



RESEARCH ARTICLE

A mixed-integer least-squares formulation of the GNSS snapshot positioning problem

Eyal Wasserman, and Sivan Toledo* 

Blavatnik School of Computer Science, Tel-Aviv University, Tel-Aviv, Israel.

*Corresponding author. E-mail: stoledo@tau.ac.il

Received: 10 January 2021; Accepted: 26 July 2021; First published online: 26 August 2021

Keywords: GNSS, Doppler, integer ambiguity resolution, satellite navigation

Abstract

This paper presents a formulation of *snapshot positioning* as a mixed-integer least-squares problem. In snapshot positioning, one estimates a position from code-phase (and possibly Doppler-shift) observations of global navigation satellite system (GNSS) signals without knowing the time of departure (timestamp) of the codes. Solving the problem allows a receiver to determine a fix from short radio-frequency snapshots missing the timestamp information embedded in the GNSS data stream. This is used to reduce the time to first fix in some receivers, and it is used in certain wildlife trackers. This paper presents two new formulations of the problem and an algorithm that solves the resulting mixed-integer least-squares problems. We also show that the new formulations can produce fixes even with huge initial errors, much larger than permitted in Van Diggelen's widely-cited *coarse-time navigation* method.

1. Introduction

The fundamental observation equation in a global navigation satellite system (GNSS) is

$$t_i - t_{D,i} = \frac{1}{c} \|\hat{\ell} - \rho_i(t_{D,i})\|_2 + \hat{b} + \delta_i + \epsilon_i,$$

where t_i is the observed (estimated) time of arrival of a code from satellite i , $t_{D,i}$ is the time of departure of the signal, c is the speed of light, $\hat{\ell}$ is the location of the receiver, $\rho_i(t_{D,i})$ is the location of the satellite at the time of transmission, \hat{b} is the offset in time-of-arrival observation caused by the inaccurate clock at the receiver and by delays in the analogue radio-frequency (RF) chain (e.g. in cables), δ_i represents atmospheric delays and the satellite's clock error, and ϵ_i is an error or noise term that accounts for both physical noise and for unmodelled effects. Normally, the GNSS solver estimates $\hat{\ell}$ and \hat{b} by minimising the norm of the error vector ϵ ; it does not know ϵ_i and does not attempt to estimate it. The quantities $t_{D,i}$ and $\rho_i(t_{D,i})$ are usually known; $t_{D,i}$ is known because the satellite timestamps its transmission, and $\rho_i(t_{D,i})$ is known because the satellite transmits the parameters that define its orbit, called the *ephemeris*. The ephemeris can also be downloaded from the Internet.

Decoding $t_{D,i}$ takes a significant amount of time, in a global positioning system (GPS) up to 6 s under good signal-to-noise ratio (SNR) conditions and longer in low-SNR conditions. GNSS receivers that need to log locations by observing the RF signals for short periods cannot decode $t_{D,i}$. Examples for such applications include tracking marine animals like sea turtles, which surface briefly and then submerge again. It turns out that techniques that are collectively called *snapshot positioning* or *coarse-time navigation* can estimate $\hat{\ell}$ and \hat{b} when $t_{D,i}$ is not known. These techniques can also reduce the time to a first fix when a receiver is turned on.

Snapshot receivers sample the incoming GNSS RF signals for a short period, called a *snapshot*. Usually (but not always), the RF samples are correlated with replicas of the codes transmitted by the satellites, therefore determining t_i for the subset of visible satellites. The correlation (and Doppler search) is sometimes performed on the receiver, which then stores or transmits the t_i data. This appears to be the case for a proprietary technology called Fastloc, which is used primarily to track marine animals (Tomkiewicz et al., 2010; Witt et al., 2010; Dujon et al., 2014). In other cases (Ramos et al., 2011; Bavaro, 2012a, 2012b; Liu et al., 2012; Cvikel et al., 2015; Eichelberger et al., 2019; Harten et al., 2020), the logger records the raw RF samples and correlation is performed after the data are uploaded to a computer.

Techniques for estimating $\hat{\ell}$ when the $t_{D,i}$ data are not known date back to a paper by Peterson et al. (1995). They proposed to view $t_{D,i}$ as a function of both t_i and a coarse clock-error unknown that they call *coarse time*, which in principle is identical to \hat{b} , but is modelled by a separate variable. They then showed that it is usually possible to estimate $\hat{\ell}$, \hat{b} and the coarse time from five or more t_i points. This method does not always resolve the $t_{D,i}$ correctly. Lannelongue and Pablos (1998) and Van Diggelen (2002, 2009) proposed methods that appear to always resolve the $t_{D,i}$ correctly when the initial estimate of $\hat{\ell}$ and \hat{b} are within some limits (adding up to approximately 150 km). Muthuraman et al. (2012) showed that the two methods are equivalent in the sense that they usually produce the same estimates. However, the method of Lannelongue and Pablos is an iterative search procedure, while Van Diggelen's is a rounding procedure that is more computationally efficient, so Van Diggelen's method became much more widely used and widely cited. Van Diggelen also showed how to use an iterative procedure over a number of possible positions when the initial estimate of $\hat{\ell}$ and \hat{b} is outside the 150 km limit.

All three methods use a system of linearised equations with five scalar unknowns (not the usual four), which are corrections to the coordinates of $\hat{\ell}$, the offset \hat{b} (which he refers to as the common bias) and the coarse time unknown, usually denoted by tc (we denote it in this paper by the single letter s).

Van Diggelen's algorithm was used and cited in numerous subsequent papers, all of which repeat his presentation without adding explanations. Liu et al. (2012), Ramos et al. (2011) and Wang et al. (2019) describe snapshot GPS loggers whose recordings are processed using the algorithm. Badia-Solé and Iacobescu Ioan (2010) report on the performance of the method. Othieno and Gleason (2012), Chen et al. (2014) and Fernández-Hernández and Borre (2016) show how to use Doppler measurements to obtain an initial estimate that satisfies the requirements of Van Diggelen's algorithm. Yoo et al. (2020) propose a technique that replaces the estimation of the coarse-time parameter by a one-dimensional search, which allows them to estimate $\hat{\ell}$ using observations from four satellites, but at considerable computational expense.

Bissig et al. (2017) use a *direct position determination* (Weiss, 2004) approach to snapshot positioning. They quantise the four unknowns $\hat{\ell}$ and \hat{b} , and maximise the likelihood of the received snapshot over this four-dimensional lattice. To make the search efficient, they use a branch and bound approach that prunes sets of unlikely solutions. It appears that this approach allows them to estimate locations using very short snapshots, but at the cost of fairly low accuracy and relatively long running times, compared with methods that estimate the t_i data first.

In this paper we derive the observation equations that underlie the methods of Peterson et al. (1995), Lannelongue and Pablos (1998) and Van Diggelen (2002, 2009). These authors show the correction equations, not the observation equations whose Jacobian constitutes the correction equations. The formulation of the observation equations, which constitute a mixed-integer least-squares problem, allows us to apply a new type of algorithm to estimate the integer unknowns. A mixed-integer least-squares problem is an optimisation problem with a least-squares objective function and both real (continuous) unknowns and integer unknowns. More specifically, we regularise the mixed-integer problem using either a priori estimates of $\hat{\ell}$ and \hat{b} or Doppler-shift observations. Our approach is inspired by the *real-time kinematic* (RTK) method, which resolves a position from both code-phase and carrier-phase GNSS observations (Teunissen, 2017); carrier-phase constraints have integer ambiguities that must be resolved.

Our experimental results using real-world data demonstrate that our new algorithms can resolve locations with much larger initial location and time errors than the method of Van Diggelen. Van Diggelen's non-iterative method only works when the initial estimate is up to approximately 150 km (or equivalent combinations), whereas our mixed-integer least-squares solver works with initial errors of up to 150 s and 200 km. When using Doppler-shift regularisation, our method works even with initial errors of 180 s and arbitrarily large initial position errors (on Earth); if the initial position error is small, the method tolerates initial time errors of up to 5000 s.

Our implementation of the new methods and the code that we used to evaluate them are publicly available.¹

The rest of this paper is organised as follows. Section 2 presents the observation equations for the snapshot-positioning GNSS problem. Section 3 explains how to incorporate the so-called *coarse-time* parameter into the observation equations and how Van Diggelen's method exploits it. Section 4.1 presents our first regularised formulation, which uses the initial guess to regularise the mixed-integer least-squares problem. Section 4.2 presents the Doppler-regularised formulation. Our experimental results are presented in Section 5. Section 6 discusses our conclusions from this research.

2. The snapshot-positioning problem: a GNSS model with whole-millisecond ambiguities

We begin by showing that when departure times are not known, the observation equations that relate the arrival times of GNSS codes to the unknown position of the receiver contain integer ambiguities.

We denote by $t_{D,i}$ the time of departure of a code from satellite i , and we assume that $t_{D,i}$ represents a whole millisecond (in the time base of the GPS system). We denote by \hat{t}_i the time of arrival of that code at the antenna of the receiver. We assume that the receiver estimates the arrival time of that code as $t_i = \hat{t}_i + \hat{b} + \epsilon_i$, where \hat{b} represents the bias that is caused by the inaccurate clock of the receiver and by delays that the signal experiences in the path from the antenna to the analogue-to-digital converter, and ϵ_i is the arrival-time estimation error. The bias \hat{b} is time dependent, because of drift in the receiver's clock, but over short observation periods this dependence is negligible, so we ignore it.

The time of arrival is governed by the equation

$$\hat{t}_i - t_{D,i} = \frac{1}{c} \|\hat{\ell} - \rho_i(t_{D,i})\|_2 + \delta_i,$$

where c is the speed of light, $\hat{\ell}$ is the location of the receiver, ρ_i is the location of the satellite (which is a function of time, since the satellites are not stationary relative to Earth observers), and δ_i represents the inaccuracy of the satellite's clock and atmospheric delays. We assume that δ_i can be modelled, for example using models of ionospheric and tropospheric delays (dual frequency receivers can estimate the ionospheric delay, but we assume a single-frequency receiver). Setting δ_i to the satellite's clock error correction from the ephemeris induces a location error of approximately 30 m due to the atmospheric delays (Borre and Strang, 2012).

A receiver that decodes the timestamp embedded in the GPS data stream can determine $t_{D,i}$, which leads to the following equation:

$$t_i - t_{D,i} = \frac{1}{c} \|\hat{\ell} - \rho_i(t_{D,i})\|_2 + \hat{b} + \delta_i + \epsilon_i,$$

the conventional GNSS code-observation equation, in which the four unknown parameters are \hat{b} and the coordinates of $\hat{\ell}$ (we assume that δ_i is modelled, possibly trivially $\delta_i = 0$, but not estimated).

¹<https://github.com/eyalw711/snapshot-positioning>

To simplify the notation, we ignore δ_i for now and write

$$t_i - t_{D,i} = \frac{1}{c} \|\hat{\ell} - \rho_i(t_{D,i})\|_2 + \hat{b} + \epsilon_i.$$

We use observations from all the satellites such that all the t_i data lie between two consecutive whole multiples of t_{code} (in GPS, two round milliseconds in the local clock). This allows us to express

$$t_i = (N + \varphi_i)t_{\text{code}}$$

with a common and easily computable $N = \lfloor t_i/t_{\text{code}} \rfloor$ and for $\varphi_i \in [0, 1)$. We denote $N_i = t_{D,i}/t_{\text{code}}$ and write

$$(N - N_i + \varphi_i)t_{\text{code}} = \frac{1}{c} \|\hat{\ell} - \rho_i(t_{D,i})\|_2 + \hat{b} + \epsilon_i.$$

Since GNSS codes are aligned with t_{code} , $N_i \in \mathbb{Z}$. We denote $n_i = N - N_i \in \mathbb{Z}$, so

$$(n_i + \varphi_i)t_{\text{code}} = \frac{1}{c} \|\hat{\ell} - \rho_i(t_{D,i})\|_2 + \hat{b} + \epsilon_i$$

or

$$\varphi_i t_{\text{code}} = \frac{1}{c} \|\hat{\ell} - \rho_i(t_{D,i})\|_2 - n_i t_{\text{code}} + \hat{b} + \epsilon_i.$$

We now face two challenges. One is that we have $4 + m$ unknown parameters: three location coordinates, \hat{b} and n_i , but only m constraints. We clearly need more constraints so that we can resolve n_i . The other is that we have a set of nonlinear constraints with continuous real unknowns, the location and \hat{b} , and with integer unknowns, the n_i . The strategy, as in other cases with this structure, is to first linearise the nonlinear term, then to resolve the integer parameters, and to then substitute them and to solve the continuous least-squares problem (either the linearised system or the original nonlinear system). We cannot linearise the nonlinear term $\|\hat{\ell} - \rho_i(t_{D,i})\|_2/c$ using a Taylor series because it is a function of both real unknowns and of the integer unknowns n_i . We cannot differentiate this term with respect to the integer n_i .

To address this difficulty, we approximate $t_{D,i}$ by approximating the range (distance) term in the equation

$$t_{D,i} = t_i - \frac{1}{c} \|\hat{\ell} - \rho_i(t_{D,i})\|_2 - \hat{b} - \epsilon_i.$$

For now, we denote the approximation of the propagation delay by

$$d_i \approx \frac{1}{c} \|\hat{\ell} - \rho_i(t_{D,i})\|_2 + \epsilon_i$$

so

$$\hat{t}_{D,i} = t_i - d_i - \hat{b}.$$

There are several ways to set d_i , depending on our prior knowledge of $\hat{\ell}$ and \hat{b} . One option in the GPS system is to set it to approximately 76.5 ms; this limits the error in $\hat{t}_{D,i}$ to approximately 12.5 ms for any Earth observer, and the error

$$\|\hat{\ell} - \rho_i(\hat{t}_{D,i})\|_2 - \|\hat{\ell} - \rho_i(t_{D,i})\|_2$$

to approximately 10 m (Van Diggelen, 2009). We substitute $\rho_i(\hat{t}_{D,i}) = \rho_i(t_i - d_i - \hat{b})$ for $\rho_i(t_{D,i})$,

$$\varphi_i t_{\text{code}} = \frac{1}{c} \|\hat{\ell} - \rho_i(t_i - d_i - \hat{b})\|_2 - n_i t_{\text{code}} + \hat{b} + \epsilon_i^{(D)}. \tag{2.1}$$

The superscript (D) on the error term indicates that the error term now represents not only the arrival-time estimation error, but also the error induced by the inexact departure time.

We linearise around an a priori solution $\bar{\ell}$ and \bar{b} (usually $\bar{b} = 0$, otherwise we can simply shift the t_i s),

$$\begin{aligned} \varphi_i t_{\text{code}} &= \frac{1}{c} \|\bar{\ell} - \rho_i(t_i - d_i - \bar{b})\|_2 + \frac{1}{c} \mathbf{J}_{i,:} \begin{bmatrix} \hat{\ell} - \bar{\ell} \\ \hat{b} - \bar{b} \end{bmatrix} - n_i t_{\text{code}} + \hat{b} + \epsilon_i^{(D,L)} \\ &= \frac{1}{c} \|\bar{\ell} - \rho_i(t_i - d_i - \bar{b})\|_2 + \frac{1}{c} \mathbf{J}_{i,:} \begin{bmatrix} \hat{\ell} - \bar{\ell} \\ \hat{b} - \bar{b} \end{bmatrix} - n_i t_{\text{code}} + (\hat{b} - \bar{b}) + \bar{b} + \epsilon_i^{(D,L)}, \end{aligned} \quad (2.2)$$

where \mathbf{J} is the Jacobian of the Euclidean distances with respect to both the location of the receiver and to the bias, with the derivatives evaluated at $\bar{\ell}$ and at $t_i - d_i - \bar{b}$. The superscript (D, L) on the error term indicates that it now includes also the linearisation error.

There are now several ways to resolve the n_i .

3. Shadowing

Peterson et al. (1995) introduced a somewhat surprising modelling technique, which we refer to as *shadowing*. The idea is to replace the unknown b by two separate unknowns that represent essentially the same quantity, the original b and a shadow s . In principle, they should obey the equation $b = s$, but the model treats s as a free parameter; the constraint $b = s$ is dropped. In the literature, s is called the *coarse-time* parameter (and is often represented by t_c or t_c). We express this technique by splitting b and s :

$$\begin{aligned} \varphi_i t_{\text{code}} &= \frac{1}{c} \|\bar{\ell} - \rho_i(t_i - d_i - \bar{b})\|_2 + \frac{1}{c} \mathbf{J}_{i,:} \begin{bmatrix} \hat{\ell} - \bar{\ell} \\ \hat{b} - \bar{b} \end{bmatrix} - n_i t_{\text{code}} + \hat{b} + \epsilon_i^{(D,L)} \\ &= \frac{1}{c} \|\bar{\ell} - \rho_i(t_i - d_i - \bar{b})\|_2 + \frac{1}{c} \mathbf{J}_{i,:} \begin{bmatrix} \hat{\ell} - \bar{\ell} \\ s - \bar{b} \end{bmatrix} - n_i t_{\text{code}} + \hat{b} + \epsilon_i^{(D,L)}. \end{aligned}$$

We now have five unknowns, not four.

As far as we can tell, there is no clear explanation in the literature as to the benefits of shadowing. One way to justify the technique is to observe that Equation (2.2) is very sensitive to small (nanosecond scale) perturbations in the additive \hat{b} , but it is not highly sensitive to the \hat{b} (now s) by which we multiply,

$$\mathbf{J}_{i,4} = \frac{\partial}{\partial t_{D,i}} \|\bar{\ell} - \rho_i(t_i - d_i - \bar{b})\|_2. \quad (3.1)$$

For example, in GPS, the derivative is bounded by approximately 800 m/s for any $\bar{\ell}$ on Earth (Van Diggelen, 2009), so $\mathbf{J}_{i,4}/c < 3 \times 10^{-6}$ (versus 1 for the additive \hat{b}). Therefore, the dependence of the residual (the vector of $\epsilon_i^{(D,L)}$ terms for a given setting of the unknown parameters) on \hat{b} in Equation (2.2) is highly non-convex. There are many different values of \hat{b} that are almost equally good, a millisecond apart, with each of these nearly optimal hypotheses being locally well defined; if we increase \hat{b} by one millisecond and also add 1 to each n_i , the residual changes very little, because $\mathbf{J}_{i,4}$ is so small.

Shadowing turns this non-convexity into explicit rank deficiency, with which it is easier to deal. With one instance of \hat{b} replaced by the shadow s , the constraints no longer uniquely define \hat{b} , only up to a multiple of t_{code} . For any hypothetical solution ℓ, s, b, n , the solution $\ell, s, b + kt_{\text{code}}, n + k$ gives exactly the same residual. We perform a change of variables, replacing the partial sum $-n_i t_{\text{code}} + \hat{b}$ by

$-v_i t_{\text{code}} + \beta$, where $-v_i = -n_i + \lfloor \hat{b}/t_{\text{code}} \rfloor$ and $\beta = \hat{b} - \lfloor b/t_{\text{code}} \rfloor t_{\text{code}}$, so $\beta \in [0, t_{\text{code}})$:

$$\begin{aligned} \varphi_i t_{\text{code}} &= \frac{1}{c} \|\bar{\ell} - \rho_i(t_i - d_i - \bar{b})\|_2 + \frac{1}{c} J_{i,:} \begin{bmatrix} \hat{\ell} - \bar{\ell} \\ s - \bar{b} \end{bmatrix} - v_i t_{\text{code}} + \beta + \epsilon_i^{(D,L)} \\ \beta &\in [0, t_{\text{code}}). \end{aligned} \tag{3.2}$$

3.1. Resolving the integer ambiguities: Van Diggelen’s method

Van Diggelen’s method exploits the fact that the v_i values are very insensitive to $\hat{\ell}$ and to s . It therefore sets $\hat{\ell} = \bar{\ell}$ and $s = \bar{b}$, truncating the Jacobian term from Equation (3.2):

$$\begin{aligned} \varphi_i t_{\text{code}} &= \frac{1}{c} \|\bar{\ell} - \rho_i(t_i - d_i - \bar{b})\|_2 - v_i t_{\text{code}} + \beta + \epsilon_i^{(D,L,A)} \\ \beta &\in [0, t_{\text{code}}). \end{aligned} \tag{3.3}$$

The new subscript (D, L, A) indicates that the error term now compensates also for the use of the a priori estimates \bar{b} and $\bar{\ell}$ for s and $\hat{\ell}$.

Van Diggelen uses these constraints to set the v_i values in a particular way. The method selects one index j that is used to set v_j and β , and then resolves all the other v_i values so they are consistent with this β . That is, he assumes that $\epsilon_j^{(D,L,A)} = 0$ so

$$\begin{aligned} v_j &= \left\lfloor \frac{\frac{1}{c} \|\bar{\ell} - \rho_j(t_j - d_j - \bar{b})\| - \varphi_j t_{\text{code}}}{t_{\text{code}}} \right\rfloor \\ \beta &= (v_j + \varphi_j) t_{\text{code}} - \frac{1}{c} \|\bar{\ell} - \rho_j(t_j - d_j - \bar{b})\|_2. \end{aligned}$$

The method now substitutes this β in all the other constraints and assigns the other v_i values by setting $\epsilon_i^{(D,L,A)} = 0$ and rounding,

$$v_i = \left\lfloor \frac{\frac{1}{c} \|\bar{\ell} - \rho_i(t_i - d_i - \bar{b})\| - \varphi_i t_{\text{code}} + \beta}{t_{\text{code}}} \right\rfloor. \tag{3.4}$$

When $\|\hat{\ell} - \bar{\ell}\|$ and $|s - \bar{b}|$ are small enough, this gives a set of v_i values that are correct in the sense that they all differ from the correct v_i values by the same integer.

Van Diggelen chooses j in a particular way: he chooses the j that minimises the magnitude of Equation (3.1), which corresponds to the satellite closest to the zenith of $\bar{\ell}$ at $t_j - \bar{b}$. In our notation, Van Diggelen’s justification for this choice is as follows. He searches for a j for which Equation (3.3) approximates well Equation (3.2). The difference between the two is

$$\frac{1}{c} J_{j,:} \begin{bmatrix} \hat{\ell} - \bar{\ell} \\ s - \bar{b} \end{bmatrix}.$$

For each satellite, $J_{i,1:3}$ is the negation of the so-called line-of-sight vector e_i^T , which is the normalised direction from the satellite to the receiver; element $J_{i,4}$ is the range rate. Van Diggelen’s choice of j leads to a row of J in which the first three elements are almost orthogonal to $\hat{\ell} - \bar{\ell}$ and in which the fourth element, the range rate, is small. This leads to an estimated β that is relatively accurate, which helps resolve the correct v_i values.

Van Diggelen also shows that if we resolve the v_i values by setting each $\epsilon_i^{(D,L,A)} = 0$ separately, then the resolved β values might be close to 0 in one equation and close to t_{code} in another; this leads to inconsistent v_i values and to a huge position error.

3.2. Final resolution of the receiver’s location

Van Diggelen’s method resolves the integer v_i values in Equation (3.3). Now we need to resolve the continuous unknowns. We do so using Gauss–Newton iterations on Equation (3.2), iterating on $\delta_\ell = \hat{\ell} - \bar{\ell}$, $\delta_s = s - \bar{b}$ and β but keeping v fixed. We start with δ_ℓ , δ_s and β set to zero.

In every iteration, we use the current iterates to produce estimates of the location and bias,

$$\begin{aligned} \hat{\ell} &= \bar{\ell} + \delta_\ell \\ \hat{b} &= \bar{b} + \delta_s. \end{aligned}$$

We use them to improve the estimate of the ranges d_i , setting

$$\hat{d}_i = \|\hat{\ell} - \rho_i(\hat{t}_{D,i})\|_2.$$

This allows us to reduce the errors in Equation (2.1),

$$\begin{aligned} \varphi_i t_{\text{code}} &= \frac{1}{c} \|\hat{\ell} - \rho_i(t_i - \hat{d}_i - \hat{b})\|_2 - n_i t_{\text{code}} + \hat{b} + \epsilon_i^{(\hat{D})} \\ &= \frac{1}{c} \|\hat{\ell} - \rho_i(t_i - \hat{d}_i - \hat{b})\|_2 - v_i t_{\text{code}} + \beta + \epsilon_i^{(\hat{D})} \end{aligned}$$

(the second line holds because $n_i t_{\text{code}} + \hat{b} = v_i t_{\text{code}} + \beta$, by definition). We again linearise this and solve the constraints

$$\varphi_i t_{\text{code}} = \frac{1}{c} \|\hat{\ell} - \rho_i(t_i - \hat{d}_i - \hat{b})\|_2 + \frac{1}{c} \mathbf{J}_{i,:} \begin{bmatrix} \hat{\ell} - \hat{\ell} \\ s - \hat{b} \end{bmatrix} - v_i t_{\text{code}} + \beta + \epsilon_i^{(\hat{D},L)}. \tag{3.5}$$

for $\hat{\ell}$, s and β using in the generalised least-squares sense, where the Jacobian is evaluated at $\hat{\ell}$ and $t - \hat{d} - \hat{b}$.

We can now explain why Van Diggelen’s method resolves the integers only once and iterates only on the continuous unknowns. The v_i values that Van Diggelen’s method resolves are not equal to the n_i values in the nonlinear Equation (2.1). But when the linearisation error is small enough, the two integer vectors differ by a constant, $\lfloor \hat{b}/t_{\text{code}} \rfloor$. This difference is compensated for by the integer part of the continuous variable β , which is not constrained to $[0, t_{\text{code}})$ in the Gauss–Newton iterations. This is the actual function of shadowing; to allow β to compensate not only for the clock error, but also for the constant error in v . When the initial linearisation error is so large that $n - v$ is no longer a constant, the method breaks down.

4. A mixed-integer least-squares approach

A different approach, which has never been proposed for snapshot positioning, is to add regularisation constraints that will allow us to resolve all the $4 + m$ unknowns in the m instances of Equation (2.2)

$$\varphi_i t_{\text{code}} = \frac{1}{c} \|\bar{\ell} - \rho_i(t_i - d_i - \bar{b})\|_2 + \frac{1}{c} \mathbf{J}_{i,:} \begin{bmatrix} \hat{\ell} - \bar{\ell} \\ \hat{b} - \bar{b} \end{bmatrix} - (\hat{n}_i - \bar{n}_i) t_{\text{code}} - \bar{n}_i t_{\text{code}} + (\hat{b} - \bar{b}) + \bar{b} + \epsilon_i^{(D,L)}$$

using mixed-integer least-squares techniques. Note that we have rewritten Equation (2.2) in a way that emphasises a change of variables that facilitate iterative improvements: the new unknowns are $\hat{\ell} - \bar{\ell}$, $\hat{b} - \bar{b}$ and $\hat{n} - \bar{n}$. We initially set \bar{b} and \bar{n} to zero.

We denote the vector of delays by g ,

$$g_i = \frac{1}{c} \|\bar{\ell} - \rho_i(t_i - d_i - \bar{b})\|_2.$$

This section proposes two sets of regularising equations and explains how to use this approach in an iterative Gauss–Newton solver.

4.1. Resolving the ambiguities: regularisation using a priori estimates

The first set of regularising equations that we propose are

$$\frac{1}{c} \mathbf{J}_{i,:} \begin{bmatrix} \hat{\ell} - \bar{\ell} \\ \hat{b} - \bar{b} \end{bmatrix} = 0.$$

We do not enforce them exactly, only in a (weak) least-squares sense. They favour solutions of the mixed-integer least-squares problem that are in the vicinity of the a priori solution. This leads to the following weighted mixed-integer least-squares problem:

$$\hat{\ell}, \hat{b}, \hat{n} = \arg \min_{\ell, b, n} \left\| W \begin{bmatrix} \frac{1}{c} \mathbf{J} + [\mathbf{0}_{m \times 3} & \mathbf{1}_{m \times 1}] & -t_{\text{code}} \mathbf{I}_{m \times m} \\ & \frac{1}{c} \mathbf{J} & \mathbf{0}_{m \times m} \end{bmatrix} \begin{bmatrix} \hat{\ell} - \bar{\ell} \\ \hat{b} - \bar{b} \\ \hat{n} - \bar{n} \end{bmatrix} - \begin{bmatrix} t_{\text{code}} \varphi - g + \bar{n} t_{\text{code}} - \bar{b} \\ \mathbf{0} \end{bmatrix} \right\|_2^2,$$

where W is a block-diagonal weight matrix derived from the covariance matrix C of the error terms ϵ , $W^T W = C^{-1}$. Now we have $2m$ constraints, which for $m \geq 4$ should allow us to resolve the integer n_i values.

We propose to choose a diagonal W as follows. We set the first m diagonal elements of W to the standard deviation of the arrival-time estimator, say $W_{i,i} = 1/\sigma(t_i) \approx 1/(10 \text{ ns})$. To set the rest, we use box constraints on the a priori estimates $\bar{\ell}$ and \bar{b} , denoted as

$$\begin{aligned} |x - \bar{x}| &\leq x_{\text{max}} \\ |y - \bar{y}| &\leq y_{\text{max}} \\ |z - \bar{z}| &\leq z_{\text{max}} \\ |\hat{b} - \bar{b}| &\leq b_{\text{max}}. \end{aligned}$$

By the triangle inequality

$$\left| \mathbf{J}_{i,:} \begin{bmatrix} \hat{\ell} - \bar{\ell} \\ \hat{b} - \bar{b} \end{bmatrix} \right| \leq |\mathbf{J}_{i,1}|x_{\text{max}} + |\mathbf{J}_{i,2}|y_{\text{max}} + |\mathbf{J}_{i,3}|z_{\text{max}} + |\mathbf{J}_{i,4}|b_{\text{max}}.$$

We define

$$r_i = |\mathbf{J}_{i,1}|x_{\text{max}} + |\mathbf{J}_{i,2}|y_{\text{max}} + |\mathbf{J}_{i,3}|z_{\text{max}} + |\mathbf{J}_{i,4}|b_{\text{max}},$$

so

$$\left| \mathbf{J}_{i,:} \begin{bmatrix} \hat{\ell} - \bar{\ell} \\ \hat{b} - \bar{b} \end{bmatrix} \right| \leq r_i.$$

We convert the hard box constraints into soft-weighted least-squares in order to allow for using a mixed-integer least-squares solver. We need to set $W_{m+i,m+i}$; if we assume that the error in the constraint is Gaussian and that an error of r_i/c is acceptable (from the inequality above), then setting $W_{m+i,m+i} = c/r_i$, say, makes sense. In practice, we use $W_{m+i,m+i} = c/(100 \text{ km})$ in the experiments below.

This mixed-integer least-squares minimisation problem can be solved by a generic solver, such as one of the solvers that have been developed for RTK.

4.2. Doppler regularisation

It turns out that Doppler shifts allow us to regularise Equation (2.2) in a more effective way. GNSS receivers estimate not only the time of arrival of the signal, but also its Doppler shift. The estimated Doppler shift is biased, because of the inaccuracy of the receiver’s local (or master) oscillator; it is also inexact. We now show a novel technique to use the Doppler-shift observations to regularise Equation (2.2).

Our technique is based on two assumptions. One is that the receiver is stationary, or more precisely, that its velocity is negligible relative to the range rate, which is up to approximately 800 m/s. This assumption can be easily removed, but its removal leads to additional unknowns and more complicated expressions that we do not present here. The other assumption is that the local oscillator and the sampling clock in the receiver are derived from a single master oscillator in a certain (very common) way. Again, this assumption can be removed if another unknown is added.

The Doppler-shift formula for velocities much lower than the speed of light is

$$\dot{D}_i \approx -\frac{1}{c} \frac{d}{dt} \|\dot{\ell} - \rho_i\|_2 f_0.$$

The Doppler observations that the receiver makes are

$$D_i = -\frac{1}{c} \frac{d}{dt} \|\dot{\ell} - \rho_i\|_2 f_0 + \dot{f} + \epsilon_i^{(\delta)}, \tag{4.1}$$

where \dot{f} is the frequency offset (bias) of the receiver and $\epsilon_i^{(\delta)}$ is an error term that represents the observation error and the (negligible) slow-speed approximation. Therefore, the quantities $-cD_i/f_0$ are biased estimates of the range rate. We denote the a priori estimates of the Doppler shifts by \bar{D}_i .

We differentiate Equation (2.2) by time,

$$\frac{d}{dt}(\varphi_i t_{\text{code}}) = \frac{d}{dt} \left(\frac{1}{c} \|\dot{\ell} - \rho_i(t_i - d_i - \bar{b})\|_2 + \frac{1}{c} \mathbf{J}_{i,:} \begin{bmatrix} \dot{\ell} - \bar{\ell} \\ \dot{b} - \bar{b} \end{bmatrix} - n_i t_{\text{code}} + (\dot{b} - \bar{b}) + \bar{b} \right) + \epsilon_i^{(D,L,\theta)}.$$

We first manipulate the equation a little, to make it easier to differentiate:

$$\begin{aligned} & \frac{d}{dt} (c\varphi_i t_{\text{code}} - \|\dot{\ell} - \rho_i(t_i - d_i - \bar{b})\|_2 - c\bar{b}) \\ &= \frac{d}{dt} \left(\mathbf{J}_{i,:} \begin{bmatrix} \dot{\ell} - \bar{\ell} \\ \dot{b} - \bar{b} \end{bmatrix} - cn_i t_{\text{code}} + c(\dot{b} - \bar{b}) \right) + \epsilon_i^{(D,L,\theta)}. \end{aligned} \tag{4.2}$$

We denote

$$H = \begin{bmatrix} \mathbf{J}_{1,1} & \mathbf{J}_{1,2} & \mathbf{J}_{1,3} & \mathbf{J}_{1,4} + c & -ct_{\text{code}} & & & & \\ \vdots & \vdots & \vdots & \vdots & & & & \ddots & \\ \mathbf{J}_{i,1} & \mathbf{J}_{i,2} & \mathbf{J}_{i,3} & \mathbf{J}_{i,4} + c & & & & & -ct_{\text{code}} \\ \vdots & \vdots & \vdots & \vdots & & & & & \ddots \\ \mathbf{J}_{m,1} & \mathbf{J}_{m,2} & \mathbf{J}_{m,3} & \mathbf{J}_{m,4} + c & & & & & -ct_{\text{code}} \end{bmatrix}.$$

The first three columns of H are identical to those of J , the next is the fourth column of J but shifted by c and the last m columns consist of a scaled identity matrix. We now express the derivative on the

right-hand side of Equation (4.2) as

$$\frac{d}{dt} \left(H \begin{bmatrix} \dot{\ell} - \bar{\ell} \\ \dot{b} - \bar{b} \\ n_1 \\ \dots \\ n_m \end{bmatrix} \right) = H \left(\frac{d}{dt} \begin{bmatrix} \dot{\ell} - \bar{\ell} \\ \dot{b} - \bar{b} \\ n_1 \\ \dots \\ n_m \end{bmatrix} \right) + \left(\frac{d}{dt} H \right) \begin{bmatrix} \dot{\ell} - \bar{\ell} \\ \dot{b} - \bar{b} \\ n_1 \\ \dots \\ n_m \end{bmatrix}.$$

We assume that the receiver is stationary, so $\dot{\ell} - \bar{\ell}$ is time-independent, so $(d/dt)(\dot{\ell} - \bar{\ell}) = 0$. The derivatives of the integers n_1, \dots, n_m are also zero. The derivative of the remaining element in the vector, $(d/dt)(\dot{b} - \bar{b})$, is not zero and will need to be estimated. It represents the frequency offset of the receiver, which biases the observed Doppler shift. It is multiplied by a column whose elements are very close to c (the range rate is tiny relative to the speed of light), which allows it to compensate for the frequency bias.

To differentiate H , we exploit the known structure of J . For each satellite, $J_{i,1:3}$ is the negation of the so-called line-of-sight vector e_i^T , which is the normalised direction from the satellite to the receiver; element $J_{i,4}$ is the range rate. The derivatives of these quantities are shown by Van Diggelen (2009, Equation 8.6), Fernández-Hernández and Borre (2016) and other sources:

$$\frac{d}{dt} J_{i,1:3} = \frac{e^T \frac{d}{dt} \|\bar{\ell} - \rho_i\| + \left(\frac{d}{dt} (\bar{\ell} - \rho_i) \right)^T}{\|\bar{\ell} - \rho_i\|},$$

where the satellite position ρ_i and its velocity $(d/dt)\rho_i$ are taken at \hat{t}_D . To reduce the number of unknowns, we assume that the receiver is stationary, so $(d/dt)\bar{\ell} = 0$, so

$$\frac{d}{dt} J_{i,1:3} = \frac{e^T \frac{d}{dt} \|\bar{\ell} - \rho_i\| - \left(\frac{d}{dt} \rho_i \right)^T}{\|\bar{\ell} - \rho_i\|}.$$

Element $J_{i,4}$ is the range rate of satellite i , so its derivative with respect to time is the satellite’s range acceleration,

$$\frac{d}{dt} J_{i,4} = \frac{d^2}{dt^2} \|\bar{\ell} - \rho_i\|.$$

We use finite differences to evaluate this second derivative. The fourth column of H is $J_{:,4} + c$, but the derivative of c is obviously zero. The derivative of $-ct_{\text{code}}$ is also zero, so

$$\left(\frac{d}{dt} H \right) \begin{bmatrix} \dot{\ell} - \bar{\ell} \\ \dot{b} - \bar{b} \\ n_1 \\ \dots \\ n_m \end{bmatrix} = \left(\frac{d}{dt} H_{:,1:4} \right) \begin{bmatrix} \dot{\ell} - \bar{\ell} \\ \dot{b} - \bar{b} \end{bmatrix}.$$

We now derive the left-hand side of Equation (4.2),

$$\frac{d}{dt} (c\varphi_i t_{\text{code}} - \|\bar{\ell} - \rho_i(t_i - d_i - \bar{b})\|_2 - c\bar{b}).$$

The derivative of the a priori range estimate $\|\bar{\ell} - \rho_i(t_i - d_i - \bar{b})\|_2$ is the a priori range rate, which we can compute. The derivative of $c\bar{b}$ is zero.

To understand the first term, recall that

$$(n_i + \varphi_i)t_{\text{code}} = \frac{1}{c} \|\dot{\ell} - \rho_i(t_{D,i})\|_2 + \dot{b} + \epsilon_i,$$

so

$$c \frac{d}{dt} \varphi_i t_{\text{code}} = \frac{d}{dt} \|\hat{\ell} - \rho_i(t_{D,i})\|_2 + c \frac{d}{dt} \hat{b} + c \frac{d}{dt} \epsilon_i.$$

We now rewrite Equation (4.1) as

$$\frac{1}{c} \frac{d}{dt} \|\hat{\ell} - \rho_i\|_2 f_0 = -D_i + \hat{f} + \epsilon_i^{(\delta)}$$

or

$$\frac{d}{dt} \|\hat{\ell} - \rho_i\|_2 = -\frac{cD_i}{f_0} + \frac{c\hat{f}}{f_0} + \frac{c}{f_0} \epsilon_i^{(\delta)}.$$

We now substitute in the left-hand side of Equation (4.2):

$$\frac{d}{dt} c \varphi_i t_{\text{code}} = -\frac{cD_i}{f_0} + \frac{c\hat{f}}{f_0} + \frac{c}{f_0} \epsilon_i^{(\delta)} + c \frac{d}{dt} \hat{b} + c \frac{d}{dt} \epsilon_i.$$

The term \hat{f}/f_0 is the relative local-oscillator error in the receiver. If the oscillator runs too fast, \hat{f} is negative. Assuming that all the clocks in the receiver are derived from a master oscillator, if it runs too fast, \hat{b} grows over time. Under this assumption,

$$-\frac{c\hat{f}}{f_0} = c \frac{d}{dt} \hat{b},$$

so these terms cancel each other. If our assumption on the receiver does not hold, we would need to estimate

$$\frac{c\hat{f}}{f_0} + c \frac{d}{dt} \hat{b}.$$

We have arrived at a system of m linear equations that we use to regularise the mixed-integer equations. The equations are:

$$-\frac{c}{f_0} D - \frac{d}{dt} \|\hat{\ell} - \rho\|_2 = H_{:,4}(\hat{u} - \bar{u}) + \left(\frac{d}{dt} H_{:,1:4}\right) \begin{bmatrix} \hat{\ell} - \bar{\ell} \\ \hat{b} - \bar{b} \end{bmatrix}. \tag{4.3}$$

In this equation, D represents the vector of observed Doppler shifts, $(d/dt)\|\hat{\ell} - \rho\|_2$ is the vector of the a priori range rates and $\hat{u} - \bar{u} = (d/dt)(\hat{b} - \bar{b})$ is a new scalar unknown. We have explained above how to compute $H_{:,4}$ and $(d/dt)H_{:,1:4}$. The full regularised weighted least-squares that we solve is

$$\hat{\ell}, \hat{b}, \hat{n}, \hat{u} = \arg \min_{\ell, b, n, u} \left\| W \begin{bmatrix} \frac{1}{c} \mathbf{J} + [\mathbf{0}_{m \times 3} \quad \mathbf{1}_{m \times 1}] & -t_{\text{code}} \mathbf{I}_{m \times m} & \mathbf{0}_{m \times 1} \\ \left(\frac{d}{dt} H_{:,1:4}\right) & \mathbf{0}_{m \times m} & H_{:,4} \end{bmatrix} \begin{bmatrix} \hat{\ell} - \bar{\ell} \\ \hat{b} - \bar{b} \\ \hat{n} - \bar{n} \\ \hat{u} - \bar{u} \end{bmatrix} - \begin{bmatrix} t_{\text{code}} \varphi - g + \bar{n} t_{\text{code}} - \bar{b} \\ -\frac{c}{f_0} D - \frac{d}{dt} \|\hat{\ell} - \rho\|_2 \end{bmatrix} \right\|_2^2.$$

4.3. Iterating to cope with large a priori errors

Solving the linearised and regularised mixed-integer least-squares problem improves the initial a priori estimates of $\hat{\ell}$ and \hat{b} , but not to the extent possible given the code phases. The most important factor that limits the accuracy of the corrections is the fact that when the a priori estimates are large, the

resolved integers, the n_i values, are inexact. Therefore, we incorporate the mixed-integer solver into a Gauss–Newton-like iteration in which we correct all the unknowns, including the integer ambiguities, more than once.

More specifically, once we solve the mixed-integer least-squares problem for $\hat{n} - \bar{n}$, $\hat{\ell} - \bar{\ell}$ and $\hat{b} - \bar{b}$ (and for $\hat{u} - \bar{u}$ in the Doppler formulation), we use the corrections to improve the estimates of the receiver's location and of the departure times, and we linearise Equation (2.1) again. We now solve the newly linearised least-squares problem again for additional corrections, and so on.

5. Implementation and evaluation

We have implemented all the methods that we described above in MATLAB.

We use Borre's Easy Suite (Borre, 2003, 2009) to perform many routine calculations. In particular, we use it to correct the GPS time (check_t), to correct for Earth rotation during signal propagation time (e_r_corr), to read an ephemeris from a RINEX file and to extract the data for a particular satellite (rinexe, get_eph and find_eph), to transform Julian dates to GPS time (gps_time), to represent Julian dates as one number (julday), to compute the coordinate of a satellite at a given time in ECEF coordinates (satpos), to compute the azimuth, elevation and distance to a satellite (topocent, which calls togeod to transform ECEF to WG84 coordinates), and to approximate the tropospheric delay (tropo). We also use a MATLAB function by Eric Ogier (ionosphericDelay.m, available on the MathWorks File Exchange) to approximate the ionospheric delay using the Klobuchar model. We take the parameters for the Klobuchar model from files published by the GNSS Research Center at Curtin University.²

During the Gauss–Newton phase of the algorithm (after the integers have been determined), if we have only four observations, we add a pseudo-measurement constraint that constrains the correction to maintain the height of the target, in the least-squares sense (Van Diggelen, 2009).

We use Chang and Zhou's MILES package (Chang and Zhou, 2007) to solve mixed-integer least-squares problems.

We take ephemeris data from RINEX navigation files published by NASA.³

We filter satellites that are lower than 10 degrees above the horizon which have the lowest SNR and are more likely to suffer from multipath interference.

We evaluated the code on data from several sources:

- Publicly available observation data files in a standard format (RINEX) distributed by NASA. We used these to test our algorithms in the initial phases of the research. These results are not shown here.
- GPS simulations. We generated satellite positions ephemeris files and used them to compute times of arrival and code phases. These simulations do not include ionospheric or tropospheric delays, so they help us separate the issues arising from these delays from other algorithmic issues.
- Code-phase and Doppler-shift measurements collected by us using a u-blox ZED-F9P GNSS receiver, connected to an ANN-MB-00 u-blox antenna mounted on a steel plate on top of a roof with excellent sky view. We established the precise coordinates of the antenna (to compute errors) using differential carrier-phase corrections from a commercial virtual reference station (VRS).⁴ The WGS84 coordinates of the antenna are 32.1121756, 34.8055775 with height above sea level of 61.15 m. The code phase measurements are included in the UBX-RXM-MEASX emitted by the receiver. The data set includes approximately 700 epochs, one every minute (so they span a little more than 11 h). The number of satellites per epoch ranges from 8 to 13 and after filtering by elevation, between 7 and 11.
- Recordings of RF samples made by a bat-tracking GPS snapshot logger. The tag model is called Vesper. It was designed and produced by Alex Schwartz Developments on the basis of an earlier tag called Robin, designed and produced by a company called CellGuide that no longer exists. The tag records 1-bit RF samples at a rate of 1,023,000 samples per second. (The sampling rate is a multiple

²<http://saegnss2.curtin.edu/ldc/rinex/daily/>

³<https://cdsis.nasa.gov/archive/gnss/data/daily/>

⁴<https://axis-gps.com>

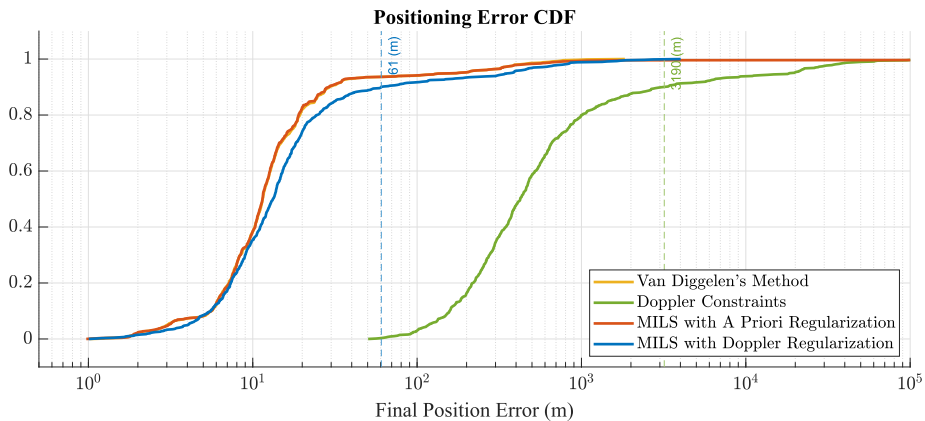


Figure 1. Cumulative distribution function of the absolute positioning errors of four algorithms: Van Diggelen's non-iterative method, the Doppler constraints alone (the first phase of Fernández-Hernández and Borre's method), and mixed-integer least-squares (MILS) with either a priori or Doppler regularization.

of $1/t_{\text{code}}$; this is known to make time-of-arrival estimate difficult (Tran et al., 2018) but the rate cannot be changed in this logger.) The tag was configured to record a 256 ms sample every 10 min for a few hours. It was placed next to the ANN-MB-00 antenna.

Figure 1 shows the cumulative distribution function (CDF) of four algorithms: Van Diggelen's non-iterative method, the Doppler constraints alone (as used in the first phase of Fernández-Hernández and Borre's method), and MILS with either a priori or Doppler regularization. The data from the u-blox receiver were used to produce these graphs. We used all 694 epochs. The initial error was of 20–21 s (uniform distribution) and 20 km in a random uniform horizontal direction. In the MILS algorithms, the final position was computed with the regularization constraints; this is why the Doppler regularization produced less accurate results. We can see that the accuracy of MILS with a priori regularization and of Van Diggelen's method is essentially identical.

Figure 2 compares the probability of success achieved by our regularised MILS solver with that achieved by Van Diggelen's non-iterative method and by the Doppler constraints alone. We considered fixes that are within 1 km of the true location to be a success in obtaining the correct integer values. Each pixel in these heat maps represents 16 different runs. Each run uses a random epoch, a random initial location estimate and a random initial time estimate. The initial location estimates have a given distance to the true location (the x axis of the heat map) but a random azimuth. The initial time estimate is a slight perturbation (uniform between zero and one second) of the given time error, which is the y axis of the heat map. In each pixel, half of the initial time errors are positive and half are negative. We used all the satellites in view in each epoch.

The results clearly show that the MILS algorithm, even with the simple a priori time and location regularization from Section 4.1, outperforms Van Diggelen's non-iterative method. Van Diggelen's method obtains a correct fix in almost all cases (success probability close to 1) when the initial location error is small and the initial time error is 150 s or less, when the initial location time error is small and the initial location error is 100 km or less, and in other equivalent combinations of time and location errors. The corresponding limits for the MILS algorithm with a priori regularization are approximately 150 s and 250 km.

Doppler-shift observations expand dramatically the region of convergence in both approaches. The MILS algorithm with Doppler regularization obtains a correct fix as long as the initial time error is at up to approximately 180 s (3 min); this works even with great-circle distances of 20,000 km, which means that the initial position can essentially be anywhere on Earth. If the initial position error is small, the method can tolerate initial time errors of up to approximately 80 min (5000 s). The heat map of the

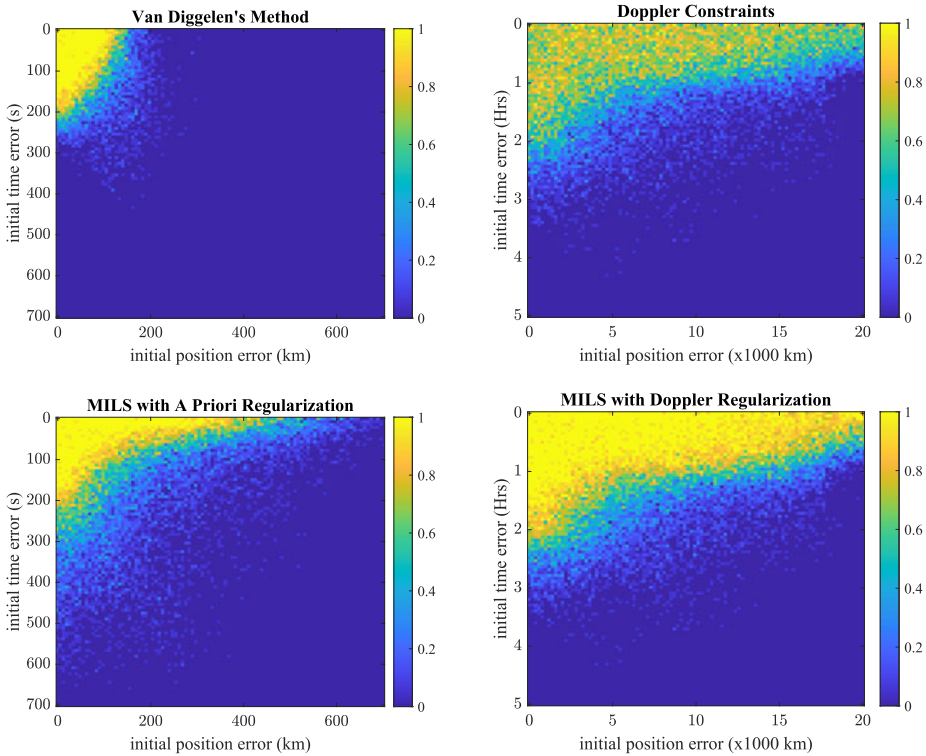


Figure 2. The probability of obtaining a fix with an error smaller than 1 km from the u-blox data set using four different algorithms.

Doppler constraints alone, together with the CDF in Figure 1, indicate that these constraints produce an estimate good enough for initialising Van Diggelen's method, but are not accurate enough on their own. Indeed, Van Diggelen writes about the Doppler constraints alone: 'For less than 1 Hz of measurement error, we expect a position error of the order of 1 km' (Van Diggelen, 2009, Section 8.3); this explains why the probabilities in the top-right plot in Figure 2 are usually far from 1, even with small initial errors. In general, both approaches have similar regions of convergence and they produce similarly accurate fixes.

Figure 3 explores how the number of satellites (observations) affects the success rates of the four methods. We repeated the experiment whose results are shown in the heat maps in Figure 2, but only on the 20 epochs in which 13 satellites were in view. We selected random subsets of the satellites in view and random initial errors, within the bounds shown in Figure 2, and computed the fraction of successful experiments. We can see that when only code phases are used, Van Diggelen's method is better when using 6–8 observations, probably because the weighting of the observations in the MILS method sometimes leads to incorrect integers when Van Diggelen's method resolves the integers correctly. However, with 5 satellites in view or more than 8, the MILS method is better. MILS with Doppler regularisation is superior to all the other methods.

While our Matlab implementation is not designed to carefully evaluate running times and computational efficiency, we did measure the running times and we can draw from them some useful conclusions. The running times of a single Gauss–Newton correction step in Van Diggelen's method and in the solution of the Doppler equations is 8–10 μ s, while the running time of a single Gauss–Newton correction step in the MILS formulation is approximately 2.5 ms when using a priori regularisation and 0.6 ms when using Doppler regularisation. While the MILS methods are clearly more expensive, they also appear to be fast enough for real-time applications.

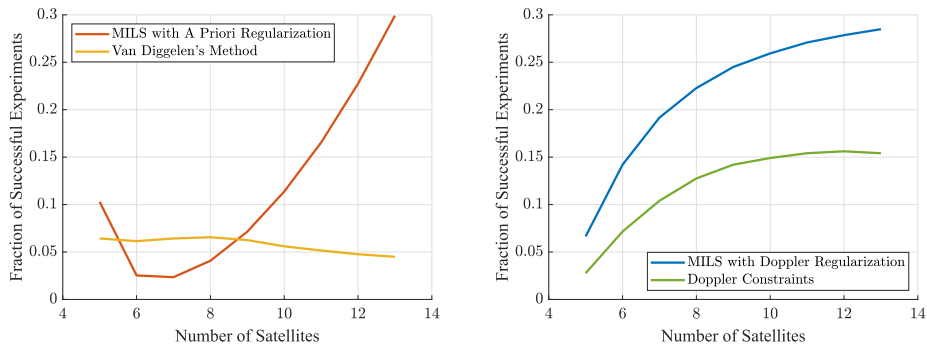


Figure 3. The fraction of successful positioning (error of at most 1 km) in the spaces of initial errors shown in Figure 2 as a function of the number of satellites (observations) used.

6. Conclusions and discussion

We have shown that Van Diggelen's ingenious *coarse-time navigation* algorithm (Van Diggelen, 2002, 2009) that estimates a location from GNSS observations without departure times is essentially a specialised solver for a mixed-integer least-squares problem. Even though Van Diggelen's algorithms have been cited and used by many authors, the actual form of the mixed-integer optimisation problem has never been presented; we present it in this paper for the first time.

We also show that the integer ambiguities can be resolved by regularising the mixed-integer least-squares problem. We proposed two regularisation techniques, one that biases solutions towards an initial a priori estimate. This extends Van Diggelen's use of the a priori estimate to resolve the integers, but our regularisation approach can resolve the integers with larger initial errors than those of Van Diggelen. In effect, the general mixed-integer formulation uses the available information more effectively than Van Diggelen's specialised solver.

We also proposed a regularisation method based on Doppler-shift observations. This method allows our solver to resolve the correct integers even with huge initial time or position errors. Doppler shifts have been used in snapshot positioning before, but they were always used to produce an initial position and time estimate that is subsequently used as an a priori estimate in Van Diggelen's algorithm. This approach, due to Fernández-Hernández and Borre (2016), is also extremely effective.

Our algorithm iterates over the entire mixed-integer least-squares problem more than once. If one resolves the integers once, in the first iteration, and continues to iterate only on the continuous unknowns, using the resolved integers, the method converges, but to fixes with larger errors.

In effect, by cleanly formulating the mixed-integer optimisation problem that underlies snapshot positioning, we have enabled the exploration of a wide range of solvers, including the two regularised solvers that we presented here. We believe that additional solvers can be discovered for this formulation. In contrast, all prior research treated Van Diggelen's algorithm as a clever black box, limiting the range of algorithms that can be developed.

Our new methods are inspired by the RTK method, which resolves a position from both code-phase and carrier-phase GNSS observations (Teunissen, 2017). In RTK, the position is eventually resolved by carrier-phase constraints, which have integer ambiguities; these constraints are regularised by pseudo-range constraints, which are less precise but have no integer ambiguities. Here the position is resolved by integer-ambiguous code-phase constraints, which are regularised by either a priori estimates or by Doppler-shift observations. RTK also requires so-called differential constraints at a fixed receiver, because the carrier phase of the satellites are not locked to each other. Here we do not require differential corrections because the code departure times are locked to whole milliseconds in all the satellites.

Acknowledgments. This study was supported by grant 1919/19 from the Israel Science Foundation. Thanks to Aya Goldshtein for assisting with collecting data from the bat-tracking GPS snapshot logger. Thanks to Amir Beck for helpful discussions. Thanks to the three reviewers for comments and suggestions that helped us improve the paper.

References

- Badia-Solé O. and Iacobescu Ioan T.** (2010). *GPS snapshot techniques*. Technical report, Danish GPS Center, Aalborg University. <http://kom.aau.dk/group/10gr815/Report.pdf>.
- Bavaro M.** (2012a). *Fruit-bat tracking with Robin GPS logger*. Michele's GNSS blog. <http://michelebavaro.blogspot.com/2012/08/the-gps-tracker-to-beat-smallest.html>. Accessed 18 May 2020.
- Bavaro M.** (2012b). *The GPS tracker to beat (smallest, lightest, and most durable)*. Michele's GNSS blog. michelebavaro.blogspot.com/2012/08/the-gps-tracker-to-beat-smallest.html. Accessed 18 May 2020.
- Bissig P., Eichelberger M. and Wattenhofer R.** (2017). Fast and Robust GPS Fix using One Millisecond of Data. In: Dutta, P. and Xing, G. (eds.). *2017 16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. Pittsburgh, PA: IEEE, 223–234.
- Borre K.** (2003). The GPS easy suite: MATLAB code for the GPS newcomer. *GPS Solutions*, **7**, 47–51. doi:10.1007/s10291-003-0049-3
- Borre K.** (2009). GPS Easy Suite II: A MATLAB Companion. *Inside GNSS*, 48–52. Available at: <https://www.insidegnss.com/pdf/EasySuite.pdf>.
- Borre K. and Strang G.** (2012). *Algorithms for Global Positioning*. Wellesley, MA: Wellesley-Cambridge.
- Chang X.-W. and Zhou T.** (2007). MILES: MATLAB package for solving mixed integer least squares problems. *GPS Solutions*, **11**, 289–294. doi:10.1007/s10291-007-0063-y
- Chen H. W., Wang H. S., Chiang Y. T. and Chang F. R.** (2014). A new coarse-time GPS positioning algorithm using combined Doppler and code-phase measurements. *GPS Solutions*, **18**, 541–551.
- Cvikel N., Berg K. E., Levin E., Hurme E., Borissov I., Boonman A., Amichai E. and Yovel Y.** (2015). Bats aggregate to improve prey search but might be impaired when their density becomes too high. *Current Biology*, **25**, 206–211.
- Dujon A. M., Lindstrom R. T. and Hays G. C.** (2014). The accuracy of Fastloc-GPS locations and implications for animal tracking. *Methods in Ecology and Evolution*, **5**(11), 1162–1169.
- Eichelberger M., von Hagen F. and Wattenhofer R.** (2019). Multi-year GPS Tracking using a Coin Cell. In: Zhong, L. (ed.). *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications (HotMobile)*. Santa Cruz, CA: ACM, 141–146. doi: 10.1145/3301293.3302367.
- Fernández-Hernández I. and Borre K.** (2016). Snapshot positioning without initial information. *GPS Solutions*, **20**, 605–616. doi:10.1007/s10291-016-0530-4
- Harten L., Katz A., Goldshtein A., Handel M. and Yovel Y.** (2020). The ontogeny of a mammalian cognitive map in the real world. *Science*, **369**(6500), 194–197.
- Lannelongue S. and Pablos P.** (1998). Fast Acquisition Techniques for GPS Receivers. In *Proceedings of the 54th Annual Meeting of The Institute of Navigation*. Denver, CO: The Institute of Navigation, 261–269.
- Liu J., Priyantha B., Hart T., Ramos H. S., Loureiro A. A. F. and Wang Q.** (2012). Energy Efficient GPS Sensing with Cloud Offloading. In: Campbell, A. and Langendoen, K. (eds.). *Proceedings of the 10th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. Toronto, Ontario, Canada: ACM, 85–98.
- Muthuraman K., Brown J. and Chansarkar M.** (2012). Coarse Time Navigation: Equivalence of Algorithms and Reliability of Time Estimates. In *Proceedings of International Technical Meeting of The Institute of Navigation*, 1115–1138. <https://www.ion.org/publications/abstract.cfm?articleID=10010>.
- Othieno N. and Gleason S.** (2012). Combined Doppler Time-free Positioning for Low Dynamics Receivers. In *Proceedings of the IEEE/ION Position, Location and Navigation Symposium*. Myrtle Beach, SC: IEEE, 6–65.
- Peterson B., Hartnett R. and Ottman G.** (1995). GPS Receiver Structures for the Urban Canyon. In *Proceedings of the 8th International Technical Meeting of the Satellite Division (ION GPS)*. Palm Springs, CA: The Institute of Navigation, 1323–1332.
- Ramos H. S., Zhang T., Liu J., Priyantha N. B. and Kansal A.** (2011). LEAP: A Low Energy Assisted GPS for Trajectory-based Services. In: Patterson, D. J., Rogers, Y. and Xie, X. (eds.). *Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp)*. Beijing, China: ACM, 335–344. doi: 10.1145/2030112.2030158.
- Teunissen P. J. G.** (2017). Carrier Phase Integer Ambiguity Resolution. In: Teunissen, P. J. G. and Montenbruck, O. (eds.). *Springer Handbook of Global Navigation Satellite Systems*, Berlin Heidelberg, Germany: Springer, 661–720.
- Tomkiewicz S. M., Fuller M. R., Kie J. G. and Bates K. K.** (2010). Global positioning system and associated technologies in animal behaviour and ecological research. *Philosophical Transactions of the Royal Society B*, **365**, 2163–2176.
- Tran V. T., Shivaramaiah N. C., Nguyen T. D., Glennon E. P. and Dempster A. G.** (2018). GNSS receiver implementations to mitigate the effects of commensurate sampling frequencies on dll code tracking. *GPS Solutions*, **22**(24), 1–11.
- Van Diggelen F.** (2002). Method and apparatus for time free processing of GPS signals. US Patent 6,417,801.
- Van Diggelen F.** (2009). *A-GPS: Assisted GPS, GNSS, and SBAS*. Boston, MA and London, UK: Artech House.
- Wang M., Qin H. and Jin T.** (2019). Massive terminal positioning system with snapshot positioning technique. *GPS Solutions*, **23**(31), 1–14. doi: 10.1007/s10291-018-0821-z.
- Weiss A. J.** (2004). Direct position determination of narrowband radio frequency transmitters. *IEEE Signal Processing Letters*, **11**, 513–516.

- Witt M., Åkesson S., Broderick A., Coyne M., Ellick J., Formia A., Hays G., Luschi P., Stroud S. and Godley B.** (2010). Assessing accuracy and utility of satellite-tracking data using Argos-linked Fastloc-GPS. *Animal Behaviour*, **80**(3), 571–581.
- Yoo W. J., Kim L., Lee Y. D. and Lee H. K.** (2020). A coarse-time positioning method for improved availability. *GPS Solutions*, **24**(2), 14. doi:10.1007/s10291-019-0919-y.