Integer Ambiguity Resolution by Mixture Kalman Filter for Improved GNSS Precision

Berntorp, Karl; Weiss, Avishai; Di Cairano, Stefano

TR2020-022 March 03, 2020

Abstract

Accurate carrier-phase integer ambiguity resolution is fundamental for high precision global navigation satellite systems (GNSSs). Real-time GNSSs typically resolve the ambiguities by a combination of recursive estimators and integer least squares solvers, which need to be reset when satellites are added or cycle slip occurs. In this paper we propose a mixture Kalman filter solution to integer ambiguity resolution. By marginalizing out the set of ambiguities and exploiting a likelihood proposal for generating the ambiguities, we can bound the possible values to a tight and dense set of integers. Thus, we extract the state and integer estimates from a mixture Kalman filter. The proposed approach yields an integrated method to detect cycle slip and initialize new satellites. A numerical analysis and experimental results indicate that the proposed method achieves reliable position estimates, repeatedly finds the correct integers in cases when other methods may fail, and is more robust to cycle slip.

IEEE Transactions on Aerospace and Electronic Systems

Integer Ambiguity Resolution by Mixture Kalman Filter for Improved GNSS Precision

Karl Berntorp[1], Avishai Weiss[1], and Stefano Di Cairano[1]

Abstract—**Accurate carrier-phase integer ambiguity resolution is fundamental for high precision global navigation satellite systems (GNSSs). Real-time GNSSs typically resolve the ambiguities by a combination of recursive estimators and integer least squares solvers, which need to be reset when satellites are added or cycle slip occurs. In this paper we propose a mixture Kalman filter solution to integer ambiguity resolution. By marginalizing out the set of ambiguities and exploiting a likelihood proposal for generating the ambiguities, we can bound the possible values to a tight and dense set of integers. Thus, we extract the state and integer estimates from a mixture Kalman filter. The proposed approach yields an integrated method to detect cycle slip and initialize new satellites. A numerical analysis and experimental results indicate that the proposed method achieves reliable position estimates, repeatedly finds the correct integers in cases when other methods may fail, and is more robust to cycle slip.**

I. Introduction

Global navigation satellite systems (GNSSs), such as GPS, Galileo, and, in the future, QZSS, are used in many positioning and navigation applications world-wide. GNSS receivers can be found in airplanes, cars, and cell phones. A GNSS receiver determines its position using two types of range measurements from several satellites orbiting the earth: pseudorange (or code) measurements and carrier-phase measurements. The code measurement is determined by multiplying the signal travel time from the satellite to the receiver with the speed of light. Code measurements are inexact because they include several sources of errors, such as satellite clock timing error, ionospheric and tropospheric refraction effects, receiver tracking noise, and multipath error. To reduce these errors, differential corrections are used in many GNSS applications [1]. The carrier-phase measurement is obtained by integrating a reconstructed carrier of the signal as it arrives at the receiver. The carrier-signal observations are more precise than the code measurements and can be tracked within a percent or less of the wavelength ($\lambda \approx 0.19$m for the L_1 frequency band). However, because of the unknown number of wave periods in transit between the satellite and the receiver, when the receiver starts tracking the carrier phase of the signal, there is an integer ambiguity in the carrier phase measurement. Furthermore, cycle slip, which is a sudden loss of lock of the carrier signal, for example due to a power loss or receiver failure, causes a jump in the carrier-phase measurements and is a common error source in GNSSs.

A standard approach to resolve the ambiguities, at start and after a cycle slip has occurred, is to use a recursive estimator such as an extended Kalman filter (EKF) combined with the least-squares ambiguity decorrelation method (LAMBDA) method, which first relaxes the ambiguity to be real valued (by EKF) and then obtains an integer solution for the ambiguity by a local search around such real value (by LAMBDA) [2]. An overview of traditional approaches for ambiguity resolution can be found in [3], and a summary of integer estimation theory is presented in [4]. Many GNSS ambiguity resolution methods are based on two-stage approaches. First an estimation of a real-valued ambiguity by an augmentation of the receiver state vector with the ambiguities, using recursive methods such as the extended Kalman filter (EKF) [5] or least squares [4], which forms the basis for an integer least squares (ILS) solution based on the real-valued estimates. An established example of this is the LAMBDA method [2], [6], [7]. The LAMBDA method has been further developed into the modified LAMBDA (MLAMBDA, also denoted LAMBDA version 3) [8], and related approaches can be found in [9], [10].

In this paper, we formulate the GNSS ambiguity resolution problem in a Bayesian framework as a joint GNSS receiver state and ambiguity parameter estimation problem. By exploiting marginalization, we can solve for the GNSS receiver state using a mixture Kalman filter, where each Kalman filter (KF) is conditioned on a unique fixed ambiguity for each satellite. Thus, based on the previous estimates of position and ambiguities, the proposed method produces the likely values for the ambiguities, which are used in the mixture Kalman Filter to update the position estimate. The approach employs the marginalized particle filter [11] to determine the ambiguities, which leads to a dimension reduction in the particle filter, resulting in a computationally tractable method.

Bayesian approaches have previously been considered in the context of GNSS ambiguity resolution, see for example [12], which presents a solution similar to the case of fixed multiple models, and [13], which uses Bayesian statistics to derive confidence regions for a GPS application. An overview of multiple-model methods for GNSS ambiguity resolution can be found in [14]. The work in [15] uses fixed multiple-model KFs, where the ambiguities are the integers the models depend on. A similar approach is found in [16], where each of the filters uses a different set of ambiguities, and switched multiple-model estimators for detection of cycle slip are found in [17]. A difficulty with the multiple-model approaches is that the ambiguities can take any integer value, and straightforward application of a multiple-model approach is therefore computationally intractable. In [14], a procedure using the LAMBDA method to search for the integers to use in the different models

[1] The authors are with Mitsubishi Electric Research Laboratories (MERL), 02139 Cambridge, MA, USA. Email: karl.o.berntorp@ieee.org

is mentioned, starting from the real-valued solution. However, this procedure needs to be restarted as soon as a cycle slip or loss of contact with the satellite occurs.

Particle filters have also been considered in relation to GNSS ambiguity resolution. An early work is [18], which, however, does not employ marginalization. In [19], [20] the integer estimates are formed using the position samples generated in the particle filter by manipulations of the likelihood. The work in [21] applies particle filtering for estimating the joint GNSS receiver state and ambiguities. However, estimating the joint state and ambiguities in a particle filter, and not utilizing marginalization, leads to an unnecessarily high-dimensional estimation problem, which is problematic in a real-time application because of the curse of dimensionality in the considered particle-filter implementations.

Any motion model of the ambiguities is highly uncertain due to the unboundedness of the ambiguity set and lack of knowledge of how and when the ambiguities change. In our approach, by leveraging the optimal proposal density for the ambiguities, which for the uncertain ambiguity model corresponds to likelihood sampling, we can statistically bound the possible range of ambiguities and execute a mixture KF. The number of KFs is made adaptive on the possible range of ambiguities. Hence, as the estimator narrows the ambiguity set, the number of particles decrease. Owing to the proposal sampling, our method can handle cycle slip automatically, which gives increased robustness to the position estimate. The specific contribution of this paper that distinguishes it from prior work is that we leverage marginalization, implying that the particle filter only estimates the ambiguities, whereas the receiver state is estimated conditioned on the ambiguity trajectories. This enables estimating a range of possible ambiguities, while keeping the computational requirements manageable.

Both the combined EKF/LAMBDA and multiple-model estimators need an additional cycle-slip detection scheme to make these methods work in practice, since the methods are restarted as soon as a cycle slip occurs. While EKF/LAMBDA works well in many situations, it still suffers from performance limitations. These are especially evident when cycle slip occurs, as this may result in a degraded position estimate for several seconds, or even minutes. In contrast, our method can, at least partially, handle cycle slip through the proposal sampling in the particle filter, but can also be used in conjunction with a cycle-slip detection scheme. We validate both approaches in our evaluation.

Notation: For a discrete time signal x with sampling period T_s, $x_k = x(t_k) = x(kT_s)$. For a vector-valued discrete time signal $\boldsymbol{x}$, $\boldsymbol{x}_{m:k} = \{\boldsymbol{x}_m, \dots, \boldsymbol{x}_k\}$ is the sequence of values of $\boldsymbol{x}$ between sampling instants m and k, and $\hat{\boldsymbol{x}}_{h|k}$ denotes the estimated value of $\boldsymbol{x}$ at time step h, based on data up to time step k. With $p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k})$, we mean the posterior density function of the state trajectory $\boldsymbol{x}_{0:k}$ given the measurement sequence $\boldsymbol{y}_{0:k}$. $\mathbb{R}$ denotes the set of real numbers and $\mathbb{Z}$ is the set of integer numbers. The jth element of a vector $\boldsymbol{x}$ is detoned with x_j. The notation $\boldsymbol{I}$ denotes the identity matrix of appropriate dimensions. With $\mathcal{N}(\boldsymbol{x}_k; \boldsymbol{\mu}, \boldsymbol{\Upsilon})$ we mean the Gaussian probability density function given mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Upsilon}$.

Outline: Sec. II gives the problem setup and modeling aspects. Sec. III presents the proposed method for integer ambiguity resolution. Sec. IV contains a numerical analysis and the experimental results are found in Sec. V. Finally, Sec. VI concludes the paper.

II. Problem Setup

We consider the code and carrier-phase measurements from the jth satellite to the receiver r at each time t_k, that is, at step (epoch) k, using a standard measurement model [1], [4], [18], [21], [22],

$$P_k^j = \rho_k^j + c(\delta t_{r,k} - \delta t_k^j) + I_k^j + T_k^j + \epsilon_k^j, \tag{1a}$$

$$\Phi_k^j = \rho_k^j + c(\delta t_{r,k} - \delta t_k^j) - I_k^j + T_k^j + \lambda n^j + \eta_k^j, \tag{1b}$$

where P^j is the code measurement ρ^j is the distance between the receiver and the jth satellite, c is the speed of light, δt_r is the receiver clock bias, δt^j is the satellite clock bias, I^j is the ionospheric delay, T^j is the tropospheric delay, ϵ^j is the code observation noise, Φ^j is the carrier-phase observation, λ is the carrier wavelength, n^j is the ambiguity, and η^j is the carrier observation noise. The code and carrier-phase noise sources are assumed Gaussian distributed according to $\epsilon_k \sim \mathcal{N}(0, \sigma_\epsilon^2)$, $\eta_k \sim \mathcal{N}(0, \sigma_\eta^2)$, where the actual values will depend on several factors (e.g., the elevation angle of the satellite). The distance between the receiver and the jth satellite is

$$\rho^j = \sqrt{(p_X^j - p_{X,r})^2 + (p_Y^j - p_{Y,r})^2 + (p_Z^j - p_{Z,r})^2}, \tag{2}$$

where $\boldsymbol{p}^j = [p_X^j \ p_Y^j \ p_Z^j]^{\mathrm{T}}$ and $\boldsymbol{p}_r = [p_{X,r} \ p_{Y,r} \ p_{Z,r}]^{\mathrm{T}}$ are the coordinates of the jth satellite and the receiver r, respectively. By utilizing a base receiver (reference) b mounted at a known location broadcasting to the target receiver r, most of the error sources can be removed or mitigated. Single and double differences (DD) between satellites and the receivers reduce the error sources, either explicitly or approximately. By forming the difference of the observation equation (1) for the two receivers, the error due to the satellite clock bias can be eliminated. For very short baselines (the distance between r and the base receiver b) the DD, that is, the single difference between receivers differenced again between two satellites, eliminates the ionospheric and tropospheric delays [21], [22]. For longer baselines (e.g., above 10 km) DD suppresses the errors, and other established methods can be used [23], [24] to reduce the effects further.

Denote the single-differenced observation equations between the receivers b and r with $\Delta P_{br,k}^j = P_{b,k}^j - P_{r,k}^j$ and $\Delta \Phi_{br,k}^j = \Phi_{b,k}^j - \Phi_{r,k}^j$, respectively, and the DD between a reference (pivot) satellite l and satellite j with $\nabla\Delta(\cdot)_{br,k}^{jl}$. Then, for short baseline conditions and/or using a priori delay estimators [23], [24],

$$\nabla\Delta P_{br,k}^{jl} \approx \nabla\Delta\rho_{br,k}^{jl} + \nabla\Delta\epsilon_{br,k}^{jl}, \tag{3a}$$

$$\nabla\Delta\Phi_{br,k}^{jl} \approx \nabla\Delta\rho_{br,k}^{jl} + \lambda\nabla\Delta n_{br}^{jl} + \nabla\Delta\eta_{br,k}^{jl}. \tag{3b}$$

We assume M sets of observation equations of the form (3),[1] where M can vary over different time steps, and introduce

$$\boldsymbol{n} = \left[\nabla\Delta n_{br,k}^{l1} \ \cdots \ \nabla\Delta n_{br,k}^{lM}\right]^{\mathrm{T}} \in \mathbb{Z}^M. \tag{4}$$

[1] For a single-frequency setting, this implies $M+1$ visible satellites.

Without loss of generality, the last satellite is assumed to be the reference satellite (i.e., $l = M + 1$). We form the observation equations at each time step k from the double-differenced measurements (3)

$$\boldsymbol{y}_k = \begin{bmatrix}\nabla\Delta P^{l1}_{br,k} & \cdots & \nabla\Delta P^{lM}_{br,k} & \nabla\Delta\Phi^{l1}_{br,k} & \cdots & \nabla\Delta\Phi^{lM}_{br,k}\end{bmatrix}^{\mathrm{T}}, \tag{5}$$

yielding the corresponding measurement model

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{g}(\boldsymbol{n}_k) + \boldsymbol{e}_k, \tag{6}$$

where

$$\boldsymbol{h}(\boldsymbol{x}_k) = \begin{bmatrix}\nabla\Delta\rho^{l1}_{br,k} \\ \vdots \\ \nabla\Delta\rho^{lM}_{br,k} \\ \nabla\Delta\rho^{l1}_{br,k} \\ \vdots \\ \nabla\Delta\rho^{lM}_{br,k}\end{bmatrix}, \quad \boldsymbol{g}(\boldsymbol{n}_k) = \lambda\begin{bmatrix}0 \\ \vdots \\ 0 \\ \boldsymbol{n}_k\end{bmatrix}, \quad \boldsymbol{e} = \begin{bmatrix}\nabla\Delta\epsilon^{l1}_{br,k} \\ \vdots \\ \nabla\Delta\epsilon^{lM}_{br,k} \\ \nabla\Delta\eta^{l1}_{br,k} \\ \vdots \\ \nabla\Delta\eta^{lM}_{br,k}\end{bmatrix}. \tag{7}$$

As the proposed method is to be used for receivers employed in different types of environments, we model the receiver motion using the linear general-purpose state-transition model

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_k\boldsymbol{x}_k + \boldsymbol{B}_k\boldsymbol{w}_{\boldsymbol{x},k}, \tag{8}$$

where $\boldsymbol{F}_k$ is the state-transition matrix and $\boldsymbol{B}_k$ is the noise-transition matrix. The motion model (8) includes general-purpose kinematic motion models where little is known about the moving object. While here we focus on (8), note that since we rely on particle filtering our proposed approach can handle models that are more general than (8). In the evaluation of our method in Sec. IV, we use a constant-velocity (CV) model with the state vector

$$\boldsymbol{x}_k = \begin{bmatrix}\boldsymbol{p}_{r,k} & \boldsymbol{v}_{r,k}\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^6, \tag{9}$$

where $\boldsymbol{v}_r$ is the receiver velocity. Using zero-order hold sampling with sampling period T_s, the CV model is [25]

$$\boldsymbol{x}_{k+1} = \begin{bmatrix}\boldsymbol{I} & T_s\boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\boldsymbol{x}_k + \begin{bmatrix}\frac{T_s^2}{2}\boldsymbol{I} \\ T_s\boldsymbol{I}\end{bmatrix}\boldsymbol{w}_{\boldsymbol{x},k}, \tag{10}$$

where $\boldsymbol{w}_{\boldsymbol{x},k} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_{\boldsymbol{x},k})$.

The estimation problem includes the state vector $\boldsymbol{x}_k$, which for the CV model $\boldsymbol{x}_k \in \mathbb{R}^6$, and the ambiguity vector $\boldsymbol{n}_k \in \mathbb{Z}^M$. The objective in this paper is to resolve the unknown receiver state $\boldsymbol{x}_k$ and as a consequence the set of unknown integer ambiguities $\boldsymbol{n}_k$ at each time step k from the measurements $\boldsymbol{y}_{0:k} = \{\boldsymbol{y}_0, \ldots, \boldsymbol{y}_k\}$. The integer ambiguities are typically constant for extended periods of time, but may change abruptly as soon as a loss-of-lock of tracking of the satellite occurs, for instance, due to shadowing effects in urban areas or cycle slip.

III. GNSS Integer Ambiguity and Position Resolution

This section presents the proposed method for joint positioning and integer ambiguity resolution. We approach the problem in a fully Bayesian context and rely on a mixture KF approach where the number of KFs is adapted to the uncertainty in the ambiguity estimates.

The estimation model consisting of (6) and (8) is nonlinear in the position due to the observation equations. The observations are linear in the ambiguity vector. However, formulating a dynamic model that properly describes the time evolution of the ambiguities is difficult, since the ambiguities are approximately constant over lapses of time but there can be abrupt and usually unpredictable changes in the ambiguity values from one time step to another. To reflect the uncertainty in the time evolution of the ambiguities, we describe them with a random-walk model

$$\boldsymbol{n}_{k+1} = \boldsymbol{n}_k + \boldsymbol{w}_{\boldsymbol{n},k}, \quad \boldsymbol{w}_{\boldsymbol{n},k} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_{\boldsymbol{n}}). \tag{11}$$

In a Bayesian formulation, we write (11) as

$$\boldsymbol{n}_{k+1} \sim p(\boldsymbol{n}_{k+1}|\boldsymbol{n}_k). \tag{12}$$

A. Bounding the Range of Ambiguities

The method is based on estimating the ambiguities in the particle filter and resolving the receiver state in the EKF, where each particle is a hypothesis of the integer vector (4). We leverage marginalized particle filtering for bounding the possible range of the ambiguities. Based on the bounded ranges, we fix the set of integer vectors and execute one EKF for each value in the set. The ambiguity is linear in both the motion model and observation equations, and estimating the ambiguities in the particle filter may on the outset seem counterintuitive. However, introducing the ambiguities in the particle filter makes it possible to efficiently move around them in the ambiguity space according to the fit of the observation model to the measurements.

To bound the range of ambiguities, we start with the joint density $p(\boldsymbol{x}_k, \boldsymbol{n}_{0:k}|\boldsymbol{y}_{0:k})$ of receiver state $\boldsymbol{x}_k$ and ambiguity trajectory $\boldsymbol{n}_{0:k}$, which we decompose as

$$p(\boldsymbol{x}_k, \boldsymbol{n}_{0:k}|\boldsymbol{y}_{0:k}) = p(\boldsymbol{x}_k|\boldsymbol{n}_{0:k}, \boldsymbol{y}_{0:k})p(\boldsymbol{n}_{0:k}|\boldsymbol{y}_{0:k}). \tag{13}$$

To resolve (13), we first estimate $p(\boldsymbol{n}_{0:k}|\boldsymbol{y}_{0:k})$ with a particle filter using a set of N weighted particles, which results in the approximation

$$p(\boldsymbol{n}_{0:k}|\boldsymbol{y}_{0:k}) \approx \sum_{i=1}^{N} q_k^i \delta(\boldsymbol{n}_{0:k}^i - \boldsymbol{n}_{0:k}). \tag{14}$$

In (14), $\delta(\cdot)$ is the Dirac delta mass and q_k^i is the associated weight for the ith particle given the measurements $\boldsymbol{y}_{0:k}$.

Given the ambiguity set $\{\boldsymbol{n}_{0:k}^i\}_{i=1}^N$, we execute constrained EKFs to determine the first term on the right-hand side of (13), which results in the Gaussian approximation

$$p(\boldsymbol{x}_k|\boldsymbol{n}_{0:k}, \boldsymbol{y}_{0:k}) \approx \mathcal{N}(\boldsymbol{x}_k; \hat{\boldsymbol{x}}_{k|k}(\boldsymbol{n}_{0:k}), \boldsymbol{P}_{k|k}(\boldsymbol{n}_{0:k})) \tag{15}$$

for each particle. In (15), $\hat{\boldsymbol{x}}_{k|k}(\boldsymbol{n}_{0:k})$ is the state estimate constrained to the ambiguity trajectory $\boldsymbol{n}_{0:k}$, and $\boldsymbol{P}_{k|k}(\boldsymbol{n}_{0:k})$ is its associated covariance. For brevity, in what follows we make the dependence on the ambiguity implicit (i.e., $\boldsymbol{P}_{k|k} := \boldsymbol{P}_{k|k}(\boldsymbol{n}_{0:k})$) for all variables. The mean and covariance of the conditional probability density function (15) using the EKF is given by [11]

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_{k|k-1}) - \boldsymbol{g}(\boldsymbol{n})_k), \tag{16a}$$

$$\boldsymbol{P}_{k|k} = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k \boldsymbol{H}_k \boldsymbol{P}_{k|k-1}, \tag{16b}$$

$$\boldsymbol{S}_k = \boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^{\mathrm{T}} + \boldsymbol{R}_k, \tag{16c}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^{\mathrm{T}} \boldsymbol{S}_k^{-1}, \tag{16d}$$

$$\boldsymbol{H}_k = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k|k-1}}, \tag{16e}$$

and the one-step prediction of the mean and covariance are

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{F}_{k-1} \hat{\boldsymbol{x}}_{k-1|k-1}, \tag{17a}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_{k-1} \boldsymbol{P}_{k-1|k-1} \boldsymbol{F}_{k-1}^{\mathrm{T}} + \boldsymbol{Q}_{\boldsymbol{x},k-1}. \tag{17b}$$

The weight update in the particle filter is given by

$$q_k^i = \frac{p(\boldsymbol{y}_k|\boldsymbol{n}_k, \boldsymbol{y}_{0:k-1}) p(\boldsymbol{n}_k|\boldsymbol{n}_{k-1}^i)}{\pi(\boldsymbol{n}_k|\boldsymbol{n}_{k-1}^i, \boldsymbol{y}_{0:k})} q_{k-1}^i, \tag{18}$$

where

$$\pi(\boldsymbol{n}_k|\boldsymbol{n}_{k-1}^i, \boldsymbol{y}_{0:k}). \tag{19}$$

A key design choice in the particle filter is the proposal density (19). The standard choice is to use the prior (12), that is, $\pi(\boldsymbol{n}_k|\boldsymbol{n}_{k-1}^i, \boldsymbol{y}_{0:k}) = p(\boldsymbol{n}_k|\boldsymbol{n}_{k-1})$. However, this would demand an unnecessarily large amount of particles, since the prediction model (12) of the ambiguities is uninformative. Instead, we choose the conditional distribution as proposal density,

$$\pi(\boldsymbol{n}_k|\boldsymbol{n}_{k-1}^i, \boldsymbol{y}_{0:k}) = p(\boldsymbol{n}_k|\boldsymbol{n}_{k-1}^i, \boldsymbol{y}_{0:k}). \tag{20}$$

Inserting (20) into (18) and using the identity

$$p(\boldsymbol{n}_k|\boldsymbol{n}_{k-1}^i, \boldsymbol{y}_{0:k}) = \frac{p(\boldsymbol{y}_k|\boldsymbol{n}_k, \boldsymbol{y}_{0:k-1}) p(\boldsymbol{n}_k|\boldsymbol{n}_{k-1}^i)}{p(\boldsymbol{y}_k|\boldsymbol{n}_{k-1}^i, \boldsymbol{y}_{0:k-1})} \tag{21}$$

leads to the weight update

$$q_k^i \propto p(\boldsymbol{y}_k|\boldsymbol{n}_{k-1}^i, \boldsymbol{y}_{0:k-1}) q_{k-1}^i. \tag{22}$$

The proposal (20) is optimal in the sense that it minimizes the effect of the sampling on the weights, that is, the weights will be unaffected by $\boldsymbol{n}_k^i$, whereas other alternatives add variance among the weights [25]. It is generally difficult to sample from (20). However, the observation equation (6) is linear and Gaussian in the ambiguity vector $\boldsymbol{n}$, which is one of the few cases where exact sampling is possible [26]. For a linear and Gaussian observation equation, the optimal proposal (20) for a marginalized particle filter is

$$p(\boldsymbol{n}_k|\boldsymbol{n}_{k-1}^i, \boldsymbol{y}_{0:k}) = \mathcal{N}\left(\boldsymbol{n}_k; \hat{\boldsymbol{n}}_k^i, \boldsymbol{\Sigma}_k^i\right) \tag{23a}$$

$$\hat{\boldsymbol{n}}_k^i = \boldsymbol{n}_{k-1}^i + \boldsymbol{K}_k^i(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1}^i), \tag{23b}$$

$$\boldsymbol{K}_k^i = \boldsymbol{Q}_{\boldsymbol{n}}(\boldsymbol{G}_k^i)^{\mathrm{T}}(\boldsymbol{G}_k^i \boldsymbol{Q}_{\boldsymbol{n}}(\boldsymbol{G}_k^i)^{\mathrm{T}} + \boldsymbol{S}_k^i)^{-1}, \tag{23c}$$

$$\boldsymbol{\Sigma}_k^i = \left((\boldsymbol{S}_k^i)^{-1} + (\boldsymbol{G}_k^i \boldsymbol{Q}_{\boldsymbol{n}}(\boldsymbol{G}_k^i)^{\mathrm{T}})^{-1}\right)^{-1}, \tag{23d}$$

$$\boldsymbol{G}_k^i = \left.\frac{\partial \boldsymbol{g}(\boldsymbol{n})}{\partial \boldsymbol{n}}\right|_{\boldsymbol{n}=\boldsymbol{n}_{k-1}^i}, \tag{23e}$$

where $\boldsymbol{S}_k^i$ is obtained from (16c), conditioned on the ith ambiguity trajectory. In this paper we choose $\boldsymbol{Q}_{\boldsymbol{n}}$ large such that the ambiguity prior (12) is uninformative, which means that most information about the ambiguities is contained in the measurements. For the proposal (23),

$$\lim_{\boldsymbol{Q}_{\boldsymbol{n}} \to \infty} \boldsymbol{\Sigma}_k^i = \left((\boldsymbol{S}_k^i)^{-1} + (\boldsymbol{G}_k^i \boldsymbol{Q}_{\boldsymbol{n}}(\boldsymbol{G}_k^i)^{\mathrm{T}})^{-1}\right)^{-1} \to \boldsymbol{S}_k^i, \tag{24}$$

that is, by increasing the process noise of the ambiguities, the optimal proposal approaches sampling from the EKF likelihood (16c). With the optimal proposal (23), the likelihood for the weight update (22) is

$$\begin{aligned} p(\boldsymbol{y}_k|\boldsymbol{n}_{k-1}^i, \boldsymbol{y}_{0:k-1}) &= \int p(\boldsymbol{y}_k, \boldsymbol{x}_k|\boldsymbol{n}_{k-1}^i, \boldsymbol{y}_{0:k-1}) \,\mathrm{d}\boldsymbol{x}_k \\ &= \int p(\boldsymbol{y}_k|\boldsymbol{n}_{k-1}^i, \boldsymbol{x}_k) p(\boldsymbol{x}_k|\boldsymbol{y}_{0:k-1}) \,\mathrm{d}\boldsymbol{x}_k \\ &\approx \mathcal{N}\left(\boldsymbol{y}_k|\hat{\boldsymbol{y}}_{k|k-1}^i, \boldsymbol{G}_k^i \boldsymbol{Q}_{\boldsymbol{n}}(\boldsymbol{G}_k^i)^{\mathrm{T}} + \boldsymbol{S}_k^i\right). \end{aligned} \tag{25}$$

Note that although the optimal proposal (23a) and therefore also the likelihood (25) are linear in the ambiguities, the covariance $\boldsymbol{S}_k^i$ is obtained from the EKF recursion, which is approximate.

The generated ambiguities are real valued when using the optimal proposal (23). However, we can bound the range of possible integer ambiguities as follows. We apply particle replacement by resampling according to their importance weights q_k^i. After resampling with replacement, all particles have weight $q_k^i = 1/N$ and the particle-filter approximation (14) becomes

$$p(\boldsymbol{n}_k|\boldsymbol{y}_{0:k}) \approx \hat{p}(\boldsymbol{n}_k|\boldsymbol{y}_{0:k}) = \frac{1}{N}\sum_{i=1}^{N} \delta(\boldsymbol{n}_k^i - \boldsymbol{n}_k). \tag{26}$$

To get a measure of the tails of the distribution for a finite number of particles, we convert the discrete representation (26) to a continuous density using a kernel density smoother [27],

$$\hat{p}_K(\boldsymbol{n}_k|\boldsymbol{y}_{0:k}) = \frac{1}{N}\sum_{i=1}^{N} K_h(\boldsymbol{n}_k^i - \boldsymbol{n}_k), \tag{27}$$

where $K_h(\cdot)$ is the kernel density and h is the bandwidth. We truncate (27), resulting in the continuous truncated density $\hat{p}_{K,\mathrm{tr}}(\boldsymbol{n}_k|\boldsymbol{y}_{0:k})$. Based on $\hat{p}_{K,\mathrm{tr}}(\boldsymbol{n}_k|\boldsymbol{y}_{0:k})$ we fix the ambiguities, which gives a finite set $\mathcal{S}$ of $N_{\mathcal{S}}$ possible integer vectors $\{\boldsymbol{n}_{\mathrm{I}}^i\}_{i=1}^{N_{\mathcal{S}}}$ contained in the support of $\hat{p}_{K,\mathrm{tr}}$, that is,

$$\mathcal{S} = \{\boldsymbol{n}_{\mathrm{I}} \in \mathbb{Z}^M : \hat{p}_{K,\mathrm{tr}}(\boldsymbol{n}_{\mathrm{I}}|\boldsymbol{y}_{0:k}) > 0\}. \tag{28}$$

The weight update (22) of the particle filter and (26)–(28) do not need to be performed at every time step. In practice, we measure the difference between the predicted real-valued ambiguities according to some distance measure $\mathrm{dist}(\{\boldsymbol{n}_k^i\}_{i=1}^N, \{\boldsymbol{n}_{k-1}^i\}_{i=1}^N)$, and execute (22), (25), (26)–(28) only if the distance is larger than some threshold γ.

B. Ambiguity Resolution by Mixture Kalman Filter

Provided the $N_{\mathcal{S}}$ possible integer vectors in (28), we execute a bank of $N_{\mathcal{S}}$ EKFs to find the state vector $\boldsymbol{x}_k^{\mathrm{KF}}$, where each

EKF is conditioned on a unique integer ambiguity vector contained in (28).

The state posterior is expressed using the law of total probability as a Gaussian mixture of $N_\mathcal{S}$ components,

$$\begin{aligned} p(\boldsymbol{x}_k^{\mathrm{KF}}|\boldsymbol{y}_{0:k}) &= \sum_{i=1}^{N_\mathcal{S}} p(\boldsymbol{n}_\mathrm{I}^i, \boldsymbol{x}_k^{\mathrm{KF}}|\boldsymbol{y}_{0:k}) \\ &= \sum_{i=1}^{N_\mathcal{S}} p(\boldsymbol{n}_\mathrm{I}^i|\boldsymbol{y}_{0:k}) p(\boldsymbol{x}_k^{\mathrm{KF}}|\boldsymbol{n}_\mathrm{I}^i, \boldsymbol{y}_{0:k}) \\ &= \sum_{i=1}^{N_\mathcal{S}} \omega_k^i \mathcal{N}(\boldsymbol{x}_k^{\mathrm{KF}}|\hat{\boldsymbol{x}}_{k|k}^{\mathrm{KF},i}, \boldsymbol{P}_{k|k}^i), \end{aligned} \tag{29}$$

where

$$\omega_k^i = p(\boldsymbol{n}_\mathrm{I}^i|\boldsymbol{y}_{0:k}) \tag{30}$$

is the posterior probability of $\boldsymbol{n}_\mathrm{I}^i$. The recursions for $\hat{\boldsymbol{x}}_{k|k}^{\mathrm{KF},i}, \boldsymbol{P}_{k|k}^{\mathrm{KF},i}$ are in (16) with $\bar{\boldsymbol{n}}_k$ replaced with $\bar{\boldsymbol{n}}_\mathrm{I}^i = [0 \, \cdots \, \boldsymbol{n}_\mathrm{I}^i]^\mathrm{T}$. Note that $\boldsymbol{n}_\mathrm{I}^i$ in (29) is fixed for each EKF, hence the omission of index k. The posterior probabilities (30) can be computed from Bayes' rule,

$$\begin{aligned} \omega_k^i = p(\boldsymbol{n}_\mathrm{I}^i|\boldsymbol{y}_{0:k}) &= p(\boldsymbol{y}_k|\boldsymbol{n}_\mathrm{I}^i, \boldsymbol{y}_{0:k-1}) \frac{p(\boldsymbol{n}_\mathrm{I}^i|\boldsymbol{y}_{0:k-1})}{p(\boldsymbol{y}_k|\boldsymbol{y}_{0:k-1})} \\ &\propto \omega_{k-1}^i \mathcal{N}(\boldsymbol{y}_k|\hat{\boldsymbol{y}}_{k|k-1}^i, \boldsymbol{S}_k^i), \end{aligned} \tag{31}$$

where the mean and covariance predictions $\hat{\boldsymbol{y}}_{k|k-1}^i, \boldsymbol{S}_k^i$ are given from the corresponding EKF. From (31), we choose the maximum-likelihood (ML) estimate $\boldsymbol{n}^{\mathrm{ML}}$ to resolve the ambiguity,

$$\boldsymbol{n}^{\mathrm{ML}} = \arg\max_{\boldsymbol{n}_\mathrm{I} \in \mathcal{S}} \omega_k^i. \tag{32}$$

The state estimate and corresponding covariance can either be obtained by the corresponding ML or by the minimum-variance (MV) estimate [11]

$$\hat{\boldsymbol{x}}_{k|k}^{\mathrm{MV}} = \sum_{i=1}^{N_\mathcal{S}} \omega_k^i \hat{\boldsymbol{x}}_{k|k}^{\mathrm{KF},i}, \tag{33a}$$

$$\boldsymbol{P}_{k|k}^{\mathrm{MV}} = \sum_{i=1}^{N_\mathcal{S}} \omega_k^i \left(\boldsymbol{P}_{k|k}^i + (\hat{\boldsymbol{x}}_{k|k}^{\mathrm{KF},i} - \hat{\boldsymbol{x}}_{k|k}^{\mathrm{MV}})(\hat{\boldsymbol{x}}_{k|k}^{\mathrm{KF},i} - \hat{\boldsymbol{x}}_{k|k}^{\mathrm{MV}})^\mathrm{T} \right). \tag{33b}$$

Note that the particular choice of how to extract the estimate does not carry any importance to the algorithm, since it estimates the distribution for each time step. In our implementation we have chosen to select $\boldsymbol{n}$ from ML and $\boldsymbol{x}$ from MV, while noting that in general, $\boldsymbol{n}^{\mathrm{ML}}$ will not be used as what the algorithm carries forward is the ambiguity distribution, not its current estimate.

C. Algorithm Implementation

With the fixed finite set of integers (28) and the mixture KF solution (29) based on such set, an algorithm for joint ambiguity resolution and positioning can be formulated as in Algorithm 1. The method is initialized with N states and ambiguities (particles), with placeholders also for the mixture EKF states (Line 1). We execute a prediction step of the ambiguities using (23a) (Line 7). If the predicted ambiguities differ from the previous ones more than some threshold γ (Line 9), we compute the weights (Line 12), from which we construct a continuous representation (27) (Line 15). The range of ambiguities is then bounded and the method fixes $N_\mathcal{S}$ integer vectors (Line 16) and executes EKFs (Line 22) to compute the ML estimate (32) (Line 25) and MV state estimates (33) (Line 26). Finally, we sample a new set of indices $J(i) \in \mathbb{N}, i = 1, ...N$ to be used in the prediction of ambigities in the subsequent time step. If $N = N_\mathcal{S}$, then $J(i) = i$. However, if $N \neq N_\mathcal{S}$, we need to determine a set of N indices reflecting the updated weights from the filter bank. Note that (16) and (17) are used several times in Algorithm 1, for example, (16) is used both on Line 11 and Line 22 and (17) is used on Line 4 and Line 19. The superscripts on the variables being updated indicate what variables (16) and (17) are applied to.

There are a few design choices available in Algorithm 1. The number of particles N used in the prediction when determining whether the ambiguities have changed (Lines 3–8) is a design choice that in the current implementation is fixed a priori, but there are other alternatives, such as using $N_\mathcal{S}$. In determining whether the ambiguities have changed (Line 9), the distance function can be implemented in several ways. In our implementation, we simply check the largest difference in $\{\|\boldsymbol{n}_k^i - \boldsymbol{n}_{k-1}^i\|\}_{i=1}^N$. However, this may trigger unncessary reinitialization of the entire Kalman filter bank in case of cycle slip. Hence, this check can be improved. The reason for large changes in the ambiguity set between consecutive time steps can be come large due to, for example, cycle slip and that unrealiable measurements were not removed properly before transmitting the measurement vector to the estimator. In computing the continuous density $\hat{p}_K$, then truncated to obtain (28), we use a Gaussian kernel (Line 15), but other alternatives can also be used. The initialization at Line 17 can be done in several ways. At Line 28, we draw new indices to determine the ambiguity difference in the next iteration. Here, the number of particles can be related to $N_\mathcal{S}$ instead of fixing it to N. Moreover, the number N used at Line 27 can depend on $N_\mathcal{S}$. The ambiguity process noise covariance $\boldsymbol{Q}_{\boldsymbol{n}}$ is an important design parameter, since it will affect the search space explored by the method. From (24), letting $\boldsymbol{Q}_{\boldsymbol{\eta}} \to \infty$ implies that we are effectively sampling from the likelihood. Hence, the search space will be adjusted to the measurements instantaneously, which implies that effects such as cycle slip are accounted for immediately. However, the ambiguities to explore will also be sensitive to the measurement noise and will rely heavily on unhealthy satellites being properly discarded at each time step. A proper tuning of $\boldsymbol{Q}_{\boldsymbol{n}}$ is therefore such that the convergence is fast, yet does not fluctuate too much with the measurement noise at each time step. Since this is a single parameter that can be set equal for all ambiguities, tuning involves a simple search on a single dimensional space which is in general not very time consuming.

IV. Numerical Analysis

We present a numerical study with three double-differenced satellite measurements, that is, six measurements and three

Algorithm 1 Proposed method

```
**Initialize:** Generate $\{\boldsymbol{n}^i_{-1}\}_{i=1}^N \sim p_0(\boldsymbol{n}_0)$, $\{\hat{\boldsymbol{x}}^i_{0|-1}\}_{i=1}^N \sim p_0(\boldsymbol{x}_0)$, $\{\boldsymbol{P}^i_{0|-1}\}_{i=1}^N = \boldsymbol{P}_0$, $\{\hat{\boldsymbol{x}}_0^{\mathrm{KF},i}\}_{i=1}^{N_{\mathcal{S}}} \sim p_0(\boldsymbol{x}_0)$, $\{\boldsymbol{P}^{\mathrm{KF}i}_{0|-1}\}_{i=1}^{N_{\mathcal{S}}} = \boldsymbol{P}_0$ and set $\{w^i_{-1}\}_{i=1}^N = 1/N$.
for $k = 0$ to $T$ do
  Predict the ambiguities:
  for $i = 1$ to $N$ do
    Update $\{\hat{\boldsymbol{x}}^i_{k|k-1}, \boldsymbol{P}^i_{k|k-1}\}$ using (17).
    Set $\hat{\boldsymbol{y}}^i_{k|k-1} = \boldsymbol{h}(\hat{\boldsymbol{x}}^i_{k|k-1}) + \lambda \boldsymbol{n}^i_k$.
    Compute $\boldsymbol{S}^i_k$ from (16c).
    Generate $\boldsymbol{n}^i_k \sim p(\boldsymbol{n}_k|\boldsymbol{n}^i_{k-1}, \boldsymbol{y}_{0:k})$ from (23a).
  end for
  Bounding the range of ambiguities:
  if $\mathrm{dist}(\{\boldsymbol{n}^i_k\}_{i=1}^N, \{\boldsymbol{n}^i_{k-1}\}_{i=1}^N) > \gamma$ then
    for $i = 1$ to $N$ do
      Update $\{\hat{\boldsymbol{x}}^i_{k|k}, \boldsymbol{P}^i_{k|k}\}$ from (16).
      Update $q^i_k$ using (22) and (25).
    end for
    Resample particles to get equally weighted particles and distribution (26).
    Compute $\hat{p}_K(\boldsymbol{n}_k|\boldsymbol{y}_{0:k})$ using (27).
    Determine $\{\boldsymbol{n}^i_{\mathrm{I}}\}_{i=1}^{N_{\mathcal{S}}}$ using (28).
    Initialize $\{\hat{\boldsymbol{x}}^{\mathrm{KF},i}_{k|k-1}\}_{i=1}^{N_{\mathcal{S}}}$, $\{\boldsymbol{P}^{\mathrm{KF},i}_{k|k-1}\}_{i=1}^{N_{\mathcal{S}}}$.
  else
    Update $\{\hat{\boldsymbol{x}}^{\mathrm{KF},i}_{k|k-1}, \boldsymbol{P}^{\mathrm{KF},i}_{k|k-1}\}_{i=1}^{N_{\mathcal{S}}}$ using (17).
  end if
  Update filter bank:
  for $i = 1$ to $N_{\mathcal{S}}$ do
    Update $\{\hat{\boldsymbol{x}}^{\mathrm{KF},i}_{k|k}, \boldsymbol{P}^{\mathrm{KF},i}_{k|k}\}$ using (16).
    Update weight $\omega^i_k$ using (31).
  end for
  Optionally, resample filter bank according to the weights $\{\omega^i_k\}_{i=1^{N_{\mathcal{S}}}}$.
  Determine estimates:
  Compute $\boldsymbol{n}^{\mathrm{ML}}$ using (32).
  Compute $\{\hat{\boldsymbol{x}}^{\mathrm{MV}}_{k|k}, \boldsymbol{P}^{\mathrm{MV}}_{k|k}\}$ using (33).
  Set indices for prediction of ambiguities (for $N \neq N_{\mathcal{S}}$):
  for $i = 1$ to $N$ do
    Draw index $J(i)$ with probability $\omega_k^{J(i)}$.
  end for
  Set $\{\boldsymbol{n}^i_k, \hat{\boldsymbol{x}}^i_{k|k}, \boldsymbol{P}^i_{k|k}\}_{i=1}^N = \{\boldsymbol{n}^l_{\mathrm{I}}, \hat{\boldsymbol{x}}^{\mathrm{KF},l}_{k|k}, \boldsymbol{P}^{\mathrm{KF},l}_{k|k}\}_{l=J(1)}^{J(N)}$.
end for
```

ambiguities. The satellites measure the distance to a moving rover with sampling time $T_s = 0.1$ s. The positioning target is a rover that travels on the Earth-ground plane ($p_Z = 0$) with nominal speed 5 m/s.

We compare the method presented in this work with that of a two-stage approach based on first executing an EKF to find the real-valued ambiguities and then, using the LAMBDA method to find the integer ambiguities [4], [5], [21]. We denote this filter with EKF. We also compare with an idealized (oracle) implementation of the EKF that knows the correct ambiguities at each time instant. Hence, this method is impossible to implement in practice, but serves as ground-truth of what can be achieved in terms of tracking performance when the ambiguities are known. The purpose of the numerical analysis is to illustrate the transient behavior of the proposed method in the ideal conditions when the model and noise covariances are known and we have ground truth. Hence, neither of the methods incorporate additional schemes such as explicit cycle-slip detection, which would be incorporated in a real-world implementation, but may shadow the effect of the core estimation algorithm. We defer to the next section to actual validation in real-world conditions, that include inexact models, unknown covariances, and introduction of cycle slips. The ambiguities in both filters are initialized to zero, and the state/particles are set to the true rover state with initial covariance

$$\boldsymbol{P}_0 = \begin{bmatrix} 100 & 100 & 1 & 1 & 1 & 0.1 \end{bmatrix}^{\mathrm{T}}. \tag{34}$$

The noise parameters are the same in both filters. The code and phase measurement standard deviations are set to 0.5 m and 0.1 m, respectively. All satellites are assumed to have the same elevation angle, such that the measurement covariance is diagonal with the nominal code and phase standard deviation. Furthermore, the motion model (8) and associated process-noise covariance are modeled with respect to the Earth-centered earth-fixed (ECEF) frame, with $\boldsymbol{Q} = \mathrm{diag}(1^2, 1^2, 0.1^2)$, where $\mathrm{diag}(\cdot)$ is the diagonal matrix with the arguments on the diagonal.

We present results for two scenarios. The first scenario illustrates how the method performs under large initial position and ambiguity uncertainty, and the second scenario considers cycle-slip detections. The root-mean-square-error (RMSE) at each time step is used as a performance measure. Let $\boldsymbol{x}_{k,j}$ and $\hat{\boldsymbol{x}}_{k,j}$ denote the true and estimated quantity, respectively, at time index k of the jth of K Monte-Carlo simulations. The RMSE at time step k is then computed as

$$\mathrm{RMSE}_k = \sqrt{\frac{1}{N_{\mathrm{MC}}} \sum_{j=1}^{N_{\mathrm{MC}}} \|\boldsymbol{x}_{k,j} - \hat{\boldsymbol{x}}_{k,j}\|^2}. \tag{35}$$

A. Transient Performance

Fig. 1 shows the position error of the longitudinal position for the first 50 s for one realization, and Fig. 2 displays the corresponding ambiguity estimation error for one of the ambiguities. The results are similar for the other positions and ambiguities, respectively. The convergence of the proposed method ALG 1 is faster than that of EKF. Furthermore, whereas ALG 1 chooses the correct integer estimates after the transients, EKF fluctuates around the correct value. In the simulations, the initial position is generated from a Gaussian distribution with covariance (34). The true ambiguities are $\{n_1, n_2, n_3\} = \{-220, 210, 175\}$ but the initial estimates are set to zero. Still, ALG 1 produces a position estimate with error less than one wavelength ($\lambda \approx 0.2$ m for L_1) within a few seconds.

Fig. 3 shows the position RMSE for $N_{\mathrm{MC}} = 100$ Monte-Carlo executions. The convergence is faster for Algorithm 1 compared to EKF. Furthermore, the lower bounds set by EKFIDEAL are attained within a few epochs.

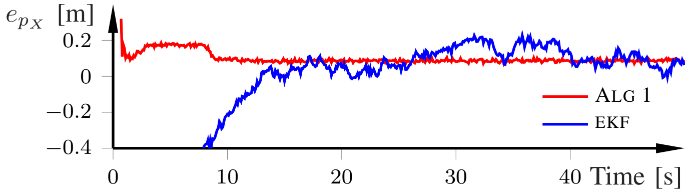

Fig. 1. Position errors of the longitudinal position for a 50 s excerpt from one realization for the numerical analysis in Sec. IV-A.

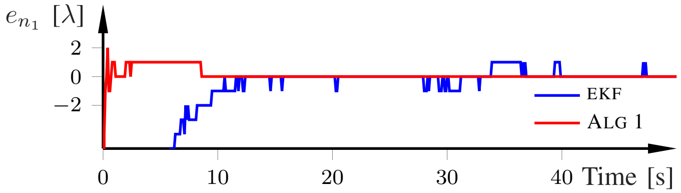

Fig. 2. Ambiguity errors for one of the satellites corresponding to Fig. 1. The true ambiguity is $n_1 = -220$ and the initial ambiguity estimate is set to zero. The correct ambiguity is attained within 8 s in this realization for our proposed method (red).

B. Cycle-Slip

Fig. 4 displays the ambiguities and Fig. 5 shows the corresponding position error for the longitudinal position when a cycle-slip in the satellites occur, one at a time. The proposed method (ALG 1) detects the cycle-slip almost immediately owing to the proposal used in the particle filter, and the filter tracks the correct ambiguities throughout. Moreover, the results also indicate one of the shortcomings with a two-stage approach where both rover states and ambiguities are estimated with the EKF. When there is a cycle slip in n_1, EKF estimates for n_2 and n_3 deviate from the true values, with similar behavior for the other cycle-slip occasions. The reason for this unwanted behavior is the coupling in the covariance matrix amongst the different ambiguities, which gives a Kalman gain that is nonzero in the elements corresponding to the cross terms. This causes the EKF to erroneously correct n_2 and n_3 for changes in n_1. Clearly, from Fig. 5, the position error grows rapidly as a result of this. This behavior is not present in the particle-filter approach, where the marginalization and subsequent handling of the ambiguities by the particle filter decorrelates the effects of the ambiguities of different satellites. Note that although it is not clear from the figure, EKF will eventually correct the positioning, but only after a long transient. In a production implementation, any ambiguity resolution method would be accompanied with add-on detection schemes, such as an explicit cycle-slip detection. In the simulations we have intentionally set the cycle slips to be significant, to clearly see the transient behavior of the two estimators. However, comparing against experimental data at our disposal and reported in the next section, the order of magnitude of the cycle slips is not unrealistic.

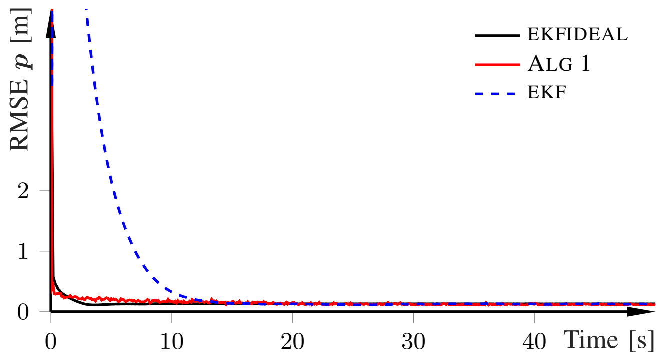

Fig. 3. Position RMSE for 100 Monte-Carlo executions for the numerical analysis in Sec. IV-A. The proposed method (red) reaches steady state faster than EKF due to the fast convergence of the ambiguities, see Fig. 2.

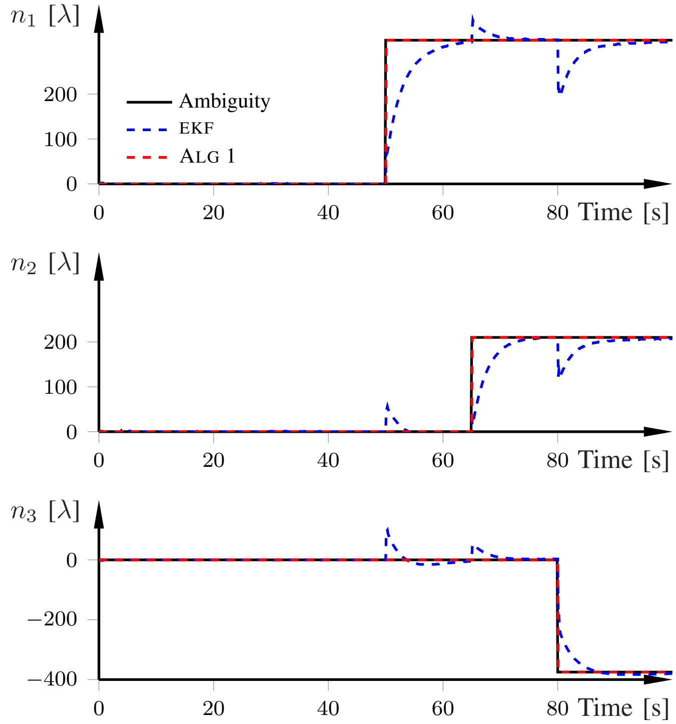

Fig. 4. Ambiguities for single-satellite cycle-slip for the numerical analysis in Sec. IV-B.

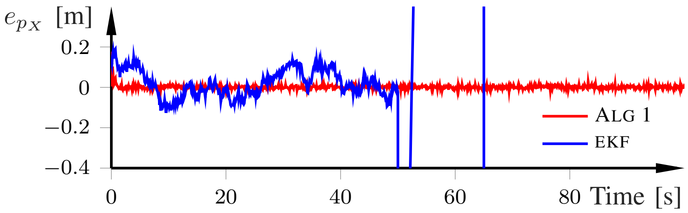

Fig. 5. Position error of the longitudinal position for single-satellite cycle-slip, corresponding to Fig. 4.

C. Computation Time

Fig. 6 shows how the computational requirements scales with the number of satellites used in the estimation. The figure indicates that the method gives real-time feasibility for 12 satellites with the current chosen sampling period ($T_s = 100$ms), and for more satellites than that for a larger discretization time. As the number of satellites grows larger, the computational time becomes increasingly larger. However,

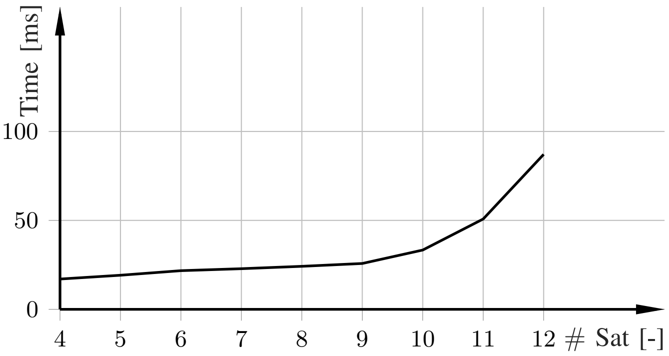

Fig. 6. Average computation time for one iteration of Algorithm 1 (Lines 2–32) for varying number of satellites used in the estimation. The computation times are caluclated on an i7 2.7 GHz laptop with a nonoptimized MATLAB implementation.

there are major speedup possiblities, since the current implementation is in MATLAB and nonoptimized.

V. EXPERIMENTAL EVALUATION

The numerical analysis in Sec. IV shows the performance under ideal conditions, where the model and noise assumptions are correct. To see how the proposed method performs on real data, in this section we present an evaluation from an experimental data set obtained using a low-cost u-blox antenna and receiver and provided by the open-source package GoGPS [28], which is employed in the evaluation. GoGPS is a positioning software application designed to process single-frequency L1 code and phase observations for positioning. GoGPS focuses on the treatment of observations by low-cost receivers, with relative positioning using double-differenced observations with respect to a known reference station, in accordance with (3). Atmospheric delays, which are present in (1), are suppressed with well-established methods [23], [24]. GoGPS contains several algorithms for GNSS positioning, such as single epoch least squares algorithm or recursive EKFs combined with ILS solvers, such as LAMBDA version 2, [6], [7] and MLAMBDA (LAMBDA version 3, [8]). GoGPS handles changes in the satellite configurations, that is, satellite additions/ losses or switching of reference (pivot) satellites used for the double differences, and cycle slips are managed for all EKF modes. GoGPS can apply different observation-weighting strategies; either based on satellite elevation or weight functions that exploit the known signal-to-noise ratio characteristics of low-cost receivers. Furthermore, the software package also includes means to discard satellites with too low of an elevation angle.

A. Preliminaries

Similar to the numerical analysis, we use a constant velocity motion model that we define in geodetic coordinates and use a diagonal covariance matrix $\bar{\boldsymbol{Q}}_{\boldsymbol{x}} \in \mathbb{R}^{3\times 3}$ acting on the velocity according to $\bar{\boldsymbol{Q}}_{\boldsymbol{x}} = \mathrm{diag}(0.5^2, 0.5^2, 0.1^2)$. However, in contrast to the simplified study in Sec. IV, $\bar{\boldsymbol{Q}}_{\boldsymbol{x}}$ is at each time step transformed to the global, Earth-fixed coordinate system in which the estimation is performed, which results in a process-noise covariance matrix with cross-terms. The noise variances for code and phase measurements are $\sigma_\epsilon = 0.4^2$ m and $\sigma_\eta = 0.01^2$ m, respectively, which are weighted according to a model available in GoGPS that is based on the elevation angle of the observations. This results in a time-varying nondiagonal measurement noise matrix $\boldsymbol{R}_k$. Furthermore, a satellite is removed from the measurement equation if the elevation angle is below 10 deg. The data sets are recordings from GPS data, and in one of the data sets there are several additions and losses of satellites, and numerous detection of cycle slips. When a new satellite is added and/or cycle slip is detected, the initial estimate is determined from a least squares initialization given old data. The threshold for determining when to reinitialize the filter bank (Line 9 in Algorithm 1) is set to $\gamma = 10$.

We compare Algorithm 1 with an EKF/MLAMBDA method (EKF), similar to the ones we have implemented for the simulations in Sec. IV, described in [8], [28] and packaged as LAMBDA version 3, as a MATLAB implementation in GoGPS. The implementation internally determines the quality of the integer estimates, by utilizing the residual errors from the real-valued estimate. Depending on the quality of the estimate, EKF occasionally outputs the real-valued estimates instead, together with the corresponding state estimate. We use the same process noise and measurement noise in both EKF and our proposed method, and both methods execute at 1 Hz.

We implement both filters in two versions, one without cycle-slip detection and one including explicit cycle-slip detection. The purpose is to show the respective method's ability to handle cycle slip intrinsically, compared with aiding the estimators with explicit cycle-slip detection. For both methods, we utilize a standard cycle-slip detection scheme available in GoGPS in determining which ambiguities need to be reinitialized. In theory, our method intrinsically handles cycle-slip detection and such a detection scheme is therefore not needed when the assumptions of the estimator hold, as validated in simulation in Sec. IV-B. However, explicit cycle-slip detection may still be essential to incorporate in a real-world experiment where various modeling errors are present, which may violate the assumptions of the algorithm. Essentially, the employed cycle-slip detection scheme in GoGPS is based on comparing the double-differenced code and phase observations and estimating an ambiguity from that estimate,

$$\nabla\Delta n^{lj} = \frac{\nabla\Delta\Phi^{lj} - \nabla\Delta P^{lj}_{br,k}}{\lambda}. \tag{36}$$

A cycle slip is detected if the estimated ambiguity (36) differs more than a predefined number of cycles from the estimate at the previous time step. In this paper, the threshold is set to 10 cycles, that is, roughly corresponding to a 2 m error between the estimated position and the position inferred from the phase observation assuming constant ambiguity. It is worth noting the the algorithm proposed here does not pose restrictions on the cycle-slip detection method used, and that methods other than (36) are applicable.

B. Experimental Results

Fig. 7 shows the estimated path for EKF without cycle-slip detection (red dashed), Algorithm 1 without cycle-slip detection (black dashed), EKF with cycle-slip detection (red solid), and Algorithm 1 with cycle-slip detection (black solid). The results are with respect to a known base station. The true path is rectangular, with the initial estimate indicated by a green cross. We do not have ground-truth coordinates for the data set. However, it is still possible to draw a few conclusions. When investigating the 2D path, it is clear that Algorithm 1 exhibits a smoother path compared to the EKF/MLAMBDA method for the respective variants. The path for EKF without cycle-slip detection is largely biased, and the path is nonsmooth and shows jumps, indicating unreliable estimates. Algorithm 1 without cycle-slip detection (i.e., as it is described in Algorithm 1) is much smoother, although there are a few glitches in its performance, most notably in the upper right corner. The implementations with proper cycle-slip detection naturally produce more reliable estimates for both methods. However, EKF exhibits several large jumps between two consecutive time steps, indicating that the cycle-slip detection cannot fully compensate for the effects due to the cycle slip.

From the 3D path (right plot), Algorithm 1 exhibits considerably less variations in Z-estimates, even without cycle-slip detection. For instance, EKF without cycle-slip detection varies about 40 m, EKF with cycle-slip detection differs about 5 m, our proposed method without cycle-slip detection varies with roughly 2 m, and the proposed method with cycle-slip detection stays within 0.5 m.

Fig. 8 displays the ambiguities corresponding to the estimated path in Fig. 7 and positions in Fig. 9 for the implementations with add-on cycle-slip detection. In this experiment, there are seven satellites visible to the receiver for most of the experiment, resulting in six ambiguities to estimate. For Algorithm 1, the set of possible ambiguities for each satellite is indicated by the gray area. There are numerous detected cycle slips for the fourth satellite and one cycle slip for the eighth satellite (at $t = 115$ s). Except for the time instants when cycle slips are detected, the ambiguities are mostly constant for our approach, whereas the estimates produced by EKF for most of the satellites fluctuate more. In Fig. 8, one can see the effects of the correlation in the covariance matrix between receiver state and ambiguities in EKF. For example, at approximately $t = 45$ s, there is a cycle-slip detection for satellite 4. However, this also leads to large jumps in the ambiguity estimates for some of the other satellites (e.g., satellites 7, 13, and 25). Although there are at times also changes in the ambiguity estimates for Algorithm 1 (see Satellite 10), these are often minor adjustments (1–2 cycles) except for when a cycle slip is detected.

Fig. 9 displays the distance between estimated positions for consecutive time steps, which gives a measure of the jumps that the different estimators produce The distances in Fig. 9 correspond to the 3D path in Fig. 7. The true path is generated by an approximately constant velocity, except for a ramp-up from zero velocity in the beginning of the experiment. Hence, ideally the distances between two consecutive position estimates should be approximately constant throughout. The time instants of the detected cycle slips are indicated by the vertical dashed lines. Since, according to (36), the detection of cycle slip is dependent on the current estimate of the ambiguity, different algorithms generally have different detections. We therefore plot all of the detected cycle slips for the two algorithms, which overlap for 9 out of the 12 cycle-slip detections.

The impact the cycle slips have on the estimation performance can be seen in the large jumps in the estimates in Fig. 9 for EKF, that often occur at the time of the cycle slips. The detection of cycle slip most often leads to much improved position estimates for EKF. However, as discussed in relation to the results in Fig. 4, the correlation between the positions and ambiguities introduced by considering the ambiguities as additional states in EKF causes changes in ambiguities for satellites where a cycle slip does not occur. This leads to jumps in the estimates and hence decreased estimation performance. Furthermore, when comparing Figs. 8 and 9, it is clear that there are many jumps in the position estimates larger than 1 m for EKF, even with add-on cycle-slip detection (red solid) that are caused by the fluctuations in ambiguity estimates. For Algorithm 1 without cycle-slip detection (black dashed in Fig. 9), for most of the time, similar estimates as when using explicit cycle-slip detection are obtained, and there are only 4 occasions of a difference between two consecutive estimates larger than 1 m, which is a large improvement compared to EKF. For Algorithm 1 with explicit cycle-slip detection, there is only one time instant when the position estimate jumps more than 1 m.

The number of ambiguity combinations (i.e., the number of EKFs) needed in the method is shown in Fig. 10. After the initial transients, the number of combinations is around 729 (3^6, i.e., 3 possible ambiguities for each of the 6 satellites), which results in a computation time of about 50 Hz in a pure Matlab-code implementation (i.e., 50 times faster than real time). For a closer account of the scaling of the computational requirements with the number of satellites, see Fig. 6.

VI. Conclusion

We addressed the GNSS positioning and ambiguity resolution problem by proposing a method based on particle filtering and mixture KFs. By modeling the ambiguities as part of the nonlinear state in a marginalized approach, we can use proposal sampling to guide the ambiguities to their statistically correct values and fix the ambiguities in such a way that the set of possible integer values contains the true ambiguities. This allows for directly finding the integer ambiguities from a mixture of KFs.

The numerical analysis showed that the method handles cycle slip automatically, and that the ambiguity estimation is error free, resulting in improved tracking performance over conventional algorithms. Furthermore, cross-correlation between the ambiguities, which is apparent in Kalman-type methods and leads to periods of degraded estimation performance, is avoided.

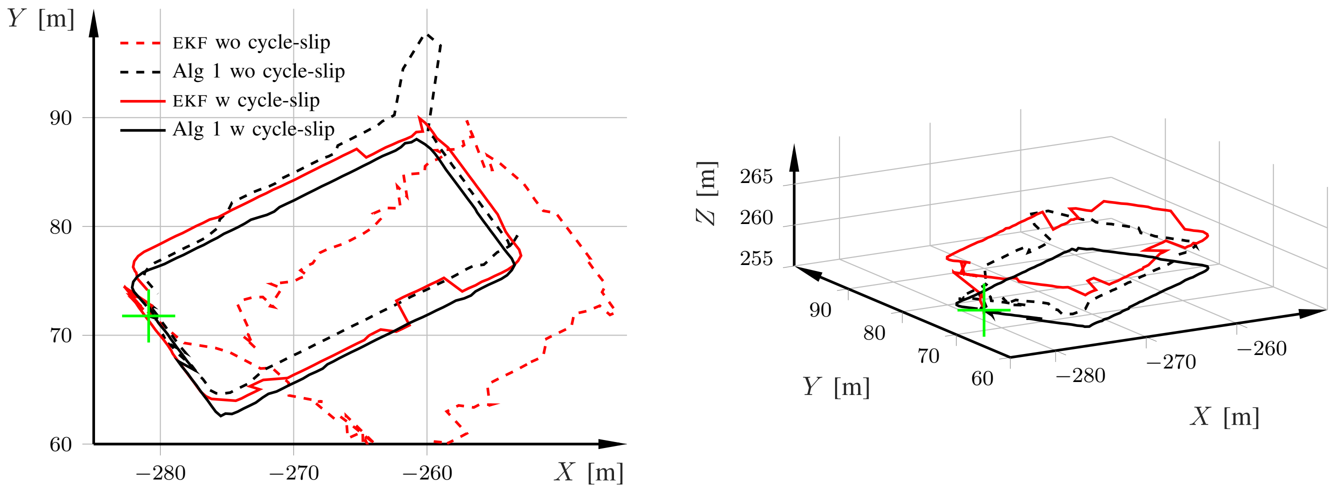

Fig. 7. Projection onto the 2D-plane (left plot) and three-dimensional path (right plot) estimated by EKF without cycle-slip detection (red dashed), Algorithm 1 without cycle-slip detection (black dashed), EKF with cycle-slip detection (red solid), and Algorithm 1 with cycle-slip detection (black solid), using experimental data available in the goGPS package [28]. The coordinates are with respect to a known reference station. The path for EKF without cycle-slip detection is not shown in the 3D plot because of its large variations in the Z estimates. The green plus indicates the initial estimate (common for all filters).

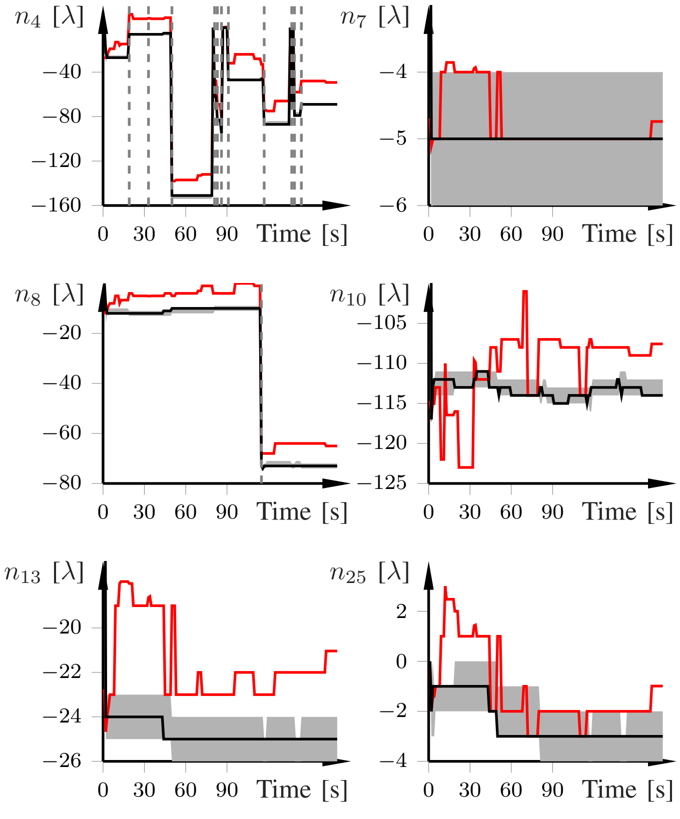

Fig. 8. Double-differenced ambiguity estimates for the satellites visible at any during the experiment, using experimental data available in the goGPS package [28]. The ambiguity estimates from Algorithm 1 are in black, the corresponding set of possible ambiguities for each satellite is indicated by the gray filled area, EKF in red, and the cycle-slip detections are indicated by the vertical dashed lines (there are only cycle slips detected for satellite 4 and 8). The estimated ambiguities correspond to the algorithm implementations with add-on cycle-slip detection as determined according to (36).

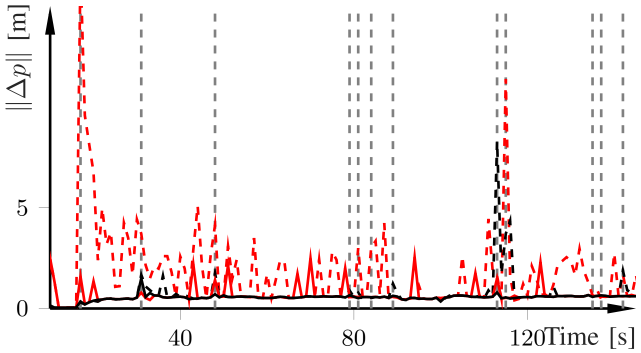

Fig. 9. Distance between two consecutive receiver position estimates as a function of time, with same color scheme as in Fig. 7, using experimental data available in the goGPS package [28]. The occurrences of cycle slips are indicated by the gray vertical dashed lines.

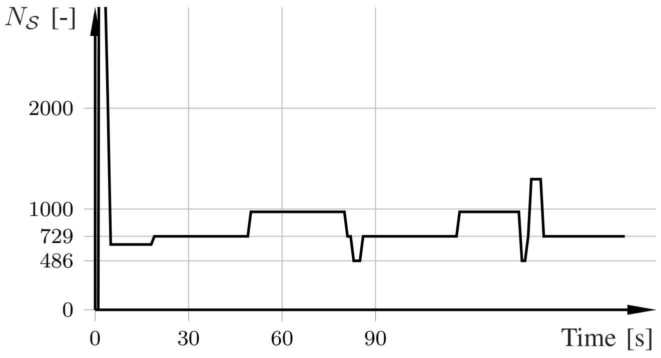

Fig. 10. The adaptation of the number of ambiguity combinations $N_{\mathcal{S}}$ from one realization, using experimental data available in the goGPS package [28] corresponding to the results in Fig. 8 (Example 2).

The experimental results carried out using publicly available data in the GoGPS package indicate that our method gives reliable and smooth tracking, and the ambiguity estimates are robust to cycle slip. In particular, for situations when there are major external disturbances on the measurement signals, such as in urban areas where the reception is poor, the proposed method shows significantly improved robustness, owing to its capabilities to handle multiple ambiguity hypotheses simultaneously.

References

[1] G. Xu and Y. Xu, *GPS: theory, algorithms and applications*. Springer, 2016.

[2] P. J. G. Teunissen, "A new method for fast carrier phase ambiguity estimation," in *Position Location and Navigation Symposium*, 1994.

[3] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System: theory and practice*. Springer-Verlag, 1997.

[4] P. Teunissen, "Theory of carrier phase ambiguity resolution," *Wuhan University Journal of Natural Sciences*, vol. 8, no. 2, p. 471, 2003.

[5] S. Zhao, X. Cui, F. Guan, and M. Lu, "A Kalman filter-based short baseline RTK algorithm for single-frequency combination of GPS and BDS," *Sensors*, vol. 14, no. 8, pp. 15 415–15 433, 2014.

[6] P. J. Teunissen, "The least-squares ambiguity decorrelation adjustment: a method for fast GPS integer ambiguity estimation," *J. Geodesy*, vol. 70, no. 1, pp. 65–82, 1995.

[7] P. D. Jonge and C. Tiberius, "The LAMBDA method for integer ambiguity estimation: implementation aspects," *Delft Geodetic Computing Centre LGR Series*, vol. 12, 1996.

[8] X.-W. Chang, X. Yang, and T. Zhou, "MLAMBDA: a modified LAMBDA method for integer least-squares estimation," *J. Geodesy*, vol. 79, no. 9, pp. 552–565, 2005.

[9] A. Hassibi and S. Boyd, "Integer parameter estimation in linear models with applications to GPS," *IEEE Trans. Signal Process.*, vol. 46, no. 11, pp. 2938–2952, 1998.

[10] E. W. Grafarend, "Mixed integer-real valued adjustment (IRA) problems: GPS initial cycle ambiguity resolution by means of the LLL algorithm," *GPS Solutions*, 2000.

[11] T. B. Schön, F. Gustafsson, and P.-J. Nordlund, "Marginalized particle filters for mixed linear nonlinear state-space models," *IEEE Trans. Signal Process.*, vol. 53, pp. 2279–2289, 2005.

[12] B. Betti, M. Crespi, and F. Sanso, "A geometric illustration of ambiguity resolution in GPS theory and a Bayesian appraoch," *Manuscr. Geod.*, vol. 18, pp. 317–330, 1993.

[13] B. Gundlich and K.-R. Koch, "Confidence regions for GPS baselines by Bayesian statistics," *J. Geodesy*, vol. 76, no. 1, pp. 55–62, 2002.

[14] B. Gundlich and P. Teunissen, "Multiple models–fixed, switching, interacting," in *nternational Association of Geodsey Symposia. Volume 127. V Hotine-Marussi Symposium on Mathematical Geodesy*, 2004.

[15] R. G. Brown and P. Y. Hwang, "A Kalman filter approach to precision GPS geodesy," *Navigation*, vol. 30, no. 4, pp. 338–349, 1983.

[16] P. E. Henderson, J. F. Raquet, and P. S. Maybeck, "A multiple filter approach for precise kinematic DGPS positioning and carrier-phase ambiguity resolution," *Navigation*, vol. 49, no. 3, pp. 149–160, 2002.

[17] J. D. Wolfe, W. R. Williamson, and J. L. Speyer, "Hypothesis testing for resolving integer ambiguity in GPS," *Navigation*, vol. 50, no. 1, pp. 45–56, 2003.

[18] B. Azimi-Sadjadi and P. Krishnaprasad, "Integer ambiguity resolution in GPS using particle filtering," in *Amer. Control Conf.*, Arlington, VA, Jun. 2001.

[19] S. S. Hwang and J. L. Speyer, "Relative GPS carrier-phase positioning using particle filters with position samples," in *Amer. Control Conf.*, St. Louis, MO, Jun. 2009.

[20] ——, "Particle filters with adaptive resampling technique applied to relative positioning using GPS carrier-phase measurements," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 6, pp. 1384–1396, 2011.

[21] M. Sahmoudi and R. Landry, "A nonlinear filtering approach for robust multi-GNSS RTK positioning in presence of multipath and ionospheric delays," *IEEE J. Sel. Topics Signal Process.*, vol. 3, no. 5, pp. 764–776, 2009.

[22] G. Blewitt, "Basics of the GPS technique: observation equations," *Geodetic Applications of GPS*, pp. 10–54, 1997.

[23] J. Klobuchar, "Ionospheric time-delay algorithm for single-frequency GPS users," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 3, no. 23, pp. 325–331, 1987.

[24] J. Saastamoinen, "Contributions to the theory of atmospheric refraction," *Bull. Géodésique*, vol. 105, no. 1, pp. 279–298, 1973.

[25] F. Gustafsson, *Statistical Sensor Fusion*. Lund, Sweden: Utbildningshuset/Studentlitteratur, 2010.

[26] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.

[27] C. M. Bishop, *Pattern Recognition and Machine Learning*. NJ, USA: Springer-Verlag New York, 2006.

[28] A. M. Herrera, H. F. Suhandri, E. Realini, M. Reguzzoni, and M. C. de Lacy, "goGPS: open-source MATLAB software," *GPS Solutions*, vol. 20, no. 3, pp. 595–603, 2016.