

GNSS-RTK Factor Graph Optimization with Adaptive Ambiguity Noise

Hu, Yingjie; Di Cairano, Stefano; Berntorp, Karl

TR2025-102 July 09, 2025

Abstract

This paper proposes a graph optimization-based real-time kinematic global navigation satellite system (GNSS) positioning approach, which consists of two stages of factor graph optimization (FGO). The first stage computes float solutions of navigation states including the carrier phase integer ambiguities, where we characterize the time evolution of integer ambiguities with an adaptive ambiguity model to accommodate cycle slips. By exploring the time-correlated constraint inherent in the integer ambiguity evolution, we achieve integer fixation with higher accuracy. The second-stage FGO takes the solutions from the first stage as prior and performs another graph optimization to obtain the fixed solutions of positions and velocities. Monte Carlo simulation results demonstrate that our proposed approach can achieve statistically smaller root mean square error in position estimates compared to Kalman filter-based method and is more robust to cycle slips.

American Control Conference (ACC) 2025

© 2025 MERL. This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

GNSS-RTK Factor Graph Optimization with Adaptive Ambiguity Noise

Yingjie Hu, Stefano Di Cairano, Karl Berntorp*

Abstract—This paper proposes a graph optimization-based real-time kinematic global navigation satellite system (GNSS) positioning approach, which consists of two stages of factor graph optimization (FGO). The first stage computes float solutions of navigation states including the carrier phase integer ambiguities, where we characterize the time evolution of integer ambiguities with an adaptive ambiguity model to accommodate cycle slips. By exploring the time-correlated constraint inherent in the integer ambiguity evolution, we achieve integer fixation with higher accuracy. The second-stage FGO takes the solutions from the first stage as prior and performs another graph optimization to obtain the fixed solutions of positions and velocities. Monte Carlo simulation results demonstrate that our proposed approach can achieve statistically smaller root mean square error in position estimates compared to Kalman filter-based method and is more robust to cycle slips.

I. INTRODUCTION

Global navigation satellite system (GNSS) using real-time kinematic (RTK) is a positioning technology that holds promise to provide centimeter-level position accuracy. In the RTK technology, a GNSS receiver tracks the carrier-phase measurements, which are the difference between the phases of the receiver-generated carrier signal and the satellite carrier signal received at the time of measurement [1]. The carrier phase can be tracked with great precision and the receiver-satellite distance is equal to a certain number of whole cycles plus the measured fractional cycle. However, the number of cycles of the carrier wave during the transit between satellite and receiver, referred to as integer ambiguity, as it is an integer multiple of the wavelength, is unknown.

Thus, accurate computation of the integer ambiguity is a crucial step of GNSS-RTK. In fact, after the initial ambiguity is resolved, cycle slips may occur, which are caused by a loss-of-lock of the carrier signal tracking and result in a sudden jump in the integer ambiguities. This is especially frequent in dynamic environments, e.g., in urban settings. Each time a cycle slip occurs, the RTK algorithm needs to re-estimate the integer ambiguity. Conventional filtering-based Kalman-type approaches estimate the GNSS receiver states using the observations at the current epoch [2], [3]. Alternatively, optimization-based approaches have been extensively studied to solve for estimation problems [4], [5]. Factor graph optimization (FGO) [6] is a nonlinear optimization framework based on maximum *a posteriori* estimation (MAP) that has been recently applied to GNSS positioning [7], [8]. FGO-based approaches solve for the optimal state trajectory

based on a batch of historic measurement data. As opposed to conventional approaches, FGO can leverage the historic information inherent in the observation batch and better exploit the time correlation between observations through different epochs. Thus, FGO-based GNSS positioning methods using pseudorange (code) measurements have shown improved performance and robustness relative to extended Kalman filters (EKFs) in challenging urban environments [8].

Several initial application of FGO to GNSS-RTK have been proposed [9]–[14]. However, these approaches still have challenges with respect to ambiguity determinations as they either do not include ambiguity resolution [10], or they assume the ambiguities to be constants during the FGO batch [9], [11], [12], [14].

To address the issue of recurrent cycle slips in GNSS-challenged, dynamic environments, we propose an FGO-based GNSS RTK positioning algorithm in this paper that can accommodate the frequent cycle slips without assuming constant integer ambiguities. Specifically, the proposed algorithm consists of two stages of FGO. The first-stage FGO is responsible for the float solutions of the state estimates including integer ambiguities. To account for the frequent cycle slips, we model the time evolution of integer ambiguities as a random walk process with adaptive noise to reflect the cycle slips. When cycle slips are detected, the variance of the ambiguity process noise will be inflated to reflect the potential integer jump. This adaptive ambiguity model is incorporated as factors into the first-stage FGO, which enables the optimization to leverage the time correlation in the integer ambiguity evolution despite the existence of cycle slips. Float solutions of GNSS receiver states and integer ambiguities can be obtained from the first-stage FGO. Integer fixation approaches, e.g., the LAMBDA method [15], can be used to obtain the integer solutions for the ambiguity variables. The resolved integer ambiguities will be used in the second-stage FGO to solve for the fixed solution of GNSS receiver states. A numerical simulation study is implemented to validate the efficacy of our proposed algorithm. Monte Carlo simulation results show that our proposed method is able to achieve smaller root-mean-square-errors (RMSE) in position estimates than both FGO without the adaptive ambiguity model and the mixed-integer filter-based GNSS RTK approach that we have previously proposed [2], [16].

Notation: \mathbb{R}^n denotes the set of real-valued vectors of dimension $n \times 1$. \mathbb{Z}^n represents the set of integer vectors of dimension $n \times 1$. $\mathbf{1}$ denotes a column vector of ones. \mathbf{I} is identity matrix. The notation $\|\mathbf{v}\|_{\mathbf{P}}^2 = \mathbf{v}^T \mathbf{P}^{-1} \mathbf{v}$ represents the Mahalanobis norm, in which $\mathbf{v} \in \mathbb{R}^n$ and \mathbf{P} is a $n \times n$ matrix. $\mathbf{x} \sim \mathcal{N}(\mathbf{x}|\bar{\mathbf{x}}, \Sigma)$ denotes that random variable \mathbf{x} is

Yingjie Hu (hu000258@umn.edu) is with the Department of Aerospace Engineering and Mechanics at the University of Minnesota, Twin Cities. Stefano Di Cairano (dicairano@merl.com) and Karl Berntorp (karl.o.berntorp@ieee.org) are with Mitsubishi Electric Research Labs (MERL), Cambridge, MA, USA.

Gaussian distributed with mean $\bar{\mathbf{x}}$ and covariance matrix Σ . $p_{\mathbf{x}}(\mathbf{x})$ represents the probability density distribution (PDF) of random variable \mathbf{x} and the subscript of x is often omitted for brevity.

II. PROBLEM SETUP

To simplify the exposition, we assume GNSS code and carrier phase measurements and ignore Doppler measurements. Consider the following code and carrier phase measurement model from satellite s to receiver r at time t_k ,

$$\begin{aligned}\rho_{r,k}^s &= r_{r,k}^s + c(\delta t_{r,k} - \delta t_k^s) + I_{r,k}^s + T_{r,k}^s + \epsilon_{r,k}^s \\ \lambda\phi_{r,k}^s &= r_{r,k}^s + c(\delta t_{r,k} - \delta t_k^s) - I_{r,k}^s + T_{r,k}^s + \lambda N_{r,k}^s + \eta_{r,k}^s\end{aligned}\quad (1)$$

where $\rho_{r,k}^s$ is the code measurement, $\phi_{r,k}^s$ is the carrier phase measurement, and $r_{r,k}^s = \|\mathbf{p}_{r,k} - \mathbf{p}_{s,k}\|$ is the geometric range between receiver r and satellite s . Symbols $\mathbf{p}_{r,k}, \mathbf{p}_{s,k} \in \mathbb{R}^3$ denote the receiver and satellite positions, c is the speed of light, $\delta t_{r,k}$ is the receiver clock bias, and δt_k^s is the satellite clock bias. The term $I_{r,k}^s$ denotes the ionospheric delay, $T_{r,k}^s$ is the tropospheric delay, and λ represents the carrier wavelength. Furthermore, $N_{r,k}^s$ represents the carrier phase integer ambiguity, and $\epsilon_{r,k}^s$ and $\eta_{r,k}^s$ denote the measurement errors, generally caused by multipath effects, NLOS receptions, receiver noise, and antenna delay [10]. $\epsilon_{r,k}^s$ and $\eta_{r,k}^s$ are assumed to be white Gaussian noise, the variance of which depends on the satellite elevation.

Usually, a SD operation is performed between the measurements of receiver r and base b to remove common-mode errors (e.g., ionosphere, troposphere, and satellite clock) within a local vicinity [17],

$$\begin{aligned}\Delta\rho_{br,k}^s &= r_{b,k}^s - r_{r,k}^s + c(\delta t_{b,k}^s - \delta t_{r,k}^s) + \Delta\epsilon_{br,k}^s \\ \lambda\Delta\phi_{br,k}^s &= r_{b,k}^s - r_{r,k}^s + c(\delta t_{b,k}^s - \delta t_{r,k}^s) + \lambda\Delta N_{br,k}^s + \Delta\eta_{br,k}^s\end{aligned}\quad (2)$$

Next, a second difference between the reference (pivot) satellite o and satellite s is performed to remove the receiver clock bias [18]. The DD code and carrier phase measurements can be expressed as

$$\begin{aligned}\nabla\Delta\rho_{br,k}^{so} &= \nabla\Delta r_{br,k}^{so} + \nabla\Delta\epsilon_{br,k}^{so} \\ \lambda\nabla\Delta\phi_{br,k}^{so} &= \nabla\Delta r_{br,k}^{so} + \lambda\nabla\Delta N_{br,k}^{so} + \nabla\Delta\eta_{br,k}^{so}\end{aligned}\quad (3)$$

where $\nabla\Delta r_{br,k}^{so} = (r_{b,k}^s - r_{r,k}^s) - (r_{b,k}^o - r_{r,k}^o)$ and $\nabla\Delta N_{br,k}^{so} = \Delta N_{br,k}^s - \Delta N_{br,k}^o$ is the DD ambiguity. If M pairs of DD code and carrier phase measurements are available at time t_k ($M+1$ satellites o, s_1, s_2, \dots, s_M are visible), we can form the DD code and carrier phase measurement equations

$$\begin{aligned}\mathbf{y}_{\rho,k} &= \mathbf{h}(\mathbf{x}_k) + \boldsymbol{\epsilon}_k \\ \mathbf{y}_{\phi,k} &= \mathbf{h}(\mathbf{x}_k) + \lambda\mathbf{n}_k + \boldsymbol{\eta}_k\end{aligned}\quad (4)$$

where the receiver state \mathbf{x}_k includes at least the position and velocity of the receiver $\mathbf{x}_k = [\mathbf{p}_{r,k}^\top \ \mathbf{v}_{r,k}^\top]^\top$, $\mathbf{n}_k \in \mathbb{Z}^M$ is the DD carrier phase integer ambiguity vector, $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\boldsymbol{\epsilon}_k | \mathbf{0}, \mathbf{R}_{\rho,k})$ and $\boldsymbol{\eta}_k \sim \mathcal{N}(\boldsymbol{\eta}_k | \mathbf{0}, \mathbf{R}_{\phi,k})$, and

$$\begin{aligned}\mathbf{h}(\mathbf{x}_k) &= [\nabla\Delta r_{br,k}^{s_1 o} \ \dots \ \nabla\Delta r_{br,k}^{s_M o}]^T \\ \mathbf{n}_k &= [\nabla\Delta N_{br,k}^{s_1 o} \ \dots \ \nabla\Delta N_{br,k}^{s_M o}]^T \\ \boldsymbol{\epsilon}_k &= [\nabla\Delta\epsilon_{br,k}^{s_1 o} \ \dots \ \nabla\Delta\epsilon_{br,k}^{s_M o}]^T \\ \boldsymbol{\eta}_k &= [\nabla\Delta\eta_{br,k}^{s_1 o} \ \dots \ \nabla\Delta\eta_{br,k}^{s_M o}]^T.\end{aligned}\quad (5)$$

In this work, a generic linear model is used to model the motion of the receiver,

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{w}_k \quad (6)$$

where \mathbf{F}_k is the state transition matrix and \mathbf{B}_k denotes the process noise matrix. In the numerical evaluation in Sec. IV, we use the constant-velocity (CV) model [19] as follows to characterize the receiver motion.

$$\mathbf{x}_{k+1} = \begin{bmatrix} \mathbf{I} & dt\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} \frac{dt^2}{2}\mathbf{I} \\ dt\mathbf{I} \end{bmatrix} \mathbf{w}_k \quad (7)$$

where dt is the sampling period and \mathbf{w}_k is the process noise $\mathbf{w}_k \sim \mathcal{N}(\mathbf{w}_k | \mathbf{0}, \mathbf{Q}_k)$. Note that our proposed methodology is not limited to linear motion models and more sophisticated models can be used for the receiver dynamics and here CV model is only used for simplicity.

III. METHODOLOGY

This section presents the proposed GNSS-RTK positioning algorithm based on FGO. We incorporate an adaptive ambiguity factor into the factor-graph formulation to accommodate frequent cycle slips, making the ambiguity resolution more robust to cycle slips without requiring data with constant ambiguities. In the exposition we use DD GNSS raw measurements, but the approach applies also to other measurements, such as SD measurements.

A. Adaptive Ambiguity Model

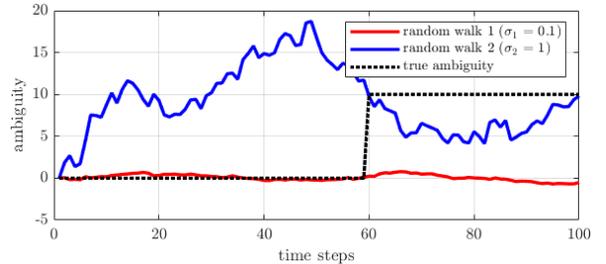


Fig. 1. An example of random walk processes and an integer ambiguity evolution

The integer ambiguity remains constant during continuous tracking of carrier signal. However, GNSS signals being momentarily blocked or shaded will cause the loss-of-lock of signal tracking, which results in a sudden jump in integer ambiguity. For instance, Fig. 1 shows a scenario where a cycle slip causes a jump from 0 to 10 in the integer ambiguity (black) at time step $k = 60$. It is challenging to properly model the ambiguity dynamics as ambiguities often remain constant over periods of time but the jumps caused by cycle slips are abrupt and usually unpredictable. A common

approach in the literature is to model the ambiguity dynamics by a random walk process with some predefined process noise [20]. As shown in Fig. 1, for the first 60 time steps, the ambiguity remains constant and the random walk 1 (red) with a smaller standard deviation better models the constant ambiguity than random walk 2 (blue) with a greater standard deviation. The ambiguity value jumps to 10 at $k = 60$. Clearly, the random walk 1 does not capture such events, whereas the larger noise for random walk 2 can capture the changed ambiguity after such large jumps. After $k = 60$, the ambiguity stays constant, and random walk with smaller noise becomes more suitable again. This example indicates that the ambiguity dynamics can be effectively modeled by a random walk with small noise if there are no cycle slips, and large noise at the time of the jumps.

Therefore, to reflect the jumps in the integer ambiguities during cycle slips, the dynamics of the float ambiguity $\hat{\mathbf{n}}_k \in \mathbb{R}^M$ is modeled by the following random walk process

$$\tilde{\mathbf{n}}_{k+1} = \tilde{\mathbf{n}}_k + \boldsymbol{\nu}_k \quad (8)$$

where $\boldsymbol{\nu}_k$ is the adaptive process noise [16], $\boldsymbol{\nu}_k \sim \mathcal{N}(\boldsymbol{\nu}_k | \mathbf{0}, \mathbf{V}_k)$. If there are no cycle slips from time step k to $k+1$, the uncertainty of $\boldsymbol{\nu}_k$ should be fairly small such that the ambiguity $\tilde{\mathbf{n}}_{k+1}$ remains stable. When cycle slips happen from time step k to $k+1$, uncertainty of $\boldsymbol{\nu}_k$ should be inflated to reflect the sudden jumps on the ambiguity values. As such, cycle slips can be accounted for in this adaptive random walk model. Here we choose to model the float ambiguity dynamics instead of the integer ambiguity dynamics because $\boldsymbol{\nu}_k \in \mathbb{R}^M$.

The adaptive ambiguity model depends on cycle slip detection to correctly adjust the uncertainty of $\boldsymbol{\nu}_k$. We first introduce the cycle slip indicator vector $\mathbf{c}_k \in \mathbb{Z}^M$ to indicate the occurrence of cycle slips. The i -th entry of \mathbf{c}_k is determined by

$$\mathbf{c}_k(i) = \begin{cases} 1 & \text{cycle slip from } \mathbf{n}_k(i) \text{ to } \mathbf{n}_{k+1}(i), \\ 0 & \text{no cycle slip from } \mathbf{n}_k(i) \text{ to } \mathbf{n}_{k+1}(i). \end{cases} \quad (9)$$

The process noise can be made adaptive to reflect to the occurrence of cycle slips by adjusting the variance \mathbf{V}_k based on the indicator vector \mathbf{c}_k . We use the method mentioned in [16] to perform cycle slip detection and then construct the indicator vector \mathbf{c}_k . The covariance matrix of the adaptive process noise $\boldsymbol{\nu}_k$ can be computed as follows.

$$\mathbf{V}_k = \text{diag}(\mathbf{c}_k \cdot \sigma_{\text{jump}}^2 + (\mathbf{1} - \mathbf{c}_k) \cdot \sigma_{\text{stay}}^2) \quad (10)$$

where σ_{jump}^2 denotes the noise variance in presence of cycle slips and theoretically it should reflect the possible range of integer jumps. σ_{stay}^2 characterizes the uncertainty of float ambiguity evolution when no cycle slips are detected. The values of σ_{jump} and σ_{stay} are determined empirically. The selection of σ_{jump} and σ_{stay} will be discussed in the Sec. IV.

Next, we develop the factor graph-based GNSS RTK approach with the adaptive ambiguity factor based on the adaptive ambiguity model (8).

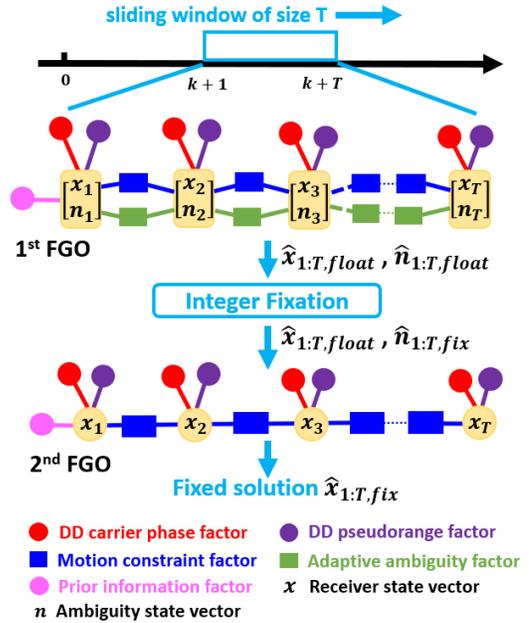


Fig. 2. Architecture of the proposed FGO-based GNSS RTK algorithm. The subscript $1:T$ denotes the indices within the sliding window.

B. FGO-Based GNSS RTK with Adaptive Ambiguity Noise

FGO approaches formulate the MAP estimation problem as a nonlinear graph optimization. Each node of the graph represents a system state and edges of the graph are called factors, encoding information of measurements and constraints of the system. By optimally solving the nonlinear optimization, an *a posteriori* estimate of the state trajectory can be determined, which maximizes the posterior probability of the system states conditioned on a batch of measurements and constraints within the graph. The architecture of our proposed algorithm is a cascade of two FGOs. The first-stage FGO resolves the float solutions of system states. After ambiguity fixation, the second-stage FGO determines the fixed solutions. Fig. 2 shows the structure of our proposed algorithm.

The proposed algorithm is implemented using a sliding window to reduce the computational load. For instance, Fig. 2 shows that the current sliding window of size T spans through time step $k+1$ to $k+T$. Only the system states and measurements within the sliding window will be used for FGO computation. The first-stage FGO first solves for the float solution of receiver states $\hat{\mathbf{x}}_{1:T,\text{float}}$ and ambiguities $\hat{\mathbf{n}}_{1:T,\text{float}}$ within the current sliding window. Next, based on the float solution, we fix the integer ambiguities $\hat{\mathbf{n}}_{1:T,\text{fix}}$, for instance, using the LAMBDA method or variants thereof [21], [22]. The fixed ambiguity estimates are then used in the second FGO to find the optimal receiver states $\hat{\mathbf{x}}_{1:T,\text{fix}}$ conditioned on the fixed integer ambiguities. The fixed solution $\hat{\mathbf{x}}_{T,\text{fix}}$ at the end of the current sliding window is output as the state estimate at time step $k+T$. Then the sliding window moves forward by one step and repeats this process, using the previous solution as prior information. In the remainder of this section, we go through each of the

factors involved in the FGO architecture to formulate the nonlinear optimization.

1) *DD carrier phase factor*: Equation (4) displays the DD code and carrier phase measurements, based on which the error factors for DD code and carrier phase measurements can be established. For the first-stage FGO, DD carrier phase factor can be expressed as

$$\|e_{\phi,i}^I\|_{\mathbf{R}_{\phi,i}}^2 = \|\mathbf{y}_{\phi,i} - \mathbf{h}(\mathbf{x}_i) - \lambda \mathbf{n}_i\|_{\mathbf{R}_{\phi,i}}^2 \quad (11)$$

where the superscript I represents the error factor of the first FGO and subscript i denotes the index within the sliding window. In the second FGO, the fixed ambiguity estimates $\hat{\mathbf{n}}_{1:T,\text{fix}}$ are treated as known information. Consequently, the DD carrier phase factor of the second FGO is

$$\|e_{\phi,i}^{II}\|_{\mathbf{R}_{\phi,i}}^2 = \|\mathbf{y}_{\phi,i} - \mathbf{h}(\mathbf{x}_i) - \lambda \hat{\mathbf{n}}_{i,\text{fix}}\|_{\mathbf{R}_{\phi,i}}^2. \quad (12)$$

2) *DD pseudorange factor*: DD pseudorange (code) measurements are only dependent on the receiver states. Thus, the error factor of DD pseudorange is the same for the first and second FGO,

$$\|e_{\rho,i}\|_{\mathbf{R}_{\rho,i}}^2 = \|\mathbf{y}_{\rho,i} - \mathbf{h}(\mathbf{x}_i)\|_{\mathbf{R}_{\rho,i}}^2. \quad (13)$$

3) *Motion constraint factor*: The receiver dynamics is modeled by the generic linear model in (6), which applies constraints on receiver states between consecutive time steps. Denote the equivalent process noise covariance matrix by $\mathbf{Q}_i = \mathbf{B}_i \mathbf{Q}_i \mathbf{B}_i^T$. Thus, the motion constraint factor is

$$\|e_{x,i}\|_{\mathbf{Q}_i}^2 = \|\mathbf{x}_{i+1} - \mathbf{F}_i \mathbf{x}_i\|_{\mathbf{Q}_i}^2. \quad (14)$$

4) *Adaptive ambiguity factor*: As discussed in Sec. III-A, the time evolution of the ambiguity vector \mathbf{n}_i is modeled as an adaptive random walk according to (8), where the covariance matrix of the random walk noise is adapted to reflect the occurrence of cycle slips. Based on this adaptive ambiguity model, the adaptive ambiguity factor can be constructed and incorporated into the first-stage FGO, which enables the graph optimization to explore the time correlation inherent in the integer ambiguity evolution. The adaptive ambiguity factor can be expressed as

$$\|e_{n,i}\|_{\mathbf{V}_i}^2 = \|\mathbf{n}_{i+1} - \mathbf{n}_i\|_{\mathbf{V}_i}^2. \quad (15)$$

5) *Prior information factor*: The implementation of the sliding window reduces the dimension of the optimization, where the window length is a tradeoff between performance and computational load. However, it limits the use of all the historical data by restricting the graph to observations within the current window. To address this issue, we introduce a prior information factor, which enables the inheritance of historical information in the current sliding window. Fig. 3 depicts the information flow between consecutive sliding windows. We can observe an overlap of system states between the two successive sliding windows. Solutions from the previous sliding window can be utilized as prior information for the current sliding window. Through the use of the prior information factor, historical information is able to propagate across sliding windows.

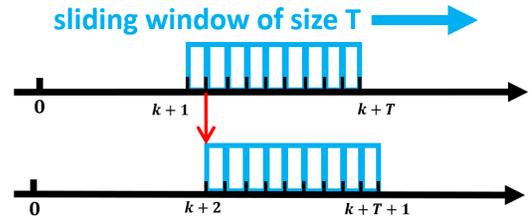


Fig. 3. Illustration of information flow between consecutive sliding windows. The arrow in red denotes that the estimates from previous window serve as prior for current sliding window.

Specifically, the fixed state estimates at the second time step of the previous sliding window is treated as prior information for the current sliding window. The prior information is denoted as $[\hat{\mathbf{x}}_{\text{pri}}^T \ \hat{\mathbf{n}}_{\text{pri}}^T]^T$ with associated covariance matrix \mathbf{P}_{pri} . The state vector at the first time step of the current window is $[\mathbf{x}_1^T \ \mathbf{n}_1^T]^T$. The prior information factor of the first-stage FGO is expressed as

$$\|e_{\text{pri}}^I\|_{\mathbf{P}_{\text{pri}}}^2 = \|\begin{bmatrix} \mathbf{x}_1^T & \mathbf{n}_1^T \end{bmatrix}^T - \begin{bmatrix} \hat{\mathbf{x}}_{\text{pri}}^T & \hat{\mathbf{n}}_{\text{pri}}^T \end{bmatrix}^T\|_{\mathbf{P}_{\text{pri}}}^2. \quad (16)$$

After solving the first-stage FGO of the current sliding window, float solutions $\hat{\mathbf{x}}_{1:T,\text{float}}$ and $\hat{\mathbf{n}}_{1:T,\text{float}}$ are obtained. Then, $\hat{\mathbf{x}}_1$, the receiver state estimate at the first time step of current sliding window, will be utilized as the prior information for the second-stage FGO. The associated covariance matrix is $\mathbf{P}_{1,\text{float}}$, resulting in the prior information factor of the second-stage FGO

$$\|e_{\text{pri}}^{II}\|_{\mathbf{P}_{1,\text{float}}}^2 = \|\mathbf{x}_1 - \hat{\mathbf{x}}_1\|_{\mathbf{P}_{1,\text{float}}}^2. \quad (17)$$

Using the factors (11)–(17), the objective function of the first-stage FGO can be written as

$$\hat{\mathbf{x}}_{1:T,\text{float}}, \hat{\mathbf{n}}_{1:T,\text{float}} = \underset{\mathbf{x}_{1:T}, \mathbf{n}_{1:T}}{\text{argmin}} \|e_{\text{pri}}^I\|_{\mathbf{P}_{\text{pri}}}^2 + \sum_{i=1}^T (\|e_{\phi,i}^I\|_{\mathbf{R}_{\phi,i}}^2 + \|e_{\rho,i}\|_{\mathbf{R}_{\rho,i}}^2) + \sum_{i=1}^{T-1} (\|e_{x,i}\|_{\mathbf{Q}_i}^2 + \|e_{n,i}\|_{\mathbf{V}_i}^2). \quad (18)$$

Similarly, the objective function of the second-stage FGO is

$$\hat{\mathbf{x}}_{1:T,\text{fix}} = \underset{\mathbf{x}_{1:T}}{\text{argmin}} \|e_{\text{pri}}^{II}\|_{\mathbf{P}_{1,\text{float}}}^2 + \sum_{i=1}^T (\|e_{\phi,i}^{II}\|_{\mathbf{R}_{\phi,i}}^2 + \|e_{\rho,i}\|_{\mathbf{R}_{\rho,i}}^2) + \sum_{i=1}^{T-1} \|e_{x,i}\|_{\mathbf{Q}_i}^2. \quad (19)$$

C. Algorithm Summary

Algorithm 1 outlines the proposed algorithm. The sliding window length increments by 1 at each time step until the current time step k reaches the predefined window size T (Line 6-10). After that, sliding window starts moving forward (line 12-17). In line 6-7 and 13-14, the adaptive ambiguity covariance is determined for the adaptive ambiguity factors in the FGO formulation. In order to accurately adapt the

covariance \mathbf{V}_k to cycle slips, the state estimates from previous sliding window are recursively utilized in detecting cycle slips for current sliding window (line 6 and 13). although initial state errors can cause incorrect detection of cycle slips, the receiver state errors are mitigated as more observation data becomes available. The improved state estimates are then used to update the cycle slip detection, which can, in turn, enhance the accuracy of state estimates. This is discussed in Sec. IV-D. In line 8-10 and 15-17, the fixed solutions of state estimates and integer ambiguities are obtained by solving the two-stage FGO.

Algorithm 1 Proposed Algorithm

```

1: Define total time length  $L$ , sliding window size  $T$ 
2: Set initial state estimate  $\hat{\mathbf{x}}_0, \hat{\mathbf{n}}_0$  and covariance  $\mathbf{P}_0$ 
3: for  $k = 1, \dots, L$  do
    // Receive new measurements
4:   Receive:  $t_k, \mathbf{y}_k$ 
5:   if  $k \leq T$  then
    // Cycle slip detection
6:      $\{\mathbf{c}_i\}_{i=0}^{k-1} \leftarrow \text{cycleSlipDetection}(\{\mathbf{y}_i\}_{i=0}^k, \{\hat{\mathbf{x}}_{i,\text{fix}}\}_{i=0}^{k-1}, \{\hat{\mathbf{n}}_{i,\text{fix}}\}_{i=0}^{k-1})$  from (9)
    // Update adaptive ambiguity covariance
7:     Determine  $\{\mathbf{V}_i\}_{i=0}^{k-1}$  from (10)
    // Solve two-stage FGO
8:      $\{\hat{\mathbf{x}}_{i=0,\text{float}}^k, \{\hat{\mathbf{n}}_{i=0,\text{float}}^k \leftarrow \text{solve 1st FGO (18)}$ 
9:      $\{\hat{\mathbf{n}}_{i=0,\text{fix}}^k \leftarrow \text{integer fixation}$ 
10:     $\{\hat{\mathbf{x}}_{i=0,\text{fix}}^k \leftarrow \text{solve 2nd FGO (19)}$ 
11:   else
12:     set index  $j = k - T + 1$ 
    // Cycle slip detection
13:     $\{\mathbf{c}_i\}_{i=j}^{k-1} \leftarrow \text{cycleSlipDetection}(\{\mathbf{y}_i\}_{i=j}^k, \{\hat{\mathbf{x}}_{i,\text{fix}}\}_{i=j}^{k-1}, \{\hat{\mathbf{n}}_{i,\text{fix}}\}_{i=j}^{k-1})$  from (9)
    // Update adaptive ambiguity covariance
14:    Determine  $\{\mathbf{V}_i\}_{i=j}^{k-1}$  from (10)
    // Solve two-stage FGO
15:     $\{\hat{\mathbf{x}}_{i=j,\text{float}}^k, \{\hat{\mathbf{n}}_{i=j,\text{float}}^k \leftarrow \text{solve 1st FGO (18)}$ 
16:     $\{\hat{\mathbf{n}}_{i=j,\text{fix}}^k \leftarrow \text{integer fixation}$ 
17:     $\{\hat{\mathbf{x}}_{i=j,\text{fix}}^k \leftarrow \text{solve 2nd FGO (19)}$ 
18:   end if
    // Output navigation solution
19:   Output  $\hat{\mathbf{x}}_{k,\text{fix}}, \hat{\mathbf{n}}_{k,\text{fix}}$ 
20: end for

```

IV. NUMERICAL EVALUATION

We conduct a numerical evaluation to assess the performance of the proposed factor graph-based GNSS RTK positioning algorithm. Simulation results of our proposed algorithm (AA-FGO) are compared with two other approaches:

- 1) The dual-density mixed-integer KF based approach in [16] with adaptive ambiguity prior (MIKF);
- 2) A two-stage FGO-based approach with nonadaptive ambiguity factor (NAA-FGO).

The comparison with MIKF is to show the potential of FGO with respect to KF-type methods, as MIKF implements

a similar strategy of adaptive ambiguity model as in our method. On the other hand, the comparison with NAA-FGO is done to show the impact of the adaptive ambiguity model.

A. Simulation Setup

In the simulation, we consider a single-band GNSS receiver obtaining code and carrier signals from $M+1$ satellites at a sampling rate of 10 Hz, with the carrier wavelength $\lambda = 0.2$ m. A static base station whose position is known is located in the local vicinity. M pairs of DD code and carrier phase measurements can be obtained at each epoch. The value of M is randomly drawn from the set $M \in [6, 7, 8]$. Correspondingly, the unknown DD integer ambiguity vector $\mathbf{n}_k \in \mathbb{Z}^M$ needs to be resolved for each epoch. To simulate the frequent cycle slips commonly encountered in dynamic environments, such as urban canyons, multiple integer jumps are introduced to the time evolution of integer ambiguities using the method outlined in Sec. IV-B. An example of the time history of simulated integer ambiguities is shown in Fig. 4. As mentioned in Sec. II, the receiver motion is modeled by the CV model described by (7) with sampling period $dt = 0.1$ s. Each simulation lasts for 300 time steps.

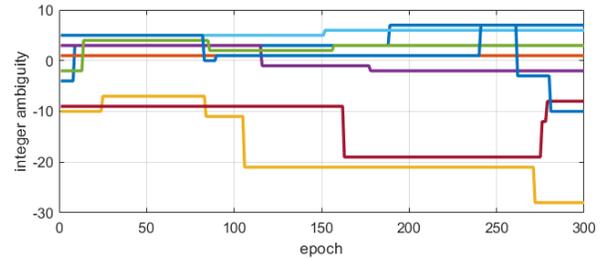


Fig. 4. An example of the integer ambiguity history with recurrent cycle slips

Our proposed algorithm is implemented using a sliding window to reduce the computational load. For the sliding window implementation, when the current time step is smaller than the predefined sliding window size, the FGO-based approaches use all the historical data up to the current epoch. When the current time step exceeds the predefined sliding window length, the sliding window starts moving forward as shown in Fig. 3. We test the impact of different sliding window sizes on the positioning performance. We have implemented the three algorithms in Matlab and use CasADi [23] through the MPCTools interface [24] and Ipopt [25] to solve the nonlinear graph optimization.

B. Cycle Slip Simulation

To simulate recurrent cycle slips in the evolution of integer ambiguities, we employ a dynamic process driven by a special discrete random walk to generate the integer jumps in ambiguity values [16]. Consider a discrete stochastic process that “jumps” with probability $b \in [0, 1]$, attaining a random value drawn from the uniform distribution over the integers on the interval $I = [-a, a] \subset \mathbb{Z}$, and is zero with probability

1 – b. Let $s \in I \subset \mathbb{Z}$ be a scalar random variable with the following associated density

$$\mathcal{J}(s|a, b) = p_s(s) = \begin{cases} (1-b)\delta(s-\tau) & \text{for } \tau = 0 \\ \frac{b}{2a}\delta(s-\tau) & \text{for } \tau \in I \setminus \{0\} \end{cases} \quad (20)$$

where δ denotes the Dirac delta function. For multivariate ambiguity vector $\mathbf{n}_k \in \mathbb{Z}^M$, the jumps are independent in each dimension, and on rectangular intervals about the origin defined by the corresponding elements of a vector $\mathbf{a} \in \mathbb{Z}^M$ with jump probabilities defined by the elements of a vector $\mathbf{b} \in \mathbb{R}^M$, in which $b_j \in [0, 1], j = 1, \dots, M$. Hence, the PDF of the integer noise vector \mathbf{s} can be expressed as

$$\mathbf{s} \sim \mathcal{J}(\mathbf{s}|\mathbf{a}, \mathbf{b}) = p_{\mathbf{s}}(\mathbf{s}) = \prod_{j=1}^M \mathcal{J}(s_j|a_j, b_j). \quad (21)$$

Using (21), we model the integer ambiguity time-evolution as

$$\mathbf{n}_{k+1} = \mathbf{n}_k + \mathbf{s}_k, \quad \mathbf{s}_k \sim \mathcal{J}(\mathbf{s}_k|\mathbf{a}, \mathbf{b}), \quad (22)$$

This stochastic discrete jump process (22) is employed in the numerical evaluation to simulate cycle slips. By tuning parameter \mathbf{b} , we can control the frequency of cycle slip occurrence in the simulation. To capture the behavior of the integer jump process, a good rule of thumb of picking the adaptive parameters in (10) is to let $\sigma_{\text{jump}} \approx a_{\text{max}}$ and $0.01 < \sigma_{\text{stay}} < 0.4$, where a_{max} represents the maximum element of \mathbf{a} .

C. Performance Evaluation

Fig. 5 displays the ambiguity resolution performance of the three different algorithms for one realization. The three algorithms are executed on the same set of measurement data. It can be seen that NAA-FGO (green) exhibits the largest ambiguity estimate errors. In comparison, the ambiguity errors of the other two approaches with adaptive ambiguity model converge and exhibit good fixation performance. Moreover, despite relatively large transient errors at the beginning, our proposed algorithm (blue) converges faster than the MIKF solutions (red).

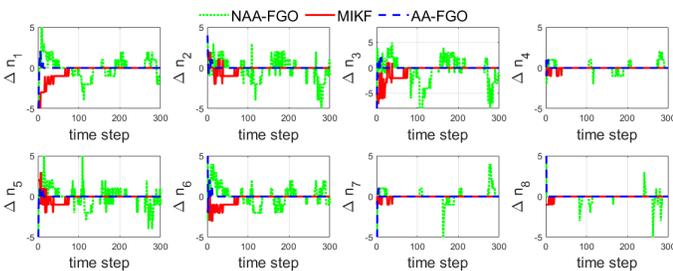


Fig. 5. Integer ambiguity resolution errors for one realization.

Fig. 6 shows the state estimation errors for the same realization corresponding to the ambiguity errors in Fig. 5. The state estimation errors are consistent with the ambiguity resolution errors. Our proposed approach (blue) and MIKF (red) exhibit superior performance compared with the NAA-FGO (green). In addition, the position errors of MIKF exhibit

a much longer transient stage (during the first 80 epochs) than our proposed algorithm before the errors converge.

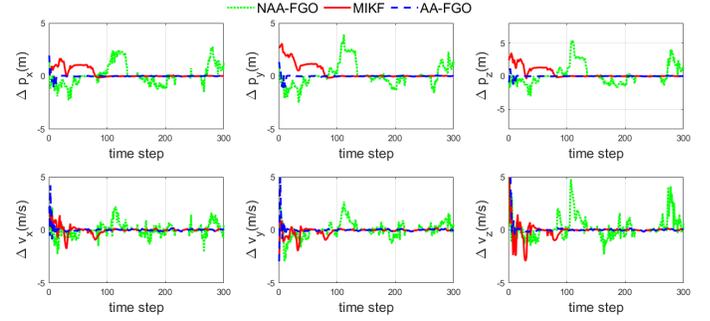


Fig. 6. State estimation errors for the same realization corresponding to Fig. 5.

To statistically assess and evaluate the performance of our proposed algorithm, we conduct a Monte Carlo (MC) simulation study, where 300 MC runs are executed with random problem initialization. The positional root-mean-square-errors (RMSE) of the three different algorithms are computed and reported in Fig. 7. The sliding window size $T = 90$ is picked in this MC simulation. We can see that the RMSE of NAA-FGO (green) converges to a level of around 0.6 m, which is significantly greater than that of our proposed algorithm (blue) and MIKF solutions (red). Our proposed approach (blue) achieves the smallest RMSE among the three and converges significantly faster.

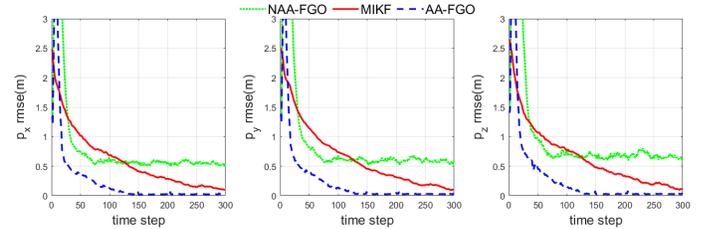


Fig. 7. Positional RMSE of three algorithms. MC runs = 300. Sliding window size = 90.

In addition, the impact of the sliding window size is tested using MC simulation. In principle, as the window size increases, more observation data will be used but the dimension of the optimization increases accordingly. This means that the improved performance comes with increased computational load. We test with sliding window size 50, 70, and 90 and the MC results of positional RMSE are displayed in Fig. 8. We can see that as the window size increases, the RMSE of the receiver positions gradually decreases in our proposed algorithm (AA-FGO). This is in alignment with our expectation. Moreover, our proposed algorithm with sliding window 50 and 70 still outperforms the rest two methods with a smaller RMSE level and faster convergence speed.

D. Discussion

We conducted numerical simulation to assess the performance of our proposed algorithm in challenging environment with frequent cycle slips. MC results show that

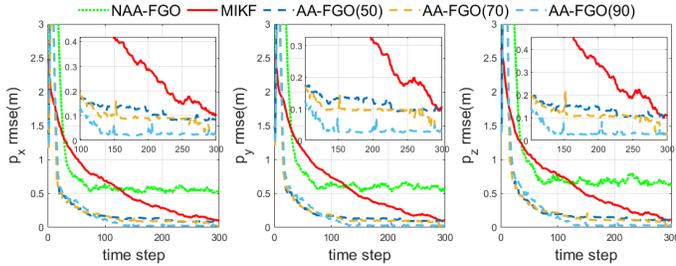


Fig. 8. Performance comparison with different sliding window sizes. Window sizes of 50, 70, and 90 are reported here. MC runs = 300.

our proposed FGO-based GNSS RTK approach with the adaptive ambiguity model (AA-FGO) is able to achieve better performance than the filtering-based approach (MIKF) and the non-adaptive FGO-based approach (NAA-FGO). The adaptive ambiguity factors enables the possibility of the optimization leveraging the time correlation inherent in the integer ambiguity evolution, increasing the robustness to cycle slips.

In Fig. 7 and Fig. 8, we can observe some relative large error spikes of our proposed algorithm's solutions at the first few time steps and then the RMSE level quickly drops and converges soon. This occurs due to the inaccurate cycle slip detection at the beginning. As shown in (10), the adaptive variance of the ambiguity model depends on the cycle slip detection. In this paper, we use the previous receiver state estimates to detect cycle slips as shown in Algorithm 1. The initial state errors can lead to erroneous detection, which contributes to the large error spikes at the beginning. As more observation data becomes available, receiver state errors get arrested, which, in return, improves the accuracy of the cycle slip detection. Therefore, we can see the error level quickly drops and converges in the solution of our proposed algorithm.

V. CONCLUSION

We proposed a FGO-based GNSS RTK positioning algorithm with adaptive ambiguity noise to address the cycle slips in GNSS-challenged, dynamics environments. The ambiguity dynamics is modeled by a random walk process with adaptive noise covariance where the variance is made adaptive to reflect the integer jumps during cycle slips. The adaptive ambiguity model is incorporated in a two-stage FGO architecture, which enables the graph optimization to explore the time correlation in the ambiguity evolution. MC simulation results show that our proposed adaptive FGO approach improves the RMSE of position estimates and convergence speed compared to the adaptive filtering approach and the non-adaptive FGO-based approach.

REFERENCES

- [1] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*. Ganga-Jamuna Press, 2001.
- [2] M. Greiff, S. Di Cairano, K. J. Kim, and K. Berntorp, "A system-level cooperative multiagent gnss positioning solution," *IEEE Transactions on Control Systems Technology*, vol. 32, no. 1, pp. 158–173, 2024.

- [3] Z. Yu, Y. Hu, and J. Huang, "Gps/ins/odometer/dr integrated navigation system aided with vehicular dynamic characteristics for autonomous vehicle application," *IFAC-PapersOnLine*, vol. 51, no. 31, pp. 936–942, 2018.
- [4] Y. Hu and et al., "Networked inertial navigation with constraints generated by neural networks," in *Proceedings of the 2022 International Technical Meeting of The Institute of Navigation*, 2022, pp. 1405–1419.
- [5] W. Hu, J. Uwineza, and J. A. Farrell, "Outlier accommodation for gnss precise point positioning using risk-averse state estimation," in *2024 American Control Conference (ACC)*. IEEE, 2024, pp. 1373–1379.
- [6] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, "Information fusion in navigation systems via factor graph based incremental smoothing," *Robotics and Autonomous Systems*, vol. 61, no. 8, pp. 721–738, 2013.
- [7] T. Suzuki, "Time-relative rtk-gnss: Gnss loop closure in pose graph optimization," *IEEE robotics and automation letters*, vol. 5, no. 3, pp. 4735–4742, 2020.
- [8] W. Wen, T. Pfeifer, X. Bai, and L.-T. Hsu, "Factor graph optimization for gnss/ins integration: A comparison with the extended kalman filter," *NAVIGATION: J. Inst. Navigation*, vol. 68, no. 2, pp. 315–331, 2021.
- [9] S. Zhao, Y. Chen, and J. A. Farrell, "High-precision vehicle navigation in urban environments using an mem's imu and single-frequency gps receiver," *IEEE Trans. Intell. Transportation systems*, vol. 17, no. 10, pp. 2854–2867, 2016.
- [10] W. Wen and L.-T. Hsu, "Towards robust gnss positioning and real-time kinematic using factor graph optimization," in *IEEE Int. Conf. Robotics and Automation*, 2021, pp. 5884–5890.
- [11] X. Bai, W. Wen, and L.-T. Hsu, "Time-correlated window-carrier-phase-aided gnss positioning using factor graph optimization for urban positioning," *IEEE Trans. Aerospace and Elec. Sys.*, vol. 58, no. 4, pp. 3370–3384, 2022.
- [12] J. Beuchert, M. Camurri, and M. Fallon, "Factor graph fusion of raw gnss sensing with imu and lidar for precise robot localization without a base station," in *IEEE Int. Conf. Robotics and Automation*, 2023, pp. 8415–8421.
- [13] H. Gao, H. Li, H. Huo, and C. Yang, "Robust gnss real-time kinematic with ambiguity resolution in factor graph optimization," in *Proc. Int. Tech. Meet. Institute of Navigation*, 2022, pp. 835–843.
- [14] J. Geng, C. Long, and G. Li, "A robust android gnss rtk positioning scheme using factor graph optimization," *IEEE Sensors Journal*, vol. 23, no. 12, pp. 13 280–13 291, 2023.
- [15] P. Teunissen, "A new method for fast carrier phase ambiguity estimation," in *Proc. IEEE Position, Location and Navigation Symp.*, 1994, pp. 562–573.
- [16] M. Greiff, K. Berntorp, S. Di Cairano, and K. J. Kim, "Mixed-integer linear regression kalman filters for gnss positioning," in *IEEE Conf. Control Tech. Appl.*, 2021, pp. 980–985.
- [17] W. Hu, Y. Hu, M. Stas, and J. A. Farrell, "Optimization-based outlier accommodation for tightly coupled rtk-aided inertial navigation systems in urban environments," *preprint arXiv:2407.13912*, 2024.
- [18] J. A. Farrell, *Aided navigation: GPS with high rate sensors*. McGraw-Hill, Inc., 2008.
- [19] F. Gustafsson, *Statistical sensor fusion*. Studentlitteratur, 2010.
- [20] K. Berntorp, A. Weiss, and S. Di Cairano, "Integer ambiguity resolution by mixture kalman filter for improved gnss precision," *IEEE Tran. Aerospace and Elec. Sys.*, vol. 56, no. 4, pp. 3170–3181, 2020.
- [21] X.-W. Chang, X. Yang, and T. Zhou, "Mlambda: a modified lambda method for integer least-squares estimation," *Journal of Geodesy*, vol. 79, no. 9, pp. 552–565, 2005.
- [22] P. Teunissen, P. De Jonge, and C. Tiberius, "The lambda method for fast GPS surveying," in *Int. Symp. GPS Technology Appl.*, 1995.
- [23] J. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.
- [24] M. Risbeck and J. B. Rawlings, "MPCTools: Nonlinear model predictive control tools for CasADi (Octave interface)," 2016.
- [25] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25–57, 2006.