_{st} . In simple cases such as that of Fig. A2, such a path can visually be obtained by moving on grid \mathcal{G} horizontally and vertically, in alternate manner from the selected closure-edge point; see

the related telescoping sums introduced in Sect. 3.3. If need be, the edges paths can be obtained automatically in an algebraic manner; see Sect. 3.4.

Appendix B The S-system approach

In this appendix, we give a survey of the general framework of the S-system approach; for further details and related applications, see Baarda 1973; Teunissen 1984; de Jonge 1998; Teunissen and Odijk 2003.

Denoting by \mathcal{E} a Euclidean space of dimension n , we consider some linear operator \mathcal{A} from \mathcal{E} into \mathbb{R}^m with $m \geq n$ for example. The problem to be solved in a sense to be defined is governed by a relation of the form

$$\mathcal{A}\xi = \gamma \tag{B1}$$

The components of ξ are the unknown parameters of the problem, whereas γ is the data vector. In many situations encountered in practice, \mathcal{A} is not of full rank; its null space (i.e., its kernel) is not reduced to $\{0\}$:

$$n_0 \stackrel{\text{def}}{=} \dim(\ker \mathcal{A}) \geq 1 \tag{B2}$$

In the S-system approach, this rank defect is removed via an appropriate reduction and redefinition of the unknown parameters. Those new parameters are the ‘estimable functions of parameters’ of some minimum-constrained problem thus defined; see, e.g., de Jonge 1998. We now give a geometrical interpretation of the S-system principle.

Let us choose some subspace \mathfrak{F} of \mathcal{E} of dimension $n - n_0$ such that $\mathfrak{F} \cap \ker \mathcal{A} = \{0\}$; \mathcal{E} can then be regarded as the direct sum of \mathfrak{F} and $\ker \mathcal{A}$ (see Fig. B1):

$$\mathcal{E} = \mathfrak{F} \oplus \ker \mathcal{A} \tag{B3}$$

The ‘estimable functional variable’ is then defined as the oblique projection of ξ on \mathfrak{F} along $\ker \mathcal{A}$: $\mathcal{S}\xi$. The oblique projection (operator) \mathcal{S} is the \mathcal{S} -transformation of the S-system method; see, e.g., de Jonge 1998.

We now show how the S-system approach can provide the matrix of \mathcal{S} in the standard basis of \mathcal{E} . The estimable functional variable $\mathcal{S}\xi$, which basically depends on the choice of \mathfrak{F} , can thus be explicitly defined.

According to its definition,

$$\mathcal{S}\xi = \xi - \eta \tag{B4}$$

where η is the vector of $\ker \mathcal{A}$ such that $\xi - \eta$ lies in \mathfrak{F} ; see Fig. B1. Denoting by $[W]$ a matrix whose column vectors form a basis of $\ker \mathcal{A}$, we have

$$[\eta] = [W][\zeta] \tag{B5}$$

where ζ is some vector of \mathbb{R}^{n_0} . Clearly, the entries of $[\eta]$, $[W]$ and $[\zeta]$ are expressed in the standard basis of \mathcal{E} . Let $[S_\perp]$ now be a matrix whose column vectors form a

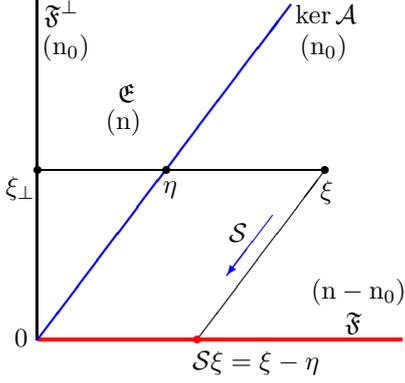


Figure B1: *S-system principle*. In this geometrical representation, \mathfrak{E} is a Euclidean space of dimension n . The unknown functional variable ξ is a vector of \mathfrak{E} . The null space of the operator \mathcal{A} involved in Eq. (B1) is of dimension n_0 : $\dim(\ker \mathcal{A}) = n_0$; \mathfrak{F} is a subspace of \mathfrak{E} of dimension $n - n_0$ such that $\mathfrak{F} \cap \ker \mathcal{A} = \{0\}$; \mathfrak{E} can then be regarded as the direct sum $\mathfrak{F} \oplus \ker \mathcal{A}$. In the S-system approach, the ‘estimable functional variable’ is then defined as the oblique projection of ξ on \mathfrak{F} along $\ker \mathcal{A}$: $S\xi$.

basis of \mathfrak{F}^\perp , the orthogonal complement of \mathfrak{F} in \mathfrak{E} . As $\xi - \eta$ is orthogonal to all the vectors of \mathfrak{F}^\perp , we have (in particular) $[S_\perp]^\top([\xi] - [W][\zeta]) = 0$, i.e.,

$$[S_\perp]^\top[W][\zeta] = [S_\perp]^\top[\xi] \quad (\text{B6})$$

As shown further on, $[S_\perp]^\top[W]$ is invertible. It then follows that $[\zeta] = ([S_\perp]^\top[W])^{-1}[S_\perp]^\top[\xi]$, hence from Eqs. (B4) and (B5),

$$[S] = [I] - [W]([S_\perp]^\top[W])^{-1}[S_\perp]^\top \quad (\text{B7})$$

where $[I]$ is the identity matrix on \mathfrak{E} .

We now show that the n_0 -by- n_0 matrix $[M_0] \stackrel{\text{def}}{=} [S_\perp]^\top[W]$ is invertible.

Proof. Let ξ_\perp be the projection of some vector η of $\ker \mathcal{A}$ on \mathfrak{F}^\perp ; see Fig. B1. By considering the case where $\xi = \xi_\perp$, Eq. (B6) yields $[M_0][\zeta] = [S_\perp]^\top[\xi_\perp]$. The condition $[M_0][\zeta] = 0$ implies $[S_\perp]^\top[\xi_\perp] = 0$, hence $\xi_\perp = 0$. As a result, η then lies in \mathfrak{F} . As $\mathfrak{F} \cap \ker \mathcal{A} = \{0\}$, it follows that $\eta = 0$, hence $\zeta = 0$. The null space of $[M_0]$ is therefore reduced to $\{0\}$; but $[M_0]$ is an n_0 -by- n_0 matrix; $[M_0]$ is therefore invertible. \square

Remark B1. In the S-system approach as it is implemented by de Jonge (1998) for example, one chooses some basis for \mathfrak{F} . The corresponding matrix is denoted by S ; \mathfrak{F} is then regarded as the range of S ; $[S_\perp]^\top \xi$ is called the ‘S-basis.’ Note that $[S_\perp]$ is then denoted by S^\perp \square

Appendix C QR implementation

In this appendix, we show how the float version of Eq. (81) can be solved in the LS sense via recursive QR factorization. Here, for simplicity, we will assume that the elementary orthogonal transformations involved in that factorization are Givens rotations; see Eqs. (2.3.10) to (2.3.13) in Björck (1996).

In the closure-ambiguity approach, the number of entries of \mathbf{v} , n_c , is a non-decreasing function of k ; see Sect. 6.3. In Sect. C1, we consider the case where n_c is constant; the cases where at some epochs k , n_c increases is dealt with in Sect. C2. In that QR framework, we finally describe in Sect. C3 the construction of the matrices \mathbf{A}_k , \mathbf{B}_k and \mathbf{b}_k involved in Eq. (81).

C1 Recursive QR factorization

As shown in this section, the float version of the following equation (Eq. (81)) can be solved in the LS sense via recursive QR factorization:

$$\begin{bmatrix} \mathbf{B}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_1 \\ \mathbf{0} & \mathbf{B}_2 & \mathbf{0} & \mathbf{0} & \mathbf{A}_2 \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_k & \mathbf{A}_k \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_k \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_k \end{bmatrix}$$

Throughout this section, n_c is assumed to be fixed; for related notions, see Sect. 6.3 of Björck 1996; Golub and van Loan 1989; Bierman 1977.

C1.1 Initialization: epoch 1

At epoch 1, the problem is to minimize the functional (see the first line of Eq. (81))

$$f_1(\mathbf{w}_1, \mathbf{v}) \stackrel{\text{def}}{=} \|(\mathbf{B}_1 \mathbf{w}_1 + \mathbf{A}_1 \mathbf{v}) - \mathbf{b}_1\|_{\mathbb{R}^{n_{e1}}}^2$$

The LS solution $(\hat{\mathbf{w}}_1, \hat{\mathbf{v}})$ is then obtained via two QR factorizations (see Fig. C1).

1) *QR factorization of \mathbf{B}_1* : the Givens rotations of this step are those required for finding the upper-triangular matrix \mathbf{K}_1 . The modified version of \mathbf{A}_1 thus obtained includes an upper block \mathbf{L}_1 and a lower block \mathbf{L}'_1 . Likewise, the modified version of \mathbf{b}_1 includes two column submatrices: \mathbf{c}_1 and \mathbf{c}'_1 .

2) *QR factorization of \mathbf{L}'_1* : the Givens rotations of that step yield the upper-triangular matrix \mathbf{R}_1 . The lower part of \mathbf{L}'_1 is reduced to $\mathbf{0}$; \mathbf{c}'_1 then yields $(\mathbf{d}_1, \mathbf{d}'_1)$; see Fig. C1. Note that \mathbf{K}_1 , \mathbf{L}_1 and \mathbf{c}_1 are not affected by those rotations.

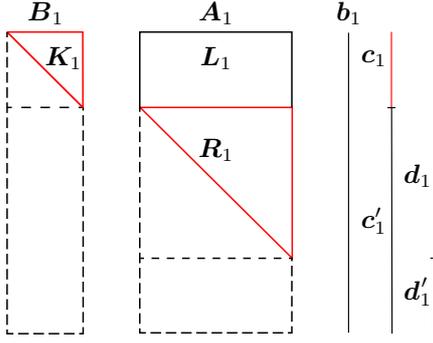


Figure C1: *QR factorization at epoch 1.* The principle of the recursive QR method is sketched here for the first epoch with the input block matrices \mathbf{B}_1 , \mathbf{A}_1 and the data column matrix \mathbf{b}_1 . The initialization process is performed in two steps: \mathbf{K}_1 , $(\mathbf{L}_1, \mathbf{L}'_1)$, $(\mathbf{c}_1, \mathbf{c}'_1)$ are built in the first step (see text for \mathbf{L}'_1), whereas \mathbf{R}_1 , $(\mathbf{d}_1, \mathbf{d}'_1)$ are built in the second one; for the LS solution thereby obtained at epoch 1, see text.

At the end of this initialization stage, we thus have

$$f_1(\mathbf{w}_1, \mathbf{v}) = \|(\mathbf{K}_1 \mathbf{w}_1 - (\mathbf{c}_1 - \mathbf{L}_1 \mathbf{v}))\|_{\mathbb{R}^{n_{b1}}}^2 + \|\mathbf{R}_1 \mathbf{v} - \mathbf{d}_1\|_{\mathbb{R}^{n_c}}^2 + \|\mathbf{d}'_1\|_{\mathbb{R}^{n_{e1} - n_{b1} - n_c}}^2$$

The float solution in \mathbf{v} at epoch 1 is therefore given by the formula $\hat{\mathbf{v}} = \mathbf{R}_1^{-1} \mathbf{d}_1$, hence $\hat{\mathbf{w}}_1 = \mathbf{K}_1^{-1} (\mathbf{c}_1 - \mathbf{L}_1 \hat{\mathbf{v}})$. These solutions can therefore be computed by back substitution. Note that $\|\mathbf{d}'_1\|_{\mathbb{R}^{n_{e1} - n_{b1} - n_c}}^2$ is the square of the LS residual norm at epoch 1.

C1.2 Next epoch: epoch 2

The functional to be minimized is then $f_1(\mathbf{w}_1, \mathbf{v}) + f_2(\mathbf{w}_2, \mathbf{v})$ where

$$f_2(\mathbf{w}_2, \mathbf{v}) \stackrel{\text{def}}{=} \|(\mathbf{B}_2 \mathbf{w}_2 + \mathbf{A}_2 \mathbf{v}) - \mathbf{b}_2\|_{\mathbb{R}^{n_{e2}}}^2$$

As sketched in Fig. C2, the LS solution $(\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2, \hat{\mathbf{v}})$ is again obtained via two QR factorizations. The first step of epoch 2 is similar to that of epoch 1; the second one is different.

1) QR factorization of \mathbf{B}_2

One thus obtains the upper-triangular matrix \mathbf{K}_2 ; see Fig. C2. The modified version of \mathbf{A}_2 then includes an upper block \mathbf{L}_2 and a lower block \mathbf{L}'_2 . Likewise, the modified version of \mathbf{b}_2 includes two column submatrices: \mathbf{c}_2 and \mathbf{c}'_2 .

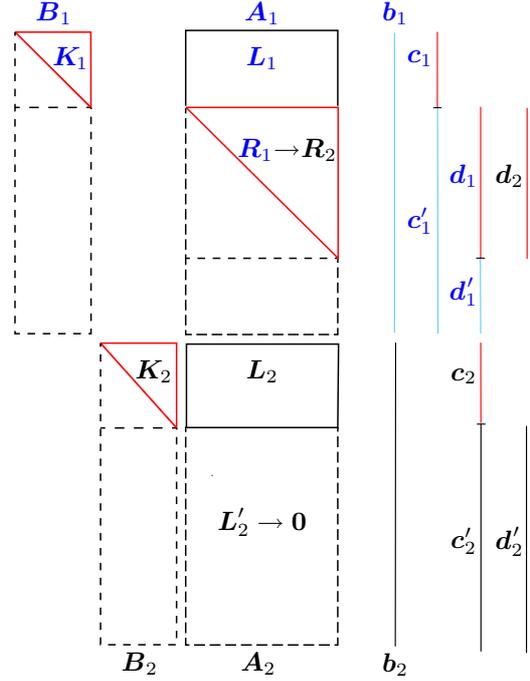


Figure C2: *QR factorization at epochs 1 and 2.* The principle of the recursive QR method is sketched here for the first two epochs: epoch 1 with the input block matrices \mathbf{B}_1 , \mathbf{A}_1 and the data column matrix \mathbf{b}_1 ; epoch 2 with the input block matrices \mathbf{B}_2 , \mathbf{A}_2 and the data column matrix \mathbf{b}_2 . The initialization process is performed in two steps as described in Fig. C1. At epoch 2, one first builds \mathbf{K}_2 , $(\mathbf{L}_2, \mathbf{L}'_2)$, $(\mathbf{c}_2, \mathbf{c}'_2)$ like at epoch 1, and then \mathbf{R}_2 , $(\mathbf{d}_2, \mathbf{d}'_2)$; for the LS solution thereby obtained at epoch 2, see text.

2) QR factorization of $\begin{bmatrix} \mathbf{R}_1 \\ \mathbf{L}'_2 \end{bmatrix}$:

The Givens rotations of the second step then operate on $(\mathbf{R}_1, \mathbf{L}'_2)$ and $(\mathbf{d}_1, \mathbf{c}'_2)$ so as to transform \mathbf{L}'_2 into a zero-block matrix. One thus gets \mathbf{R}_2 and $(\mathbf{d}_2, \mathbf{d}'_2)$.

At the end of this stage, we thus have

$$f_1(\mathbf{w}_1, \mathbf{v}) + f_2(\mathbf{w}_2, \mathbf{v}) = \|(\mathbf{K}_1 \mathbf{w}_1 - (\mathbf{c}_1 - \mathbf{L}_1 \mathbf{v}))\|_{\mathbb{R}^{n_{b1}}}^2 + \|\mathbf{R}_2 \mathbf{v} - \mathbf{d}_2\|_{\mathbb{R}^{n_c}}^2 + \|\mathbf{d}'_1\|_{\mathbb{R}^{n_{e1} - n_{b1} - n_c}}^2 + \|(\mathbf{K}_2 \mathbf{w}_2 - (\mathbf{c}_2 - \mathbf{L}_2 \mathbf{v}))\|_{\mathbb{R}^{n_{b2}}}^2 + \|\mathbf{d}'_2\|_{\mathbb{R}^{n_{e2} - n_{b2}}}^2$$

The float solution in \mathbf{v} at epoch 2 is therefore given by the formula $\hat{\mathbf{v}} = \mathbf{R}_2^{-1} \mathbf{d}_2$, hence the LS solutions in \mathbf{w}_1 and \mathbf{w}_2 :

$$\hat{\mathbf{w}}_1 = \mathbf{K}_1^{-1} (\mathbf{c}_1 - \mathbf{L}_1 \hat{\mathbf{v}}), \quad \hat{\mathbf{w}}_2 = \mathbf{K}_2^{-1} (\mathbf{c}_2 - \mathbf{L}_2 \hat{\mathbf{v}})$$

The square of the LS residual norm at epoch 2 is then equal to $\|\mathbf{d}'_1\|_{\mathbb{R}^{n_{e1} - n_{b1} - n_c}}^2 + \|\mathbf{d}'_2\|_{\mathbb{R}^{n_{e2} - n_{b2}}}^2$.

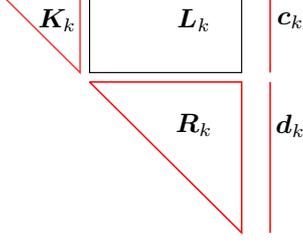


Figure C3: *Recursive QR triangular structure.* According to the principle of the recursive QR method sketched in Fig. C2, the calculation of \mathbf{R}_{k+1} and \mathbf{d}_{k+1} requires to have kept in memory the upper-triangular matrix \mathbf{R}_k and the column matrix \mathbf{d}_k ; see text.

C1.3 Next epochs

In summary, one thus operates, recursively, with the key structure shown in Fig. C3: \mathbf{K}_k , $(\mathbf{L}_k, \mathbf{L}'_k)$ and $(\mathbf{c}_k, \mathbf{c}'_k)$ are computed from \mathbf{B}_k , \mathbf{A}_k and \mathbf{b}_k , the quantities \mathbf{R}_k and $(\mathbf{d}_k, \mathbf{d}'_k)$ being then computed from $(\mathbf{R}_{k-1}, \mathbf{L}'_k)$ and $(\mathbf{d}_{k-1}, \mathbf{c}'_k)$. The generalization is straightforward; we then have

$$\begin{aligned} \sum_{\kappa=1}^k f_{\kappa}(\mathbf{w}_1, \mathbf{v}) = & \|(\mathbf{K}_1 \mathbf{w}_1 - (\mathbf{c}_1 - \mathbf{L}_1 \mathbf{v}))\|_{\mathbb{R}^{n_{b1}}}^2 \\ & + \|\mathbf{R}_k \mathbf{v} - \mathbf{d}_k\|_{\mathbb{R}^{n_c}}^2 \\ & + \|\mathbf{d}'_1\|_{\mathbb{R}^{n_{e1} - n_{b1} - n_c}}^2 \\ & + \|(\mathbf{K}_2 \mathbf{w}_2 - (\mathbf{c}_2 - \mathbf{L}_2 \mathbf{v}))\|_{\mathbb{R}^{n_{b2}}}^2 \\ & + \|\mathbf{d}'_2\|_{\mathbb{R}^{n_{e2} - n_{b2}}}^2 \\ & \vdots \\ & + \|(\mathbf{K}_k \mathbf{w}_k - (\mathbf{c}_k - \mathbf{L}_k \mathbf{v}))\|_{\mathbb{R}^{n_{bk}}}^2 \\ & + \|\mathbf{d}'_k\|_{\mathbb{R}^{n_{ek} - n_{bk}}}^2 \end{aligned}$$

The float solution in \mathbf{v} at epoch k is therefore given by the formula

$$\hat{\mathbf{v}} = \mathbf{R}_k^{-1} \mathbf{d}_k \quad (\text{C1})$$

hence the LS solutions in $\mathbf{w}_1, \dots, \mathbf{w}_k$:

$$\hat{\mathbf{w}}_{\kappa} = \mathbf{K}_{\kappa}^{-1} (\mathbf{c}_{\kappa} - \mathbf{L}_{\kappa} \hat{\mathbf{v}}) \quad (\kappa = 1, \dots, k) \quad (\text{C2})$$

The solution of the corresponding NLP problem is therefore defined as follows:

$$\tilde{\mathbf{v}} = \underset{\mathbf{v} \in \mathbb{Z}^{n_c}}{\operatorname{argmin}} \|\mathbf{R}_k (\mathbf{v} - \hat{\mathbf{v}})\|_{\mathbb{R}^{n_c}}^2 \quad (\text{C3})$$

Indeed, $\mathbf{R}_k \mathbf{v} - \mathbf{d}_k = \mathbf{R}_k (\mathbf{v} - \hat{\mathbf{v}})$. The phase biases $\tilde{\mathbf{w}}_{\kappa}$ are then given by the relations

$$\tilde{\mathbf{w}}_{\kappa} = \mathbf{K}_{\kappa}^{-1} (\mathbf{c}_{\kappa} - \mathbf{L}_{\kappa} \tilde{\mathbf{v}}) \quad (\kappa = 1, \dots, k) \quad (\text{C4})$$

Their variance-covariance matrix is equal to $\mathbf{K}_{\kappa}^{-1} [\mathbf{K}_{\kappa}^{-1}]^T$.

C2 Handling new components of the closure-ambiguity variable

We now consider the case where $\mathcal{L}_{c,k}$ includes n_c^a new closure edges (see Sect. 6.3); superscript a stands for added. One then proceeds in three steps:

- 1) n_c^a closure-ambiguity entries are added at the top of column matrix \mathbf{v} ;
- 2) as specified in Sect. C3.2, n_c^a columns are added on the right-hand side of \mathbf{B}_k ;
- 3) as shown in Fig. C4, to build \mathbf{R}_k , the last n_c^a lines of \mathbf{K} and \mathbf{L} obtained through the first QR step are added at the top of \mathbf{R} . Matrices \mathbf{d}_k , \mathbf{K}_k , \mathbf{L}_k and \mathbf{c}_k are then updated accordingly.

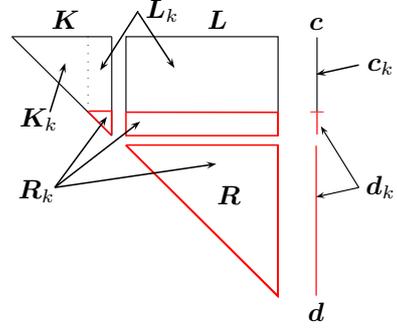


Figure C4: *Handling new components of the closure-ambiguity variable.* When new entries of \mathbf{v} appear at epoch k , the first columns of \mathbf{A}_k are processed as the last columns of \mathbf{B}_k (see Fig. C2). The recursive QR operation then yields the quantities \mathbf{K} , \mathbf{L} , \mathbf{c} , \mathbf{R} and \mathbf{d} . To get \mathbf{K}_k , \mathbf{L}_k , \mathbf{c}_k , \mathbf{R}_k and \mathbf{d}_k , one then proceeds as illustrated here.

C3 Construction of matrices \mathbf{B}_k , \mathbf{A}_k and \mathbf{b}_k

We first consider the case where the variance-covariance matrix \mathbf{V}_k of the data involved in the definition of \mathbf{b}_k is the identity: $\mathbf{V}_k = \mathbf{I}$. Denoting by \mathbf{b}_k the column matrix whose entries are the values of b_k on the edges of the observational graph \mathcal{H}_k (see Sect. 1), we then have $\mathbf{b}_k = \mathbf{b}_k$. To build $\mathbf{B}_k = \mathbf{B}_k$ and $\mathbf{A}_k = \mathbf{A}_k$, we then distinguish the cases where at epoch k , n_c does not increase (Sect. C3.1), or increases (Sect. C3.2). The case $\mathbf{V}_k \neq \mathbf{I}$ is dealt with in Sect. C3.3.

C3.1 Case where n_c does not increase

Matrix \mathbf{B}_k , whose number of columns is n_{bk} , is built from the characteristic function H_k of \mathcal{H}_k ; see Fig. 1. The p th line of \mathbf{B}_k corresponds to the p th edge (r_i, s_j) on which $H_k(i, j) = 1$. All the matrix elements of that

line are zero, except (one or) two of them (see Eq. (6) and the definition of ϖ_{sk} in Sect. 6.2):

$$\mathbf{B}_k^{p,i-1} = 1 \text{ (for } i > 1), \quad \mathbf{B}_k^{p,n_r-1+\varpi_{sk}(j)} = -1$$

Matrix \mathbf{A}_k has n_c columns: the number of elements of $\mathcal{L}_{c,k}$; see Sect 6.3 and Fig. 5. According to the action of \mathcal{R}_k^e , the entries of the column associated with some closure edge (r_i, s_j) are then all zero, except that corresponding to the line associated with that edge if $H_k(i, j) = 1$; that entry is then set equal to unity. The lines of \mathbf{A}_k are of course sorted as the lines of \mathbf{B}_k .

C3.2 Case where n_c increases

We here consider the case where n_c^a new closure edge(s) appear(s) in $\mathcal{L}_{c,k}$ at some epoch $k > 1$: $n_c := n_c + n_c^a$; see Sect 6.3.

Matrix $\mathbf{B}_k^{p,q}$ is defined as in Sect. C3.1, but n_c^a columns are then added on its right-hand side. (For example, at epoch 2 of Fig. 5, \mathbf{B}_2 has four additional columns.) The entries of the column of \mathbf{B}_k associated with some new closure edge (r_i, s_j) are all zero, except that corresponding to the line associated with that edge; that entry is set equal to unity.

Matrix \mathbf{A}_k is then built as in Sect. C3.1, except for the new closure-edges, since they are then taken into account in the augmented definition of \mathbf{B}_k .

C3.3 Case where \mathbf{V}_k is not the identity

We here consider the general case where the variance-covariance matrix \mathbf{V}_k is to be taken into account. In the QR implementation under consideration, the inverse of \mathbf{V}_k is then factorized in the form

$$\mathbf{V}_k^{-1} = \mathbf{U}_k^T \mathbf{U}_k \quad (\text{C5})$$

where \mathbf{U}_k is an upper-triangular matrix. As

$$\begin{aligned} (\mathbf{B}_k \mathbf{w}_k + \mathbf{A}_k \mathbf{v} - \mathbf{b}_k)^T \mathbf{V}_k^{-1} (\mathbf{B}_k \mathbf{w}_k + \mathbf{A}_k \mathbf{v} - \mathbf{b}_k) \\ = \|\mathbf{U}_k (\mathbf{B}_k \mathbf{w}_k + \mathbf{A}_k \mathbf{v} - \mathbf{b}_k)\|_{\mathbb{R}^{n_{ek}}}^2 \end{aligned}$$

matrices \mathbf{B}_k , \mathbf{A}_k and \mathbf{b}_k are then given by the relations

$$\mathbf{B}_k = \mathbf{U}_k \mathbf{B}_k, \quad \mathbf{A}_k = \mathbf{U}_k \mathbf{A}_k, \quad \mathbf{b}_k = \mathbf{U}_k \mathbf{b}_k \quad (\text{C6})$$

The problem is then to solve Eq. (81) in the Euclidean LS sense.

Acknowledgments

The authors are very grateful to Flavien Mercier and Félix Perosanz for helpful discussions.

References

- Baarda W (1973) S-transformations and criterion matrices. Publications in Geodesy 1. Netherlands Geodetic Commission, Delft. 5
- Bertiger W, Desai SD, Haines B, Harvey N, Moore AW (2010) Single receiver phase ambiguity resolution with GPS data. J Geod 84: 327-337
- Bierman GJ (1977) Factorization methods for discrete sequential estimation. Vol. 128 in Mathematics in science and engineering, Academic Press, Inc. New-York
- Biggs N (1996) Algebraic graph theory. 2nd edn. Cambridge University Press, Cambridge
- Björck A (1996) Numerical methods for least-squares problems. SIAM
- Blewitt G (1989) Carrier phase ambiguity resolution for the global positioning system applied to geodetic baselines up to 2000 km. J Geophys Res 94(B8): 10187-10203
- Collins P, Bisnath S, Lahaye F, Heroux P (2010) Undifferenced GPS ambiguity resolution using the decoupled clock model and ambiguity datum fixing. J Navigation 57: 123-135
- de Jonge PJ (1998) A processing strategy for the application of the GPS in networks. PhD dissertation 46. Netherlands Geodetic Commission, Delft
- Ge M, Gendt G, Dick G, Zhang FP (2005) Improving carrier-phase ambiguity resolution in global GPS network solutions. J Geod 79: 103-110
- Ge M, Gendt G, Rothacher M, Shi C, Lui J (2008) Resolution of GPS carrier-phase ambiguities in precise point positioning (PPP) with daily observations. J Geod 82: 389-399
- Ge M, Douša J, Li X, Ramatschi M, Nischan T, Wickert J (2012) A novel real-time precise positioning service system: global precise point positioning with regional augmentation. J GPS 11: 2-10
- Geng J, Meng X, Dodson AH, Teferle FN (2010) Integer ambiguity resolution in precise point positioning: method comparison. J Geod 84: 569-581
- Golub GH and van Loan CF (1989) Matrix computations, 2nd edn. The Johns Hopkins University Press, Baltimore, Maryland
- Lannes A, Gratton S (2009) GNSS networks in algebraic graph theory. J GPS 8: 53-75
- Lannes A, Prieur JL (2011) Algebraic analysis of the phase-calibration problem in the self-calibration procedures. Astron Nachr 332: 759-784

- Lannes A, Teunissen PJG (2011) GNSS algebraic structures. *J Geod* 85: 273-290
- Lannes A (2013) On the theoretical link between LLL reduction and LAMBDA decorrelation. *J Geod* 87: 323-335
- Lannes A, Prieur JL (in press) Integer-ambiguity resolution in astronomy and geodesy. *Astron Nachr*
- Laurichesse D, Mercier F (2007) Integer ambiguity resolution on undifferenced GPS phase measurements and its applications to PPP. ION GNSS 2007 20th international technical meeting of the satellite division, 25-28 Sept 2007, Forth Worth, TX, pp 839-848
- Lenstra AK, Lenstra HW, Lovász L (1982) Factorizing polynomials with rational coefficients. *Math Ann* 261: 515-534.
- Li X, Ge M, Zhang H, Wickert J (2013) A method for improving uncalibrated phase delay estimation and ambiguity-fixing in real-time precise point positioning. *J Geod* DOI:10.1007/s00190-013-0611-x
- Loehnert E, Wolf R, Pielmeier J, Werner W, Zink T (2000) Concepts and performance results on the combination of different integrity methods using UAIM and GNSS without SA. Proc. ION GPSS-2000. Salt Lake City, Utah USA: 2831-2840
- Loyer S, Perosanz F, Mercier F, Capdeville H, Marty JC (2012) Zero-difference GPS ambiguity resolution at CNES-CLS IGS Analysis Center. *J Geod* 86: 991-1003
- Odijk D, Teunissen PJG, Zhang B (2012) Single-frequency integer ambiguity resolution enabled precise point positioning. *J Surv Eng* 138: 193-202
- Teunissen PJG (1984) Generalized inverses, adjustment, the datum problem and S-transformations. In: E. Grafarend and F. Sanso (eds) *Optimization of Geodetic Networks*, Springer, Berlin, pp 11-55
- Teunissen PJG and Odijk D (2003) Rank-defect integer estimation and phase-only modernized GPS ambiguity resolution. *J Geod* 76: 523-535
- Teunissen PJG, Odijk D, Zhang B (2010) PPP-RTK: Results of CORS network-based PPP with integer ambiguity resolution. *J Aeronautics, Astronautics and Aviation* 42: 223-229
- Tiberius CCJM (1998) Recursive data processing for kinematic GPS surveying. Publications on Geodesy, New series: ISSN 0165 1706, Number 45, Netherlands Geodetic Commission, Delft
- Verhagen S, Teunissen PJG (2006) New global satellite system ambiguity resolution methods compared to existing approaches. *J Guid Control Dynam* 29: 891-991
- Zhang B, Teunissen PJG, Odijk D (2011) A novel undifferenced PPP-RTK concept. *J Navigation* 64: S180-S191
- Zumberge JF, Heflin MB, Jefferson DC, Watkins MM, Webb FH (1997) Precise point positioning for the efficient and robust analysis of the GPS data from large networks. *J Geophys Res* 102(B3): 5005-5017
-