remote sensing

MDPI

Article

A Cooperative GNSS Vector-DLL (CoVDLL) Method for Multiple UAVs Positioning

Chuntao Li [1], Xinru Wang [1], Changhui Jiang [1,*], Zikang Su [1], Shoubin Chen [2] and Yuwei Chen [3]

[1] College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China; lct13770925493@163.com (C.L.); xr_wang@nuaa.edu.cn (X.W.); zk_su@nuaa.edu.cn (Z.S.)
[2] Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen 518060, China; shoubin.chen@whu.edu.cn
[3] School of Physics and Optoelectronic Engineering, Hangzhou Institute for Advanced Study (UCAS), Hangzhou 310024, China; yuwei.chen@nls.fi
* Correspondence: changhui.jiang@nuaa.edu.cn

Abstract

Currently, the Global Navigation Satellite System (GNSS) plays a critical role in providing position information for UAVs. Traditional GNSS receivers typically employ the scalar-tracking (ST) method to track signals and extract observations. An advanced alternative to ST is the Vector Delay Lock Loop (VDLL), which enables more reliable navigation solution estimation for GNSS receivers. To enhance GNSS positioning performance for UAVs, this paper proposes a cooperative VDLL (CoVDLL) that incorporates ranging information. While single VDLL (S-VDLL) explores the inherent relationship between signal tracking parameters and navigation solutions, the CoVDLL leverages signal tracking parameters from multiple UAVs along with their ranging data. To optimize navigation solution estimation in the CoVDLL, a Factor Graph Optimization (FGO) algorithm is employed to realize the navigation solution's optimal estimation. Experiments conducted under various settings demonstrate that the CoVDLL method reduces positioning errors by an average of approximately 30% compared to the single-VDLL approach.

Keywords: GNSS; cooperative vector tracking; factor graph optimization

check for updates

Academic Editor: Gino Dardanelli

Received: 8 March 2025
Revised: 8 June 2025
Accepted: 12 June 2025
Published: 23 June 2025

Citation: Li, C.; Wang, X.; Jiang, C.; Su, Z.; Chen, S.; Chen, Y. A Cooperative GNSS Vector-DLL (CoVDLL) Method for Multiple UAVs Positioning. *Remote Sens.* **2025**, *17*, 2156. https://doi.org/10.3390/rs17132156

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The Global Navigation Satellite System (GNSS) has been extensively employed across various applications, including Unmanned Aerial Vehicles (UAVs) [1,2]. GNSS receivers provide Position, Velocity, and Time (PVT) information when enough satellites are in view. In the UAV positioning applications, this PVT information serves as critical guidance data, enabling the UAV to follow predefined trajectories and land accurately at designated destinations [1–3]. Moreover, for UAV swarm operations, precise positioning is essential for organizing and controlling the swarm to execute specific tasks effectively. The accuracy of this positioning information is a pivotal factor that determines the performance of UAV swarms in mission execution. Therefore, investigating methods to enhance GNSS positioning accuracy, particularly for UAV swarm applications, is of paramount importance.

Two promising approaches have emerged with significant potential to improve GNSS positioning accuracy for UAV swarms. The first approach involves advanced GNSS signal tracking and positioning technologies, i.e., vector tracking (VT). The second approach leverages cooperative ranging information among UAV swarms [4,5]. Cooperative positioning

and navigation have proven effective in enhancing both accuracy and robustness, facilitating mutual assistance among individual UAVs within the swarm [4–6]. Inter-ranging measurements are exploited in the cooperative GNSS for UAV swarms, in which the GNSS is built based on the scalar tracking [7–11]. UWB is usually employed to measure the inter-ranging information [7–11]. These methods hold great promise for advancing the capabilities of UAV swarms in complex operational environments.

VT is an advanced GNSS signal tracking technology that fully leverages the fact that all channels share the same Position, Velocity, and Time (PVT) information in a receiver [12,13]. In VT, a navigation filter—typically a Kalman filter—is used to simultaneously perform signal tracking and PVT information estimation [13–18]. Measurements such as frequency errors and code phase errors, derived from frequency and code discriminators, are input into the navigation filter to estimate the system states. These estimated states, along with the PVT information, are used to update and predict signal tracking parameters [18]. Previous studies have extensively investigated, compared, and presented both VT and scalar-tracking technologies [19–27]. A wealth of research and experimental data demonstrate that VT outperforms scalar tracking in various aspects [19–27]. The advantages of VT over scalar tracking can be summarized as follows: (1) Improved sensitivity: VT maintains lock on weak signals better than scalar tracking, owing to its collective processing of signals; (2) robustness against interference: by integrating data from multiple satellites, VT is more resistant to signal blockage and interference, providing higher accuracy in challenging environments such as urban canyons or dense foliage; (3) enhanced multipath mitigation: the centralized processing of signals enables VT to mitigate multipath effects, as the combined signals can help distinguish between direct and reflected signals; (4) faster reacquisition: VT accelerates the signal reacquisition process, quickly recovering signals after temporary losses due to its joint processing of signals.

In addition to advancements in GNSS signal tracking technologies, significant research has been conducted on cooperative GNSS algorithms to enhance positioning accuracy [7,9,11,28–32] Unlike conventional GNSS systems that depend on standalone receivers operating in isolation, cooperative GNSS employs a collaborative approach among multiple receivers to improve overall performance, accuracy, and reliability. This innovative methodology involves the exchange of information between multiple GNSS receivers to enhance both positioning accuracy and system robustness [33,34].

The cooperative framework can be implemented through various strategies, including the sharing of raw satellite observations, computed positions, or error correction data. The core principle of this approach lies in utilizing spatial diversity and the complementary perspectives of different receivers to address common challenges such as signal obstructions, multipath effects, and interference [7,11,28,33,34]. Cooperative GNSS systems can range from basic data-sharing configurations to sophisticated networked architectures. At its most fundamental level, receivers exchange raw GNSS data, which is then processed collectively to refine positioning solutions. More advanced implementations involve network-based systems where a centralized server aggregates data from multiple receivers, processes the information, and disseminates enhanced positioning data back to users [7,11,28].

The advantages of cooperative GNSS can be categorized as follows: (a) Enhanced positioning accuracy: by integrating data from multiple sources, cooperative GNSS can substantially mitigate errors associated with atmospheric disturbances, satellite clock biases, and orbital inaccuracies; (b) improved system robustness: the collaborative nature of this approach effectively reduces the impact of signal blockages and multipath effects, which are particularly prevalent in urban environments and other challenging operational scenarios. Their advantages are demonstrated with the cooperative scalar tracking-based GNSS [7,11,28,29,34]. In view of the difference between the VT and scalar tracking, coop-

erative VT can realize mutual aiding between the GNSS receivers and improve the VT performance. On the contrary, the scalar-tracking GNSS cannot realize the signal-tracking mutual aiding because the navigation solutions cannot be fed back to aid the signal tracking in the scalar tracking.

Recently, researchers paid attention to the robustness of the VT; some adaptive methods are proposed to enhance the navigation data estimation and signal tracking [23,29]. The cooperative scheme utilizing the mutual information from the UAV swarm is ignored, and it has great potential in improving the VT performance. Given the distinct advantages of VT and cooperative GNSS technologies, it is crucial to explore their integration in the context of UAV positioning and navigation applications. This paper introduces a novel CoVDLL method designed to enhance the positioning accuracy of UAV swarms. The key contributions of this work are outlined as follows:

(1) In view of the VT technology utilizing the navigation data to predict signal tracking parameters, full usage of the mutual navigation information from the UAV swarm has great potential to improve the positioning accuracy. This study explores the application of VT technology in UAV GNSS positioning and proposes a CoVDLL framework to improve the positioning accuracy of UAV swarms. Additionally, an inter-ranging-based cooperative positioning model for UAVs is developed and thoroughly investigated.

(2) Based on the inter-ranging information, the state vector of the CoVDLL is constructed. A Factor Graph Optimization (FGO) method is incorporated with the CoVDLL to realize the optimal estimation of the UAV swarms' position. A cost function is defined, including the VDLL measurement factor, state transformation factor, and the inter-ranging factor. A nonlinear solver is employed to solve for the state estimation with the cost function.

Reminder: the paper is organized as Section 2 describes the vector tracking mechanism, equations, and the navigation filter; Section 3 introduces the cooperative vector tracking and the FGO method; Experiments and the results are presented and analyzed in the following Section 4. Finally, the results are discussed, future works are suggested, and the conclusion is draw.

2. Phase Lock Loop Assisted Vector Delay Lock Loop (PLL-A-DLL)

As previously mentioned, a conventional GNSS receiver employs a Phase-Locked Loop (PLL) to track carrier signals and a Delay-Locked Loop (DLL) to track the C/A ranging code. Pseudo-range and pseudo-range rate measurements are independently extracted from all tracking channels, and a least squares or Kalman Filter algorithm is used to compute the Position, Velocity, and Time (PVT) information [7,27,33,34]. In this study, a PLL-assisted Vector Delay-Locked Loop (PLL-A-VDLL) architecture is adopted as the foundational framework for developing the cooperative VDLL. The structure of the PLL-A-VDLL GNSS receiver is illustrated in Figure 1.

The GNSS signal undergoes initial sampling and down-conversion to an intermediate frequency (IF). The acquired satellite signals within the IF signal are then distributed to individual channels for signal tracking. In each tracking channel, the incoming signal is multiplied with local carrier signal replicas in both the in-phase and quadrature arms [22,23]. Subsequently, the signal is correlated with three local code replicas, spaced at intervals of 0.5 chips. Following integration and dumping, carrier and code discriminators are employed to extract the carrier phase errors and code phase errors between the incoming signals and the local signal replicas.

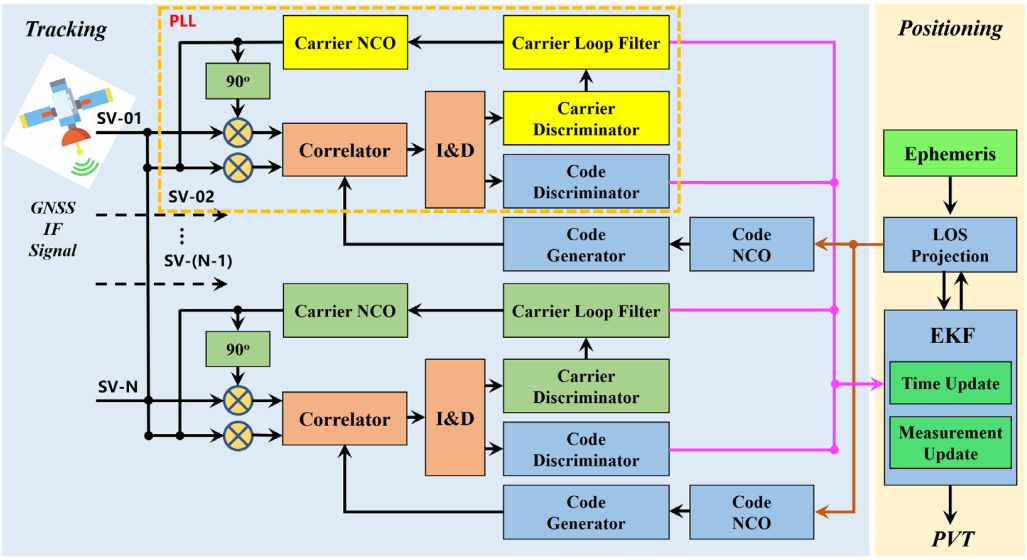


Figure 1. Working flow of the PLL-Assisted-VDLL GNSS receiver.

In the carrier signal tracking process, the discriminator outputs from all tracking channels are fed back into the carrier loop filter. The local carrier signal replica is then updated by the carrier Numerically Controlled Oscillator (NCO) based on the outputs of the carrier loop filter. The results of the carrier signal tracking, specifically the Doppler shifts, serve as measurements for the navigation processor, which utilizes an Extended Kalman Filter (EKF). For code tracking, the code discriminator outputs from all channels are used as inputs to the navigation processor [7,27,33,34].

The navigation solutions generated by the EKF are fed back to the tracking channels to adjust the code signal tracking parameters. New local signal replicas are generated and multiplied with the incoming signals. Updated measurements from the carrier and code discriminators are then processed by the navigation processor EKF to produce new navigation solutions. As illustrated in Figure 1, prior ephemeris data are required while the navigation solutions are fed back to update the signal tracking parameters. Specifically, the Line-Of-Sight (LOS) vector is calculated using the satellite positions from the ephemeris and the user receiver position. The changes in Position, Velocity, and Time (PVT) are projected onto the LOS direction between the satellite and the user receiver to determine the signal tracking parameters [23].

PVT changes are regarded as the state vector in the EKF. The state vector of the navigation processor EKF. The error state vector $\mathbf{X}$ is defined as

$$\mathbf{X} = [\Delta\mathbf{p}\ \Delta\mathbf{v}\ \Delta t_b\ \Delta t_d]^T_{8\times1} \tag{1}$$

where state $\Delta\mathbf{p} = \left[\Delta p_x\ \Delta p_y\ \Delta p_z\right]$ denotes the three-dimensional position error vector in the Earth-Centered-Earth-Fixed (ECEF) coordinates; $\Delta\mathbf{v} = \left[\Delta v_x\ \Delta v_y\ \Delta v_z\right]$ denotes the three-dimensional velocity errors vector; Δt_b denote the time bias error and Δt_d denotes the time drift error; in addition, the units of the Δt_b and Δt_d are meter and meters per second.

The state propagation model is expressed as

$$\mathbf{X}_{k+1} = \begin{bmatrix} \Delta\mathbf{p} \\ \Delta\mathbf{v} \\ \Delta t_b \\ \Delta t_d \end{bmatrix}_{k+1} = \mathbf{F}_{k,k+1}\cdot\mathbf{X}_k + \mathbf{W} = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathrm{T}\cdot\mathbf{I}_{3\times3} & \mathbf{0}_{3\times2} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times2} \\ \mathbf{0}_{2\times3} & \mathbf{0}_{2\times3} & \mathbf{T}_{2\times2} \end{bmatrix}_{k,k+1} \cdot \begin{bmatrix} \Delta\mathbf{p} \\ \Delta\mathbf{v} \\ \Delta t_b \\ \Delta t_d \end{bmatrix}_k + \mathbf{W} \tag{2}$$

$$\mathbf{T}_{2\times 2} = \begin{bmatrix} 1 & \mathrm{T} \\ 0 & 1 \end{bmatrix} \tag{3}$$

where $\mathbf{F}_{k,k+1}$ denotes the state transformation matrix, T denotes the updating period and $\mathbf{W}$ denotes the process noise vector.

In the EKF, pseudo-range error $\Delta\rho$ and pseudo-range rate error $\Delta\dot{\rho}$ from each tracking channel are employed in the measurement vector $\mathbf{Z}$. Pseudo-range error for i^{th} satellite at $(k+1)^{th}$ epoch is expressed as

$$\Delta\rho^{i}_{k+1} = \Delta\tau^{i}_{k+1} \cdot \frac{c}{f_{C/A}} \tag{4}$$

where $\Delta\tau^{i}_{k+1}$ denotes the code phase error extracted from the code discriminator; $\Delta\rho^{i}_{k+1}$ denotes the pseudo-range error; the constant c denotes the light speed; $f_{C/A}$ is the C/A code frequency, which is 1.023 MHz. In the VDLL, the relationship between the code phase error and position is illustrated in the following Figure 2. The position error vector is projected to the LOS vector between the user receiver and the satellite. The relationship between the position increment and the code phase error is modeled as

$$\Delta\tau^{i}_{k+1} = (\Delta\mathbf{p})^{T} \cdot \mathbf{LOS}^{i}_{k+1} + c \cdot \Delta t_{b,k+1} + \omega^{i}_{k+1} \tag{5}$$

where $\cdot\mathbf{LOS}^{i}_{k+1}$ denotes the Line-Of-Sight (LOS) vector at the $(k+1)^{th}$ epoch for i^{th} satellite; $\Delta t_{b,k+1}$ denotes the time bias increment; and ω^{i}_{k+1} denotes the C/A code noise. One thing to note is that the model is constructed assuming that the Ionosphere and Troposphere factor-induced code phase errors are compensated.

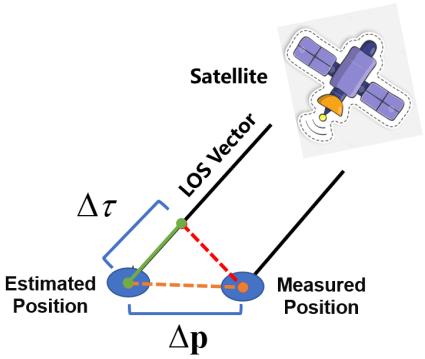


Figure 2. Relationship between the position error vector and the code phase error.

Similarly, the pseudo-range rate error $\Delta\dot{\rho}$ is expressed as

$$\Delta\dot{\rho}^{i}_{k+1} = \Delta f^{i}_{Doppler} \cdot \frac{c}{f_{L1}} \tag{6}$$

where $f^{i}_{Doppler}$ denotes the doppler shift frequency extracted from the carrier signal tracking with unit Hz; the variable f_{L1} denotes the GPS L1 carrier signal frequency (1575.42 MHz). The relationship between the position increment and the code phase error is modeled as

$$\Delta f^{i}_{Doppler} = (\Delta\mathbf{v})^{T} \cdot \mathbf{LOS}^{i}_{k+1} + c \cdot \Delta t_{d,k+1} + \eta^{i}_{k+1} \tag{7}$$

where $\mathbf{LOS}^{i}_{k+1}$ denotes the LOS vector at the $(k+1)^{th}$ epoch for i^{th} satellite; $\Delta t_{d,k+1}$ denotes the time drift increment, and η^{i}_{k+1} denotes the carrier frequency doppler measurement noise.

Assuming there are N well-tracked in-view satellites, the measurement model is expressed as

$$\mathbf{Z}_{k+1} = \begin{bmatrix} \Delta\boldsymbol{\rho}_{k+1} \\ \Delta\dot{\boldsymbol{\rho}}_{k+1} \end{bmatrix} = \mathbf{H}_{k+1} \cdot \mathbf{X}_{k+1} + \boldsymbol{\Lambda} \tag{8}$$

$$\Delta\boldsymbol{\rho}_{k+1} = \left[\Delta\rho^1_{k+1} \ldots \Delta\rho^i_{k+1} \ldots \Delta\rho^N_{k+1}\right]^T \tag{9}$$

$$\Delta\dot{\boldsymbol{\rho}}_{k+1} = \left[\Delta\dot{\rho}^1_{k+1} \ldots \Delta\dot{\rho}^i_{k+1} \ldots \Delta\dot{\rho}^N_{k+1}\right]^T \tag{10}$$

where the superscript i denote the satellite index, $\boldsymbol{\Lambda}$ denotes the measurement noise vector, $\mathbf{H}$ denotes the observation matrix expressed as

$$\mathbf{H} = \begin{bmatrix} -\mathbf{LOS}^1_x & -\mathbf{LOS}^1_y & -\mathbf{LOS}^1_z & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & & & \vdots \\ -\mathbf{LOS}^i_x & -\mathbf{LOS}^i_y & -\mathbf{LOS}^i_z & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & & & \vdots \\ -\mathbf{LOS}^N_x & -\mathbf{LOS}^N_y & -\mathbf{LOS}^N_z & 0 & 0 & 1 \\ 0 & 0 & 0 & -\mathbf{LOS}^1_x & -\mathbf{LOS}^1_y & -\mathbf{LOS}^1_z \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & -\mathbf{LOS}^i_x & -\mathbf{LOS}^i_y & -\mathbf{LOS}^i_z \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & -\mathbf{LOS}^N_x & -\mathbf{LOS}^N_y & -\mathbf{LOS}^N_z \end{bmatrix} \tag{11}$$

where $\mathbf{LOS}^i = \left[-\mathbf{LOS}^i_x - \mathbf{LOS}^i_y - \mathbf{LOS}^i_z\right]$ denotes the LOS vector from the user receiver to the i^{th} satellite.

With the system propagation and measurement models set up in the Equations (1)–(11), the Extended Kalman Filter (EKF) is executed. In the EKF, the first step includes the state vector and covariance matrix prediction.

$$\mathbf{X}^-_{k+1} = \mathbf{F}_{k,k+1} \cdot \mathbf{X}_k \tag{12}$$

$$\boldsymbol{\Gamma}^-_{k+1} = \mathbf{F}_{k,k+1} \cdot \boldsymbol{\Gamma}_k \cdot (\mathbf{F}_{k,k+1})^T + \mathbf{Q} \tag{13}$$

where $\mathbf{X}^-_{k+1}$ means the predicted state according to the system propagation model; $\mathbf{Q}$ is the prediction noise covariance diagonal matrix with the size of 8×8 and $\boldsymbol{\Gamma}^-_{k+1}$ is the predicted covariance matrix.

$$\hat{\mathbf{X}}_{k+1} = \mathbf{X}^-_{k+1} + \mathbf{K}_{k+1} \cdot \left(\mathbf{Z}_{k+1} - \mathbf{H}_{k+1} \cdot \mathbf{X}^-_{k+1}\right) \tag{14}$$

$$\mathbf{K}_{k+1} = \boldsymbol{\Gamma}^-_{k+1} \cdot (\mathbf{H}_{k+1})^T \left[\mathbf{H}_{k+1} \cdot \boldsymbol{\Gamma}^-_{k,k+1} (\mathbf{H}_{k+1})^{\mathrm{T}} + \mathbf{R}\right]^{-1} \tag{15}$$

$$\boldsymbol{\Gamma}_{k+1} = (\mathbf{I}_{2\times 2} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}) \cdot \boldsymbol{\Gamma}^-_{k+1} \tag{16}$$

where $\mathbf{K}_{k+1}$ means the Kalman gain matrix; $\mathbf{R}$ denotes the measurement noise covariance diagonal matrix with size of 8×8 size; $\mathbf{I}_{8\times 8}$ means an identity matrix with 8×8 size here.

3. Cooperative Vector Delay Lock Loop

As aforementioned, in the VDLL, signal tracking is carried out by utilizing the inherent relationship between the navigation solutions and signal tracking parameters. Superior performance of the VDLL comes from fully utilizing the characteristic that all the tracking

channels share the same navigation solutions. Predicted position is employed to calculate the code phase and accomplish the local signal replica updating. In this way, the channel with strong signals can help the channels with weak signals. In the multiple UAVs, ranging measurement between the UAV individuals could be utilized. Ranging information can be utilized to construct cooperative VDLL with multiple UAVs. Figure 3 shows the schematic illustration of the multiple UAV system. VDLLs from different UAVs are correlated through the inter-ranging information between these UAVs. Navigation solutions are estimated cooperatively with the VDLLs. An FGO method is employed to construct the CoVDLL system.

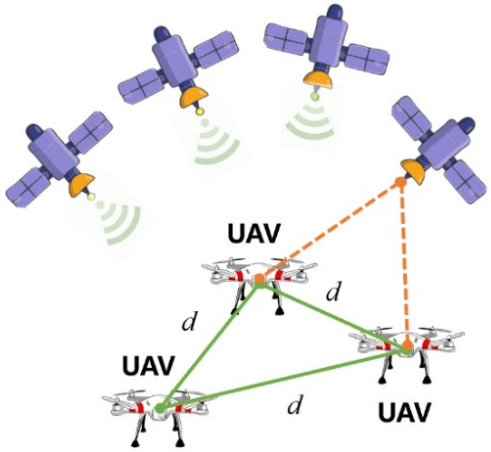


Figure 3. Illustration of the multiple UAVs.

3.1. Factor Graph Optimization

Factor Graph Optimization (FGO) is an efficient algorithm widely used to solve complex optimization problems by decomposing them into smaller, more manageable components [9,29]. In the FGO, the variables are nodes, and the constraints are regarded as factors connecting these nodes. The optimization process involves iteratively updating the variables based on messages passed between nodes and factors, aiming to converge to the optimal solution. One of the key advantages of the FGO is its ability to handle various types of constraints and dependencies in a unified framework, making it applicable to a wide range of problems across different domains such as computer vision, robotics, and machine learning [9,29]. By exploiting the graphical structure of the problem, the FGO algorithm can efficiently explore the solution space and find high-quality solutions even in large-scale, highly nonlinear optimization problems.

In the factor graph, the states' probability relationship is usually described by a joint probability density function of a set of variables.

$$f(\theta_1, \ldots, \theta_N) = \prod_{i=1}^{N} f(\theta_i | \theta_1, \ldots, \theta_{i-1}) \tag{17}$$

where $\theta = (\theta_1, \ldots, \theta_N)$ denotes a set of variables, $f(\cdot)$ denotes the joint density function.

Based on the joint probability density function, optimal estimates of the states are realized through solving for the maximum values of the transformed global function. The state's optimal estimation is expressed as

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}(G(\theta_1, \theta_2, \ldots, \theta_N)) \tag{18}$$

where $G(\cdot)$ is the global function, which is normally divided into different factors.

$$G(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N) = \prod_{j=1}^{m} f(\boldsymbol{\lambda}_j) \tag{19}$$

where $\boldsymbol{\lambda}_j \subseteq \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$.

Therefore, the state estimation can be expressed as

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}(G(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_N)) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left(\prod_{j=1}^{m} f(\boldsymbol{\lambda}_j) \right) \tag{20}$$

Each factor has a corresponding cost function, which is usually expressed as

$$f(\boldsymbol{\lambda}_j) \propto \exp\left(-\frac{1}{2} \|\boldsymbol{\xi}\|_{\sum}^2 \right) \tag{21}$$

where the $\boldsymbol{\xi}$ denotes the residual vector, and the function $\|\boldsymbol{\xi}\|_{\sum}^2$ is described as

$$\|\boldsymbol{\xi}\|_{\sum}^2 = \boldsymbol{\xi}^T \sum^{-1} \boldsymbol{\xi} \tag{22}$$

Finding the optimal estimates of the states can be described as

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}}(-\log G(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_N)) = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{2} \sum_i \|\boldsymbol{\xi}_i\|_{\sum_i}^2 \tag{23}$$

Normally, we solve for the state increments and correct the set initial states. Optimal state estimates are realized while the residuals or the iteration count meets set thresholds. Increments estimation can be described as

$$\Delta\boldsymbol{\theta}_i = \underset{\Delta\boldsymbol{\theta}_i}{\operatorname{argmin}}(-\ln f(\Delta\boldsymbol{\theta}_i)) = \left\| \mathbf{J}\left(\hat{\boldsymbol{\theta}}_i\right) \cdot \Delta\boldsymbol{\theta}_i - \mathbf{B}\left(\hat{\boldsymbol{\theta}}_i\right) \right\|^2 \tag{24}$$

where $\mathbf{J}\left(\hat{\boldsymbol{\theta}}_i\right)$ is the Jacobian matrix, $\mathbf{B}\left(\hat{\boldsymbol{\theta}}_i\right)$ denotes the residuals. The iteration process is summarized as:

First Step: set up the cost functions for each state constraint and define the state's initial state values $\hat{\boldsymbol{\theta}}_i = \boldsymbol{\theta}_0$

Second Step: determining the Jacobian matrix $\mathbf{J}\left(\hat{\boldsymbol{\theta}}_i\right)$ and the residuals $\mathbf{B}\left(\hat{\boldsymbol{\theta}}_i\right)$ at the value $\hat{\boldsymbol{\theta}}_i = \boldsymbol{\theta}_0$;

Third Step: calculating and updating the state increments $\Delta\mathbf{X}_i$ using the equation $\mathbf{J}\left(\hat{\boldsymbol{\theta}}_i\right) \cdot \Delta\boldsymbol{\theta}_i = \mathbf{B}\left(\hat{\boldsymbol{\theta}}_i\right)$

Fourth Step: correcting and updating the state vector $\hat{\boldsymbol{\theta}}_i = \boldsymbol{\theta}_i + \Delta\boldsymbol{\theta}_i$.

Repeat the steps (1–4) until the iteration count satisfies the pre-defined threshold or the residuals meet the pre-defined threshold. The iterative process is presented in the Figure 4.

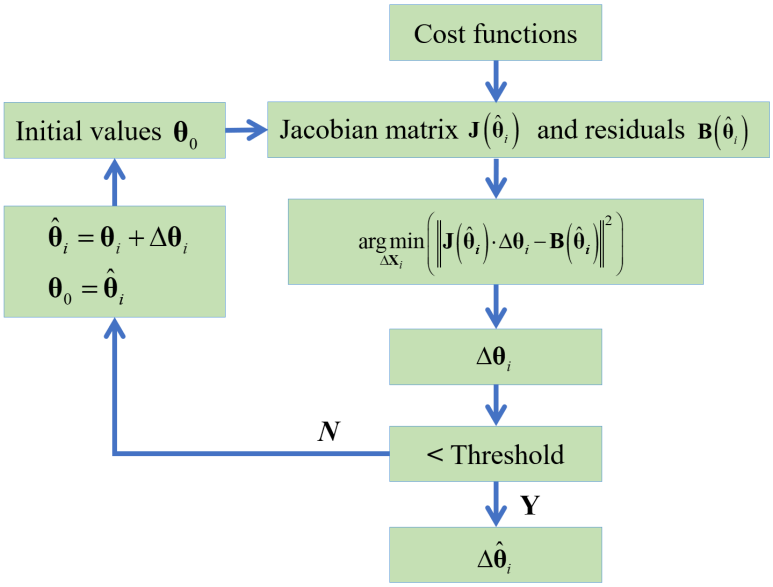


Figure 4. Flowchart of the FGO algorithm.

3.2. CoVDLL

In the CoVDLL system, different VDLLs in the UAVs are correlated with the ranging measurements. In another way, the VDLL states are correlated with the ranging measurements, which are regarded as constraints in the FGO algorithm. Based on the factor graph theory presented in sub-Section 3.1, the CoVDLL factor graph is presented in Figure 5. There are three factors in the CoVDLL system, i.e., state transformation factor, measurement factor, and cooperative ranging factor. For clearly presenting the relationship between the constraints and factors, Figure 5 presents the factor graph with the assumption that there are three UAVs. In addition, the following descriptions are illustrated with the assumption that there are three UAVs in the system. The state vector of the multiple VDLLs is defined as

$$\mathbf{X} = \left(\left[\mathbf{X}^{U_1}, \mathbf{X}^{U_2}, \mathbf{X}^{U_3}\right]^T\right) \tag{25}$$

where

$$\mathbf{X}^{U_1} = \left(\left[\mathbf{X}_1^{U_1}, \mathbf{X}_2^{U_1}, \ldots, \mathbf{X}_k^{U_1}\right]^T\right)$$
$$\mathbf{X}^{U_2} = \left(\left[\mathbf{X}_1^{U_2}, \mathbf{X}_2^{U_2}, \ldots, \mathbf{X}_k^{U_2}\right]^T\right)$$
$$\mathbf{X}^{U_3} = \left(\left[\mathbf{X}_1^{U_3}, \mathbf{X}_2^{U_3}, \ldots, \mathbf{X}_k^{U_3}\right]^T\right)$$

the superscript U_1, U_2 and U_3 denote the index of the UAV, and the subscript k denotes the epoch.

In the CoVDLL system, there are a total of three factors defined as the state transformation factor, the measurement factor, and the cooperative ranging factor.

(1) State transformation factor

As described in the Equation (2), the consecutive state vectors are correlated with the state propagation model. The state transformation factor is described as

$$\mathbf{X}_{k+1}^{U_i} = \mathbf{F}_{k,k+1}^{U_i} \cdot X_k^{U_i} + N\left(0, \boldsymbol{\Sigma}_{k+1}^{U_i,trans}\right) \tag{26}$$

$$\mathbf{X}_{k+1} = h_{trans}(\mathbf{X}_k) + N\left(0, \boldsymbol{\Sigma}_{k+1}^{trans}\right) \tag{27}$$

The cost function is defined as

$$\left\|\mathbf{E}_{k+1}^{trans}\right\|_{\mathbf{\Sigma}_{k+1}^{trans}}^{2}=\left\|\mathbf{X}_{k+1}-h_{trans}(\mathbf{X}_{k})\right\|_{\mathbf{\Sigma}_{k+1}^{trans}}^{2} \tag{28}$$

where $\mathbf{E}_{k+1}^{trans}$ denotes the costs at $k+1$ epoch, the $\mathbf{\Sigma}_{k+1}^{trans}$ is the covariance matrix.

(2) Measurement Factor

Relationship between the state vector and the measurements is defined as

$$\mathbf{Z}_{k+1}^{U_i}=\mathbf{H}_{k+1}^{U_i}\cdot X_{k+1}^{U_i}+N\left(0,\mathbf{\Sigma}_{k+1}^{U_i,M}\right) \tag{29}$$

$$\mathbf{Z}_{i+1}=h_{M}(\mathbf{X}_{i+1})+N\left(0,\mathbf{\Sigma}_{i+1}^{M}\right) \tag{30}$$

Similarly, the cost function of the measurement factor is defined as

$$\left\|\mathbf{E}_{i+1}^{M}\right\|_{\mathbf{\Sigma}_{i+1}^{M}}^{2}=\left\|\mathbf{Z}_{i+1}-h_{M}(\mathbf{X}_{i+1})\right\|_{\mathbf{\Sigma}_{i+1}^{M}}^{2} \tag{31}$$

where $\mathbf{E}_{k+1}^{M}$ denotes the costs at $k+1$ epoch, the $\mathbf{\Sigma}_{k+1}^{trans}$ is the covariance matrix.

(3) Cooperative Ranging Factor

Relationship between the cooperative ranging measurement $\mathrm{d}_{k}^{U_i^j}$ is defined as

$$\left|\mathrm{d}_{k+1}^{U_i^j}-\mathrm{d}_{k}^{U_i^j}\right|=\left\|\Delta\mathbf{p}_{k+1}^{U_i}-\Delta\mathbf{p}_{k+1}^{U_j}\right\|+N\left(0,\mathbf{\Sigma}_{k+1}^{U_i^j,D}\right) \tag{32}$$

$$\left\|\mathbf{E}_{k+1}^{D_i^j}\right\|_{\mathbf{\Sigma}_{k+1}^{D}}^{2}=\left\|\left|d_{k+1}^{U_i^j}-d_{k}^{U_i^j}\right|-\left\|\Delta\mathbf{p}_{k+1}^{U_i}-\Delta\mathbf{p}_{k+1}^{U_j}\right\|\right\|_{\mathbf{\Sigma}_{k+1}^{D_i^j}}^{2}=\left\|\mathbf{Z}_{k+1}^{D_i^j}-h_{D_i^j}(\mathbf{X}_{k+1})\right\|_{\mathbf{\Sigma}_{k+1}^{D_i^j}}^{2} \tag{33}$$

In summary, in the CoVDLL system, the global cost function is defined as

$$\mathbf{X}^{*}=\operatorname{argmin}\left(\sum_{k+1}^{N}\left(\left\|\mathbf{E}_{k+1}^{trans}\right\|_{\mathbf{\Sigma}_{k+1}^{trans}}^{2}+\left\|\mathbf{E}_{i+1}^{M}\right\|_{\mathbf{\Sigma}_{i+1}^{M}}^{2}+\left\|\mathbf{E}_{1}^{D_1^2}{}_{k+1}\right\|_{\mathbf{\Sigma}_{k+1}^{D}}^{2}+\left\|\mathbf{E}_{1}^{D_1^3}{}_{k+1}\right\|_{\mathbf{\Sigma}_{k+1}^{D}}^{2}+\left\|\mathbf{E}_{2}^{D_2^3}{}_{k+1}\right\|_{\mathbf{\Sigma}_{k+1}^{D}}^{2}\right)\right) \tag{34}$$

where the function $\operatorname{argmin}(\cdot)$ means minimizing the cost function.

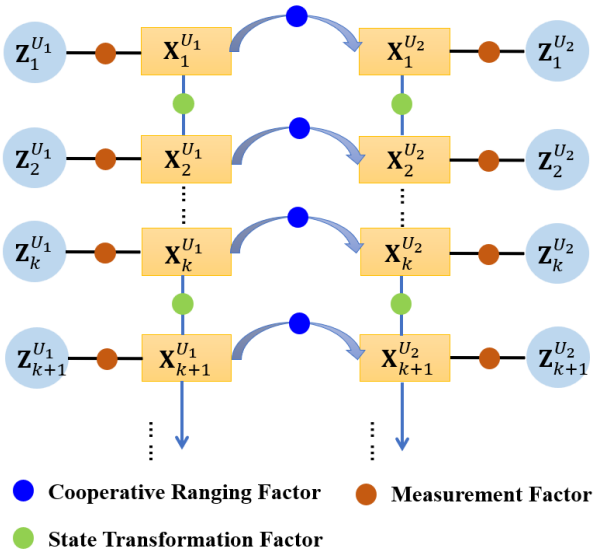


Figure 5. Factor graph of the CoVDLL.

4. Experimental Results and Analysis

With the methods described in the previous sections, GNSS intermediate frequency (IF) datasets are generated to verify the performance of the proposed cooperative VDLLs with UAV swarms' applications. A GNSS IF signal simulation software V1 from Beijing EPOO Technology (No. 145, Tongle Road, Nan Dulehe Town, Pinggu District, Beijing, China) is employed here to generate the GNSS IF signal with a pre-designed trajectory. Before simulation, the trajectory is designed and the parameters are set up in the simulation software with the corresponding function button. Parameters of the trajectory include time duration, acceleration, moving type (linear or circle), etc. Figure 6 shows the designed trajectories.

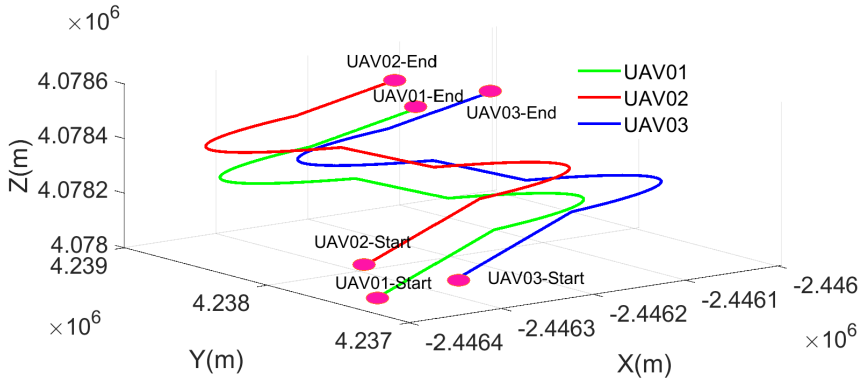


Figure 6. Three-dimensional plot of the trajectories in ECEF coordinates.

After setting up the trajectory parameters, the software runs and generates the GNSS IF signal dataset for further processing. The sampling frequency of the IF dataset is 5.714 MHz, and the IF signal frequency value is 4.309 MHz. Signal strength is set to 45 dB-Hz during the simulation. A GNSS software-defined receiver (GNSS-SDR) is utilized here to process the GNSS IF dataset and generate navigation solutions. Along with the IF dataset, the trajectory reference is generated at the same time with the frequency value of 50 Hz. After time synchronization, the trajectory reference is utilized to calculate the position errors of the CoVDLLs.

4.1. Experiment with Fixed Inter-Ranging Measurements

In the first experiment, we simulated three different UAVs flying trajectories. In detail, the trajectories are generated with fixed distances of 100 m, which are presented in Figure 7. The start points of the UAV01, UAV02, and UAV03 trajectories are set as (−2,446,392.31 m, 4,237,275.21 m, 4,078,049.95 m), (−2,446,430.61 m, 4,237,441.55 m, 4,078,214.23 m), and (−2,446,368.91 m, 4,237,507.89 m, 4,078,178.50 m) in the Earth-Centered-Earth-Fixed (ECEF) coordinates. The following Table 1 shows the details of the trajectory, which is composed of six different parts.

In the simulated GNSS IF datasets, a total of 10 available satellites were included. Figure 7 shows the in-view satellites, including PRN#2, PRN#6, PRN#7, PRN#8, PRN#9, PRN#10, PRN#18, PRN#21, PRN#26, and PRN#29. A total of twelve satellites are well-tracked and included in the navigation solutions estimation. One thing to note is that sometimes the satellites with lower elevation angles are excluded from the navigation solutions estimation. In this experiment, we did not exclude the satellites with a predefined elevation angle threshold. In addition, we used the GTSAM library (MATLAB version) to realize the FGO method. The standard deviation parameters of the velocity, pseudorange, and the cooperative ranging measurements are set as 0.1 m/s, 10 m, and 1 m.

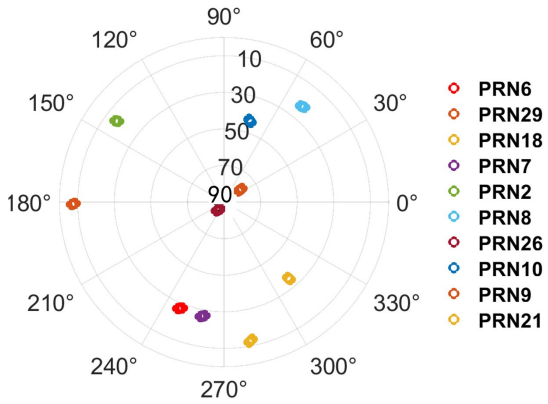


Figure 7. Satellite distribution.

Table 1. Trajectory Description.

Time	Action
0 ~ 50 s	static
50 s ~ 60 s	uniform acceleration
60 s ~ 90 s	circular motion
90 s ~ 100 s	uniform motion
100 s ~ 130 s	circular motion
130 s ~ 180 s	uniform motion

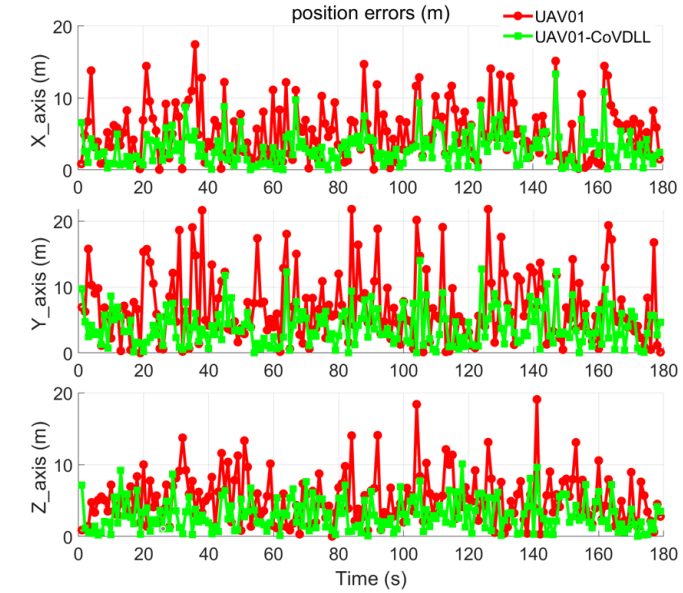
With trajectory reference, horizontal position errors are calculated, analyzed and presented in the Figures 8–10. Figure 8a, Figure 9a, and Figure 10a present the three-axis position errors from the UAV01, UAV02, and UAV03 individually in the ECEF coordinates. Figure 8b, Figure 9b, and Figure 10b present the cumulative distribution of the three-axis position errors from the UAV01, UAV02, and UAV03 individually. By comparing the positioning results presented in the Figures, it is obvious that the CoVDLL can effectively reduce the three-axis positioning errors. Table 2 shows the statistical analysis results, including the mean and Standard Deviation (STD) values of the three-axis position errors. Mean values of the three-axis position errors of the UAV01 decreased by 44.3%, 44.1%, and 42.5%, and the STD values decreased by 40.4%, 44.8%, and 34.7% while comparing the VDLL and the CoVDLL methods; for UAV02, mean values of the three-axis position errors decreased by 39.9%, 36.0%, and 25.7%, and the STD values decreased by 39.8%, 21.2%, and 36.3%; and for UAV03, the mean values of the three-axis position errors decreased by 25.7%, 43.2%, and 19.0%, and the STD values decreased by 27.3%, 41.0%, and 31.0%.

Table 2. STD values from the fixed inter-ranging experiment (Horizontal position errors).

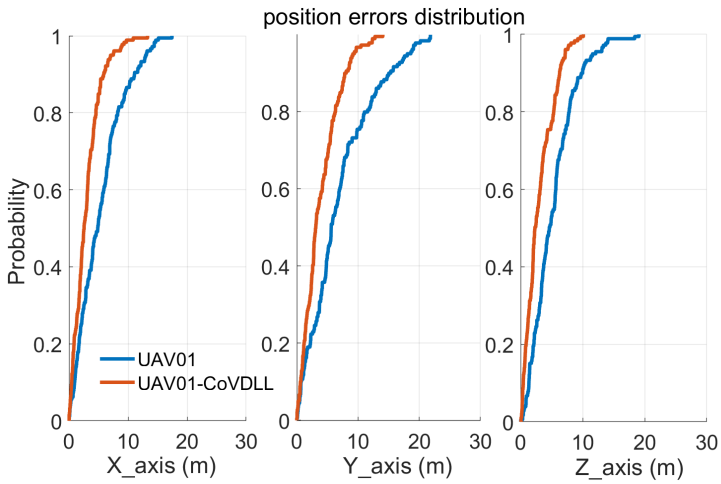
		Mean Values		STD Values	
		VDLL (m)	**CoVDLL (m)**	**VDLL (m)**	**CoVDLL (m)**
UAV01	**X-Axis**	5.33	2.97	3.86	2.30
	Y-Axis	6.98	3.90	5.34	2.95
	Z-Axis	5.22	3.00	3.49	2.28
UAV02	**X-Axis**	5.04	3.03	3.67	2.21
	Y-Axis	7.94	5.08	5.20	4.10
	Z-Axis	6.04	4.49	4.30	2.74

Table 2. *Cont.*

		Mean Values		STD Values	
		VDLL (m)	**CoVDLL (m)**	**VDLL (m)**	**CoVDLL (m)**
UAV03	**X-Axis**	4.51	3.35	3.15	2.29
	Y-Axis	7.89	4.48	5.63	3.32
	Z-Axis	6.49	5.26	4.45	3.07

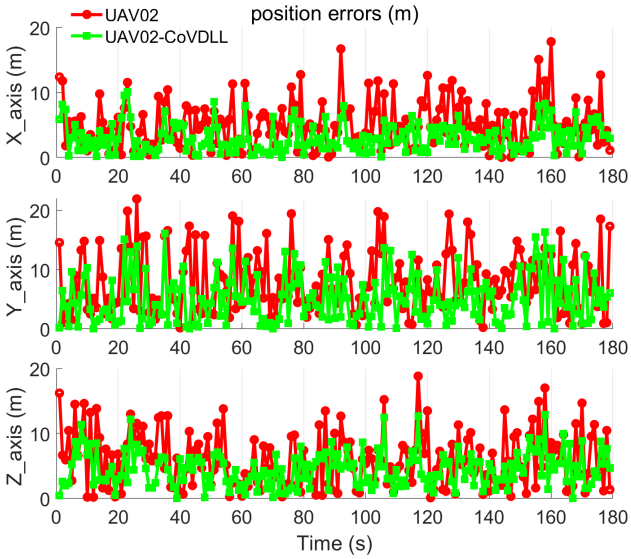


(**a**) three-axis position errors (UAV01)

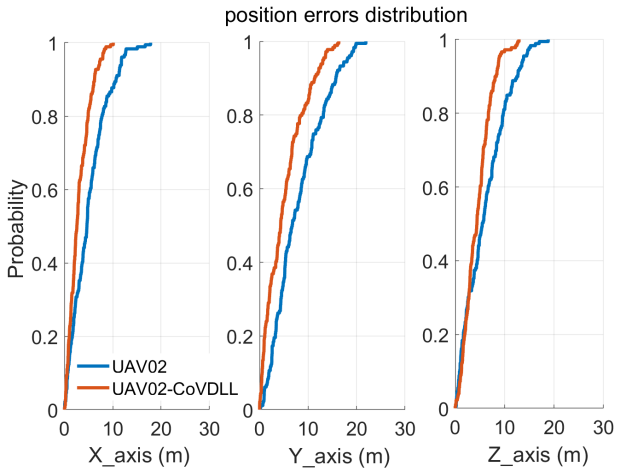


(**b**) three-axis position errors distribution (UAV01)

Figure 8. Position errors for UAV01 in the fixed inter-ranging experiment.

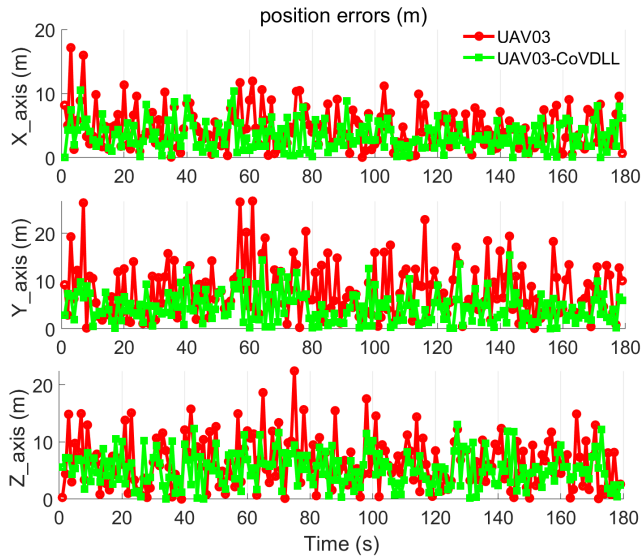


(**a**) three-axis position errors (UAV02)



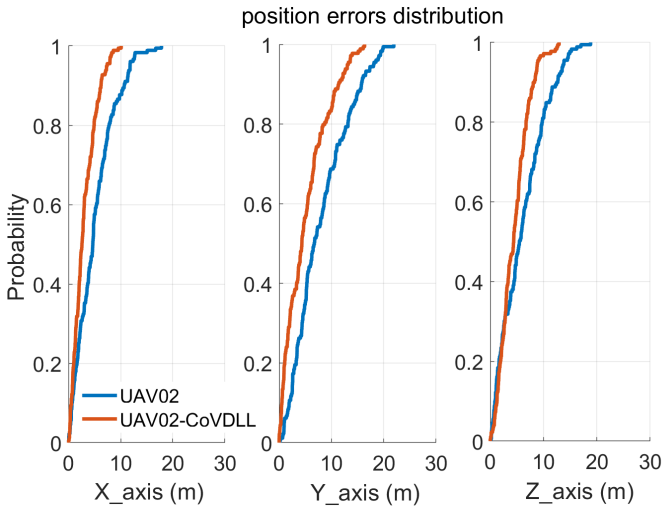
(**b**) three-axis position errors (UAV02)

Figure 9. Position errors for UAV02 in the fixed inter-ranging experiment.



(**a**) three-axis position errors (UAV03)

Figure 10. *Cont.*



(**b**) three-axis position errors (UAV03)

Figure 10. Position errors for UAV03 in the fixed inter-ranging experiment.

4.2. Experiment with Time-Varying Inter-Ranging Measurements

In this first experiment, we designed an experiment with fixed inter-ranging UAVs. This subsection describes an experiment with time-varying inter-ranging UAVs. Dynamic trajectories are presented in Figure 11 with the ECEF coordinates, in which the green line represents the UAV01 trajectory, the red line represents the UAV02 trajectory, and the blue line represents the trajectory of the UAV03. It can be observed that the trajectory curves are quite different, which means the trajectories have different velocity, acceleration, and heading angle information. Figure 12 shows the velocity information changing with time. Maximum values of the UAV01 velocity reach 40 m/s, and the UAV02 and UAV03 velocities reach 20 m/s. In addition, Figure 12b gives details describing the three-axis velocity information. Part of the UAV02 and UAV03 velocity are similar since trajectory settings are similar. UAV02 and UAV03 have different heading angles; therefore, their trajectories are different (Figure 11) even if part of their velocity values is similar.

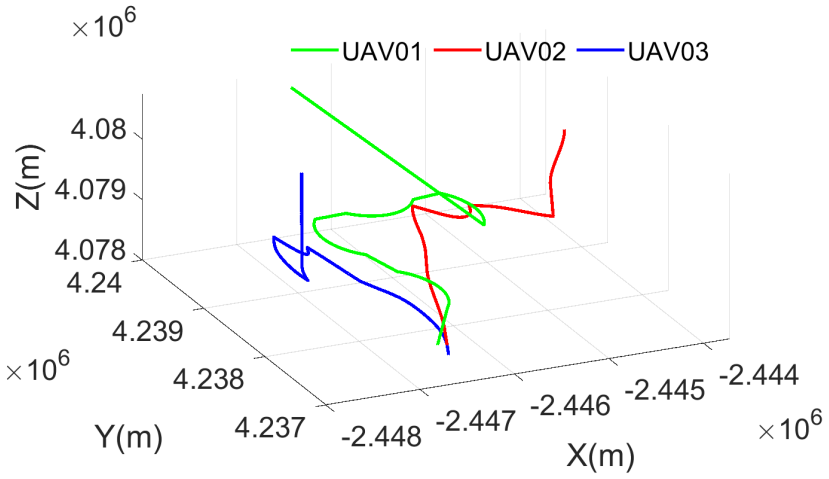
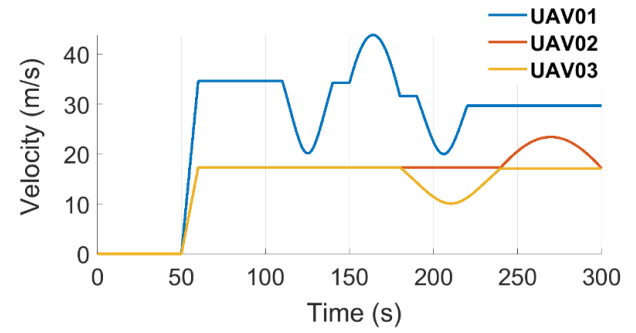
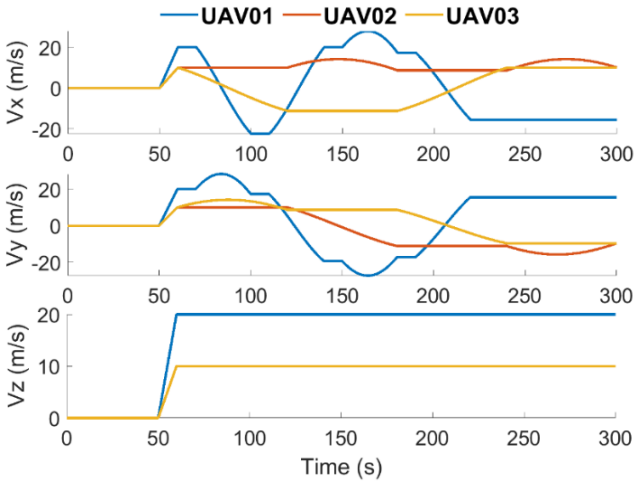


Figure 11. Dynamic trajectories.



(**a**) Velocity



(**b**) three-axis velocity

Figure 12. Velocity information vs. time.

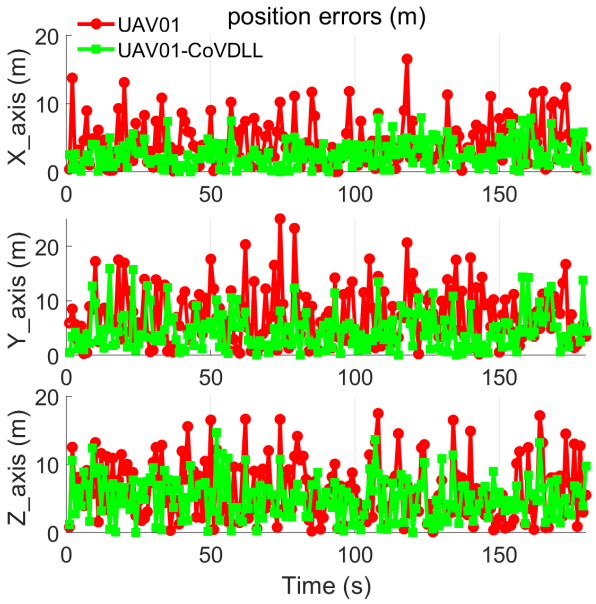
In aspects of the satellites, similar in-view satellites are included in the position determination and calculation due to the fact that the two experiments have the same ephemeris setting. Horizontal position errors from the DLL and CoVDLL are analyzed and compared in the Figures 13–15. The red line represents the horizontal position results from the single VDLL method, and the green line represents the horizontal position results from the CoVDLL. It can be observed that the CoVDLL method contributes to three-axis positioning results compared with the VDLL method, which indicates that the cooperative method is effective at reducing the position errors. Statistical analysis results, i.e., mean values and the standard deviation values of the three-axis position errors, are listed in Table 3. In summary, mean values of the three-axis position errors of the UAV01 decreased by 37.7%, 36.1%, and 17.6%, and the STD values decreased by 39.3%, 31.7%, and 27.9% while comparing the VDLL and the CoVDLL methods; for UAV02, mean values of the three-axis position errors decreased by 21.5%, 36.0%, and 4.7%, and the STD values decreased by 20.7%, 39.0%, and 11.9%; and for UAV03, the mean values of the three-axis position errors decreased by 19.1%, 20.3%, and 10.0%, and the STD values decreased by 23.7%, 19.7%, and 15.4%.

Table 3. STD values (Horizontal position errors).

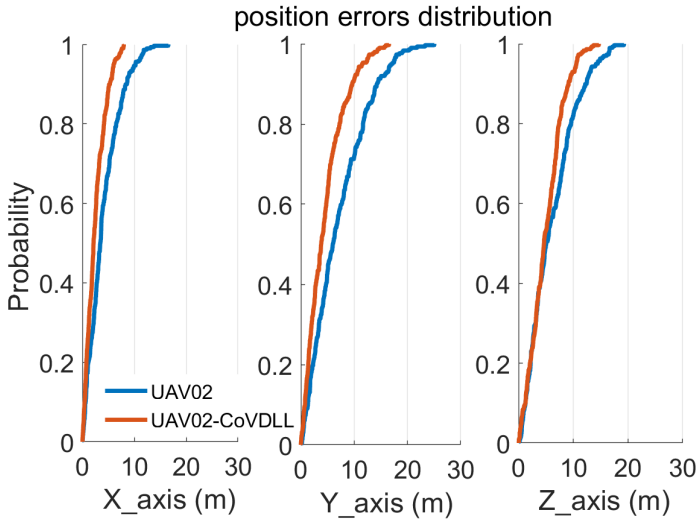
		Mean Values		STD Values	
		VDLL (m)	CoVDLL (m)	VDLL (m)	CoVDLL (m)
UAV01	X-Axis	4.06	2.53	3.08	1.87
	Y-Axis	7.10	4.54	5.17	3.53
	Z-Axis	6.07	5.00	4.23	3.05

Table 3. *Cont.*

		Mean Values		STD Values	
		VDLL (m)	CoVDLL (m)	VDLL (m)	CoVDLL (m)
UAV02	**X-Axis**	4.46	3.50	3.29	2.61
	Y-Axis	9.28	5.94	7.28	4.44
	Z-Axis	5.56	5.30	3.96	3.49
UAV03	**X-Axis**	4.19	3.39	3.17	2.42
	Y-Axis	6.55	5.22	5.02	4.03
	Z-Axis	5.51	4.96	4.02	3.40

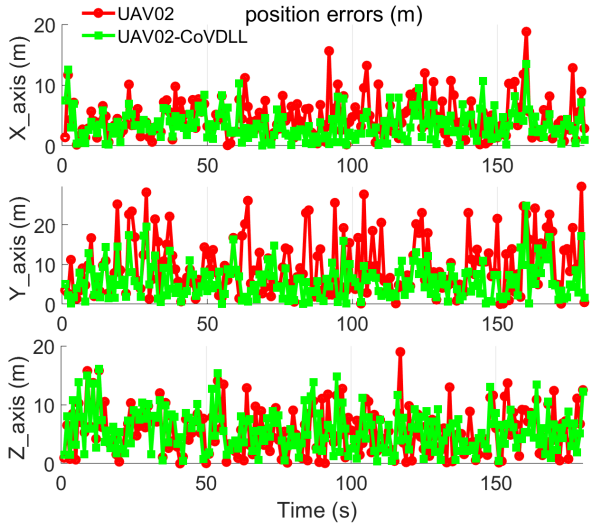


(**a**) three-axis position errors (UAV01)

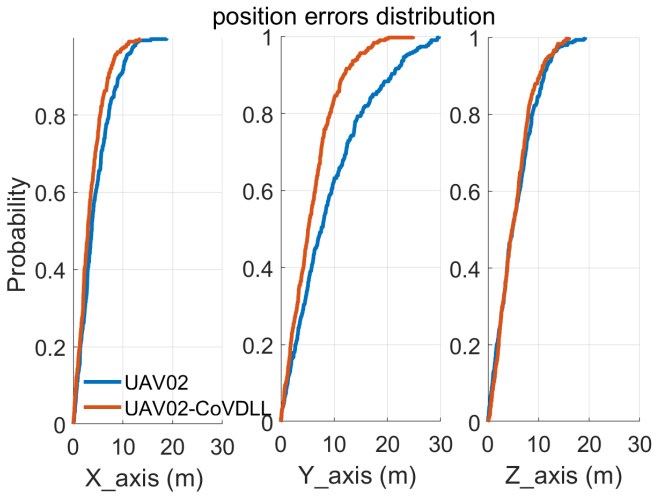


(**b**) three-axis position errors distribution (UAV01)

Figure 13. Position errors for UAV01 in the time-varying inter-ranging experiment.

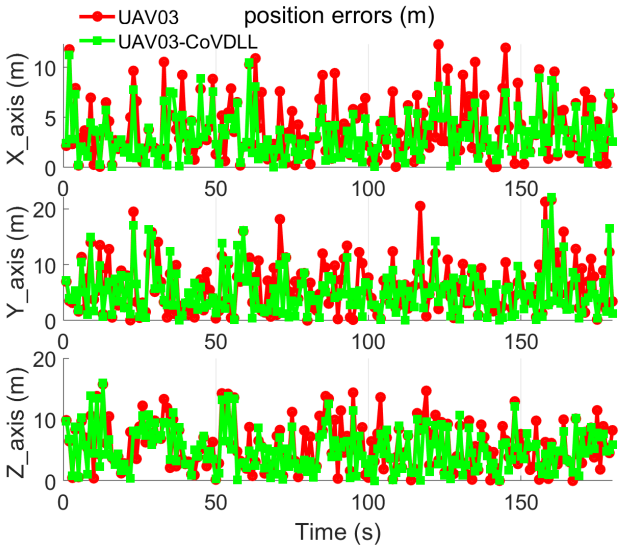


(**a**) three-axis position errors (UAV02)



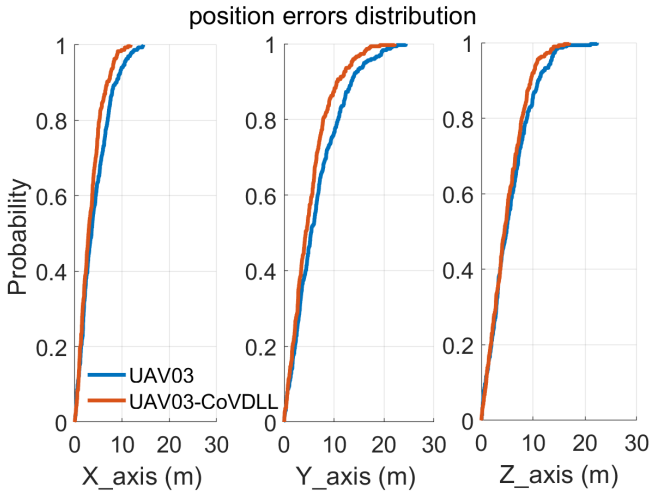
(**b**) three-axis position errors distribution (UAV02)

Figure 14. Position errors for UAV02 in the time-varying inter-ranging experiment.



(**a**) three-axis position errors (UAV03)

Figure 15. *Cont.*



(**b**) three-axis position errors distribution (UAV03)

Figure 15. Position errors for UAV03 in the time-varying inter-ranging experiment.

5. Limitations and Future Works

This paper presents a CoVDLL positioning method for multiple UAVs, utilizing an FGO algorithm. Experimental results from dynamic tests highlight its effectiveness in enhancing positioning accuracy for UAVs. However, we believe several promising areas warrant further investigation:

(1) Due to hardware computational limitations, the proposed method was validated only through a three-UAV simulation. In addition, limited by the hardware equipment, we did not have enough equipment to collect real-world flying datasets to evaluate the CoVDLL. Future work should include field tests with real-world UAVs and GNSS signals to thoroughly evaluate the algorithm's performance. Additionally, expanding the experiments to include more UAVs would provide a more comprehensive assessment. Furthermore, since the computational load of FGO is significantly higher than that of the Kalman Filter, optimizing the moving-window size is essential to reduce computational demands while maintaining performance.

(2) Limited by the simulation settings and software, UAVs are typically equipped with various onboard sensors, such as Inertial Measurement Units (IMUs), LiDAR, and visual cameras. Software including the GNSS IF signal simulation, IMU measurements simulation, and other sensor simulations has the potential to help to evaluate the proposed method. These sensors can capture UAV motion and environmental data, which can be leveraged to estimate position and other dynamic information. In our experiments, we observed only marginal improvements in z-axis position errors (Figure 15). Integrating VDLL with other sensors to utilize dynamic information—such as velocity and observation angles—could lead to the development of a multi-sensor fusion system; potentially enhancing both positioning accuracy and system reliability.

(3) GNSS IF signal simulation software here has limited function, i.e., no spoofing simulation function and no signal blockage or signal degradation simulation function. Therefore, in the simulation, we did not carry out the assessment of the CoVDLL under a signal-challenging environment. This study assumes ideal GNSS signal conditions to achieve satisfactory positioning results. It would be valuable to test the proposed method in challenging GNSS environments to evaluate how the cooperative scheme can improve positioning accuracy under adverse signal conditions, i.e., signal spoofing and weak signal.

6. Conclusions

In this paper, we propose a CoVDLL method for UAV swarm applications. The FGO method is used to integrate the VDLL data from different UAV agents along with inter-ranging measurements. A nonlinear optimizer is employed to estimate the navigation solutions. Numerical simulations are conducted to evaluate the performance of the method with three UAV trajectories. The experimental results support the following conclusions:

(1) The CoVDLL method effectively reduces position errors compared to individual VDLL methods, and including inter-ranging measurements significantly enhances the construction of the CoVDLL.
(2) The nonlinear optimization technique proves to be effective in achieving optimal position estimations by utilizing the correlated ranging information.

Author Contributions: Conceptualization, C.L., C.J. and S.C.; methodology, C.J.; software, C.L. and Y.C.; validation, C.L., C.J. and X.W.; formal analysis, Z.S.; investigation, C.J.; resources, C.J. and Y.C.; data curation, S.C. and X.W.; writing—original draft preparation, C.L. and C.J.; writing—review and editing, C.L., X.W. and S.C.; visualization, Z.S.; supervision, S.C.; project administration, S.C. and X.W.; funding acquisition, S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the "National Natural Science Foundation of China (42101445)", "Jiangsu Province Natural Science Foundation of China (BK20241388)", and the "Nanjing University of Aeronautics and Astronautics Startup Funding for Yong Talents" (YQR23046). Thanks EPOO Technology Company a lot for providing the GNSS IF signal generation software.

Data Availability Statement: The datasets are available; please e-mail the corresponding author.

Acknowledgments: Thanks EPOO technology company a lot for providing the GNSS IF signal generation software.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Gao, B.; Hu, G.; Zhong, Y.; Zhu, X. Cubature Kalman filter with both adaptability and robustness for tightly-coupled GNSS/INS integration. *IEEE Sens. J.* **2021**, *21*, 14997–15011. [CrossRef]
2. Gao, G.; Gao, B.; Gao, S.; Hu, G.; Zhong, Y. A hypothesis test-constrained robust Kalman filter for INS/GNSS integration with abnormal measurement. *IEEE Trans. Veh. Technol.* **2022**, *72*, 1662–1673. [CrossRef]
3. Ruan, L.; Li, G.; Dai, W.; Tian, S.; Fan, G.; Wang, J.; Dai, X. Cooperative relative localization for UAV swarm in GNSS-denied environment: A coalition formation game approach. *IEEE Internet Things J.* **2021**, *9*, 11560–11577. [CrossRef]
4. Ng, Y.; Gao, G.X. GNSS multi-receiver vector tracking. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *53*, 2583–2593. [CrossRef]
5. Xu, B.; Jia, Q.; Hsu, L.T. Vector tracking loop-based GNSS NLOS detection and correction: Algorithm design and performance analysis. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 4604–4619. [CrossRef]
6. Zhao, S.; Akos, D. An open-source GPS/GNSS vector tracking loop-implementation, filter tuning, and results. In Proceedings of the 2011 International Technical Meeting of The Institute of Navigation, San Diego, CA, USA, 24–26 January 2011; pp. 1293–1305.
7. Shen, J.; Wang, S.; Zhai, Y.; Zhan, X. Cooperative relative navigation for multi-UAV systems by exploiting GNSS and peer-to-peer ranging measurements. *IET Radar Sonar Navig.* **2021**, *15*, 21–36. [CrossRef]
8. Vetrella, A.R.; Fasano, G.; Accardo, D.; Moccia, A. Differential GNSS and vision-based tracking to improve navigation performance in cooperative multi-UAV systems. *Sensors* **2016**, *16*, 2164. [CrossRef]
9. Causa, F.; Fasano, G.; Grassi, M. Improving autonomy in GNSS-challenging environments by multi-UAV cooperation. In Proceedings of the 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC), San Diego, CA, USA, 8–12 September 2019; IEEE: Piscataway, NJ, USA; pp. 1–10.
10. Sivaneri, V.O.; Gross, J.N. Flight-testing of a cooperative UGV-to-UAV strategy for improved positioning in challenging GNSS environments. *Aerosp. Sci. Technol.* **2018**, *82*, 575–582. [CrossRef]
11. Goel, S.; Kealy, A.; Gikas, V.; Retscher, G.; Toth, C.; Brzezinska, D.G.; Lohani, B. Cooperative localization of unmanned aerial vehicles using GNSS, MEMS inertial, and UWB sensors. *J. Surv. Eng.* **2017**, *143*, 04017007. [CrossRef]

12. Marçal, J.; Nunes, F. Robust vector tracking for GNSS carrier phase signals. In Proceedings of the 2016 International Conference on Localization and GNSS (ICL-GNSS), Barcelona, Spain, 28–30 June 2016; IEEE: Piscataway, NJ, USA; pp. 1–6.
13. Jiang, C.; Chen, S.; Chen, Y.; Liu, D.; Bo, Y. GNSS vector tracking method using graph optimization. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *68*, 1313–1317. [CrossRef]
14. Luo, Y.; Wang, Y.; Wu, S.; Wang, P. Multipath effects on vector tracking algorithm for GNSS signal. *Sci. China Inf. Sci.* **2014**, *57*, 1–13. [CrossRef]
15. Ren, T.; Petovello, M.; Basnayake, C. Improving GNSS bit synchronization and decoding using vector tracking. In Proceedings of the 26th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2013), Nashville, TN, USA, 16–20 September 2013; pp. 121–134.
16. Yang, H.; Zhou, B.; Wang, L.; Wei, Q.; Ji, F.; Zhang, R. Performance and evaluation of GNSS receiver vector tracking loop based on adaptive cascade filter. *Remote Sens.* **2021**, *13*, 1477. [CrossRef]
17. Jiang, C.; Chen, Y.; Xu, B.; Jia, J.; Sun, H.; Chen, C.; Duan, Z.; Bo, Y.; Hyyppä, J. Vector tracking based on factor graph optimization for GNSS NLOS bias estimation and correction. *IEEE Internet Things J.* **2022**, *9*, 16209–16221. [CrossRef]
18. Abedi, A.A.; Mosavi, M.R. Low computational-complexity vector tracking for low-cost GNSS receivers. *Measurement* **2022**, *195*, 111171. [CrossRef]
19. Jiang, C.; Chen, S.; Chen, Y.; Bo, Y.; Wang, C.; Tao, W. Performance analysis of GNSS vector tracking loop based GNSS/CSAC integrated navigation system. *J. Aeronaut. Astronaut. Aviat.* **2017**, *49*, 289–297.
20. Bhattacharyya, S.; Gebre-Egziabher, D. Integrity monitoring with vector GNSS receivers. *IEEE Trans. Aerosp. Electron. Syst.* **2014**, *50*, 2779–2793. [CrossRef]
21. Xu, B.; Hsu, L.T. Open-source MATLAB code for GPS vector tracking on a software-defined receiver. *GPS Solut.* **2019**, *23*, 46. [CrossRef]
22. Lashley, M.; Bevly, D.M.; Hung, J.Y. Performance analysis of vector tracking algorithms for weak GPS signals in high dynamics. *IEEE J. Sel. Top. Signal Process.* **2009**, *3*, 661–673. [CrossRef]
23. Jiang, C.; Chen, Y.; Chen, C.; Hyyppa, J. Walking Gait Assisted Vector Tracking GNSS for Pedestrian Position. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 8506711. [CrossRef]
24. Chan, B.; Petovello, M.G. Collaborative vector tracking of GNSS signals with ultra-wideband augmentation in degraded signal environments. In Proceedings of the 2011 International Technical Meeting of The Institute of Navigation, San Diego, CA, USA, 24–26 January 2011; pp. 404–413.
25. Gao, W.; Yang, R.; Huang, J.; Zhan, X. Quasi-deep integration for DPE/INS in GNSS navigation domain: Framework design and optimization. *IEEE Trans. Aerosp. Electron. Syst.* **2025**, *61*, 6774–6793. [CrossRef]
26. Yang, R.; Ling, K.V.; Poh, E.K.; Morton, Y. Generalized GNSS signal carrier tracking: Part I—Modeling and analysis. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *53*, 1781–1797. [CrossRef]
27. Yan, Z.; Chen, X.; Tang, X.; Zhu, X. The implementation and comparison between Kalman filter-based and vector tracking loops. In Proceedings of the IEEE International Conference on Sensing, Measurement & Data Analytics in the Era of Artificial Intelligence (ICSMD), Xi'an, China, 15–17 October 2020; pp. 45–50.
28. Burchfield, S.; Givhan, C.; Martin, S. A Beam Steering Vector Tracking GNSS Software-Defined Receiver for Robust Positioning. *Sensors* **2025**, *25*, 1951. [CrossRef] [PubMed]
29. Goel, S.; Kealy, A.; Lohani, B. Development and experimental evaluation of a low-cost cooperative UAV localization network prototype. *J. Sens. Actuator Netw.* **2018**, *7*, 42. [CrossRef]
30. Jin, R.; Zhang, G.; Hsu, L.T.; Hu, Y. A Survey on Cooperative Positioning Using GNSS Measurements. *IEEE Transactions on Intelligent Vehicles* **2024**, 1–20. [CrossRef]
31. Chen, M.; Xiong, Z.; Xiong, J.; Wang, R. A hybrid cooperative navigation method for UAV swarm based on factor graph and Kalman filter. *Int. J. Distrib. Sens. Netw.* **2022**, *18*, 15501477211064758. [CrossRef]
32. Ding, X.; Li, K.; Yang, Y. A navigation-data-tolerated EKF carrier loop for GNSS signals. *IEICE Trans. Commun. March* **2025**. [CrossRef]
33. Jiang, C.; Xu, B.; Hsu, L.T. Probabilistic approach to detect and correct GNSS NLOS signals using an augmented state vector in the extended Kalman filter. *GPS Solut.* **2021**, *25*, 72. [CrossRef]
34. Gleason, S. A real-time on-orbit signal tracking algorithm for GNSS surface observations. *Remote Sens.* **2019**, *11*, 1858. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.