Instantaneous GNSS Ambiguity Resolution and Attitude Determination via Riemannian Manifold Optimization

XING LIU, Member, IEEE
TARIG BALLAL, Member, IEEE
MOHANAD AHMED
TAREQ Y. AL-NAFFOURI, Senior Member, IEEE

King Abdullah University of Science and Technology, Thuwal, Saudi Arabia

In this article, we consider the problem of the global navigation satellite system (GNSS)-based attitude determination. A GNSS attitude model with nonlinear orthonormality constraints is used to rigorously incorporate the a priori knowledge of the receiver geometry. Given the characteristics of the employed nonlinear constraints, we formulate GNSS attitude determination as an optimization problem on a Riemannian manifold. We design a Riemannian algorithm to deliver the constrained float attitude matrix solution. Subsequently, the constrained float solution, combined with a proposed decomposition of the objective function, is utilized to enhance the efficiency of the integer ambiguity search. Both simulation and experimental evidences demonstrate the superiority of the proposed method. The results reveal that the proposed method can maintain the high probability of resolving the integer ambiguities correctly while enjoying the low computational complexity compared with the state-of-the-art techniques.

1. INTRODUCTION

Global navigation satellite system (GNSS)-based attitude determination has recently received much attention as it plays a critical role in various navigation, guidance, and control applications [1], [2], [3], [4], [5], [6]. Nowadays, GNSS attitude determination has been ubiquitously explored in numerous land, airborne, and maritime scenarios [7], [8], [9], [10], [11], [12]. The goal of GNSS attitude determination is to estimate a vehicle’s orientation with respect to a reference coordinate system [6], [13], utilizing multiple GNSS antennas/receivers rigidly mounted on the body frame. Compared to other sensors used for attitude determination, such as gyroscopes or star-trackers, a GNSS-based solution offers several advantages, including the driftless characteristic, low-power consumption, and minor maintenance requirements [14], [15].

A GNSS receiver can generate pseudorange and carrier-phase observations based on the signals transmitted from the navigation satellites. The measurement accuracy of the pseudorange observations is two orders of magnitude lower than that of carrier-phase observations [16]. It is essential to employ the precise GNSS data, the carrier phase, to obtain high-accuracy attitude estimates. The main challenge in fully utilizing the carrier-phase observations is to successfully resolve the unknown integer parts (number of whole cycles), a process usually referred to as integer ambiguity resolution. The integer ambiguities can be resolved by employing the change of the receiver-satellite geometry induced by the vehicle’s movement in the techniques known as motion-based methods [17], [18], [19]. However, this class of methods requires an initialization period of time, which can be excessively long for static or low-dynamic users. More recently, instantaneous ambiguity resolution approaches have been developed to solve the problem by searching in the float (attitude/position) or integer domains [7], [20], [21], [22], [23], [24], [25], [26], [27], [28], requiring only a single-epoch GNSS observation.

The least-squares ambiguity decorrelation adjustment (LAMBDA) method [20], [29] has become the standard approach for GNSS ambiguity resolution with unconstrained or linearly constrained models [16], [30], [31]. It can efficiently recover the ambiguities by searching for the integers within an ellipsoidal region. For unconstrained or linearly constrained models, LAMBDA solutions are optimal in terms of the ambiguity resolution success rate, i.e., it offers the highest probability of resolving the ambiguities correctly [32]. However, this technique fails to take advantage of a priori knowledge of the antenna geometry, which introduces nonlinear (orthonormality) constraints to the optimization problem.

The multivariate constrained LAMBDA (MC-LAMBDA) method [33] was proposed to leverage the prior knowledge of the antenna relative positions to enhance attitude determination. The literature shows that this technique can significantly improve the success-rate performance compared to the LAMBDA method [31],
Resolving the unknown carrier-phase ambiguities correctly allows for accurate attitude determination, whereas the orthonormality constraints facilitate high ambiguity resolution success rates and reliable attitude estimation [30]. However, incorporating the nonlinear constraints into the optimization problem results in a more complex search space that is not ellipsoidal anymore, increasing the complexity of the search process significantly [31], [35].

The performance of the MC-LAMBDA approach relies on the quality of their (initial) float solution, that is, the least-squares (LS) estimations with the integer constraint of ambiguities ignored. Under satellite-deprived environments, such as urban canyons, searching the integers requires a huge computational load due to the poor float solutions initially obtained through unconstrained LS estimation [36].

Based on a specific relaxation of the nonlinear constraints pertaining to the antenna-array geometry, the affine-constrained attitude model (AC-AM) has been proposed as a compromise between the unconstrained LAMBDA and MC-LAMBDA methods [16], [30]. The AC-AM relaxes the orthonormality constraints to linear ones through an affine transformation. Therefore, if the number of baselines is greater than the dimension of the range of the baseline matrix, the AC-AM provides a better float solution compared to the unconstrained model such that it outperforms the standard LAMBDA in terms of the ambiguity resolution success rate [30]. On the other hand, the AC-AM is weaker than the orthonormality-constrained attitude model (OC-AM) used in the MC-LAMBDA method. Given a large number of baselines, this method can generally provide a high success rate, allowing it to be a fast alternative to the MC-LAMBDA method [16]. However, with a limited number of baselines, there is still much room for success-rate improvement.

This article considers the OC-AM that incorporates both the orthonormality and integer constraints. We improve the computational efficiency of the MC-LAMBDA method in two ways. First, we improve the float solution by enforcing the constraints. Using the prior knowledge of the geometric constraints is expected to improve the float solution and make it closer to the true solution. This can further enhance the integer search. Since the search strategy of the MC-LAMBDA depends on the float solution being at the global minimum of the unconstrained objective function, using the improved float solution requires a modified integer search strategy. We develop such a strategy based on a new proposed decomposition of the objective function.

As we will see in the following sections, the float solution needs to be computed not only once at the initialization stage but also repetitively as part of the integer search routine. Hence, as a second way to improve the computational efficiency of attitude estimation using orthonormality constraints, we employ a more efficient solver to compute the constrained float solution. The nonlinearly constrained LS problem can be solved, for example, through parameterization [37], [38], [39] or using the Lagrange multipliers method [40], [41]. The solution is usually implemented using an iterative technique such as the Newton’s or Gauss–Newton methods [40], [42]. In this article, we apply Riemannian optimization to solve the nonlinearly constrained LS problem. Riemannian optimization treats the constrained optimization problem in the Euclidean space as an unconstrained optimization on a Riemannian manifold. This simplifies our perception of the problem and allows us to apply tools from unconstrained optimization to solve the problem on the manifold [43]. Additionally, the transformation of the constrained optimization into an unconstrained optimization on a manifold offers the following advantages.

1) Since the iterates execute on the manifold where all the constraints are satisfied, Riemannian optimization does not need to consider Lagrange multipliers or penalty functions, inherently maintaining convergence properties of unconstrained optimization algorithms [43].
2) The dimension of the manifold is smaller than or equal to that of the original Euclidean space [44]. The (possibly) reduced dimensionality, as well as exploiting the underlying geometric structure of the manifold, allows Riemannian optimization algorithms to be more efficient than their (constrained) Euclidean counterparts [45].
3) Another benefit of Riemannian algorithms lies in their flexibility with respect to changing or modifying the objective function without requiring a substantial change in the whole algorithm. The Riemannian algorithm design is largely dictated by the manifold, i.e., the constraints, not the objective function. This can be particularly useful when the objective function needs to be modified to cope with changes in the noise model or incorporate new measurements, etc. [43].

To differentiate between the proposed method and existing methods, we have summarized our main contributions as follows:

1) Enforcing the orthonormality constraints in the (initial) float solution to improve the integer search.
2) Proposing a decomposition of the objective function that allows us to employ the constrained float solution to accelerate the search process.
3) Developing a Riemannian optimization algorithm to efficiently compute the constrained float solution and improve the computational efficiency of the overall attitude determination algorithm.

The rest of this article is organized as follows. Section II describes the main GNSS attitude determination models and the related methods from the literature. Section III briefly introduces the necessary ingredients for designing optimization algorithms on Riemannian manifolds. Section IV presents the proposed attitude determination algorithm. Section IV-A presents the problem formulation and the proposed method to obtain the constrained float solution using Riemannian manifold optimization. Section IV-B
describes how to apply the constrained float solution to solve the ambiguity resolution problem using a proposed objective-function decomposition. Section V presents numerical results, comparing the performance of the proposed approach and a number of benchmark methods using different evaluation criteria. Finally, Section VI concludes this article.

II. RELATED WORK

A. Unconstrained Attitude Determination

Consider a vehicle equipped with $A + 1$ GNSS antennas that track $S + 1$ satellites simultaneously. The double-difference (DD) operation is applied to cancel out the common-mode errors, such as the receiver and satellite clock biases, hardware offsets, and atmospheric delays. In a single-frequency and single-epoch case, the GNSS attitude observation equations are given by

$$
\Psi = HX + WN + \Pi
$$

$$
P = HX + \Xi
$$

where $\Psi \in \mathbb{R}^{S \times A}$ and $P \in \mathbb{R}^{S \times A}$ denote the DD carrier-phase and pseudorange matrices, respectively, $H \in \mathbb{R}^{S \times 3}$ is composed of the satellite line-of-sight vectors, $W \in \mathbb{R}^{S \times S}$ contains the wavelength of the GNSS frequency, the columns of $X \in \mathbb{R}^{3 \times A}$ represent the unknown baseline coordinates in the chosen reference frame, $N \in \mathbb{R}^{S \times A}$ denotes the unknown DD integer ambiguities, and $\Pi \in \mathbb{R}^{S \times A}$ and $\Xi \in \mathbb{R}^{S \times A}$ are unmodeled errors and noise.

Model (1) is called the unconstrained attitude model (UC-AM) since it disregards the prior knowledge of the baseline matrix. Note that “unconstrained” is only in terms of the antenna-array geometry; the integer constraints are given full consideration. To estimate the unknown baseline coordinates and integer ambiguities, the problem can be formulated as

$$
\min_{X \in \mathbb{R}^{3 \times A}, N \in \mathbb{Z}^{S \times A}} \|\text{vec}(Y - AX - BN)\|_{Q_{YY}}^2
$$

where

$$
Y \triangleq \begin{bmatrix} \Psi \\ P \end{bmatrix}, \quad A \triangleq \begin{bmatrix} H \\ \Xi \end{bmatrix}, \quad B \triangleq \begin{bmatrix} W \\ 0 \end{bmatrix}
$$

The matrices $A$ and $B$ link the GNSS measurements and the unknown parameters. The notation $\text{vec}(\cdot)$ represents the vectorization operation, $\|\cdot\|_{Q_{YY}}^2 = (\cdot)^T Q_{YY} (\cdot)$, and $Q_{YY}$ is the covariance matrix of $\text{vec}(Y)$, that is

$$
Q_{YY} = \mathbb{E} \left[ \text{vec}(Y - \mathbb{E}(Y)) [\text{vec}(Y - \mathbb{E}(Y))]^T \right]
$$

where $\mathbb{E}(\cdot)$ is the expectation operator. Since a common reference antenna is used for the differencing operation, we have

$$
Q_{YY} = D \otimes Q_{yy}
$$

where $Q_{yy}$ is the covariance matrix of GNSS observations for a single baseline, $D$ denotes correlation between single-baseline observations, and $\otimes$ indicates the Kronecker product.

To determine the solution of (2), the following orthogonal decomposition can be applied to simplify the problem [16], (3) shown at the bottom of this page.

In (3), $E_{uc} = Y - A\hat{X}_{uc} - B\hat{N}_{uc}$ is the residual matrix of the LS solution, i.e., the float solution without the integer constraints given by

$$
\hat{X}_{uc}, \hat{N}_{uc} = \arg\min_{X \in \mathbb{R}^{3 \times A}, N \in \mathbb{Z}^{S \times A}} \|\text{vec}(Y - AX - BN)\|_{Q_{YY}}^2.
$$

The covariance matrices that characterize the dispersion of $\text{vec}(\hat{X}_{uc})$ and $\text{vec}(\hat{N}_{uc})$ are $Q_{\hat{X}_{uc}\hat{X}_{uc}}$, $Q_{\hat{X}_{uc}\hat{N}_{uc}}$, and $Q_{\hat{N}_{uc}\hat{N}_{uc}}$. The other variables in (3) are expressed as

$$
\text{vec}(\hat{X}_{uc}(N)) = \text{vec}(\hat{X}_{uc}) - Q_{\hat{X}_{uc}\hat{X}_{uc}}^{-1} Q_{\hat{X}_{uc}\hat{N}_{uc}} \text{vec}(\hat{N}_{uc} - N)
$$

$$
Q_{\hat{X}_{uc}(N)\hat{X}_{uc}(N)} = Q_{\hat{X}_{uc}\hat{X}_{uc}} - Q_{\hat{X}_{uc}\hat{N}_{uc}} Q_{\hat{N}_{uc}\hat{N}_{uc}}^{-1} Q_{\hat{X}_{uc}\hat{N}_{uc}}.
$$

For further details, refer to [16].

The residual term is independent of the parameters. Since the UC-AM disregards the constraints on $X$, we can let $X = \hat{X}_{uc}(N)$ so that the third term on the right-hand side of (3) is equal to zero, i.e., minimizing (3). Hence, the original optimization in (2) is equivalent to the following integer LS problem

$$
\min_{N \in \mathbb{Z}^{S \times A}} \|\text{vec}(N - \hat{N}_{uc})\|_{Q_{\hat{X}_{uc}\hat{X}_{uc}}}^2
$$

The LAMBDA method can solve (6) efficiently, with $\hat{N}_{uc}$ being the center of the search space (an ellipsoidal set). Once the integer ambiguities are resolved, the recovered DD carrier phase can be utilized to estimate the baseline coordinates.

B. Affine-Constrained Attitude Determination

One can accurately measure the antenna-array geometry because the GNSS antennas are firmly mounted on the vehicle. Therefore, the antenna-array coordinates in the body frame can be considered as known parameters that can be incorporated in GNSS attitude determination. In [16] and [30], to leverage the a priori knowledge, the authors take advantage of an affine transformation, which converts the baseline coordinates from the body frame to the reference
frame through

\[ X = RX_b, \quad R \in \mathbb{R}^{3 \times q} \]  \hspace{1cm} (7)

where \( X_b \in \mathbb{R}^{q \times A} \) and \( q = \min(3, A) \). Then, (1) and (7) establish the AC-AM.

Based on the affine transformation, the attitude determination problem can be expressed as

\[
\min_{R \in \mathbb{R}^{3 \times 3}, N \in \mathbb{Z}^{3 \times A}} \| \text{vec}(Y - AX_b - BN) \|_{Q_{YY}^{-1}}^2 .
\]  \hspace{1cm} (8)

Similar to (3), the following orthogonal decomposition is used (9), shown at bottom of this page, where \( E_{AC} = Y - \hat{R}_{AC} X_b - BN_{AC} \) is the LS residual matrix of the float solution given by

\[
\hat{R}_{AC}, \hat{N}_{AC} = \arg \min_{R \in \mathbb{R}^{3 \times 3}, N \in \mathbb{Z}^{3 \times A}} \| \text{vec}(Y - ARX_b - BN) \|_{Q_{YY}^{-1}}^2
\]  \hspace{1cm} (10)

with the covariance matrix \( Q_{AC} N_{AC} \), \( Q_{AC} \), \( Q_{AC} N_{AC} \), and \( Q_{AC} \). The other variables in (9) are given by

\[
\text{vec}(\hat{R}_{AC})(N) = \text{vec}(\hat{R}_{AC}) - Q_{AC} N_{AC} \text{vec}(\hat{N}_{AC} - N)
\]  \hspace{1cm} (11)

Finally, the optimization problem in (8) is identical to the following minimization [30]:

\[
\min_{N \in \mathbb{Z}^{3 \times A}} \| \text{vec}(N - \hat{N}_{AC}) \|_{Q_{AC}^{-1}}^2 .
\]  \hspace{1cm} (12)

If the number of baselines exceeds the dimensionality of the baseline span, we have \( Q_{AC} N_{AC} < Q_{AC} \) [16]. In that case, the AC-AM can improve the float solution to provide better ambiguity estimation than the UC-AM.

C. Orthonormality-Constrained Attitude Determination

The AC-AM does not rigorously integrate the known geometry of the GNSS antenna configuration into the objective function. To fully leverage the nonlinear constraints of antenna-array geometry, an orthonormal matrix \( R \) is used to link \( X \) and \( X_b \) through

\[ X = RX_b, \quad R \in \mathbb{O}^{3 \times q} \]  \hspace{1cm} (13)

where \( R^T R = I_q \) and \( I_q \) represents an identity matrix of size \( q \). The matrix \( R \) indicates the orientation of the platform’s body frame relative to the reference coordinate. The combination of (1) and (13) yields the OC-AM.

The OC-AM involves two kinds of constraints, namely, the integer constraints for the ambiguities and the orthonormality constraints for matrix \( R \). The optimization problem for the OC-AM can be expressed as

\[
\min_{R \in \mathbb{O}^{3 \times q}, N \in \mathbb{Z}^{3 \times A}} \| \text{vec}(Y - ARX_b - BN) \|_{Q_{YY}^{-1}}^2
\]  \hspace{1cm} (14)

which can be decomposed into three terms [31], [33], (15) shown at bottom of this page.

Both (9) and (15) use the same float solution. The only difference between (9) and (15) is the solution space for the matrix \( R \). Due to the orthonormality constraint, one has to take the last term of (15) into account. For the OC-AM, the optimization in (14) is equivalent to [33]

\[
\min_{N \in \mathbb{Z}^{3 \times A}} \left( \| \text{vec}(N - \hat{N}_{AC}) \|_{Q_{AC}^{-1}}^2 + \min_{R \in \mathbb{O}^{3 \times q}} \| \text{vec}(R - \hat{R}_{AC})(N) \|_{Q_{AC}^{-1}}^2 \right)
\]  \hspace{1cm} (16)

The OC-AM is stronger than the UC-AM and AC-AM. However, the search space of (16) is no longer an ellipsoidal set, which leads to more complexity than that of (6) and (12). Nowadays, the MC-LAMBDA method is the most notable approach to determine attitude based on the OC-AM. The MC-LAMBDA method solves (16) using special search-and-shrink or search-and-expand procedures to speed up the integer search process [35]. The MC-LAMBDA method can solve (16) efficiently in moderate conditions; however, it may incur an extravagantly high computational load in challenging scenarios, e.g., with a small number of satellites and/or high noise levels. Therefore, developing a more efficient orthonormality-constrained attitude determination method is essential to alleviate the high computational burden encountered in many practical situations.

III. OVERVIEW OF RIEMANNIAN MANIFOLD OPTIMIZATION

In this article, to further improve the computational efficiency, we formulate the GNSS attitude determination
problem as an optimization on Riemannian manifolds. Before presenting the details of the proposed method, we first briefly introduce all the ingredients needed to develop Riemannian algorithms on matrix manifolds. This includes the manifold terminology, the Riemannian derivatives, and the optimization algorithm design.

A. Manifold Optimization Terminology

A matrix manifold $\mathcal{M}$ refers to a topological space that is in bijection with an open space of the Euclidean space. At each point $X \in \mathcal{M}$, the manifold $\mathcal{M}$ locally resembles a linear space, i.e., a Euclidean space, which is known as the tangent space $T_X \mathcal{M}$. The dimension of $T_X \mathcal{M}$ represents the dimension of $\mathcal{M}$, namely, the degrees of freedom.

The tangent space $T_X \mathcal{M}$ can be endowed with a positive-definite inner product $\langle \cdot, \cdot \rangle_X$ that is called the Riemannian metric, which makes it possible to define the notion of length for tangent vectors. Although there are various Riemannian metrics for a manifold $\mathcal{M}$, a typical option is to employ the inner product in canonical form, permitting the simple expressions of the Riemannian gradient and Hessian.

Let $f : \mathcal{M} \rightarrow \mathbb{R}$ be a smooth function over a matrix manifold $\mathcal{M}$. The directional derivative of $f$ at the point $X \in \mathcal{M}$ in the direction $\xi \in T_X \mathcal{M}$, expressed by $D(f(X))|\xi|_X$, is defined as

$$D(f(X))|\xi|_X = \lim_{t \rightarrow 0} \frac{f(X + t\xi) - f(X)}{t}.$$  \hspace{1cm} (17)

Given that the tangent space $T_X \mathcal{M}$ is a linear approximation of the Riemannian manifold near the point $X$, only the tangent vectors $\xi \in T_X \mathcal{M}$ can be utilized as valid direction vectors. The operator $D(f(X)) : T_X \mathcal{M} \rightarrow \mathbb{R}$ is defined as the indefinite directional derivative of $f$ at $X$, which gives the directional derivative $D(f(X))|\xi|_X$ for a specific tangent vector $\xi$.

B. Riemannian Optimization Algorithms

To design optimization algorithms over Riemannian manifolds, one needs the notions of Riemannian gradient $\nabla_X f$ and Riemannian Hessian $\nabla_X^2 f$, which are derivative operators defined only on the tangent space $T_X \mathcal{M}$, unlike Euclidean gradient $\nabla f$ and Euclidean Hessian $\nabla^2 f$ that are valid in Euclidean space and not exclusively on $T_X \mathcal{M}$.

At the point $X \in \mathcal{M}$, the Euclidean gradient $\nabla f$ is related to the directional derivative $\xi \in T_X \mathcal{M}$ as

$$\langle \nabla f, \xi \rangle_X = D(f(X))|\xi|_X.$$ \hspace{1cm} (18)

When the canonical Riemannian metric is used, one can easily achieve the Riemannian gradient $\nabla_X f$ via projecting the Euclidean gradient $\nabla f$ onto the tangent space, that is

$$\nabla_X f = \Pi_X (\nabla f)$$ \hspace{1cm} (19)

where $\Pi_X$ denotes the corresponding orthogonal projection.

Similar to the definition of Euclidean Hessian, one can define the Riemannian Hessian $\nabla_X^2 f$ as the directional derivative of the Riemannian gradient. However, the directional derivative of $\nabla_X f$ does not have to lie in the tangent space $T_X \mathcal{M}$ such that further operation is required to project the directional derivative of $\nabla_X f$ onto the tangent space, i.e.,

$$\nabla_X^2 f|\xi|_X = \Pi_X (D(\nabla_X f)|\xi|_X)$$ \hspace{1cm} (20)

in which the same orthogonal projection $\Pi_X$ is applied.

Rather than the nonlinear constrained optimization in Euclidean space, Riemannian optimization generally involves unconstrained optimization on Riemannian manifold, i.e., unconstrained optimization over a constraints set. Therefore, Riemannian optimization algorithms follow similar procedures as the unconstrained ones in Euclidean space, but some discrepancies need to be addressed. Initially, one needs to locally approximate the manifold around the point $X$ on the given manifold to achieve a linear space. In other words, the first step is to derive the formulation of the tangent space $T_X \mathcal{M}$ and endow it with the Riemannian metric. Then, based on the Riemannian gradient and Hessian, we can obtain a descent direction $\xi \in T_X \mathcal{M}$ and a step size $\alpha$ so that we can find a new point $X + \alpha \xi$ contained in the tangent space. However, the newly found point is not a feasible solution since it is not on the manifold. Finally, a retraction operator $\Re_X$ is utilized to project the new point to the manifold.

The procedures of Riemannian optimization are summarized in Algorithm 1. The Riemannian algorithms that rely on only the gradient information refer to the first-order algorithms, where the steepest descent direction can be chosen as $\xi = -\frac{\nabla_X f}{\|\nabla_X f\|}$. In contrast, those who take advantage of the Riemannian Hessian belong to the second-order algorithms, among which, one example is Newton’s method on Riemannian manifolds that figures out the search direction by solving $\nabla_X^2 f|\xi|_X = -\nabla_X f$.

According to the discussion above, the Riemannian optimization algorithms require a few ingredients, namely, the tangent space, the orthogonal projector, the Riemannian gradient, the Riemannian Hessian, and the retraction operator. For more details about Riemannian optimization, refer to [46], [47], [48].

IV. PROPOSED RIEMANNIAN-MANIFOLD-BASED ATTITUDE DETERMINATION METHOD

For the integer search-based methods discussed in Section II, ambiguity resolution is carried out via searching

| Algorithm 1: Procedures of Riemannian Optimization. |
|------------------|------------------|
| 1: Initialize $X \in \mathcal{M}$.               |
| 2: while $\|\nabla_X f\|_X \neq 0$ do             |
| 3: Find the search direction $\xi \in T_X \mathcal{M}$ using $\nabla_X f$ and/or $\nabla_X^2 f$. |
| 4: Compute the step size $\alpha$ using backtracking. |
| 5: Retract $X = \Re_X (\alpha \xi)$.             |
| 6: end while                                     |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
around the float solution in the integer domain; that is, the float solution acts as an intermediate to simplify the integer search. As a consequence, the performance of ambiguity resolution profoundly relies on the quality of the float solution. Hence, it is critical to ensure that the float solution is of sufficient accuracy. Indeed, the more accurate the float solution, the better. To facilitate resolving the carrier-phase ambiguities more efficiently and reliably, we propose an approach based on optimization techniques on Riemannian manifolds to enhance the float solution. Subsequently, we develop an efficient integer search algorithm based on the improved float solution. The proposed method employs the OC-AM with (14) being the objective function. We refer to the proposed attitude determination method as the Riemannian-manifold-based orthogonality-constrained attitude determination (RieMOCAD) method.

A. Intermediate Float Solution and Riemannian Algorithm Design

In this section, we first formulate the float solution used by the proposed RieMOCAD method. Then, we study and characterize the geometry of the manifolds of interest and present Riemannian optimization algorithms employed to achieve the float solution.

1) Constrained Float Solution: The key principle behind the proposed method is to maintain the orthogonality constraint \( R \in \mathbb{O}^{3 \times q} \) when pursuing the initial float solution. The motivation here is that preserving this constraint can enhance the strength of the model used to estimate the float solution (compared to the one with the constraint ignored). A constrained float solution was suggested in [40] for determining the initial size of integer search space. In this work, we use this constrained float solution to determine the starting point of the search space rather than its size.

Incorporating the constraints, however, usually leads to increased computational complexity. Although the float solution needs to be computed only once, the orthonormal matrix \( R \) needs to be estimated repetitively in the search stage. Hence, we have to take the complexity into account. To develop a computationally efficient solution, we leverage advanced tools from manifold optimization [46], [47], [48].

Due to the orthogonality characteristics, the solution set of \( R \) is a manifold, which allows us to solve the problem using efficient Riemannian algorithms. In other words, we calculate the float solution using Riemannian manifold optimization, which gives us

\[
\hat{R}_{RM}, \hat{N}_{RM} = \arg \min_{R \in \mathbb{O}^{3 \times q}, N \in \mathbb{R}^{3 \times d}} \| \text{vec} (Y - ARX_b - BN) \|_{Q_{YY}}^2.
\]  

(21)

The LS solutions \( \hat{R}_{ac} \) and \( \hat{N}_{ac} \) are required in the integer search process (presented later). Hence, the first step is to compute the LS solution \( \hat{R}_{ac} \) and \( \hat{N}_{ac} \), which allows us to obtain \( \hat{R}_{RM} \) and \( \hat{N}_{RM} \) based on \( \hat{R}_{ac} \) and \( \hat{N}_{ac} \) rather than solving (21) directly. According to [40], the optimization problem in (21) is equivalent to

\[
\hat{R}_{RM} = \arg \min_{R \in \mathbb{O}^{3 \times q}} \| \text{vec} (R - \hat{R}_{ac}) \|_{Q_{ac}}^2.
\]  

(22)

\[
\hat{N}_{RM} = \hat{N}_{ac} (\hat{R}_{RM}).
\]  

(23)

That is to say, solving (22) is where Riemannian optimization comes into play. This reformulation helps us in exploiting Riemannian optimization with an easy-compute form.

2) Optimization Algorithm Design on Riemannian Manifolds: As stated in Section II-C, the attitude matrix \( R \) is an orthonormal matrix whose columns are orthogonal regarding the inner product. Here, for the sake of simplicity and avoiding confusion, we still utilize the symbol \( X \) to denote the point on the manifold rather than symbol \( R \). Then, we can use the terminologies discussed in Section III and present the required elements for Riemannian algorithms.

The potential solution set \( \mathcal{M} \) is an embedded submanifold of \( \mathbb{R}^{3 \times q} \) \( (q = 1, 2, 3) \), which is well known as the Stiefel manifold given by [43]

\[
\mathcal{M} = \{ X \in \mathbb{R}^{3 \times q} | X^T X = I_q \}.
\]  

(24)

The dimension of the Stiefel manifold \( \mathcal{M} \) is

\[
dim \mathcal{M} = \dim \mathbb{R}^{3 \times q} - \dim \text{Sym}(q) = 3q - \frac{q(q + 1)}{2}
\]  

(25)

where \( \text{Sym}(q) \) represents the linear space of symmetric matrices of size \( q \).

The tangent spaces of the Stiefel manifold \( \mathcal{M} \), a subspace of \( \mathbb{R}^{3 \times q} \), is given by

\[
\mathcal{T}_X \mathcal{M} = \{ V \in \mathbb{R}^{3 \times q} | X^T V + V^T X = 0 \}.
\]  

(26)

Alternatively, one can also express the tangent vectors in an explicit form as

\[
\mathcal{T}_X \mathcal{M} = \{ XS + X_1 K | S \in \text{Skew}(q), K \in \mathbb{R}^{(3-q) \times q} \}
\]  

(27)

where \( X \) and \( X_1 \) constitute the orthonormal basis of \( \mathbb{R}^{3 \times 3} \), and

\[
\text{Skew}(q) = \{ S \in \mathbb{R}^{q \times q} | S^T = -S \}
\]  

(28)

denotes the set of skew-symmetric matrices of size \( q \).

At a point \( X \) on the Stiefel manifold \( \mathcal{M} \), the orthogonal projection to the tangent space \( \mathcal{T}_X \mathcal{M} \) can be formulated as

\[
\Pi_X(U) = U - X \frac{X^T U + U^T X}{2} = (I - XX^T) U + \frac{X^T U - U^T X}{2}.
\]  

(29)

Note that the orthogonal projector \( \Pi_X \) guarantees that \( U - \Pi_X(U) \) falls into the normal space \( \mathcal{T}_X^\perp \mathcal{M} \) of the tangent space \( \mathcal{T}_X \mathcal{M} \). According to (18), (19), and (20), we can compute the Riemannian gradient \( \nabla_X f \) and Riemannian Hessian \( \nabla_X^2 f \). Then, we obtain

\[
\nabla_X f = \nabla_X f - \frac{X^T \nabla_X \nabla_X f + (\nabla_X f)^T X}{2}
\]  

(30)

\[
\nabla_X^2 f [\xi] = \Pi_X(\nabla_X^2 f [\xi]) - \xi^T \nabla_X f + (\nabla_X f)^T X
\]  

(31)
in which the QR decomposition is employed such that
\( Q R_{id} = X + V \) with \( Q \in \mathcal{M} \) and \( R_{id} \in \mathbb{R}^{q \times q} \). Here, \( R_{id} \) represents an upper triangular with nonnegative diagonal entries. On the other hand, the polar retraction can also be used [43], which is expressed as

\[
\mathfrak{R}_k(V) = (X + V) (X + V)^T (X + V)^{-\frac{1}{2}} = (X + V) (I_q + V^T V)^{-\frac{1}{2}}
\]

where \((-\cdot)^{-\frac{1}{2}}\) represents the inverse matrix square root. For both retraction operators, it readily seen that \( \mathfrak{R}_k(0) = X \). In this work, we employ the polar retraction.

So far, we have discussed the ingredients needed to perform Riemannian optimization on the Stiefel manifold.

Recalling that the goal is to solve (22), the objective function can be expressed as

\[
f(R) = \| \text{vec} (R - \hat{R}_{ac}) \|^2_{Q_{ac}^{-1} b_{ac}}.
\]

The covariance matrix \( Q_{\hat{R}_{ac} \hat{R}_{ac}} \) is factored into the Euclidean gradient and Hessian of \( f(R) \), so as to their Riemannian counterparts. Following the procedures in Algorithm 1, we can design efficient Riemannian algorithm on the Stiefel manifold to compute \( \hat{R}_{im} \) based on (29)–(31) and (33).

B. Ambiguity Resolution Aided by the Constrained Float Solution

1) Decomposition of the Objective Function: As stated in Section II, decomposing the objective function to a few more simple terms plays a crucial part in ambiguity resolution by defining the search space and simplifying the search process. However, the orthogonal decomposition (3) and (9) rely on an essential fact that the first-order and high-order (higher than two) derivatives of the objective function at the LS solution are zero [42]. There is no doubt that the orthogonal decomposition is not feasible at any other point except for the point where the gradient vanishes corresponding to the LS solution. This prompts us to develop a special decomposition to utilize the float solution given by \( (\hat{R}_{im}, \hat{N}_{im}) \) in the integer search.

In order to utilize the high-quality float solution \( (\hat{R}_{im}, \hat{N}_{im}) \), we first consider decomposing (14) at an arbitrary point. The following lemma will describe our proposed decomposition.

**Lemma 1** For \( \forall \hat{R} \in \mathbb{R}^{3 \times q} \) and \( \forall \hat{N} \in \mathbb{R}^{5 \times A} \), we can rewrite the objective function of (14) as a sum of five terms expressed by \( \hat{R} \) and \( \hat{N} \) as

\[
\| \text{vec} (Y - ARX_b - BN) \|^2_{Q_{ac}^{-1} b_{ac}} = \| \text{vec} (E) \|^2_{Q_{ac}^{-1} b_{ac}}
\]

\[
+ \| \text{vec} (N - \hat{N}) \|^2_{Q_{ac}^{-1} b_{ac}} + \| \text{vec} (R - \hat{R}(N)) \|^2_{Q_{ac}^{-1} b_{ac}}
\]

\[
+ 2 \text{vec}(N - \hat{N})^T Q_{ac}^{-1} \text{vec}(\hat{N} - \hat{N}_{ac})
\]

\[
- \| \text{vec}(\hat{R} - \hat{R}_{ac}(N)) \|^2_{Q_{ac}^{-1} b_{ac}}.
\]

**Proof** See Appendix A.

Compared with (15), (34) contains five terms on the right-hand side, among which the first represents the squares norm of the residual corresponding to \( \hat{R} \) and \( \hat{N} \), the second measures the distance from the integer matrix \( N \) to \( \hat{N} \) in the metric of \( Q_{ac}^{-1} \), the third weighs the distance from the orthonormal matrix \( R \) to \( \hat{R}(N) \) in the metric of \( Q_{ac}^{-1} \), and the last two terms are related to distance from the selected points \( \hat{R} \) and \( \hat{N} \) to the LS solution \( \hat{R}_{im} \) and \( \hat{N}_{im} \). Note that the covariance matrices of the LS solution are used in (34), and we can always compute these matrices using a close form. If \( \hat{R} = \hat{R}_{ac} \) and \( \hat{N} = \hat{N}_{ac} \), the last two terms in (34) equal to zero, leading to the identical expression as (15).

Fig. 1 shows a geometric illustration of the decomposition of the objective function. The essence of the optimization (14) is to seek a solution that minimizes \( d^2 \) in Fig. 1. Decomposing \( d^2 \) into a few easy-evaluated terms allows the full usage of the float solutions as an intermediate to simplify the search process. This figure depicts one of the most simple cases and indicates the difference between the two decomposition methods.

To simplify ambiguity resolution, we try to reformulate (34) as an alternative form that is easy to be bounded and evaluate.

**Lemma 2** For \( \forall \hat{R} \in \mathbb{R}^{3 \times q} \) and \( \forall \hat{N} \in \mathbb{R}^{5 \times A} \), the following relationship is an alternative form of (34)

\[
\| \text{vec}(Y - ARX_b - BN) \|^2_{Q_{ac}^{-1} b_{ac}} = \| \text{vec}(E) \|^2_{Q_{ac}^{-1} b_{ac}}
\]

\[
+ \| \text{vec}(N - \hat{N}) \|^2_{Q_{ac}^{-1} b_{ac}} + \| \text{vec}(R - \hat{R}(N)) \|^2_{Q_{ac}^{-1} b_{ac}}
\]

\[
+ 2 \text{vec}(N - \hat{N})^T Q_{ac}^{-1} \text{vec}(\hat{N} - \hat{N}_{ac})
\]

\[
- \| \text{vec}(\hat{R} - \hat{R}_{ac}(N)) \|^2_{Q_{ac}^{-1} b_{ac}}.
\]

**Proof** See Appendix B.

Lemmas 1 and 2 hold for an arbitrary point, so as to \( \hat{R}_{im} \) and \( \hat{N}_{im} \). If we replace \( \hat{R} \) and \( \hat{N} \) using \( \hat{R}_{im} \) and \( \hat{N}_{im} \), the first and last terms on the right-hand side of (35) are independent of the parameters. Then, we achieve the equivalent form of
(14) as follows:

$$\min_{N \in \mathbb{Z}^{S \times A}} C(N)$$

(36)

with

$$C(N) = \left\| \text{vec}(N - \hat{N}_{RM}) \right\|^2_{Q^{-1}_{AC}S_{AC}} + 2 \text{vec}(N - \hat{N}_{RM})^T Q^{-1}_{AC} \text{vec}(\hat{N}_{RM} - \hat{N}_{AC}) + \min_{R \in \mathbb{O}^{q \times q}} \left\| \text{vec}(R - \hat{R}_{AC}(N)) \right\|^2_{Q^{-1}_{RAC}(N)}.$$  

(37)

The function $C(N)$ includes three coupled terms.

If we ignore the last two terms in (37), we can define a loose-form solution of RieMOCAD (RieMOCAD-LF); that is, we calculate the closest integer estimations to the float solution $\hat{N}_{RM}$ in the metric of the associated weight matrix

$$\min_{N \in \mathbb{Z}^{S \times A}} \left\| \text{vec}(N - \hat{N}_{RM}) \right\|^2_{Q^{-1}_{AC}S_{AC}}.$$  

(38)

RieMOCAD-LF, (38), only partially employs the orthonormality constraint, which can be used as a low-complexity option in many practical applications. To fully incorporate this constraint, one needs to achieve the tight-form solution of RieMOCAD (RieMOCAD-TF), i.e., minimizing (36). Section IV-B2 presents the details to solve the optimization problem (36). Note that (36) provides only one possible tight-form solution of the OC-AM, and (16) is another alternative. In this contribution, (36) is applied rather than (16) to accelerate the process of the integer ambiguity resolution.

2) Integer Search Strategy: To find the optimal integer matrix that minimizes (36), a search process is required and the search space is defined as

$$\Omega(\chi) = \left\{ N \in \mathbb{Z}^{S \times A} \mid C(N) < \chi \right\}. $$

(39)

It requires high computational load to evaluate the integer matrix in $\Omega(\chi)$ due to the complex optimization caused by the presence of the orthonormality constraint. In other words, we need to evaluate every candidate in $\Omega(\chi)$ by solving the orthonormality-constrained optimization using the methods discussed in Section IV-A2. Similar to the MC-LAMBDA method, the objective function $C(N)$ is also bounded by two easier-to-evaluate functions, which are formulated as

$$C_L(N) = \left\| \text{vec}(N - \hat{N}_{RM}) \right\|^2_{Q^{-1}_{AC}S_{AC}} + 2 \text{vec}(N - \hat{N}_{RM})^T Q^{-1}_{AC} \text{vec}(\hat{N}_{RM} - \hat{N}_{AC}) + \xi_{\min} \sum_{i=1}^{q} (\| \tilde{r}_i(N) \| - 1)^2,$$

$$C_U(N) = \left\| \text{vec}(N - \hat{N}_{RM}) \right\|^2_{Q^{-1}_{AC}S_{AC}} + 2 \text{vec}(N - \hat{N}_{RM})^T Q^{-1}_{AC} \text{vec}(\hat{N}_{RM} - \hat{N}_{AC}) + \xi_{\max} \sum_{i=1}^{q} (\| \tilde{r}_i(N) \| - 1)^2,$$

(40)

with

$$C_L(N) \leq C(N) \leq C_U(N)$$

(41)

where $\tilde{r}_i(N)$ is the $i$th column of $\hat{R}_{AC}(N)$, and $\xi_{\min}$ and $\xi_{\max}$ denote the smallest and largest eigenvalues of $Q^{-1}_{RAC}(N)\hat{R}_{AC}(N)$, respectively.

Based on the lower and upper bounds of $C(N)$, we can define the corresponding search space as

$$\Omega_L(\chi) = \left\{ N \in \mathbb{Z}^{S \times A} \mid C_L(N) < \chi \right\},$$

$$\Omega_U(\chi) = \left\{ N \in \mathbb{Z}^{S \times A} \mid C_U(N) < \chi \right\},$$

(42)

with

$$\Omega_L(\chi) \subseteq \Omega(\chi) \subseteq \Omega_U(\chi).$$

(43)

Then, two search approaches, namely the search-and-shrink and search-and-expand strategies, can be applied to adjust the size of the search space adaptively so as to reduce the computational complexity significantly.

The search-and-shrink strategy takes advantage of the upper bound $C_U(N)$ and the corresponding set $\Omega_U(\chi)$. Initially, a large $\chi_0 > 0$ is chosen to initialize $\chi$ such that $\Omega_L(\chi)$ and $\Omega(\chi)$ are nonempty. Find an integer matrix $N_1 \in \Omega_L(\chi)$...
with \( C_1(N_1) = \chi_1 < \chi \). Then, shrink the search space by replacing \( \chi = \chi_1 \) and continue to look for an integer matrix \( N_2 \in \Omega_{\chi}(\chi) \) with \( C_1(N_2) = \chi_2 < \chi \). Iterate the search procedures until there is only one integer matrix \( N \) in the search space \( \Omega_{\chi}(\chi) \) (assume \( k \) iterations and \( \chi = \chi_k \)). \( N = N_k \) minimizes \( C(N) \) and \( \chi = C(N_k) \). Note that \( N = N_k \) does not minimize \( C(N) \). Therefore, we need to evaluate \( \Omega(\chi) \) using Riemannian optimization algorithms. Finally, we seek for \( N \in \Omega(\chi) \) that minimizes \( C(N) \) as the estimation.

In contrast, the search-and-expand strategy utilizes the lower bound \( C_1(N) \) and the associated search space \( \Omega_{\chi}(\chi) \). Initially, a small \( \chi_0 > 0 \) is chosen to initialize \( \chi \). If \( \Omega_{\chi}(\chi) \) is empty, replace \( \chi \) using a larger value \( \chi_1 \) and check \( \Omega_{\chi}(\chi) \) again. Iterate the search procedures (assume \( k \) steps) until \( \Omega_{\chi}(\chi) \) with \( \chi = \chi_k \) is nonempty. Then, evaluate all the candidates in \( \Omega_{\chi}(\chi) \) using \( C(N) \), that is

\[
\Omega(\chi) = \{ N \in \Omega_{\chi}(\chi) \mid C(N) < \chi \}.
\]

If \( \Omega(\chi) \) is nonempty, we extract \( N \in \Omega(\chi) \) that minimizes \( C(N) \) as the final estimation. Otherwise, let \( \chi = \chi_{k+1} = \chi_k \), \( k = k + 1 \) and repeat the search procedures until \( \Omega(\chi) \) is nonempty.

Algorithms 2 and 3 summarize the RieMOCAD method (tight form) using the search-and-shrink and search-and-expand strategies, respectively. Compared with the existing methods, the major differences of the proposed approach are as follows. The RieMOCAD method requires the float solutions LS and Riemannian optimization, whereas the MC-LAMBDA method requires only the LS. The decomposition of the objective function is different between the RieMOCAD and MC-LAMBDA methods, leading to different cost functions used in the feasible set. Instead of solving the orthonormality-constrained LS problem through the iterative techniques in Euclidean space, the RieMOCAD method employs the optimization over the Riemannian manifold in the search process.

### V. PERFORMANCE EVALUATION

This section tests the proposed method and compares it with several benchmarks using simulations as well as experimental data. For different scenarios, we analyze performance in terms of the quality of the float solution, success rate, and computational efficiency in the single-epoch and single-frequency cases, with only GPS satellites utilized. Riemannian optimization algorithms are implemented based on the Manopt toolbox [50].

#### TABLE I

| Methods               | \( \sigma \varphi \) (mm) |
|-----------------------|---------------------------|
|                       | 1            | 2            | 3            | 4            |
| Unconstrained LS      | 0.076        | 0.441        | 0.743        | 1.006        |
| Lagrange multiplier   | 0.017        | 0.179        | 0.335        | 0.458        |
| Quaternion            | 0.047        | 0.293        | 0.465        | 0.523        |
| Riemannian optimization | 0.017    | 0.153        | 0.321        | 0.415        |

#### TABLE II

| Methods               | \( \sigma \varphi \) (mm) |
|-----------------------|---------------------------|
|                       | 1            | 2            | 3            | 4            |
| Lagrange multiplier   | 1.00         | 0.998        | 0.971        | 0.964        |
| Quaternion            | 1.00         | 0.987        | 0.911        | 0.841        |
| Riemannian optimization | 1.00    | 1.00         | 1.00         | 1.00         |

Algorithm 2: RieMOCAD Using the Search-and-Shrink Strategy.

1: Compute \( \hat{R}_{\chi} \) and \( \hat{N}_{\chi} \) based on (10).
2: Solve (22) to obtain \( \hat{R}_{\chi} \) using the Riemannian algorithms.
3: Compute \( N_{\chi} \) using (23).
4: Initialize \( k = 0 \) and \( \chi = \chi_0 > 0 \) such that \( \Omega_{\chi}(\chi) \neq \emptyset \) and \( \Omega(\chi) \neq \emptyset \).
5: while card(\( \Omega_{\chi}(\chi) \)) > 1 do
6: \( k = k + 1 \).
7: Find \( N_k \in \Omega_{\chi}(\chi) \) with \( C(N_k) = \chi_k < \chi \).
8: \( \chi = \chi_k \).
9: Update \( \Omega_{\chi}(\chi) \) based on (40) and (42).
10: end while
11: Establish \( \Omega(\chi) \) using the Riemannian algorithms.
12: Choose \( N \in \Omega(\chi) \) that minimizes \( C(N) \).

Algorithm 3: RieMOCAD Using the Search-and-Expand Strategy.

1: Compute \( \hat{R}_{\chi} \) and \( \hat{N}_{\chi} \) based on (10).
2: Solve (22) to obtain \( \hat{R}_{\chi} \) using the Riemannian algorithms.
3: Compute \( N_{\chi} \) using (23).
4: Initialize \( \chi = \chi_0 > 0 \) and \( k = 0 \).
5: Let \( \Omega(\chi) = \emptyset \).
6: while \( \Omega(\chi) = \emptyset \) do
7: Set \( \Omega(\chi) \) based on (40) and (42).
8: for all \( N \in \Omega(\chi) \) do
9: Calculate \( C(N) \) using the Riemannian algorithms.
10: if \( C(N) < \chi \) then
11: \( \Omega(\chi) = \Omega(\chi) \cup N \).
12: end if
13: end for
14: if \( \Omega(\chi) = \emptyset \) then
15: \( \chi = \chi_{k+1} = \chi_k \).
16: \( k = k + 1 \).
17: end if
18: end while
19: Choose \( N \in \Omega(\chi) \) that minimizes \( C(N) \).
A. Simulation Analysis

Simulated data is used to test different methods under error-controlled conditions. The simulations are implemented using the assumed platform’s attitude and the actual satellite orbit information in the GPS Yuma Almanacs file for November 7, 2021. Several scenarios are simulated with different numbers of tracked satellites (#Sat), different numbers of baselines (#BL), various baseline lengths ($L$), and various measurement noise levels. The noise level is controlled by adding a zero-mean Gaussian noise with a specified standard deviation to the undifferenced GNSS observations. We assume that the noise of pseudorange data is two orders of magnitude higher than that of the carrier phase, that is, $\sigma_P = 100\sigma_{\phi}$. For each simulated scenario, a set of $10^4$ data is generated, with a random rotation and random (noncollinear) antenna positions in the body frame in each trial.

1) Comparison of Different Optimization Techniques in Estimating $R$: Note that in addition to Riemannian optimization, (22) can be solved using a plethora of methods such as the solutions based on the Lagrange multipliers and parameterization [39], [40]. Before studying the constrained float solution’s impact on ambiguity resolution, we first compare the Riemannian algorithm with the Lagrange multipliers method and a nonlinear estimation of a quaternion. The orthonormal matrix form and its quaternion parameterization are used to formulate the problem, with the nonlinear constraints incorporated by the Lagrange multiplier. Newton’s method is applied to solve the Lagrange functions. For more details on these methods, the readers are referred to [40]. The comparison is performed in terms of the estimation accuracy of the orthonormal matrix, algorithm convergence characteristics, and computational complexity based on simulated data ($\hat{R}_{AC}$ and $Q_{\hat{R}_{AC}}$).

The estimation accuracy of different methods is evaluated using the root-mean-square error (RMSE) to measure the average accuracy of estimating the elements of the unknown matrix $R$. We define this RMSE as follows:

$$
\text{RMSE}(\hat{X}) = \sqrt{\frac{\sum_{i=1}^{T} \left\| \text{vec}(\hat{X}_i - X_i) \right\|_2^2}{mn}} / T
$$

where $\hat{X}_i \in \mathbb{R}^{m \times m}$ denotes the matrix estimate from the $i$th epoch, $X_i$ represents the ground truth, and $T$ is the total number of epochs.

Table I presents the RMSE values of estimations of $R$ for the Riemannian optimization, Lagrange multipliers, and the quaternion methods. For reference, we also include the
Fig. 2. RMSE of float ambiguity estimation (one to three baselines). (a) $\sigma_{\Psi} = 1$ mm. (b) $\sigma_{\Psi} = 3$ mm. (c) $\sigma_{\Psi} = 5$ mm. (d) $\sigma_{\Psi} = 7$ mm.

| #BL | #Sat | $\sigma_{\Psi}$ (mm) | MC-LAMBDA | Riemocad-TF |
|-----|-----|---------------------|------------|-------------|
| 1   | 4   | 1                   | 81.18      | 81.18       |
|     | 3   | 28.00               | 28.01      | 28.01       |
|     | 5   | 14.82               | 14.82      | 14.82       |
|     | 7   | 10.19               | 10.19      | 10.19       |
|     | 5   | 37.77               | 37.77      | 37.77       |
|     | 7   | 21.92               | 21.91      | 21.91       |
|     | 6   | 5                   | 71.67      | 71.67       |
|     | 7   | 46.54               | 46.54      | 46.54       |
| 2   | 4   | 1                   | 99.21      | 99.21       |
|     | 5   | 95.34               | 95.33      | 95.33       |
|     | 6   | 98.35               | 98.35      | 98.35       |
|     | 7   | 99.11               | 99.11      | 99.11       |
|     | 8   | 99.23               | 99.23      | 99.23       |
| 3   | 4   | 1                   | 100        | 100         |
|     | 5   | 100                 | 100        | 100         |
|     | 6   | 99.99               | 99.99      | 99.99       |
|     | 7   | 100                 | 100        | 100         |
|     | 8   | 100                 | 100        | 100         |

The results in Table I demonstrate that the constrained estimators offer smaller RMSEs than the unconstrained LS. Riemannian optimization clearly outperforms the other two constrained optimization methods. Riemannian algorithms execute the iterates on the manifold such that the solutions always satisfy all the constraints. Although the Lagrange multipliers method is used to incorporate the nonlinear constraints (the orthonormality constraint or unit norm constraint for a quaternion), it is not guaranteed to converge to a solution in the constraints set. Hence, it occasionally produces substandard results.

To investigate how various constrained optimization methods fare when it comes to satisfying the constraints, Table II evaluates the probability of each method converging to a solution in the constraints set. These probability values are estimated using the same setup of Table I. From Table II, we can infer that the Riemannian optimization algorithm always converges to a solution in the constraints set. The same cannot be said about the two other constrained optimization methods. The success rate (%) for different measurement precision $\sigma_{\Psi}$ (mm), number of tracked satellites (#Sat), and number of baselines (#BL) are listed in Table VII.
methods, as they occasionally produce results that are inconsistent with the constraints. This kind of divergence (from the constraints set) might result in severe estimation errors, which explains the results in Table I. Therefore, it can be said that the inherent ability of Riemannian optimization...
to consistently and systematically converge to solutions in the desired constraints set presents a significant advantage compared with the other two methods.

To evaluate the computational complexity, the number of computations for different methods is summarized in Table III. The symbols \( L_1 \), \( L_q \), and \( L_r \) denote the number of iterations required to reach a predefined precision level for the Lagrange Multiplier, quaternion, and Riemannian optimization methods, respectively. The symbol \( K_l \) is the number of unknown parameters for the Lagrange multiplier method, which is the number of elements of the orthonormal matrix and the number of Lagrange multipliers. Since \( R \in \mathbb{O}^{3\times q} \), \( K_l = 3q + q(q + 1)/2 \). The parameter \( K_q \) represents the number of elements of a quaternion plus one Lagrange multiplier, that is, \( K_q = 5 \). The parameter \( K_r \) is equal to the number of elements of the orthonormal matrix, i.e., \( K_r = 3q \). The symbol \( L_u \) denotes the number of iterations required to find the search direction in the tangent space. Table IV shows the average values of \( L_1 \), \( L_q \), \( L_r \), and \( L_u \) to solve (22) based on simulated data generated using three baselines with random length (0 to 3 m), random observation noise (0 to 7 mm), and random number of satellites (4 to 8). Subsequently, we obtain the total number of operations for the Lagrange Multiplier, quaternion, and Riemannian optimization methods as presented in Table V. According to Table V, reparameterizing the orthonormal matrix with a quaternion estimation reduces computational complexity significantly. However, Riemannian optimization reduces the computational complexity even more.

2) Attitude Determination Performance Evaluation:

Given its clear advantage demonstrated in Section V-A1, we adopt the Riemannian optimization algorithm to implement both the RieMOCAD and MC-LAMBDA methods. This allows us to analyze the constraints float solution’s impact and fairly compare the proposed integer search strategy with MC-LAMBDA.

First, we will study the quality of various float ambiguity solutions, i.e., \( \hat{N}_{\text{UC}} \), \( \hat{N}_{\text{AC}} \), and \( \hat{N}_{\text{RM}} \) obtained from (4), (10), and (21), respectively. We characterize the quality of the float solution in terms of two metrics. The first metric is the RMSE that captures the differences between the float ambiguity solutions and the ground truth integer ambiguities. The second metric is defined as the success rate of the optimal integer solutions of (6), (12), and (38), which gives us an idea of how good the closest integer estimations to different float solutions in the metric of the associated weight matrix.

Figs. 2 and 3 plot the RMSE of the float ambiguities in various simulated environments. When the number of baselines is less or equal to the dimension of the range of the baseline matrix, the UC-AM and AC-AM produce exactly the same float solution, that is, \( \hat{N}_{\text{UC}} = \hat{N}_{\text{AC}} \), resulting in the same fixed integers of (6) and (12). According to Fig. 2, the error of the LS method does not show any noticeable variation with number of baselines. On the contrary, the RieM-based method’s errors decreases as the number of baselines increases. This is due to the rigorous way of incorporating the geometric information in the constrained optimization approach—more geometric information tends to strengthen these methods.

Fig. 3 indicates that for a number of baselines greater than the dimension of the range of the baseline matrix, the AC-AM demonstrates its advantage compared with the UC-AM since the affine transformation (linear constraint) can take advantage of the redundancy of the GNSS data. The RieMOCAD-LF outperforms the other two benchmarks due to the additional nonlinear constraint. Unlike the UC-AM, the performance of the AC-AM and RieMOCAD-LF improves as the number of baselines increases.

Table VI lists the success rate of the UC-AM, AC-AM, and RieMOCAD-LF methods [based on (6), (12), and (38), respectively] in selected simulation environments. Consistent with the RMSE of the float solutions shown in Fig. 2, the UC-AM and AC-AM achieve identical success rates in setups with three or fewer baselines. The AC-AM outperforms the UC-AM in terms of the success-rate if more than three baselines are used. Overall, RieMOCAD-LF offers the highest success rate compared with the UC-AM and AC-AM based methods.

According to Table VI, the AC-AM and RieMOCAD-LF offer acceptable results in many scenarios when using a large number of baselines. However, their estimations are not up to par in many challenging environments, especially when the number of baselines is less than the dimension of the range space of the baseline matrix. Therefore, the OC-AM is required to improve the results.

RieMOCAD-TF and the MC-LAMBDA methods are two alternative solutions for the optimization (14) based on the OC-AM. Table VII shows their success rates in different simulated environments that are excessively challenging for the UC-AM, AC-AM, and RieMOCAD-LF to resolve the integer ambiguities correctly. We can see that the MC-LAMBDA and RieMOCAD-TF methods offer similar success rates, outperforming the other three approaches.
Fig. 6. Euler angle error distribution based on the experimental data. (a) Yaw. (b) Pitch. (c) Roll.

Table VIII summarizes the corresponding average number of integer candidates that need to be evaluated by each algorithm. The integer search space of the RieMOCAD-TF method is significantly smaller than that of the MC-LAMBDA method, indicating an improvement in computational efficiency. In dual- or triple-baseline setups, the advantage of the proposed method becomes more notable.

In summary, compared with the MC-LAMBDA method, RieMOCAD-TF has the same success-rate performance but shrinks the search space in the integer domain at the expense of just an additional nonlinear optimization.

VI. CONCLUSION

By modeling the GNSS attitude problem as an optimization on the Stiefel manifold, we alleviate the difficulties of resolving the unknown carrier-phase ambiguities. First, to replace the widely used LS solution, we calculate a constrained float solution using Riemannian manifold optimization as an improved starting point for the integer search process. To take advantage of the improved intermediate solution, we propose a new decomposition of the objective function at an arbitrary point to replace the widely used orthogonal decomposition appropriate for the LS approach. By leveraging Riemannian optimization, the improved float solution, and the proposed decomposition, we are able to search for the carrier-phase ambiguities in the integer domain with high efficiency and reliability. The presented algorithm has two variants: a loose form and a tight form abbreviated as RieMOCAD-LF and RieMOCAD-TF, respectively. The proposed algorithms are compared to relevant state-of-the-art methods using both simulation and experimental data. We evaluate performance in terms of key performance indicators such as success rate, search space size, and attitude error distribution. The results confirm the feasibility and effectiveness of the proposed approach, especially in reducing the computational complexity while maintaining high success rates.

APPENDIX A

Proof of Lemma 1: We first vectorize all the unknown parameters in a single vector $\theta = \text{vec}(\mathbf{RN})$ and define the objective function as

$$\mathcal{F}(\theta) = \|\text{vec} (\mathbf{Y} - \mathbf{ARX}_b - \mathbf{BN})\|_{Q_y^{-1}}^2. \quad (A.1)$$

To achieve the LS solution $\hat{\theta} = \text{vec}(\hat{\mathbf{R}}_{ac}\hat{\mathbf{N}}_{ac})$, we let the first order derivative equal to zero [42]

$$\mathcal{F}'(\hat{\theta}) = 0. \quad (A.2)$$

Table IX shows the RMSE (cycles) of float ambiguity estimation, empirical success rate (%), and the number of integers in the search space based on the experimental data. We estimate the ground truth by postprocessing the data of the entire test period. The empirical success rate is defined as the percentage of occurrences in which the computed integer estimation equals the estimated ground truth. We can see that the AC-AM has a poor success rate (likewise for the UC-AM). As expected, RieMOCAD-LF offers a higher success rate compared to the AC-AM due to the improved float solution. The two tight-form implementations of the OC-AM, i.e., the MC-LAMBDA method and RieMOCAD-TF, offer identical success rates, higher than the other two methods. Fig. 6 shows the cumulative errors of the attitude angles (Euler angles). Consistent with the success-rate performance described above, RieMOCAD-LF outperforms the AC-AM. On the other hand, the cumulative error curves for the RieMOCAD-TF and the MC-LAMBDA methods do not exhibit any noticeable difference. The size of the integer search space (the total number of integer candidates) highlights the substantial reduction in computational complexity offered by the proposed RieMOCAD-TF compared with the MC-LAMBDA method.

B. Experimental Evaluation

In addition to the simulations, we evaluate the proposed method’s performance using experimental data and compare it with selected benchmark methods. Our experiment was carried out as follows. Three ANAVS multisensor modules [51] were arranged in an equilateral triangle with a 0.63-m side, as shown in Fig. 4. The receivers were firmly mounted on a flat surface of a platform. The platform remained stationary at position ($N^{22.18.32}$, $E^{39.6.27}$) throughout the experiment duration from 20:34 to 21:50 (UTC) on 25-May-2021. A set of 25500 epochs of GNSS measurements were collected with a 5-Hz sampling rate. The satellite visibility during the test is shown in Fig. 5, with a $10^\circ$ cut-off elevation angle. The presented algorithm has two variants: a loose form and a tight form abbreviated as RieMOCAD-LF and RieMOCAD-TF, respectively. The proposed algorithms are compared to relevant state-of-the-art methods using both simulation and experimental data. We evaluate performance in terms of key performance indicators such as success rate, search space size, and attitude error distribution. The results confirm the feasibility and effectiveness of the proposed approach, especially in reducing the computational complexity while maintaining high success rates.
Since \( \mathcal{F}(\theta) \) is a quadratic form of \( \theta \), its third or higher order derivatives are all equal to zero, i.e., \( \mathcal{F}^{(3)}(\theta) = \mathcal{F}^{(4)}(\theta) = \cdots = \mathcal{F}^{(n)}(\theta) = 0 \). Then, we can expand \( \mathcal{F}(\theta) \) as a Taylor series at \( \hat{\theta} \):

\[
\mathcal{F}(\theta) = \mathcal{F}(\hat{\theta}) + \frac{1}{2} (\theta - \hat{\theta})^T \mathcal{F}''(\hat{\theta}) (\theta - \hat{\theta}) \quad (A.3)
\]

with

\[
\mathcal{F}''(\hat{\theta}) = 2M
\]

\[
M = \begin{bmatrix}
X_b D^{-1} \otimes A^T Q_{yy}^{-1} B & X_b D^{-1} X_b^T \otimes A^T Q_{yy}^{-1} A \\
D^{-1} \otimes B^T Q_{yy}^{-1} B & D^{-1} X_b^T \otimes B^T Q_{yy}^{-1} A
\end{bmatrix}.
\]

Consider an arbitrary point \( \bar{\theta} = \text{vec}([\bar{R}, \bar{N}]) \). According to (A.3), we have

\[
\mathcal{F}(\theta) = \mathcal{F}(\hat{\theta}) + (\theta - \bar{\theta} + \bar{\theta} - \hat{\theta})^T M (\theta - \bar{\theta} + \bar{\theta} - \hat{\theta})
\]

\[
= \mathcal{F}(\hat{\theta}) + (\theta - \bar{\theta})^T M (\theta - \bar{\theta}) + (\bar{\theta} - \hat{\theta})^T M (\bar{\theta} - \hat{\theta}) + 2 (\theta - \bar{\theta})^T M (\bar{\theta} - \bar{\theta})
\]

\[
= \mathcal{F}(\hat{\theta}) + (\theta - \bar{\theta})^T M (\theta - \bar{\theta}) + 2 (\theta - \bar{\theta})^T M (\bar{\theta} - \bar{\theta}).
\]

(A.5)

We define the block-triangular transformation

\[
G = \begin{bmatrix}
I & (X_b^T)^T \otimes A^+ B \\
O & I
\end{bmatrix}.
\]

(A.6)

Then, we have

\[
G (\theta - \bar{\theta}) = \begin{bmatrix}
\text{vec} (R - \bar{R} (N)) \\
\text{vec} (N - \bar{N})
\end{bmatrix}
\]

(A.7)

\[
G (\bar{\theta} - \hat{\theta}) = \begin{bmatrix}
\text{vec} (\bar{R} - \bar{R}_{ac} (\bar{N})) \\
\text{vec} (\bar{N} - \bar{N}_{ac})
\end{bmatrix}
\]

(A.8)

\[
G^T M G^{-1} = \begin{bmatrix}
Q_{acac}^{-1} & O \\
O & Q_{nnac}^{-1}
\end{bmatrix}.
\]

(A.9)

Therefore, we obtain

\[
(\theta - \bar{\theta})^T M (\theta - \bar{\theta}) = (\theta - \bar{\theta})^T G^T G^{-1} M G (\theta - \bar{\theta})
\]

\[
= \| \text{vec} (N - \bar{N}) \|^2_{Q_{nnac}^{-1}} + \| \text{vec} (R - \bar{R} (N)) \|^2_{Q_{acac}^{-1}}
\]

(A.10)

\[
+ 2 \text{vec} (R - \bar{R} (N))^T Q_{acac}^{-1} \text{vec} (R - \bar{R}_{ac} (\bar{N}))
\]

(A.11)

The combination of (A.5), (A.10), and (A.11) concludes the proof.

APPENDIX B

Proof of Lemma 2: According to Lemma 1, \( \mathcal{F}(\theta) \) can be expressed as

\[
\mathcal{F}(\theta) = \mathcal{F}(\hat{\theta}) + \| \text{vec} (N - \bar{N}) \|^2_{Q_{nnac}^{-1}}
\]

\[
+ \| \text{vec} (R - \bar{R} (N)) \|^2_{Q_{acac}^{-1}}
\]

\[
+ 2 \| \text{vec} (R - \bar{R} (N)) \|^2_{Q_{acac}^{-1}}
\]

\[
+ 2 \text{vec} (R - \bar{R} (N))^T Q_{acac}^{-1} \text{vec} (R - \bar{R}_{ac} (\bar{N}))
\]

(B.1)

Note that

\[
\| \text{vec} (R - \bar{R}_{ac} (N)) \|^2_{Q_{acac}^{-1}}
\]

\[
= \| \text{vec} (R - \bar{R} (N) + \bar{R} (N) - \bar{R}_{ac} (N)) \|^2_{Q_{acac}^{-1}}
\]

\[
= \| \text{vec} (R - \bar{R} (N)) \|^2_{Q_{acac}^{-1}}
\]

\[
+ \| \text{vec} (\bar{R} (N) - \bar{R}_{ac} (N)) \|^2_{Q_{acac}^{-1}}
\]

\[
+ 2 \| \text{vec} (R - \bar{R} (N)) \|^2_{Q_{acac}^{-1}}
\]

\[
\times (R (N) - \bar{R}_{ac} (N))
\]

(B.2)

Since

\[
R - \bar{R}_{ac} (N) = R (N) - \bar{R}_{ac} (N)
\]

(B.3)

we have

\[
\| \text{vec} (R - \bar{R}_{ac} (N)) \|^2_{Q_{acac}^{-1}}
\]

\[
= \| \text{vec} (R - \bar{R} (N)) \|^2_{Q_{acac}^{-1}}
\]

\[
+ \| \text{vec} (\bar{R} (N) - \bar{R}_{ac} (N)) \|^2_{Q_{acac}^{-1}}
\]

\[
+ 2 \| \text{vec} (R - \bar{R} (N)) \|^2_{Q_{acac}^{-1}}
\]

\[
\times (R (N) - \bar{R}_{ac} (N))
\]

(B.4)
REFERENCES

[1] F. Causa and G. Fasano, “Improving navigation in GNSS-Challenging environments: Multi-UAS cooperation and generalized dilution of precision,” IEEE Trans. Aerosp. Electron. Syst., vol. 57, no. 3, pp. 1462–1479, Jun. 2021.

[2] L. Xiong et al., “IMU-based automated vehicle body sideslip angle and attitude estimation aided by GNSS using parallel adaptive Kalman filters,” IEEE Trans. Veh. Technol., vol. 69, no. 10, pp. 10668–10680, Oct. 2020.

[3] A. A. Ardalan and M. H. Rezvani, “An iterative method for attitude determination based on misaligned GNSS baselines,” IEEE Trans. Aerosp. Electron. Syst., vol. 51, no. 1, pp. 97–107, Jan. 2015.

[4] F. Aghili and A. Salerno, “Driftless 3-D attitude determination and positioning of mobile robots by integration of IMU with two RTK GPSs,” IEEE/ASME Trans. Mechatronics, vol. 18, no. 1, pp. 21–31, Feb. 2013.

[5] M. Farkas, B. Vanek, and S. Roza, “Small UAV's position and attitude estimation using tightly coupled multi baseline multi constellation GNSS and inertial sensor fusion,” in Proc. IEEE 5th Int. Workshop Metrol. Aeronav., 2019, pp. 176–181.

[6] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, GNSS—Global Navigation Satellite Systems: GPS, GLONASS, Galileo, and More., Berlin, Germany: Springer, 2007.

[7] Y. Li, K. Zhang, C. Roberts, and M. Murata, “On-the-fly GPS-based attitude determination using single- and double-differenced carrier phase measurements,” GPS Solutions, vol. 8, no. 2, pp. 93–102, Jul. 2004.

[8] N. O. Leite and F. Walter, “Flight test evaluation of a new GPS attitude determination algorithm,” IEEE Aerosp. Electron. Syst. Mag., vol. 22, no. 12, pp. 3–10, Dec. 2007.

[9] S. Dahiyia, V. Saini, and A. K. Singh, “GNSS signal processing based attitude determination of spinning projectiles,” IEEE Trans. Aerosp. Electron. Syst., vol. 58, no. 5, pp. 4506–4516, Oct. 2022.

[10] L. Lau, P. Cross, and M. Steen, “Flight tests of error-bounded heading and pitch determination with two GPS receivers,” IEEE Trans. Aerosp. Electron. Syst., vol. 48, no. 1, pp. 388–404, Jan. 2012.

[11] K. Chiang, M. Psiaki, S. Powell, R. Miceli, and B. O. Hanlon, “GPS-based attitude determination for a spinning rocket,” IEEE Trans. Aerosp. Electron. Syst., vol. 50, no. 4, pp. 2654–2663, Oct. 2014.

[12] G. Giorgi, P. J. G. Teunissen, and T. P. Gourlay, “Instantaneous global navigation satellite system (GNSS)-based attitude determination for maritime applications,” IEEE J. Ocean. Eng., vol. 37, no. 3, pp. 348–362, Jul. 2012.

[13] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, Global Positioning System: Theory and Practice. Berlin, Germany: Springer, 2012.

[14] J. Madsen and E. G. Lightsey, “Robust spacecraft attitude determination using global positioning system receivers,” J. Spacecraft Rockets, vol. 41, no. 4, pp. 655–664, 2004.

[15] J. L. Crassidis and F. L. Markley, “New algorithm for attitude determination using global positioning system signals,” J. Guid. Control Dyn., vol. 20, no. 5, pp. 891–896, 1997.

[16] G. Giorgi and P. J. G. Teunissen, “Low-complexity instantaneous ambiguity resolution with the affine-constrained GNSS attitude model,” IEEE Trans. Aerosp. Electron. Syst., vol. 49, no. 3, pp. 1745–1759, Jul. 2013.

[17] J. L. Crassidis, F. L. Markley, and E. G. Lightsey, “Global positioning system integer ambiguity resolution without attitude knowledge,” J. Guid. Control Dyn., vol. 22, no. 2, pp. 212–218, 1999.

[18] C. Chun and F. C. Park, “Dynamics-based attitude determination using the global positioning system,” J. Guid. Control Dyn., vol. 24, no. 3, pp. 466–473, May 2001.

[19] B. Wang, Z. Deng, S. Wang, and M. Fu, “A motion-based integer ambiguity resolution method for attitude determination using the global positioning system (GPS),” Meas. Sci. Technol., vol. 21, no. 6, 2010, Art. no. 0 65102.

[20] P. J. Teunissen, “Least-squares estimation of the integer GPS ambiguities,” in Proc. Section IV Theory Methodol., IAG General Meeting, Beijing, China, 1993, pp. 1–16.

[21] 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.

[22] S. Purivagraipong, S. Hodgatt, M. Unwin, and S. Kuntanapreeda, “Resolving integer ambiguity of GPS carrier phase difference,” IEEE Trans. Aerosp. Electron. Syst., vol. 46, no. 2, pp. 832–847, Apr. 2010.

[23] A. Douik, X. Liu, T. Ballal, T. Y. Al-Naffouri, and B. Hassibi, “Precise 3-D GNSS attitude determination based on Riemannian manifold optimization algorithms,” IEEE Trans. Signal Process., vol. 68, pp. 284–299, 2020.

[24] X. Liu, T. Ballal, and T. Y. Al-Naffouri, “GNSS attitude determination using a constrained wrapped least squares approach,” in Proc. IEEE/ION Position Location Navigat. Symp., 2020, pp. 1135–1139.

[25] K. Berntorp, A. Weiss, and S. D. Cairano, “Integer ambiguity resolution by mixture Kalman filter for improved GNSS precision,” IEEE Trans. Aerosp. Electron. Syst., vol. 56, no. 4, pp. 3170–3181, Aug. 2020.

[26] X. Liu, T. Ballal, and T. Y. Al-Naffouri, “GNSS ambiguity resolution and attitude determination by leveraging relative baseline and frequency information,” in Proc. 31st Int. Tech. Meeting Satell. Div. Inst. Navigat., 2018, pp. 1647–1660.

[27] Y. Li, X. Zou, B. Luo, W. Pan, L. Yan, and R. Kang, “AGAR: Array-geometry-aided ambiguity resolution for baseline growing with global navigation satellite systems,” IEEE Trans. Aerosp. Electron. Syst., vol. 58, no. 4, pp. 2632–2648, Aug. 2021.

[28] X. Liu, B. K. Ahmed, and T. Y. Al-Naffouri, “Attitude determination based on global navigation satellite system information,” U.S. Patent App. 17/269 390, Mar. 5, 2020.

[29] P. J. Teunissen, “A new method for fast carrier phase ambiguity estimation,” in Proc. IEEE Position Location Navigat. Symp., 1994, pp. 562–573.

[30] P. Teunissen, “The affine constrained GNSS attitude model and its multivariate integer least-squares solution,” J. Geodesy, vol. 86, no. 7, pp. 547–563, 2012.

[31] N. Nadarajah, P. J. G. Teunissen, and N. Raziq, “Instantaneous GPS–Galileo attitude determination: Single-frequency performance in satellite-deprived environments,” IEEE Trans. Veh. Technol., vol. 62, no. 7, pp. 2963–2976, Sep. 2013.

[32] P. J. Teunissen, “An optimality property of the integer least-squares estimator,” J. Geodesy, vol. 73, no. 11, pp. 587–593, 1999.

[33] G. Giorgi, P. J. Teunissen, S. Verhagen, and P. J. Buist, “Instantaneous ambiguity resolution in global-navigation-satellite-system-based attitude determination applications: A multivariate constrained approach,” J. Guid. Control Dyn., vol. 35, no. 1, pp. 51–67, 2012.

[34] P. J. G. Teunissen, “Integer least-squares theory for the GNSS compass,” J. Geodesy, vol. 84, no. 7, pp. 433–447, 2010.

[35] G. Giorgi and P. J. Teunissen, “Carrier phase GNSS attitude determination with the multivariate constrained LAMBDA method,” in Proc. IEEE Aerospace Conf., 2010, pp. 1–12.

[36] X. Liu, T. Ballal, H. Chen, and T. Y. Al-Naffouri, “Constrained wrapped least squares: A tool for high-accuracy GNSS attitude determination,” IEEE Trans. Instrum. Meas., vol. 71, 2022, Art no. 8005313.

[37] D. Medina, V. Centrone, R. Ziebold, and J. García, “Attitude determination via GNSS carrier phase and inertial aiding,” in Proc. 32nd Int. Tech. Meeting Satell. Div. Inst. Navigat., 2019, pp. 2964–2979.

[38] J. Kuang and S. Tan, “GPS-based attitude determination of gyrostat satellite by quaternion estimation algorithms,” Acta Astronautica, vol. 51, no. 11, pp. 743–759, 2002.

[39] N. Lovren and J. Pieper, “Error analysis of direction cosines and quaternion parameters techniques for aircraft attitude determination,” IEEE Trans. Aerosp. Electron. Syst., vol. 34, no. 3, pp. 983–989, Jul. 1998.
[40] G. Giorgi, “GNSS carrier phase-based attitude determination: Estimation and applications,” Doctoral thesis, Delft Institute of Earth Observation and Space Systems, Delft University of Technology, Delft, The Netherlands, 2011.
[41] M. D. Shuster, “Constraint in attitude estimation part II: Unconstrained estimation,” J. Astronautical Sci., vol. 51, no. 1, pp. 75–101, 2003.
[42] P. J. G. Teunissen, “A-PPP: Array-aided precise point positioning with global navigation satellite systems,” IEEE Trans. Signal Process., vol. 60, no. 6, pp. 2870–2881, Jun. 2012.
[43] N. Boumal, An Introduction to Optimization on Smooth Manifolds. Cambridge, U.K.: Cambridge Univ. Press, Aug. 2020.
[44] H. Sato, Riemannian Optimization and Its Applications. Berlin, Germany: Springer, 2021.
[45] W. Ring and B. Wirth, “Optimization methods on Riemannian manifolds and their application to shape space,” SIAM J. Optim., vol. 22, no. 2, pp. 596–627, 2012.
[46] S. T. Smith, “Optimization techniques on Riemannian manifolds,” Fields Inst. Commun., vol. 3, no. 3, pp. 113–135, 1994.
[47] P.-A. Absil, R. Mahony, and R. Sepulchre, Optimization Algorithms on Matrix Manifolds. Princeton, NJ, USA: Princeton Univ. Press, 2009.
[48] C. Liu and N. Boumal, “Simple algorithms for optimization on Riemannian manifolds with constraints,” Appl. Math. Optim., vol. 82, no. 3, pp. 949–981, 2020.
[49] P. J. Teunissen and A. Kleusberg, GPS for Geodesy. Berlin, Germany: Springer, 2012.
[50] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, “Manopt, a MATLAB toolbox for optimization on manifolds,” J. Mach. Learn. Res., vol. 15, no. 42, pp. 1455–1459, 2014. [Online]. Available: https://www.manopt.org
[51] ANavS Multi-Sensor Modules, ANavS GmbH–Advanced Navigation Solutions, Munich, Germany. [Online]. Available: https://anavs.com/multi-sensor-rtk-module/

Xing Liu (Member, IEEE) received the B.S. degree in electronic and information engineering from the China University of Petroleum (East China), Qingdao, China, in 2014 and the M.S. degree in signal and information processing from the University of Chinese Academy of Sciences, Beijing, China, in 2017. He is currently working toward the Ph.D. degree in electrical and computer engineering with the King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia. His research focuses on GNSS positioning and attitude determination.

Tareq Y. Al-Naffouri (Senior Member, IEEE) received the B.S. (hons.) degree in mathematics from the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, in 2004. Between 2004 and 2006, he was with the Department of Electrical Engineering, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia. He has more than 300 publications in journal and conference proceedings and 20 issued/pending patents. His research interests include signal processing and their applications to wireless communications and localization, machine learning, and network information theory.

Dr. Al-Naffouri is the recipient of the IEEE Education Society Chapter Achievement Award in 2008 and Al-Marai Award for innovative research in communication in 2009. He was an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING 2013–2018.

Mohanad Ahmed received the B.Sc. degree in electrical and electronic engineering from the University of Khartoum, Khartoum, Sudan, in 2011 and the M.Sc. degree in electrical engineering from the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, in 2015. Between 2016 and 2018, he was a Lecturer with the University of Khartoum. He is currently a Research Engineer with the King Abdullah University of Science and Technology, Thuwal, Saudi Arabia. His research interests include signal processing and embedded systems focus on indoor localization.

Tarig Ballal (Member, IEEE) received the B.Sc. (Hons.) degree in electrical engineering from the University of Khartoum, Khartoum, Sudan, in 2001 and the M.Sc. degree in telecommunications from the Blekinge Institute of Technology, Karlskrona, Sweden, in 2005, and the Ph.D. degree in computer science from the School of Computer Science and Informatics, University College Dublin, Dublin, Ireland, in 2011. From April 2011 to July 2012, he was a Research Engineer with BiancaMed Ltd., Dublin, Ireland, and University College Dublin, Dublin, Ireland. Since September 2012, he has been with the Electrical and Computer Engineering Department, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia, as a Postdoctoral Fellow and currently as a Research Scientist. His current research focuses on regularisation and robust estimation methods, image and signal processing, machine learning, acoustic sensing, and tracking and localization.