

Article

# Design and Analysis of a Non-Iterative Estimator for Target Location in Multistatic Sonar Systems with Sensor Position Uncertainties

Xin Wang <sup>1,\*</sup> , Zhi Yu <sup>1</sup> , Le Yang <sup>2</sup>  and Ji Li <sup>3</sup>

<sup>1</sup> Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Department of Electronic Engineering, Jiangnan University, Wuxi 214122, China; 6181918027@stu.jiangnan.edu.cn

<sup>2</sup> Department of Electrical and Computer Engineering, College of Engineering, University of Canterbury, Christchurch 8020, New Zealand; le.yang@canterbury.ac.nz

<sup>3</sup> Locaris Technology Co., Ltd., Zhengzhou 450000, China; ji.li@locaris-tech.com

\* Correspondence: wangxin@jiangnan.edu.cn

Received: 3 December 2019; Accepted: 10 January 2020; Published: 15 January 2020



**Abstract:** Target location is the basic application of a multistatic sonar system. Determining the position/velocity vector of a target from the related sonar observations is a nonlinear estimation problem. The presence of possible sensor position uncertainties turns this problem into a more challenging hybrid parameter estimation problem. Conventional gradient-based iterative estimators suffer from the problems of initialization difficulties and local convergence. Even if there is no problem with initialization and convergence, a large computational cost is required in most cases. In view of these drawbacks, we develop a computationally efficient non-iterative position/velocity estimator. The main numerical computation involved is the weighted least squares optimization, which makes the estimator computationally efficient. Parameter transformation, model linearization and two-stage processing are exploited to prevent the estimator from iterative computation. Through performance analysis and experimental verification, we find that the proposed estimator reaches the hybrid Cramér–Rao bound and has linear computational complexity.

**Keywords:** multistatic sonar; target location; hybrid Cramér–Rao bound; weighted least squares; nonlinear estimation; non-iterative estimator; perturbation analysis; linear model; bias analysis; complexity analysis

## 1. Introduction

In recent years, there has been a lively interest in target location using multistatic sonars [1–13]. In a multistatic sonar system, the sum of each pair of transmitter–target range and target–receiver range defines an ellipse. Then the target is at the intersection of all these ellipses [7]. The elliptical location encountered in the multistatic sonar systems has also been considered in the MIMO radar [14–20], multistatic radar [21–25] and indoor positioning systems [26,27].

A considerable amount of literature has been published on the problem of estimating the coordinates of the intersection of the ellipses, which can be statistically modelled as a nonlinear estimation problem. To resolve the essential nonlinearity in the problem, linearization is a natural idea. In particular, the measurement equations were linearized by Taylor expansion, resulting in an iterative algorithm [20]. Alternative to the Taylor expansion, introducing nuisance parameters is another approach to linearization. For example, the classic spherical-interpolation [28] and spherical-intersection [29] methods were ported to the elliptical location problems [25]. However, the estimation accuracy is not optimum in [25]. Slightly more complex than the linear models,

a quadratically constrained least squares model was constructed, which is generally difficult to solve effectively [27,30]. More recently, as another major methodology for parameter estimation, a Bayes estimator was presented for elliptical location, involving formidable numerical integration [4]. Intuitively, integrating other kinds of observations helps improve the positioning accuracy. For instance, the Doppler shift measurements were incorporated to improve the position estimate and identify the velocity additionally [5].

In addition to the difficulties raised by the high nonlinearity in the statistical models, another obstacle in the multistatic sonar location is that the complex ocean environments introduce uncertainties in the positions of the transmitters and receivers. Preliminary work considering sensor location errors in elliptical location was reported in the literature [6,10,13]. Recent advances have seen an efficient non-iterative estimator for the multistatic sonar location [5,6] inspired by the renowned work of [31].

Perturbation analysis of least squares problems is a major topic in numerical linear algebra. Related work has focused on establishing various error bounds [32–34]. We combine the basic techniques of perturbation analysis with multivariate statistics [35] to quantitatively evaluate the estimators for a nonlinear estimation problem.

On the basis of the above work, our technical contributions are summarized here.

1. We establish a statistical model of determining both the position and velocity of a moving target in a multistatic sonar system using differential delays and Doppler shifts. The uncertainties in the sensor positions are carefully taken into account in our model. The performance limit is developed for this problem.
2. To tackle the proposed nonlinear hybrid parameter estimation problem, we design an efficient non-iterative solution using parameter transformation, model linearization and two-stage processing.
3. We further analyze the bias vector and covariance matrix of our estimator theoretically using the second/first-order perturbation analysis and multivariate statistics.
4. We prove that the proposed estimator has approximate statistical efficiency and linear complexity.

The rest of this paper is organized as follows. Section 2 lists the notational conventions that will be used throughout the paper. Section 3 provides the location scenario and formulates the problem as a nonlinear estimation problem. In Section 4, we evaluate the performance limit for the proposed problem. Section 5 is devoted to developing our estimator. Then, Section 6 analyzes the bias vector and covariance matrix of our estimator up to the second/first-order random errors. Section 7 contains comprehensive Monte Carlo simulation results, and finally Section 8 draws the conclusion.

## 2. Notational Conventions

We will use bold lowercase letters to denote the *column* vectors and bold uppercase ones to denote the matrices. Specifically,  $\mathbf{0}_{p \times q}$  is a  $p \times q$  zeros matrix,  $\mathbf{1}_{p \times q}$  is a  $p \times q$  ones matrix, and  $\mathbf{I}$  is an identity matrix of appropriate size. The operators  $\otimes$  and  $\circ$  represent the Kronecker product and Hadamard product respectively. The expression  $(\mathbf{M}_1 \succeq \mathbf{M}_2)$  means that  $(\mathbf{M}_1 - \mathbf{M}_2)$  is a positive semidefinite matrix.  $\text{diag}(\mathbf{v})$  is the square diagonal matrix with the elements of vector  $\mathbf{v}$  on the main diagonal.  $\mathbf{B} = \text{blkdiag}(\mathbf{M}_1, \dots, \mathbf{M}_N)$  is the block diagonal matrix created by aligning the matrices  $\mathbf{M}_1, \dots, \mathbf{M}_N$  along the diagonal of  $\mathbf{B}$ . When we want to access selected elements of a vector/matrix, we imitate the syntax of MATLAB programming language. For simplicity of presentation, we use numerous symbols and notations. They are summarized in Table 1 for quick reference. For the sake of readability, the text also includes relevant explanations about these symbols and notations.

**Table 1.** List of symbols and notations (□ as a placeholder).

Symbols/Notations	Remarks
$\square^o$	zero-order approximation of $\square$
$\bar{\square}$	expected or nominal value of $\square$
$\hat{\square}$	estimator of $\square$
$\Delta\square$	random error or differential of $\square$
$\mathbf{Q}_{\square}$	covariance of $\square$
$M$	known number of transmitters
$N$	known number of receivers
$\mathbf{t}_i$	actual unobservable position of the $i$ -th transmitter, $i = 1, \dots, M$
$\mathbf{s}_j$	actual unobservable position of the $j$ -th receiver, $j = 1, \dots, N$
$\mathbf{t}$	$[\mathbf{t}_1^T, \mathbf{t}_2^T, \dots, \mathbf{t}_M^T]^T$
$\mathbf{s}$	$[\mathbf{s}_1^T, \mathbf{s}_2^T, \dots, \mathbf{s}_N^T]^T$
$\mathbf{z}$	$[\mathbf{t}^T, \mathbf{s}^T]^T$
$\bar{\mathbf{z}}$	known nominal value of $\mathbf{z}$
$c$	known signal speed
$\mathbf{u}$	unknown position of the target
$\dot{\mathbf{u}}$	unknown velocity of the target
$\boldsymbol{\theta}$	$[\mathbf{u}^T, \dot{\mathbf{u}}^T]^T$
$\tau_{i,j}$	observed differential delay time between $\mathbf{t}_i$ and $\mathbf{s}_j$
$f_{i,j}$	observed range rate between $\mathbf{t}_i$ and $\mathbf{s}_j$
$\boldsymbol{\tau}_i$	$[\tau_{i,1}, \tau_{i,2}, \dots, \tau_{i,N}]^T$
$\mathbf{f}_i$	$[f_{i,1}, f_{i,2}, \dots, f_{i,N}]^T$
$\mathbf{m}_i$	$[\boldsymbol{\tau}_i^T, \mathbf{f}_i^T]^T$
$\mathbf{m}$	$[\mathbf{m}_1^T, \mathbf{m}_2^T, \dots, \mathbf{m}_M^T]^T$
$\bar{\mathbf{m}}$	expected value of $\mathbf{m}$
$\Delta\mathbf{z}$	$\mathbf{z} - \bar{\mathbf{z}}$ , position uncertainties of transmitters and receivers
$\Delta\mathbf{m}$	$\mathbf{m} - \bar{\mathbf{m}}$ , observation errors
$\rho_{x,y}$	$(x - y) / \ x - y\ $ , gradient of $\ x - y\ $ with respected to $x$
$\mathbf{A}_{x,y}$	$\mathbf{I} / \ x - y\  - (x - y)(x - y)^T / \ x - y\ ^3$ , Hessian of $\ x - y\ $ with respected to $x$

### 3. Problem Formulation and Statistical Model

We now turn to the mathematical formulation of the problem. In the multistatic sonar location scenario here, the transmitters and receivers are stationary and the target is moving. Let  $M$  be the number of transmitters and  $N$  be the number of receivers. We consider a two-dimensional location scenario. The unknown position vector and velocity vector of the target are denoted by  $\mathbf{u} = [x_u, y_u]^T$  and  $\dot{\mathbf{u}} = [x_{\dot{u}}, y_{\dot{u}}]^T$ . For simplicity, the complete unknown parameter vector will be denoted by

$$\boldsymbol{\theta} = [\mathbf{u}^T, \dot{\mathbf{u}}^T]^T. \tag{1}$$

To characterize the sensor location errors, the position vectors of the  $i$ -th transmitter and  $j$ -th receiver are modeled as random vectors  $\mathbf{t}_i = [x_{t_i}, y_{t_i}]^T$  and  $\mathbf{s}_j = [x_{s_j}, y_{s_j}]^T$  respectively, where  $i = 1, 2, \dots, M$  and  $j = 1, 2, \dots, N$ . We write compactly

$$\mathbf{z} = [\mathbf{t}^T, \mathbf{s}^T]^T, \tag{2}$$

where  $\mathbf{t} = [\mathbf{t}_1^T, \mathbf{t}_2^T, \dots, \mathbf{t}_M^T]^T$  and  $\mathbf{s} = [\mathbf{s}_1^T, \mathbf{s}_2^T, \dots, \mathbf{s}_N^T]^T$ . Generally, it can be assumed that

$$\mathbf{z} \sim \mathcal{N}(\bar{\mathbf{z}}, \mathbf{Q}_z), \tag{3}$$

where the nominal positions of the sensors  $\bar{\mathbf{z}}$  and the covariance matrix  $\mathbf{Q}_z$  are known [6]. Then the sensor position errors vector is  $\Delta\mathbf{z} = \mathbf{z} - \bar{\mathbf{z}}$ .

Physically, each transmitter radiates a sonar signal and all receivers observe the signals both from direct propagation and from indirect reflection of the target. Thus, the observation model of differential delay time between  $\mathbf{t}_i$  and  $\mathbf{s}_j$  is

$$\tau_{i,j} = \frac{1}{c} (\|\mathbf{u} - \mathbf{t}_i\| + \|\mathbf{u} - \mathbf{s}_j\| - \|\mathbf{t}_i - \mathbf{s}_j\|) + \Delta\tau_{i,j}, \tag{4}$$

where  $c$  is the signal propagation speed and  $\Delta\tau_{i,j}$  is the observation noise of  $\tau_{i,j}$  [6]. Furthermore, as the target is moving, we can also obtain the observation model of range rate (i.e., the Doppler shift measurements divided by the carrier frequency) between  $\mathbf{t}_i$  and  $\mathbf{s}_j$ , that is,

$$f_{i,j} = \frac{1}{c} (\rho_{\mathbf{u},\mathbf{t}_i} + \rho_{\mathbf{u},\mathbf{s}_j})^T \dot{\mathbf{u}} + \Delta f_{i,j}, \tag{5}$$

where  $\Delta f_{i,j}$  is the observation noise of  $f_{i,j}$ . For the notations  $\rho_{\mathbf{u},\mathbf{t}_i}$  and  $\rho_{\mathbf{u},\mathbf{s}_j}$ , see Table 1.

For the transmitter at position  $\mathbf{t}_i$ , all the related observations can be collected in an observation vector

$$\mathbf{m}_i = [\boldsymbol{\tau}_i^T, \mathbf{f}_i^T]^T, \tag{6}$$

where  $\boldsymbol{\tau}_i = [\tau_{i,1}, \tau_{i,2}, \dots, \tau_{i,N}]^T$ , and  $\mathbf{f}_i = [f_{i,1}, f_{i,2}, \dots, f_{i,N}]^T$  for  $i = 1, 2, \dots, M$ . Then, the observations related to all the transmitters can be denoted by

$$\mathbf{m} = [\mathbf{m}_1^T, \mathbf{m}_2^T, \dots, \mathbf{m}_M^T]^T. \tag{7}$$

Furthermore, it is assumed that the conditional distribution (given  $\mathbf{z}$ ) of the observed vector  $\mathbf{m}$  is of the form

$$\mathbf{m} | \mathbf{z} \sim \mathcal{N}(\bar{\mathbf{m}}, \mathbf{Q}_m), \tag{8}$$

where  $\bar{\mathbf{m}}$  is the ideal error-free observation vector and  $\mathbf{Q}_m$  is the covariance matrix of  $\mathbf{m}$ . The corresponding observation error vector can be denoted by  $\Delta\mathbf{m} = \mathbf{m} - \bar{\mathbf{m}}$ .

As part of the observation model, the following small error assumptions are claimed.

1.  $\|\Delta\mathbf{t}_i\| \ll \|\mathbf{u} - \bar{\mathbf{t}}_i\|$ ,
2.  $\|\Delta\mathbf{t}_i\| \ll \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_j\|$ ,
3.  $\|\Delta\mathbf{s}_j\| \ll \|\mathbf{u} - \bar{\mathbf{s}}_j\|$ ,
4.  $\|\Delta\mathbf{s}_j\| \ll \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_j\|$ ,
5.  $|\Delta\tau_{i,j}| \ll \bar{\tau}_{i,j}$ ,
6.  $|\Delta f_{i,j}| \ll \bar{f}_{i,j}$ .

The physical motivation for these assumptions is that the position uncertainty of a given transmitter is small relative to its distance to the target and its distances to all the receivers, the position uncertainty of a given receiver is small relative to its distance to the target and its distances to all the transmitters, and the relative measurement errors are small. Besides,  $\Delta\mathbf{z}$  and  $\Delta\mathbf{m}$  are assumed to be statistically independent for ease of illustration.

Given the statistical model in Equation (8), the problem is to estimate the target position vector  $\mathbf{u}$  and velocity vector  $\dot{\mathbf{u}}$ , i.e.,  $\boldsymbol{\theta}$ , in real time and at a reasonable computational cost. Another significant work is the theoretical analysis of statistical performance of the designed estimator.

We conclude this section with some comments. Generally, the small error assumptions can be satisfied by increasing the observation period in obtaining the differential delay time and range rate measurements in a nonsingular location geometry. In addition, as we will see in Section 5, our estimator requires accurate knowledge of the positive definite covariance matrices  $\mathbf{Q}_m$  and  $\mathbf{Q}_z$ . They can usually be obtained during the calibration stage of a multistatic sonar system. Specifically, some scattering models from the environment may also help determine  $\mathbf{Q}_m$ .

#### 4. Hybrid Cramér–Rao Bound

In order to set a benchmark before designing an estimator, we now evaluate the Hybrid Cramér–Rao Bound (HCRB) [36–39] for the hybrid parameter estimation problem proposed in Section 3. The HCRB provides a lower bound on the error covariance matrix of the estimator of a hybrid unknown parameter vector.

In our statistical model, the wanted parameter vector  $\theta$  and the nuisance parameter vector (i.e., the actual sensor positions)  $\mathbf{z}$  are both unknown. What makes them different is that  $\theta$  is deterministic, and  $\mathbf{z}$  is a random parameter vector. Such models arise in many applications where we want to investigate model uncertainty or environmental mismatch. Here, we consider  $\theta$  and  $\mathbf{z}$  together as a hybrid parameter vector

$$\gamma = [\theta^T, \mathbf{z}^T]^T. \tag{9}$$

Before moving on to the estimator design, we outline the procedures for deriving the HCRB. In such a hybrid parameter case, the HCRB is calculated using the joint probability density of the observed measurement vector  $\mathbf{m}$  and the sensor position vector  $\mathbf{z}$ . The hybrid information matrix  $\mathbf{J}_H$  can be expressed as the sum

$$\mathbf{J}_H = \mathbf{J}_D + \mathbf{J}_P, \tag{10}$$

where  $\mathbf{J}_D$  represents the contribution of observations  $\mathbf{m}$  and  $\mathbf{J}_P$  represents the contribution of the prior knowledge on  $\mathbf{z}$ . Note that the unknown parameter vector  $\gamma = [\theta^T, \mathbf{z}^T]^T$  is in the mean vector  $\bar{\mathbf{m}}$  of the multivariate normal distribution in Equation (8) and  $\mathbf{z}$  is a multivariate normal random vector itself as in Equation (3). Section 3 reveals that  $\bar{\mathbf{m}}$  depends on  $\theta$ ,  $\mathbf{z}$  and  $c$ . In our model, the random parameter vector  $\mathbf{z}$  does not depend on the deterministic parameter vector  $\theta$ . Thus,  $\mathbf{J}_D$  and  $\mathbf{J}_P$  is fairly easy to get [40]. Consequently,

$$\mathbf{J}_D = E_{\mathbf{z}} \left\{ \begin{bmatrix} \frac{\partial \bar{\mathbf{m}}}{\partial \gamma} \end{bmatrix}^T \mathbf{Q}_m^{-1} \begin{bmatrix} \frac{\partial \bar{\mathbf{m}}}{\partial \gamma} \end{bmatrix} \right\}, \tag{11}$$

$$\mathbf{J}_P = \text{blkdiag}(\mathbf{0}_{4 \times 4}, \mathbf{Q}_z^{-1}). \tag{12}$$

When the levels of sensor positions' uncertainties are small, according to the approximation principle suggested by [41], the expected value matrix in Equation (11) can be approximated by replacing random vector  $\mathbf{z}$  with its expected value vector  $\bar{\mathbf{z}}$ . Then, from the blockwise inversion of  $\mathbf{J}_H|_{\mathbf{z}=\bar{\mathbf{z}}}$  and the matrix inversion lemma, we have the HCRB for the estimation of  $\theta = [\mathbf{u}^T, \hat{\mathbf{u}}^T]^T$  as follows:

$$\text{HCRB}_{\theta} \approx \left\{ \left( \frac{\partial \bar{\mathbf{m}}}{\partial \theta} \right)^T \left[ \mathbf{Q}_m + \frac{\partial \bar{\mathbf{m}}}{\partial \mathbf{z}} \mathbf{Q}_z \left( \frac{\partial \bar{\mathbf{m}}}{\partial \mathbf{z}} \right)^T \right]^{-1} \frac{\partial \bar{\mathbf{m}}}{\partial \theta} \right\}^{-1} \Bigg|_{\mathbf{z}=\bar{\mathbf{z}}}. \tag{13}$$

For numerical computation using Equation (13),  $\frac{\partial \bar{\mathbf{m}}}{\partial \theta}$  and  $\frac{\partial \bar{\mathbf{m}}}{\partial \mathbf{z}}$  are required. More information is available in Appendix A.

#### 5. Estimator Design

In this section, we use Taylor expansion, introduce auxiliary variables and apply multi-stage processing to deal with the nonlinear estimation problem proposed in Section 3. In particular, our algorithm can be divided into two stages, each involving an unconstrained linear weighted least squares (WLS) computation which is computationally attractive. During the algorithm design and performance analysis of our estimator, it is necessary to use many matrix symbols to simplify the presentation. These matrices are shown in Table 2 for easy reference. When justifying the introduction of these matrices, we find that these matrices naturally arise in a general weighted least squares problem. To prevent ourselves from obscuring the design of the estimator, the reader is referred to the Appendix B.

**Table 2.** List of matrix symbols.

Matrix Notations	Expressions
$\mathbf{P}_1$	$\mathbf{G}_1^T \mathbf{W}_1 \mathbf{G}_1$
$\mathbf{P}_2$	$\mathbf{G}_2^T \mathbf{W}_2 \mathbf{G}_2$
$\mathbf{H}_1$	$\mathbf{P}_1^{-1} \mathbf{G}_1^T \mathbf{W}_1$
$\mathbf{H}_2$	$\mathbf{P}_2^{-1} \mathbf{G}_2^T \mathbf{W}_2$
$\mathbf{K}_1$	$\mathbf{W}_1 (\mathbf{I} - \mathbf{G}_1^0 \mathbf{H}_1^0)$
$\mathbf{K}_2$	$\mathbf{W}_2 (\mathbf{I} - \mathbf{G}_2^0 \mathbf{H}_2^0)$
$\mathbf{U}$	$\mathbf{W}_2^{-1} \mathbf{K}_2$
$\mathbf{V}$	$\mathbf{H}_2^0 \mathbf{W}_2^{-1}$

Based on the Conditions 1 through 4 in Section 3, it follows from the first-order Taylor’s formula that

$$\|\mathbf{u} - \mathbf{t}_i\| \approx \|\mathbf{u} - \bar{\mathbf{t}}_i\| + \rho_{\bar{\mathbf{t}}_i, \mathbf{u}}^T \Delta \mathbf{t}_i, \tag{14}$$

$$\|\mathbf{u} - \mathbf{s}_j\| \approx \|\mathbf{u} - \bar{\mathbf{s}}_j\| + \rho_{\bar{\mathbf{s}}_j, \mathbf{u}}^T \Delta \mathbf{s}_j, \tag{15}$$

$$\|\mathbf{t}_i - \mathbf{s}_j\| \approx \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_j\| + \rho_{\bar{\mathbf{t}}_i, \bar{\mathbf{s}}_j}^T (\Delta \mathbf{t}_i - \Delta \mathbf{s}_j), \tag{16}$$

$$\rho_{\mathbf{u}, \mathbf{t}_i} \approx \rho_{\mathbf{u}, \bar{\mathbf{t}}_i} - \mathbf{A}_{\mathbf{u}, \bar{\mathbf{t}}_i} \Delta \mathbf{t}_i, \tag{17}$$

$$\rho_{\mathbf{u}, \mathbf{s}_j} \approx \rho_{\mathbf{u}, \bar{\mathbf{s}}_j} - \mathbf{A}_{\mathbf{u}, \bar{\mathbf{s}}_j} \Delta \mathbf{s}_j. \tag{18}$$

If we plug Equation (14) through Equation (16) into Equation (4), we obtain

$$c \tau_{i,j} \approx \|\mathbf{u} - \bar{\mathbf{t}}_i\| + \|\mathbf{u} - \bar{\mathbf{s}}_j\| - \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_j\| + \epsilon_{\tau,i,j}, \tag{19}$$

where

$$\epsilon_{\tau,i,j} = \rho_{\bar{\mathbf{t}}_i, \mathbf{u}}^T \Delta \mathbf{t}_i + \rho_{\bar{\mathbf{s}}_j, \mathbf{u}}^T \Delta \mathbf{s}_j - \rho_{\bar{\mathbf{t}}_i, \bar{\mathbf{s}}_j}^T (\Delta \mathbf{t}_i - \Delta \mathbf{s}_j) + c \Delta \tau_{i,j}. \tag{20}$$

Furthermore, inserting Equation (17) and Equation (18) into Equation (5) gives

$$c f_{i,j} \approx \rho_{\mathbf{u}, \bar{\mathbf{t}}_i}^T \dot{\mathbf{u}} + \rho_{\mathbf{u}, \bar{\mathbf{s}}_j}^T \dot{\mathbf{u}} + \epsilon_{f,i,j}, \tag{21}$$

where

$$\epsilon_{f,i,j} = -(\mathbf{A}_{\mathbf{u}, \bar{\mathbf{t}}_i} \dot{\mathbf{u}})^T \Delta \mathbf{t}_i - (\mathbf{A}_{\mathbf{u}, \bar{\mathbf{s}}_j} \dot{\mathbf{u}})^T \Delta \mathbf{s}_j + c \Delta f_{i,j}. \tag{22}$$

### 5.1. First Stage

Without loss of generality, let  $M < N$ . Move  $(\|\mathbf{u} - \bar{\mathbf{t}}_i\| - \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_j\|)$  from the right side to the left side in Equation (19), and square both sides. Then, we see that

$$\begin{aligned} & 2 \|\mathbf{u} - \bar{\mathbf{s}}_j\| \epsilon_{\tau,i,j} + \epsilon_{\tau,i,j}^2 \\ & \approx 2 \bar{\mathbf{t}}_i^T (\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_j) + 2c \tau_{i,j} \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_j\| + c^2 \tau_{i,j}^2 \\ & \quad - 2 (\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_j)^T \mathbf{u} - 2 (c \tau_{i,j} + \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_j\|) \|\mathbf{u} - \bar{\mathbf{t}}_i\|. \end{aligned} \tag{23}$$

Applying similar procedures to Equation (21) gives

$$\begin{aligned} & \|\mathbf{u} - \bar{\mathbf{s}}_j\| \epsilon_{f,i,j} + \rho_{\mathbf{u}, \bar{\mathbf{s}}_j}^T \dot{\mathbf{u}} \epsilon_{\tau,i,j} + \epsilon_{\tau,i,j} \epsilon_{f,i,j} \\ & \approx c f_{i,j} \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_j\| + c^2 \tau_{i,j} f_{i,j} \\ & \quad - c f_{i,j} \|\mathbf{u} - \bar{\mathbf{t}}_i\| - (\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_j)^T \dot{\mathbf{u}} \\ & \quad - (c \tau_{i,j} + \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_j\|) \rho_{\mathbf{u}, \bar{\mathbf{t}}_i}^T \dot{\mathbf{u}}. \end{aligned} \tag{24}$$

If we define an unknown parameter vector as

$$\boldsymbol{\phi}_1 = [\mathbf{u}^T, \boldsymbol{\alpha}^T, \dot{\mathbf{u}}^T, \boldsymbol{\beta}^T]^T, \tag{25}$$

where

$$\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_M]^T, \tag{26}$$

$$\boldsymbol{\beta} = [\beta_1, \beta_2, \dots, \beta_M]^T, \tag{27}$$

$$\alpha_i = \|\mathbf{u} - \bar{\mathbf{t}}_i\|, i = 1, 2, \dots, M, \tag{28}$$

$$\beta_i = \rho_{\mathbf{u}, \bar{\mathbf{t}}_i}^T \dot{\mathbf{u}}, i = 1, 2, \dots, M, \tag{29}$$

then a linear system of equations can be obtained from Equation (23) and Equation (24) as

$$\mathbf{B}_1 \boldsymbol{\epsilon}_1^{(1)} + \boldsymbol{\epsilon}_1^{(2)} \approx \mathbf{h}_1 - \mathbf{G}_1 \boldsymbol{\phi}_1. \tag{30}$$

We leave the details of  $\mathbf{h}_1$ ,  $\mathbf{G}_1$ ,  $\mathbf{B}_1$ ,  $\boldsymbol{\epsilon}_1^{(1)}$  and  $\boldsymbol{\epsilon}_1^{(2)}$  presented in Appendix C. Note that  $\boldsymbol{\epsilon}_1^{(1)}$  and  $\boldsymbol{\epsilon}_1^{(2)}$  are first-order and second-order approximation error respectively.

By ignoring the second-order error term  $\boldsymbol{\epsilon}_1^{(2)}$ , the WLS solution to Equation (30) is

$$\hat{\boldsymbol{\phi}}_1 = \mathbf{H}_1 \mathbf{h}_1, \tag{31}$$

and has covariance matrix

$$\text{cov}(\hat{\boldsymbol{\phi}}_1) = \text{cov}(\Delta \boldsymbol{\phi}_1) \approx \mathbf{P}_1^{o-1}, \tag{32}$$

where  $\mathbf{H}_1 = \mathbf{P}_1^{-1} \mathbf{G}_1^T \mathbf{W}_1$ ,  $\mathbf{P}_1 = \mathbf{G}_1^T \mathbf{W}_1 \mathbf{G}_1$  and  $\mathbf{P}_1^o$  is the zero-order approximation of  $\mathbf{P}_1$ . The weighting matrix  $\mathbf{W}_1$  is the inverse of the covariance matrix of the approximation error  $\mathbf{B}_1 \boldsymbol{\epsilon}_1^{(1)}$ , that is,

$$\begin{aligned} \mathbf{W}_1 &= [\text{E}(\mathbf{B}_1 \boldsymbol{\epsilon}_1^{(1)} \boldsymbol{\epsilon}_1^{(1)T} \mathbf{B}_1^T)]^{-1} \\ &= \mathbf{B}_1^{-T} \text{cov}(\boldsymbol{\epsilon}_1^{(1)})^{-1} \mathbf{B}_1^{-1}. \end{aligned} \tag{33}$$

The computation of  $\text{cov}(\boldsymbol{\epsilon}_1^{(1)})$  is straightforward. Because of the assumed statistical independence between  $\Delta \mathbf{z}$  and  $\Delta \mathbf{m}$  in Equation (A15),

$$\text{cov}(\boldsymbol{\epsilon}_1^{(1)}) = \mathbf{D}_z \mathbf{Q}_z \mathbf{D}_z^T + c^2 \mathbf{Q}_m, \tag{34}$$

where  $\mathbf{D}_z$  is shown in Equation (A16).

We get Equation (32) by the first-order perturbation analysis.

$$\begin{aligned} \text{cov}(\Delta \boldsymbol{\phi}_1) &\approx \text{E}[\mathbf{H}_1^o \mathbf{B}_1 \boldsymbol{\epsilon}_1^{(1)} \cdot (\mathbf{H}_1^o \mathbf{B}_1 \boldsymbol{\epsilon}_1^{(1)})^T] \\ &= \mathbf{H}_1^o \cdot \mathbf{B}_1 \text{cov}(\boldsymbol{\epsilon}_1^{(1)}) \mathbf{B}_1^T \cdot \mathbf{H}_1^{oT} \\ &= \mathbf{P}_1^{o-1} \mathbf{G}_1^{oT} \mathbf{W}_1 \cdot \mathbf{W}_1^{-1} \cdot \mathbf{W}_1^T \mathbf{G}_1^o \mathbf{P}_1^{o-T} \\ &= \mathbf{P}_1^{o-1} \cdot \mathbf{G}_1^{oT} \mathbf{W}_1^T \mathbf{G}_1^o \cdot \mathbf{P}_1^{o-T} \\ &= \mathbf{P}_1^{o-1} \mathbf{P}_1^o \mathbf{P}_1^{o-T} \\ &= \mathbf{P}_1^{o-1}. \end{aligned} \tag{35}$$

### 5.2. Second Stage

With  $\hat{\phi}_1$  and its covariance matrix  $\mathbf{P}_1^{o-1}$ , the aim of the second stage is to estimate the estimation error vector introduced in the first stage. In order to use symbols similar to the first stage, we denote this estimation error vector as  $\phi_2$ , i.e.,

$$\phi_2 = [\Delta \mathbf{u}^T, \Delta \hat{\mathbf{u}}^T]^T = \Delta \phi_1 ([1, 2, M + 3, M + 4]). \tag{36}$$

By substituting  $\mathbf{u} = \hat{\mathbf{u}} - \Delta \mathbf{u}$  and  $\alpha_i = \hat{\alpha}_i - \Delta \alpha_i$  into  $\alpha_i^2 = \|\mathbf{u} - \bar{\mathbf{t}}_i\|^2$ , we obtain

$$\begin{aligned} & 2\alpha_i \Delta \alpha_i + \|\Delta \mathbf{u}\|^2 + (\Delta \alpha_i)^2 \\ & = \hat{\alpha}_i^2 - \|\hat{\mathbf{u}} - \bar{\mathbf{t}}_i\|^2 + 2(\hat{\mathbf{u}} - \bar{\mathbf{t}}_i)^T \Delta \mathbf{u}. \end{aligned} \tag{37}$$

Furthermore, plugging  $\alpha_i = \hat{\alpha}_i - \Delta \alpha_i$ ,  $\beta_i = \hat{\beta}_i - \Delta \beta_i$ ,  $\mathbf{u} = \hat{\mathbf{u}} - \Delta \mathbf{u}$  and  $\hat{\mathbf{u}} = \hat{\mathbf{u}} - \Delta \hat{\mathbf{u}}$  into  $\alpha_i \beta_i = (\mathbf{u} - \bar{\mathbf{t}}_i)^T \hat{\mathbf{u}}$  gives

$$\begin{aligned} & \beta_i \Delta \alpha_i + \alpha_i \Delta \beta_i + \Delta \alpha_i \Delta \beta_i + \Delta \mathbf{u}^T \Delta \hat{\mathbf{u}} \\ & = \hat{\alpha}_i \hat{\beta}_i - (\hat{\mathbf{u}} - \bar{\mathbf{t}}_i)^T \hat{\mathbf{u}} + \hat{\mathbf{u}}^T \Delta \mathbf{u} + (\hat{\mathbf{u}} - \bar{\mathbf{t}}_i)^T \Delta \hat{\mathbf{u}}. \end{aligned} \tag{38}$$

In matrix notation, from Equation (37) and Equation (38), we have

$$\mathbf{B}_2 \epsilon_2^{(1)} + \epsilon_2^{(2)} = \mathbf{h}_2 - \mathbf{G}_2 \phi_2, \tag{39}$$

where

$$\epsilon_2^{(1)} = \Delta \phi_1, \tag{40}$$

i.e., the estimation error in the first stage  $\Delta \phi_1$  is considered as the first-order approximation error in the second stage. This is a key point of our estimator. The details of  $\mathbf{h}_2$ ,  $\mathbf{G}_2$ ,  $\mathbf{B}_2$  and the second-order approximation error  $\epsilon_2^{(2)}$  can be found in Appendix D.

By ignoring the second-order error term  $\epsilon_2^{(2)}$  and following the first stage’s approach, the WLS solution to Equation (39) is

$$\hat{\phi}_2 = \mathbf{H}_2 \mathbf{h}_2, \tag{41}$$

and has error covariance matrix

$$\text{cov}(\Delta \phi_2) \approx \mathbf{P}_2^{o-1}, \tag{42}$$

where  $\mathbf{H}_2 = \mathbf{P}_2^{-1} \mathbf{G}_2^T \mathbf{W}_2$ ,  $\mathbf{P}_2 = \mathbf{G}_2^T \mathbf{W}_2 \mathbf{G}_2$  and  $\mathbf{P}_2^o$  is the zero-order approximation of  $\mathbf{P}_2$ . The weighting matrix  $\mathbf{W}_2$  is the inverse of the covariance matrix of the approximation error  $\mathbf{B}_2 \epsilon_2^{(1)}$ , that is,

$$\begin{aligned} \mathbf{W}_2 & \approx \text{E}(\mathbf{B}_2 \epsilon_2^{(1)} \epsilon_2^{(1)T} \mathbf{B}_2^T)^{-1} \\ & \approx \mathbf{B}_2^{-T} \mathbf{P}_1^o \mathbf{B}_2^{-1}. \end{aligned} \tag{43}$$

Finally, our estimator can be constructed from  $\hat{\phi}_1$  in Equation (31) and  $\hat{\phi}_2$  in Equation (41) as

$$\hat{\theta} = [\hat{\phi}_1(1 : 2)^T, \hat{\phi}_1(M + 3 : M + 4)^T]^T - \hat{\phi}_2. \tag{44}$$

Last but not least, some obstacles arise in the practical computation of our estimator. In the first stage,  $\mathbf{B}_1$  and  $\mathbf{D}_z$ , as shown in Equation (A13) and Equation (A16), involve  $\mathbf{u}$  and  $\hat{\mathbf{u}}$  which are unavailable for the algorithm. To resolve this problem, we first assign an identity matrix to  $\mathbf{B}_1$  and an all-zero matrix to  $\mathbf{D}_z$  to get coarse estimates of  $\mathbf{u}$  and  $\hat{\mathbf{u}}$  from Equation (31), and then substitute the coarse estimates into Equation (A13) and Equation (A16) to update  $\mathbf{B}_1$  and  $\mathbf{D}_z$ . Confronted with similar problems in the second stage in Equation (A20), we substitute  $\hat{\alpha}$  and  $\hat{\beta}$  for computing  $\mathbf{B}_2$ , resulting  $\hat{\mathbf{B}}_2$  and  $\hat{\mathbf{W}}_2$ . These approximations will be considered properly in the statistical performance analysis in Section 6.

5.3. Summary

As a guide to implementation, the flowchart of the proposed estimator is shown in Figure 1 and the algorithm of the first stage of our estimator is summarized in Algorithm 1.

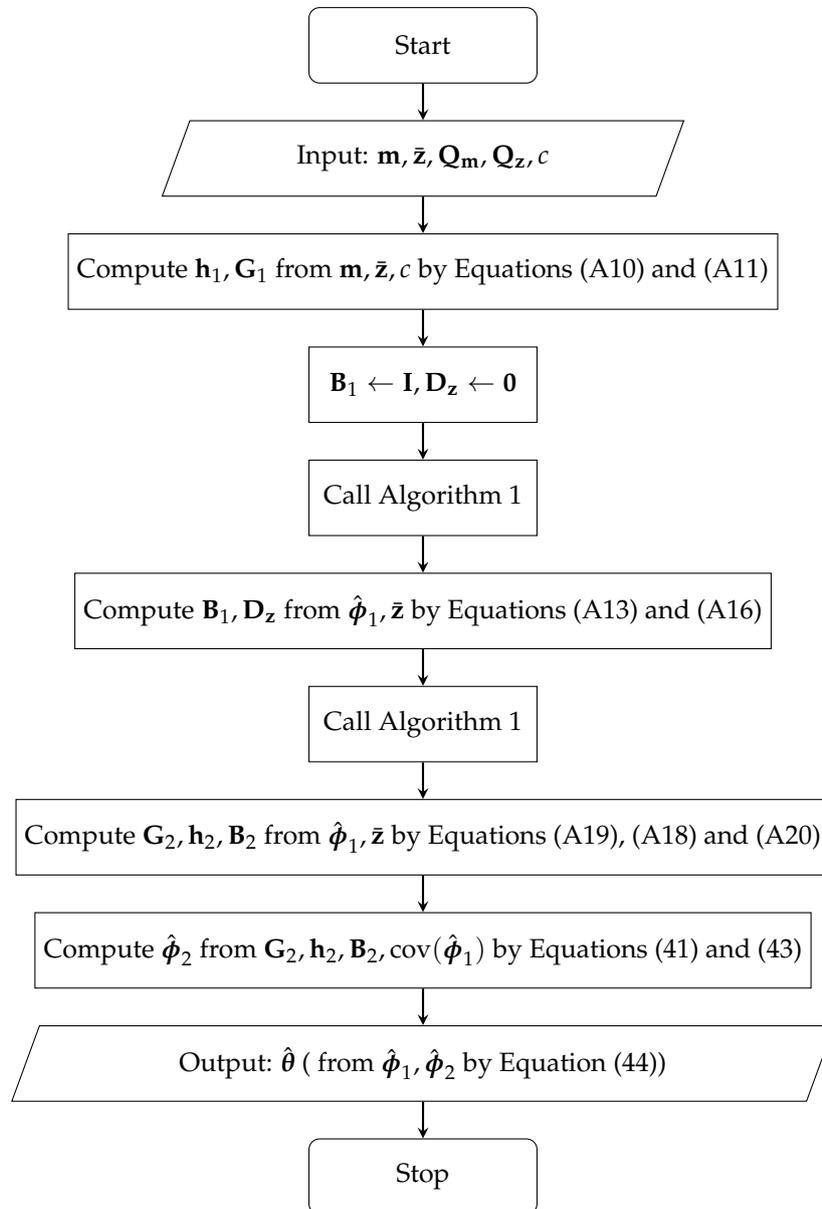


Figure 1. Flowchart of the proposed estimator. Algorithm 1 is called in the flowchart.

---

**Algorithm 1** First stage of the estimator.

---

- 1: **procedure** ESTIMATOR-FIRST-STAGE(  $Q_m, Q_z, c, G_1, h_1, D_z, B_1$  )
  - 2:    Compute  $cov(\epsilon_1^{(1)})$  from  $Q_m, Q_z, D_z, c$  by Equation (34)
  - 3:    Compute  $W_1$  from  $cov(\epsilon_1), B_1$  by Equation (33)
  - 4:    **return**  $\hat{\phi}_1$  from  $G_1, h_1, W_1$  by Equation (31)
  - 5:    **return**  $cov(\hat{\phi}_1)$  from  $G_1, W_1$  by Equation (32)
  - 6: **end procedure**
-

## 6. Performance Analysis

The covariance matrix and the bias vector are the two most important numerical characterization of a vector estimator. The perturbations in the design matrices (i.e.,  $\mathbf{G}_1$  and  $\mathbf{G}_2$ ) and the higher-order noise terms in the observation vector (i.e.,  $\mathbf{h}_1$  and  $\mathbf{h}_2$ ) in Equation (30) and Equation (39) make the conditions of Gauss–Markov theorem no longer true.

### 6.1. Bias Vector

We first derive the covariance matrix of our estimator in Equation (44). The bias analysis here will be up to the second-order statistics of the observation errors and the sensor position errors, i.e., the random terms higher than the second-order are ignored. Matrix differential calculus presented in [42] is intensively used in this subsection.

It can be seen from Equation (44) that the total bias vector of our estimator  $\hat{\theta}$  is

$$[\boldsymbol{\mu}_1(1 : 2)^T, \boldsymbol{\mu}_1(M + 3 : M + 4)^T]^T - \boldsymbol{\mu}_2, \tag{45}$$

where  $\boldsymbol{\mu}_1$  and  $\boldsymbol{\mu}_2$  are the bias vectors of  $\hat{\boldsymbol{\phi}}_1$  and  $\hat{\boldsymbol{\phi}}_2$ , respectively. The rest of our task is to compute the bias vector in the first stage and second stage, i.e.,  $\boldsymbol{\mu}_1$  and  $\boldsymbol{\mu}_2$ . We reiterate that the random terms higher than the second-order will be ignored at each occurrence of the approximately equal symbol. Please reference Table 2 for the matrix symbols involved.

The error of  $\hat{\boldsymbol{\phi}}_1$  is

$$\begin{aligned} \Delta\boldsymbol{\phi}_1 &= \hat{\boldsymbol{\phi}}_1 - \boldsymbol{\phi}_1 \\ &\approx \mathbf{H}_1(\mathbf{B}_1\boldsymbol{\epsilon}_1^{(1)} + \boldsymbol{\epsilon}_1^{(2)}) \\ &\approx \mathbf{H}_1^o(\mathbf{B}_1\boldsymbol{\epsilon}_1^{(1)} + \boldsymbol{\epsilon}_1^{(2)}) + \Delta\mathbf{H}_1 \cdot \mathbf{B}_1\boldsymbol{\epsilon}_1^{(1)}, \end{aligned} \tag{46}$$

where  $\Delta\mathbf{H}_1$  and its related differentials can be obtained by matrix differential as follows:

$$\Delta\mathbf{H}_1 = \mathbf{P}_1^{o-1}\Delta\mathbf{G}_1^T\mathbf{W}_1 + \Delta(\mathbf{P}_1^{-1})\mathbf{G}_1^{oT}\mathbf{W}_1, \tag{47}$$

$$\Delta(\mathbf{P}_1^{-1}) = -\mathbf{P}_1^{o-1}\Delta\mathbf{P}_1\mathbf{P}_1^{o-1}, \tag{48}$$

$$\Delta\mathbf{P}_1 = \Delta\mathbf{G}_1^T\mathbf{W}_1\mathbf{G}_1^o + \mathbf{G}_1^{oT}\mathbf{W}_1\Delta\mathbf{G}_1. \tag{49}$$

Putting Equation (47) through Equation (49) into Equation (46) gives the estimation error vector in the first stage

$$\Delta\boldsymbol{\phi}_1 \approx \left(\mathbf{H}_1^o \cdot \mathbf{B}_1\boldsymbol{\epsilon}_1^{(1)} + \mathbf{H}_1^o\boldsymbol{\epsilon}_1^{(2)}\right) + \left(\mathbf{P}_1^{o-1}\Delta\mathbf{G}_1^T\mathbf{K}_1 \cdot \mathbf{B}_1\boldsymbol{\epsilon}_1^{(1)} - \mathbf{H}_1^o\Delta\mathbf{G}_1\mathbf{H}_1^o \cdot \mathbf{B}_1\boldsymbol{\epsilon}_1^{(1)}\right). \tag{50}$$

We note that the weight matrix  $\mathbf{W}_1$  has no random errors in the first stage.

As in the first stage, the estimation error vector in the second stage is

$$\begin{aligned} \Delta\boldsymbol{\phi}_2 &= \hat{\boldsymbol{\phi}}_2 - \boldsymbol{\phi}_2 \\ &= \mathbf{H}_2(\mathbf{B}_2\boldsymbol{\epsilon}_2^{(1)} + \boldsymbol{\epsilon}_2^{(2)}) \\ &\approx \mathbf{H}_2^o(\mathbf{B}_2\boldsymbol{\epsilon}_2^{(1)} + \boldsymbol{\epsilon}_2^{(2)}) + \Delta\mathbf{H}_2 \cdot \mathbf{B}_2\boldsymbol{\epsilon}_2^{(1)}. \end{aligned} \tag{51}$$

Before moving on to the expression of  $\Delta\mathbf{H}_2$ , we note that we use  $\hat{\mathbf{B}}_2$  rather than  $\mathbf{B}_2$  in the practical implementation. So,  $\Delta\mathbf{H}_2$  can be obtained by matrix differential as follows:

$$\Delta\mathbf{H}_2 = \Delta(\mathbf{P}_2^{-1})\mathbf{G}_2^{oT}\mathbf{W}_2 + \mathbf{P}_2^{o-1}\Delta\mathbf{G}_2^T\mathbf{W}_2 + \mathbf{P}_2^{o-1}\mathbf{G}_2^{oT}\Delta\mathbf{W}_2, \tag{52}$$

$$\Delta(\mathbf{P}_2^{-1}) = -\mathbf{P}_2^{o-1}\Delta\mathbf{P}_2\mathbf{P}_2^{o-1}, \tag{53}$$

$$\Delta\mathbf{P}_2 = \Delta\mathbf{G}_2^T\mathbf{W}_2\mathbf{G}_2^o + \mathbf{G}_2^{oT}\mathbf{W}_2\Delta\mathbf{G}_2 + \mathbf{G}_2^{oT}\Delta\mathbf{W}_2\mathbf{G}_2^o, \tag{54}$$

$$\Delta\mathbf{W}_2 = \mathbf{B}_2^{-T}\Delta\mathbf{P}_1\mathbf{B}_2^{-1} - \mathbf{W}_2\Delta\mathbf{B}_2\mathbf{B}_2^{-1} - \mathbf{B}_2^{-T}\Delta\mathbf{B}_2^T\mathbf{W}_2. \tag{55}$$

By comparing Equation (52) through Equation (54) with Equation (47) through Equation (49), we see that  $\Delta\mathbf{W}_2$  in Equation (55) contributes new perturbation resulted from practical implementation, which are introduced by  $\Delta\mathbf{P}_1$  and  $\Delta\mathbf{B}_2$ . Furthermore,  $\Delta\mathbf{P}_1$  can be expressed with  $\Delta\mathbf{G}_1$  by Equation (49).

Then from Equation (51), combining Equation (52) through Equation (55) together with Equation (49) we have

$$\begin{aligned} \Delta\boldsymbol{\phi}_2 \approx & \left( \mathbf{H}_2^o \cdot \mathbf{B}_2\boldsymbol{\epsilon}_2^{(1)} + \mathbf{H}_2^o\boldsymbol{\epsilon}_2^{(2)} \right) + \left( \mathbf{P}_2^{o-1}\Delta\mathbf{G}_2^T\mathbf{K}_2 \cdot \mathbf{B}_2\boldsymbol{\epsilon}_2^{(1)} - \mathbf{H}_2^o\Delta\mathbf{G}_2\mathbf{H}_2^o \cdot \mathbf{B}_2\boldsymbol{\epsilon}_2^{(1)} \right) \\ & + \left( -\mathbf{H}_2^o\Delta\mathbf{B}_2\mathbf{B}_2^{-1}\mathbf{U} \cdot \mathbf{B}_2\boldsymbol{\epsilon}_2^{(1)} - \mathbf{V} \cdot \mathbf{B}_2^{-T}\Delta\mathbf{B}_2^T\mathbf{K}_2 \cdot \mathbf{B}_2\boldsymbol{\epsilon}_2^{(1)} \right) \\ & + \left( \mathbf{V} \cdot \mathbf{B}_2^{-T} \cdot \mathbf{G}_1^{oT}\mathbf{W}_1\Delta\mathbf{G}_1 \cdot \mathbf{B}_2^{-1}\mathbf{U} \cdot \mathbf{B}_2\mathbf{H}_1^o\mathbf{B}_1\boldsymbol{\epsilon}_1^{(1)} + \mathbf{V} \cdot \mathbf{B}_2^{-T} \cdot \Delta\mathbf{G}_1^T\mathbf{W}_1\mathbf{G}_1^o \cdot \mathbf{B}_2^{-1}\mathbf{U} \cdot \mathbf{B}_2\mathbf{H}_1^o\mathbf{B}_1\boldsymbol{\epsilon}_1^{(1)} \right). \end{aligned} \tag{56}$$

The last line above is by Equation (46) and Equation (49).

Finally, taking expectation of  $\Delta\boldsymbol{\phi}_1$  and  $\Delta\boldsymbol{\phi}_2$  will yield  $\boldsymbol{\mu}_1$  and  $\boldsymbol{\mu}_2$ . This involves complicated matrix partitioning and multivariate statistical analysis. Interested readers may refer to Appendix E.

### 6.2. Covariance Matrix

We then derive the covariance matrix of our estimator. The covariance analysis here will be up to the first-order statistics of the estimator errors, i.e., the error terms higher than the first-order are ignored.

$$\begin{aligned} \text{cov}(\hat{\boldsymbol{\theta}}) &= \text{cov}(\hat{\boldsymbol{\phi}}_1(1, 2, M + 3, M + 4) - \hat{\boldsymbol{\phi}}_2) \\ &= \text{cov}(\boldsymbol{\phi}_1(1, 2, M + 3, M + 4) + \Delta\boldsymbol{\phi}_1(1, 2, M + 3, M + 4) - \boldsymbol{\phi}_2 - \Delta\boldsymbol{\phi}_2) \\ &= \text{cov}(\boldsymbol{\phi}_1(1, 2, M + 3, M + 4) - \Delta\boldsymbol{\phi}_2) \quad (\text{by (36)}) \\ &= \text{cov}(\Delta\boldsymbol{\phi}_2) \\ &\approx \mathbf{P}_2^{o-1}. \end{aligned} \tag{57}$$

We note that Yang et al. have explicitly given the covariance matrix of the two-stage weighted least squares (TS-WLS) method as follows:

$$\text{cov}(\hat{\boldsymbol{\theta}}_{\text{TS-WLS}}) \approx (\mathbf{G}_2^{oT}\mathbf{B}_2^{-T}\mathbf{V}_1^{-1}\mathbf{B}_2^{-1}\mathbf{G}_2^o)^{-1}, \tag{58}$$

where

$$\mathbf{V}_1 = (\mathbf{G}_1^{oT}\tilde{\mathbf{W}}_1\mathbf{G}_1^o)^{-1}\mathbf{G}_1^{oT}\tilde{\mathbf{W}}_1\mathbf{B}_1\text{cov}(\boldsymbol{\epsilon}_1^{(1)})\mathbf{B}_1^T\tilde{\mathbf{W}}_1^T\mathbf{G}_1^o(\mathbf{G}_1^{oT}\tilde{\mathbf{W}}_1\mathbf{G}_1^o)^{-T}, \tag{59}$$

$$\tilde{\mathbf{W}}_1 = \mathbf{B}_1^{-T}(c^2\mathbf{Q}_m)^{-1}\mathbf{B}_1^{-1}. \tag{60}$$

We now claim the following proposition.

**Proposition 1.**

$$\text{cov}(\hat{\boldsymbol{\theta}}_{\text{TS-WLS}}) \succeq \text{cov}(\hat{\boldsymbol{\theta}}). \tag{61}$$

**Proof.** By Equation (57) and Equation (58), it is enough to prove

$$\mathbf{P}_2^o \succeq \mathbf{G}_2^{oT} \mathbf{B}_2^{-T} \mathbf{V}_1^{-1} \mathbf{B}_2^{-1} \mathbf{G}_2^o. \tag{62}$$

To get started, we expand the left side of Equation (62).

$$\begin{aligned} \mathbf{P}_2^o &= \mathbf{G}_2^{oT} \mathbf{W}_2 \mathbf{G}_2^o \\ &\approx \mathbf{G}_2^{oT} (\mathbf{B}_2^{-T} \mathbf{P}_1^o \mathbf{B}_2^{-1}) \mathbf{G}_2^o \quad (\text{by Equation (43)}) \\ &= \mathbf{G}_2^{oT} [\mathbf{B}_2 (\mathbf{G}_1^{oT} \mathbf{W}_1 \mathbf{G}_1^o)^{-1} \mathbf{B}_2^T]^{-1} \mathbf{G}_2^o \\ &= \mathbf{G}_2^{oT} \mathbf{B}_2^{-T} \mathbf{G}_1^{oT} \mathbf{W}_1 \mathbf{G}_1^o \mathbf{B}_2^{-1} \mathbf{G}_2^o \\ &= \mathbf{G}_2^{oT} \mathbf{B}_2^{-T} \mathbf{V}_2^{-1} \mathbf{B}_2^{-1} \mathbf{G}_2^o \quad (\text{by Equation (33)}), \end{aligned} \tag{63}$$

where

$$\mathbf{V}_2 = [\mathbf{G}_1^{oT} \mathbf{B}_1^{-T} \text{cov}(\epsilon_1^{(1)})^{-1} \mathbf{B}_1^{-1} \mathbf{G}_1^o]^{-1}. \tag{64}$$

As  $\mathbf{B}_2^{-1} \mathbf{G}_2^o$  has full column rank, to complete the proof, we need only show that

$$\mathbf{V}_1 \succeq \mathbf{V}_2. \tag{65}$$

We can prove Equation (65) by Matrix Schwarz inequality (Lemma 1.1 in [43]). Because  $\mathbf{B}_1 \text{cov}(\epsilon_1^{(1)}) \mathbf{B}_1^T$  is positive-definite, we can provide a Cholesky factorization as follows:

$$\mathbf{B}_1 \text{cov}(\epsilon_1^{(1)}) \mathbf{B}_1^T = \mathbf{L} \mathbf{L}^T, \tag{66}$$

where  $\mathbf{L}$  is a unique lower triangular matrix. Let

$$\mathbf{P} = \mathbf{L}^{-1} \mathbf{G}_1^o, \quad \mathbf{Q} = \mathbf{L}^T \tilde{\mathbf{W}}_1^T \mathbf{G}_1^o (\mathbf{G}_1^{oT} \tilde{\mathbf{W}}_1 \mathbf{G}_1^o)^{-T}. \tag{67}$$

By Matrix Schwarz inequality, we get

$$\mathbf{Q}^T \mathbf{Q} \succeq (\mathbf{P}^T \mathbf{Q})^T (\mathbf{P}^T \mathbf{P})^{-1} (\mathbf{P}^T \mathbf{Q}). \tag{68}$$

Then with a straightforward verification, we have

$$\mathbf{Q}^T \mathbf{Q} = \mathbf{V}_1, \quad \mathbf{P}^T \mathbf{P} = \mathbf{V}_2^{-1}, \quad \mathbf{P}^T \mathbf{Q} = \mathbf{I}. \tag{69}$$

The equations above imply that  $\mathbf{V}_1 \succeq \mathbf{V}_2$ , as desired.  $\square$

We close this section by establishing the approximate statistical efficiency of our estimator as the proposition below shows.

**Proposition 2.** *When the small error conditions in 1 through 6 are satisfied,*

$$\text{HCRB}_\theta \approx \text{cov}(\hat{\theta}). \tag{70}$$

**Proof.** In view of Equation (13) and Equation (57), we will prove that

$$\left\{ \left( \frac{\partial \bar{\mathbf{m}}}{\partial \theta} \right)^T \left[ \mathbf{Q}_m + \frac{\partial \bar{\mathbf{m}}}{\partial \mathbf{z}} \mathbf{Q}_z \left( \frac{\partial \bar{\mathbf{m}}}{\partial \mathbf{z}} \right)^T \right]^{-1} \frac{\partial \bar{\mathbf{m}}}{\partial \theta} \right\} \Bigg|_{\mathbf{z}=\bar{\mathbf{z}}} \approx \mathbf{P}_2^o. \tag{71}$$

By Equations (63), (64) and (34), the right side of Equation (71) is

$$\mathbf{P}_2^o = \mathbf{G}_3^T (\mathbf{D}_z \mathbf{Q}_z \mathbf{D}_z^T + c^2 \mathbf{Q}_m)^{-1} \mathbf{G}_3, \tag{72}$$

where

$$\mathbf{G}_3 = \mathbf{B}_1^{-1} \mathbf{G}_1^o \mathbf{B}_2^{-1} \mathbf{G}_2^o. \tag{73}$$

Expanding Equation (73) (in Appendix F) and comparing it with Equation (A1), we can show that, under the small error conditions,

$$\mathbf{G}_3 \approx -c \left. \frac{\partial \bar{\mathbf{m}}}{\partial \boldsymbol{\theta}} \right|_{\mathbf{z}=\bar{\mathbf{z}}}. \tag{74}$$

Furthermore, comparing Equation (A16) with Equation (A2), we arrive at

$$\mathbf{D}_z \approx c \left. \frac{\partial \bar{\mathbf{m}}}{\partial \mathbf{z}} \right|_{\mathbf{z}=\bar{\mathbf{z}}}. \tag{75}$$

If we plug Equation (74) and Equation (75) into Equation (72), we get Equation (71).  $\square$

### 6.3. Time and Space Complexity

The computational load of our algorithm is focused on solving WLS problems. Singular value decomposition (SVD) is an efficient algorithm for solving WLS problems. With the method of truncated SVD, the time complexity is  $\mathcal{O}(2mn^2 + n^3 + mn + n)$  and the space complexity is  $\mathcal{O}(3n^2 + 2mn + 3n)$  for a matrix of size  $m \times n$  [44].

To facilitate complexity analysis, we list the matrices involved in the algorithm and their sizes in Table 3. As can be seen from the table, under the assumptions that  $M \ll N$ , the computational complexity of the first stage is dominant in the total computational complexity. For the design matrix  $\mathbf{G}_1$  in the first stage,  $m = 2MN$  and  $n = 2M + 4$ . By keeping the highest order term of  $N$  and its coefficient, it can be seen that our algorithm takes  $\mathcal{O}(N[4(4M^3 + 17M^2 + 18M)])$  time and  $\mathcal{O}(N[8M(M + 2)])$  space. In summary, our algorithm has linear complexity both in time and space.

Table 3. Size of matrices.

Matrix Notations	Matrix Sizes
$\mathbf{G}_1$	$(2MN) \times (2M + 4)$
$\boldsymbol{\phi}_1$	$(2M + 4) \times 1$
$\mathbf{h}_1$	$(2MN) \times 1$
$\mathbf{B}_1$	$(2MN) \times (2MN)$
$\mathbf{W}_1$	$(2MN) \times (2MN)$
$\mathbf{G}_{1,i}$	$(2N) \times (2M + 4)$
$\mathbf{B}_{1,i}$	$(2N) \times (2N)$
$\mathbf{G}_2$	$(2M + 4) \times 4$
$\boldsymbol{\phi}_2$	$4 \times 1$
$\mathbf{h}_2$	$(2M + 4) \times 1$
$\mathbf{B}_2$	$(2M + 4) \times (2M + 4)$
$\mathbf{W}_2$	$(2M + 4) \times (2M + 4)$
$\mathbf{Q}_m$	$(2MN) \times (2MN)$
$\mathbf{Q}_z$	$(2M + 2N) \times (2M + 2N)$
$\mathbf{D}_z$	$(2MN) \times (2M + 2N)$

## 7. Results and Discussion

In the previous section, we have theoretically analyzed the performance of our estimator. Now we ascertain the performance of our estimator via computer simulations. Our simulations are divided into four subsections. Section 7.1 compares the error covariance matrix of our estimator with HCRB

and the ones of two typical estimators, i.e., the spherical-interpolation initialized Taylor series method (SI-TS) [28,45] and TS-WLS [5]. Then, surface plots of the biases are shown in Section 7.2. In Section 7.3, we empirically explore the time complexity of our estimator for locating multiple disjoint targets. Finally, we use 80 randomly generated large-scale localization scenarios to further test the proposed estimator in Section 7.4.

The first three subsections base on the simulation settings of [6]. Specifically, the simulations use  $M = 3$  transmitters and  $N = 5$  receivers to determine the unknown position  $\mathbf{u}$  and velocity  $\dot{\mathbf{u}}$  of a moving target. As in [6], the nominal positions of the sensors are known and given as follows.  $\bar{\mathbf{t}}_1 = [1500, 1500]^T$  m,  $\bar{\mathbf{t}}_2 = [-900, 4000]^T$  m,  $\bar{\mathbf{t}}_3 = [-3000, -4000]^T$  m,  $\bar{\mathbf{s}}_1 = [-1000, 3000]^T$  m,  $\bar{\mathbf{s}}_2 = [2500, -500]^T$  m,  $\bar{\mathbf{s}}_3 = [-3000, 1000]^T$  m,  $\bar{\mathbf{s}}_4 = [2000, -4000]^T$  m, and  $\bar{\mathbf{s}}_5 = [-2000, -2000]^T$  m. Graphically, the nominal location geometry is shown in Figure 2.

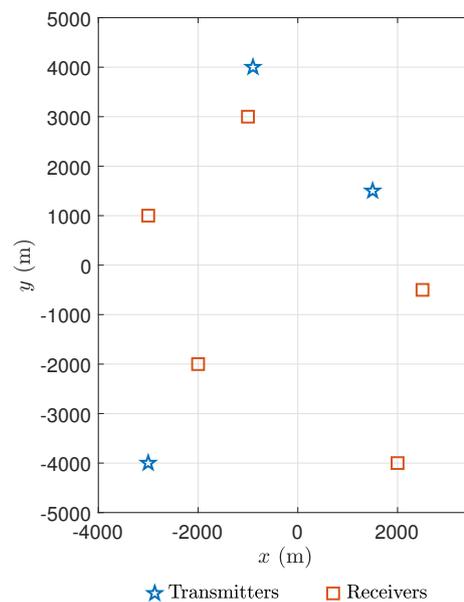


Figure 2. Nominal location geometry for computer simulations.

The additional common settings for Sections 7.1 and 7.2 are as follows. The target is at  $\mathbf{u} = [0, 2000]^T$  m with velocity  $\dot{\mathbf{u}} = [20, 10]^T$  m/s and the signal propagation speed is  $c = 1500$  m/s. The observation error covariance matrix related to the transmitter at  $\mathbf{t}_i$  is  $\mathbf{Q}_{\mathbf{m}_i} = \text{blkdiag}(\sigma_\tau^2 \mathbf{R}, \sigma_f^2 \mathbf{R})$  for  $i = 1, 2, \dots, M$ , where  $\sigma_\tau$  is a given positive constant,  $\sigma_f^2 = \sigma_\tau^2/10$ , and  $\mathbf{R} = 0.5 \mathbf{1}_{N \times N} + 0.5 \mathbf{I}_N$  [31]. The sensors' position error covariance matrix is  $\mathbf{Q}_z = \sigma_z^2 \mathbf{I}_{2(M+N)}$ , where  $\sigma_z$  is a given positive constant.

We list the settings of the Monte Carlo simulations in Table 4 to illustrate our experiments more clearly. Using Equation (8) based on Table 4, we generate data for simulations.

**Table 4.** Monte Carlo simulation settings.

Quantities	Values
$M$	3
$N$	5
$c$	1500 m/s
$\mathbf{t}_1$	$[1500, 1500]^T$ m
$\mathbf{t}_2$	$[-900, 4000]^T$ m
$\mathbf{t}_3$	$[-3000, -4000]^T$ m
$\mathbf{s}_1$	$[-1000, 3000]^T$ m
$\mathbf{s}_2$	$[2500, -500]^T$ m
$\mathbf{s}_3$	$[-3000, 1000]^T$ m
$\mathbf{s}_4$	$[2000, -4000]^T$ m
$\mathbf{s}_5$	$[-2000, -2000]^T$ m
$\mathbf{R}$	$0.5 \mathbf{1}_{N \times N} + 0.5 \mathbf{I}_N$
$\sigma_\tau$	0.02 : 0.02 : 0.2 s
$\sigma_f$	$\sigma_\tau / 10$
$\sigma_z$	20 : 20 : 200 m
$\mathbf{u}$	$[0, 2000]^T$ m
$\dot{\mathbf{u}}$	$[20, 10]^T$ m

### 7.1. Performance Comparison

We now turn to the performance comparison of several estimators. For a specific estimator  $\hat{\boldsymbol{\theta}}$  of the unknown parameter vector  $\boldsymbol{\theta}$ , its performance can be measured by the root-mean-square error (RMSE), which is defined as follows.

$$\text{RMSE} = \sqrt{\frac{1}{L} \left( \sum_{\ell=1}^L \|\boldsymbol{\theta}_\ell - \boldsymbol{\theta}\|^2 \right)},$$

where  $L$  is the number of Monte Carlo simulations and  $\boldsymbol{\theta}_\ell$  is the  $\ell$ -th random realization of  $\hat{\boldsymbol{\theta}}$ .

The RMSEs of our estimator, SI-TS and TS-WLS are compared with HCRB here. The simulation settings are as follows.  $\sigma_\tau$  is 0.02 s,  $\sigma_z$  is from 0 m to 200 m with a step size of 20 m, and the number of Monte Carlo simulations is  $10^4$  for each value of  $\sigma_z$ . The comparison curves for both the position estimator and the velocity estimator are plotted respectively in Figures 3 and 4. It is evident that our estimator has the least RMSE and can attain the HCRB accuracy at lower noise levels for determining both the position and the velocity.

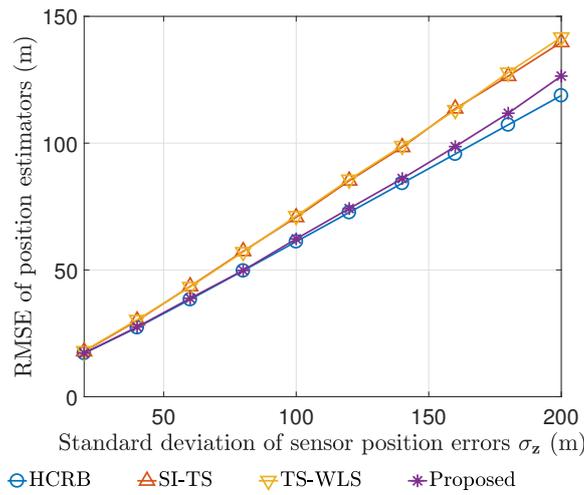


Figure 3. RMSE and HCRB for position estimator.

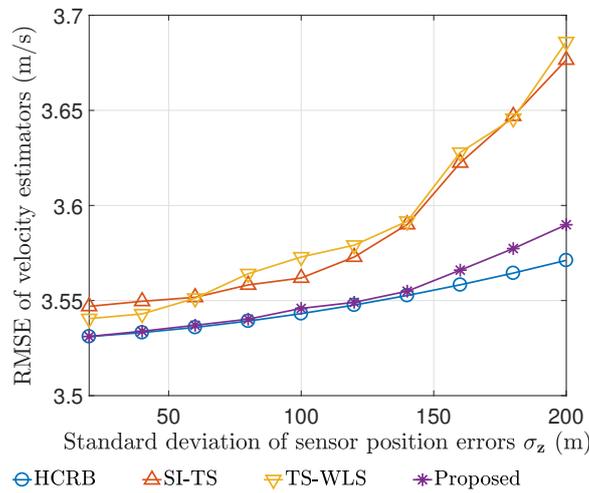


Figure 4. RMSE and HCRB for velocity estimator.

7.2. Bias Calculation

In this subsection, we evaluate the bias of our estimator. The simulation settings are as follows.  $\sigma_\tau$  is from 0.02 s to 0.2 s with a step size of 0.02 s, and  $\sigma_z$  is from 20 m to 200 m with a step size of 20 m. The norms of the theoretical bias vectors of  $\hat{\mathbf{u}}$  and  $\hat{\dot{\mathbf{u}}}$  are calculated using results from Section 6 and further visualized as surface plots in Figures 5 and 6. It is consistent with intuition that the biases of both  $\hat{\mathbf{u}}$  and  $\hat{\dot{\mathbf{u}}}$  increase with both  $\sigma_\tau$  and  $\sigma_z$ . It should be noted that the biases are relatively small compared with the norms of  $\mathbf{u} = [0, 2000]^T$  m and  $\dot{\mathbf{u}} = [20, 10]^T$  m/s, even if the noise levels are high, e.g.,  $\sigma_\tau = 0.2$  s and  $\sigma_z = 200$  m.

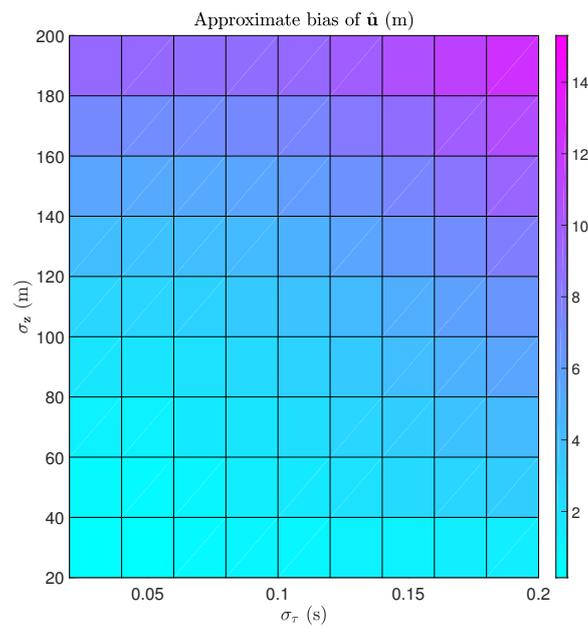


Figure 5. Surface plot of norm of the approximate bias of  $\hat{\mathbf{u}}$ .

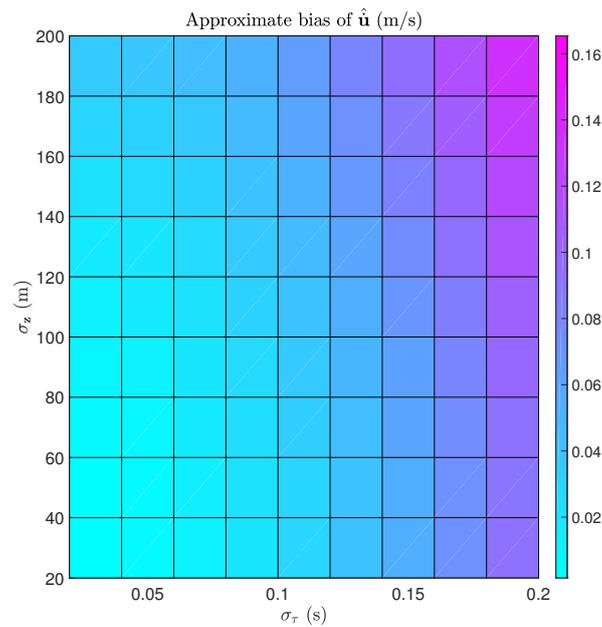
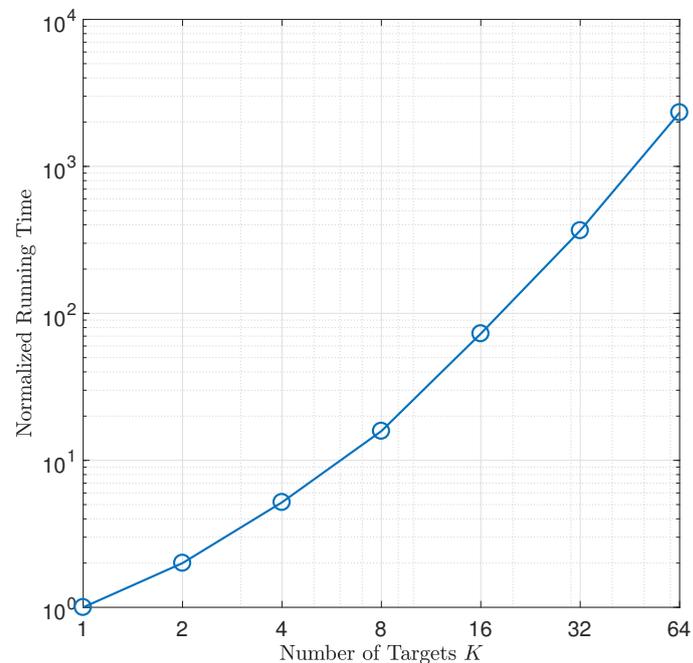


Figure 6. Surface plot of norm of the approximate bias of  $\hat{\mathbf{u}}$ .

### 7.3. Localizing Multiple Disjoint Targets

The aim of this section is to evaluate the computational complexity of the algorithm in the sense of scalability, since the WLS algorithm involved in our estimator is computationally efficient. One advantage of our estimator is that it is ready to be extended to location of multiple disjoint targets by concatenating the data matrices in Section 5. Let the number of the disjoint targets be  $K$ , and  $10^3$  Monte Carlo experiments of joint location are performed for each value of  $K$  ( $= 1, 2, 4, 8, 16, 32, 64$ ). Then, the running time of the  $10^3$  experiments are recorded. For convenience of comparison, we normalize the running time for each  $K$  with the one for  $K = 1$ . The normalized running times are plotted in Figure 7 using log-log scale. It can be seen that the running time grows almost exponentially with respect to the number of targets. This observation indicates that localizing

multiple targets sequentially is more time-efficient than localizing them simultaneously using our estimator. Such defects may root in the fact that our joint estimator does not share the nuisance parameters across the multiple targets.



**Figure 7.** Normalized running time for locating multiple disjoint targets.

#### 7.4. Large-Scale Simulation Experiments

The location scenario in Section 7.1 through Section 7.3 is the one examined in [6]. In order to evaluate the performance of the proposed estimator more comprehensively, we design the following large-scale random experiments. In view of the symmetry of the transmitter  $t_i$  and receiver  $s_j$  in the observation model, we fix the number of transmitters to 1, and increase the number of receivers from 21 to 100. The transmitter's position is fixed at  $[0, 0]^T$  m. Both the  $x$ -ordinate and  $y$ -ordinate of the individual receiver's position have the uniform probability distribution within the interval  $[-5000, 5000]^T$  m.  $\sigma_z = 20$  m, and  $\sigma_\tau = 0.02$  s. Other unspecified settings in these experiments are referred in Table 4. In each location scenario, we conduct  $10^4$  Monte Carlo simulations. Then we explore the effect of the number of receivers on the bias/RMSE and computational complexity of the proposed estimator in Figures 8 and 9.

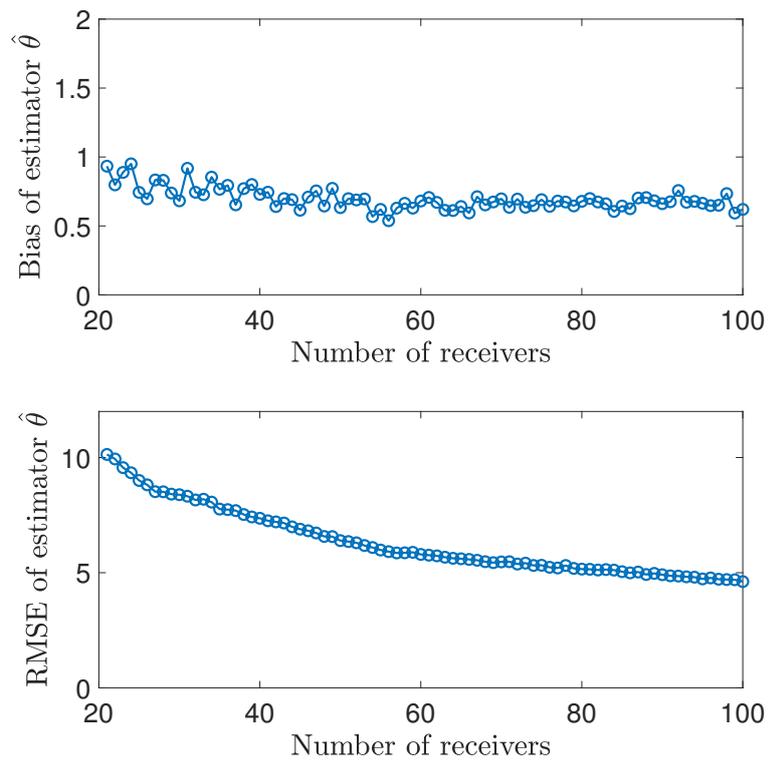


Figure 8. Normalized running time for locating multiple disjoint targets.

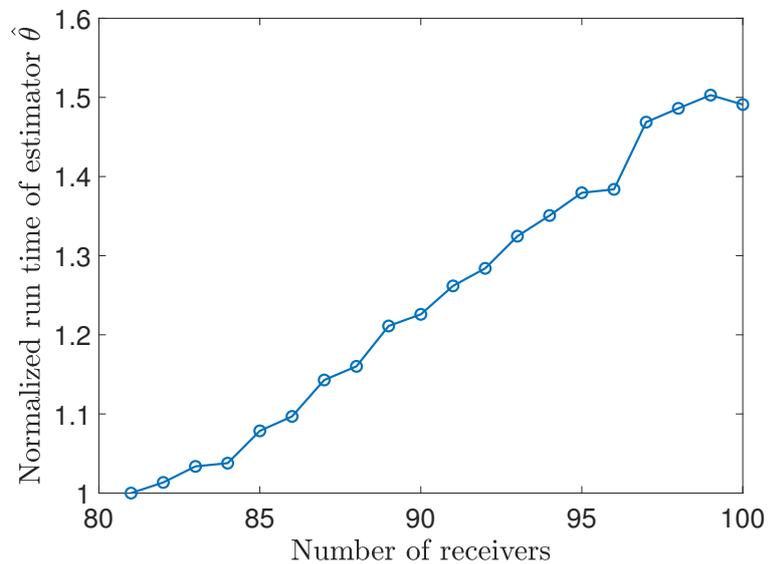


Figure 9. Normalized running time for locating multiple disjoint targets.

Figure 8 shows that increasing the number of receivers helps to reduce the RMSE of the estimator. It should be noted that increasing the number of receivers does not lead to a decrease of bias. This fact may imply that designing unbiased estimators is an inherently difficult problem in nonlinear estimation.

In addition, as can be seen in Figure 9, the estimator’s relative running time scales linearly as more receivers are used when the number of the receivers is large enough (e.g.,  $N > 80$  here). This trend coincides with the theoretical linear complexity obtained in Section 6.3.

### 8. Conclusions

This paper develops a non-iterative solution to the nonlinear hybrid parameter estimation problem of determining the position and velocity of a moving target in a multistatic sonar system in the presence of sensor position uncertainties. It outperforms conventional methods such as SI-TS and TS-WLS in RMSE, and can achieve the HCRB for moderate Gaussian observation noises and sensor position errors. Our estimator involves only two WLS minimizations. Thus, it is computationally efficient, and does not need to deal with the difficulties of initialization and local convergence. Moreover, we obtain the bias vector and covariance matrix of this estimator using perturbation analysis and multivariate statistics.

**Author Contributions:** Conceptualization, X.W. and J.L.; methodology, X.W.; software, X.W. and Z.Y.; validation, L.Y.; writing—original draft preparation, X.W. and Z.Y.; writing—review and editing, X.W. and L.Y.; funding acquisition, X.W. and L.Y. 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 (Grant No. 61703185), the Natural Science Foundation of Jiangsu Province in China (Grant No. BK20140166) and the 111 Project (Grant no. B12018).

**Acknowledgments:** The authors would like to thank the anonymous reviewers and the editor for their careful reviews and constructive suggestions to help us improve the quality of this paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

### Abbreviations

The following abbreviations are used in this manuscript:

HCRB	Hybrid Cramér–Rao bound
WLS	Weighted least-squares
SI-TS	Spherical-interpolation initialized Taylor series method
SVD	Singular value decomposition
TS-WLS	Two-stage weighted least squares method
RMSE	Root-mean-square error

### Appendix A. Jacobian Matrices for HCRB

#### Appendix A.1. Jacobian Matrix of Target Position and Velocity

$$\frac{\partial \bar{\mathbf{m}}}{\partial \theta} = \begin{bmatrix} \partial \bar{\mathbf{m}}_1 / \partial \mathbf{u} & \partial \bar{\mathbf{m}}_1 / \partial \dot{\mathbf{u}} \\ \partial \bar{\mathbf{m}}_2 / \partial \mathbf{u} & \partial \bar{\mathbf{m}}_2 / \partial \dot{\mathbf{u}} \\ \vdots & \vdots \\ \partial \bar{\mathbf{m}}_M / \partial \mathbf{u} & \partial \bar{\mathbf{m}}_M / \partial \dot{\mathbf{u}} \end{bmatrix}, \tag{A1}$$

where

$$\frac{\partial \bar{\mathbf{m}}_i}{\partial \mathbf{u}} = \left[ \left( \frac{\partial \tau_{i,1}^o}{\partial \mathbf{u}} \right)^T, \dots, \left( \frac{\partial \tau_{i,N}^o}{\partial \mathbf{u}} \right)^T, \left( \frac{\partial f_{i,1}^o}{\partial \mathbf{u}} \right)^T, \dots, \left( \frac{\partial f_{i,N}^o}{\partial \mathbf{u}} \right)^T \right]^T,$$

$$\frac{\partial \bar{\mathbf{m}}_i}{\partial \dot{\mathbf{u}}} = \left[ \left( \frac{\partial \tau_{i,1}^o}{\partial \dot{\mathbf{u}}} \right)^T, \dots, \left( \frac{\partial \tau_{i,N}^o}{\partial \dot{\mathbf{u}}} \right)^T, \left( \frac{\partial f_{i,1}^o}{\partial \dot{\mathbf{u}}} \right)^T, \dots, \left( \frac{\partial f_{i,N}^o}{\partial \dot{\mathbf{u}}} \right)^T \right]^T,$$

for  $i = 1, 2, \dots, M$ .

$$\begin{aligned} \partial\tau_{i,j}^o/\partial\mathbf{u} &= \frac{1}{c}(\rho_{\mathbf{u},t_i} + \rho_{\mathbf{u},s_j})^T, \\ \partial\tau_{i,j}^o/\partial\dot{\mathbf{u}} &= \mathbf{0}_{1\times 2}, \\ \partial f_{i,j}^o/\partial\mathbf{u} &= \frac{1}{c}\dot{\mathbf{u}}^T(\mathbf{A}_{\mathbf{u},t_i} + \mathbf{A}_{\mathbf{u},s_j}), \\ \partial f_{i,j}^o/\partial\dot{\mathbf{u}} &= \frac{1}{c}(\rho_{\mathbf{u},t_i} + \rho_{\mathbf{u},s_j})^T, \end{aligned}$$

for  $i = 1, 2, \dots, M$  and  $j = 1, 2, \dots, N$ .

Appendix A.2. Jacobian Matrix of Sensor Positions

$$\frac{\partial\bar{\mathbf{m}}}{\partial\mathbf{z}} = \begin{bmatrix} \partial\bar{\mathbf{m}}_1/\partial\mathbf{t} & \partial\bar{\mathbf{m}}_1/\partial\mathbf{s} \\ \partial\bar{\mathbf{m}}_2/\partial\mathbf{t} & \partial\bar{\mathbf{m}}_2/\partial\mathbf{s} \\ \vdots & \vdots \\ \partial\bar{\mathbf{m}}_M/\partial\mathbf{t} & \partial\bar{\mathbf{m}}_M/\partial\mathbf{s} \end{bmatrix}, \tag{A2}$$

where

$$\begin{aligned} \frac{\partial\bar{\mathbf{m}}_i}{\partial\mathbf{t}} &= \left[ \left( \frac{\partial\tau_{i,1}^o}{\partial\mathbf{t}} \right)^T, \dots, \left( \frac{\partial\tau_{i,N}^o}{\partial\mathbf{t}} \right)^T, \left( \frac{\partial f_{i,1}^o}{\partial\mathbf{t}} \right)^T, \dots, \left( \frac{\partial f_{i,N}^o}{\partial\mathbf{t}} \right)^T \right]^T, \\ \frac{\partial\bar{\mathbf{m}}_i}{\partial\mathbf{s}} &= \left[ \left( \frac{\partial\tau_{i,1}^o}{\partial\mathbf{s}} \right)^T, \dots, \left( \frac{\partial\tau_{i,N}^o}{\partial\mathbf{s}} \right)^T, \left( \frac{\partial f_{i,1}^o}{\partial\mathbf{s}} \right)^T, \dots, \left( \frac{\partial f_{i,N}^o}{\partial\mathbf{s}} \right)^T \right]^T, \end{aligned}$$

for  $i = 1, 2, \dots, M$ .

$$\begin{aligned} \partial\tau_{i,j}^o/\partial\mathbf{t}_k &= \begin{cases} (\rho_{\mathbf{t}_i,\mathbf{u}} - \rho_{\mathbf{t}_i,s_j})^T/c, & \text{if } i = k, \\ \mathbf{0}_{1\times 2}, & \text{if } i \neq k, \end{cases} \\ \partial\tau_{i,j}^o/\partial\mathbf{s}_l &= \begin{cases} (\rho_{\mathbf{s}_j,\mathbf{u}} - \rho_{\mathbf{s}_j,t_i})^T/c, & \text{if } j = l, \\ \mathbf{0}_{1\times 2}, & \text{if } j \neq l, \end{cases} \\ \partial f_{i,j}^o/\partial\mathbf{t}_k &= \begin{cases} -\dot{\mathbf{u}}^T\mathbf{A}_{\mathbf{t}_i,\mathbf{u}}/c, & \text{if } i = k, \\ \mathbf{0}_{1\times 2}, & \text{if } i \neq k, \end{cases} \\ \partial f_{i,j}^o/\partial\mathbf{s}_l &= \begin{cases} -\dot{\mathbf{u}}^T\mathbf{A}_{\mathbf{s}_j,\mathbf{u}}/c, & \text{if } j = l, \\ \mathbf{0}_{1\times 2}, & \text{if } j \neq l, \end{cases} \end{aligned}$$

for  $i = 1, 2, \dots, M, j = 1, 2, \dots, N, k = 1, 2, \dots, M$  and  $l = 1, 2, \dots, N$ .

Appendix B. Matrices Related to Weighted Least Squares

A weighted least squares problem is an optimization problem as follows.

$$\text{minimize}_{\boldsymbol{\phi}} \quad (\mathbf{h} - \mathbf{G}\boldsymbol{\phi})^T\mathbf{W}(\mathbf{h} - \mathbf{G}\boldsymbol{\phi}), \tag{A3}$$

where  $\mathbf{G}$  is the design matrix (of full column rank),  $\mathbf{h}$  is the observation vector,  $\boldsymbol{\phi}$  is the parameter vector, and  $\mathbf{W}$  is the (positive definite) weighted matrix. We refer  $(\mathbf{h} - \mathbf{G}\boldsymbol{\phi})$  as residual vector.

We introduce the weighted residual vector  $\mathbf{r}_W$  as follows.

$$\mathbf{r}_W = \mathbf{W}(\mathbf{h} - \mathbf{G}\boldsymbol{\phi}). \tag{A4}$$

It follows from Equation (A4) that

$$\mathbf{W}^{-1} \mathbf{r}_W + \mathbf{G} \boldsymbol{\phi} = \mathbf{h}. \tag{A5}$$

By the orthogonality projection principle of least squares method,

$$\mathbf{G}^T \mathbf{r}_W = \mathbf{0}. \tag{A6}$$

Combining Equation (A5) and Equation (A6), we get

$$\mathbf{M} \begin{bmatrix} \mathbf{r}_W \\ \boldsymbol{\phi} \end{bmatrix} = \begin{bmatrix} \mathbf{h} \\ \mathbf{0} \end{bmatrix}, \tag{A7}$$

where

$$\mathbf{M} = \begin{bmatrix} \mathbf{W}^{-1} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{0} \end{bmatrix}. \tag{A8}$$

After finding the inverse of  $\mathbf{M}$  by matrix inversion lemma [46], we find an interesting fact that the matrices listed in Table 2 have the same form as specific blocks of  $\mathbf{M}^{-1}$  in Equation (A9).

$$\mathbf{M}^{-1} = \begin{bmatrix} \mathbf{W}[\mathbf{I} - \mathbf{G}(\mathbf{G}^T \mathbf{W} \mathbf{G})^{-1} \mathbf{G}^T \mathbf{W}] & \mathbf{W}^T \mathbf{G}(\mathbf{G}^T \mathbf{W} \mathbf{G})^{-1} \\ (\mathbf{G}^T \mathbf{W} \mathbf{G})^{-1} \mathbf{G}^T \mathbf{W} & -(\mathbf{G}^T \mathbf{W} \mathbf{G})^{-1} \end{bmatrix}. \tag{A9}$$

### Appendix C. Linear Model for the First Stage of Our Algorithm

$$\mathbf{h}_1 = [\mathbf{h}_{1,1}^T, \mathbf{h}_{1,2}^T, \dots, \mathbf{h}_{1,M}^T]^T, \tag{A10}$$

where

$$\mathbf{h}_{1,i} = \begin{bmatrix} 2 [\bar{\mathbf{t}}_i^T (\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_1) + c \tau_{i,1} \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_1\|] + c^2 \tau_{i,1}^2 \\ 2 [\bar{\mathbf{t}}_i^T (\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_2) + c \tau_{i,2} \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_2\|] + c^2 \tau_{i,2}^2 \\ \vdots \\ 2 [\bar{\mathbf{t}}_i^T (\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_N) + c \tau_{i,N} \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_N\|] + c^2 \tau_{i,N}^2 \\ c f_{i,1} \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_1\| + c^2 \tau_{i,1} f_{i,1} \\ c f_{i,2} \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_2\| + c^2 \tau_{i,2} f_{i,2} \\ \vdots \\ c f_{i,N} \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_N\| + c^2 \tau_{i,N} f_{i,N} \end{bmatrix},$$

for  $i = 1, 2, \dots, M$ .

$$\mathbf{G}_1 = [\mathbf{G}_{1,1}^T, \mathbf{G}_{1,2}^T, \dots, \mathbf{G}_{1,M}^T]^T, \tag{A11}$$

where

$$\begin{aligned} \mathbf{G}_{1,i} &= \begin{bmatrix} \mathbf{G}_{1,i}(1 : N, 1 : M + 2) & \mathbf{0}_{N \times (M+2)} \\ \mathbf{G}_{1,i}(N + 1 : 2N, 1 : M + 2) & \mathbf{G}_{1,i}(N + 1 : 2N, M + 3 : 2M + 4) \end{bmatrix}, \tag{A12} \\ &= \mathbf{G}_{1,i}(1 : N, 1 : M + 2) \\ &= 2 \mathbf{G}_{1,i}(N + 1 : 2N, M + 3 : 2M + 4) \\ &= \begin{bmatrix} 2 (\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_1)^T & \mathbf{0}_{1 \times (i-1)} & 2(c\tau_{i,1} + \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_1\|) & \mathbf{0}_{1 \times (M-i)} \\ \vdots & \vdots & \vdots & \vdots \\ 2 (\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_N)^T & \mathbf{0}_{1 \times (i-1)} & 2(c\tau_{i,N} + \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_N\|) & \mathbf{0}_{1 \times (M-i)} \end{bmatrix}, \\ &\mathbf{G}_{1,i}(N + 1 : 2N, 1 : M + 2) = [\mathbf{0}_{N \times (i+1)}, c \mathbf{f}_i, \mathbf{0}_{N \times (M-i)}], \end{aligned}$$

for  $i = 1, 2, \dots, M$ .

$$\mathbf{B}_1 = \mathbf{I}_M \otimes \mathbf{B}_{1,i}, \tag{A13}$$

where

$$\mathbf{B}_{1,i} = \begin{bmatrix} 2 \operatorname{diag}(\|\mathbf{u} - \bar{\mathbf{s}}_1\|, \dots, \|\mathbf{u} - \bar{\mathbf{s}}_N\|) & \mathbf{0}_{N \times N} \\ \operatorname{diag}(\rho_{\mathbf{u}, \bar{\mathbf{s}}_1}^T \hat{\mathbf{u}}, \dots, \rho_{\mathbf{u}, \bar{\mathbf{s}}_N}^T \hat{\mathbf{u}}) & \operatorname{diag}(\|\mathbf{u} - \bar{\mathbf{s}}_1\|, \dots, \|\mathbf{u} - \bar{\mathbf{s}}_N\|) \end{bmatrix}. \tag{A14}$$

for  $i = 1, 2, \dots, M$ .

It follows from Equation (20) and Equation (22) that first-order error vector in the first stage is

$$\boldsymbol{\epsilon}_1^{(1)} = \mathbf{D}_z \Delta \mathbf{z} + c \Delta \mathbf{m} = [\mathbf{D}_z, c \mathbf{I}] \begin{bmatrix} \Delta \mathbf{z} \\ \Delta \mathbf{m} \end{bmatrix}, \tag{A15}$$

where

$$\begin{aligned} \mathbf{D}_z &= [\mathbf{D}_t, \mathbf{D}_s], \\ \mathbf{D}_t &= \operatorname{blkdiag}(\mathbf{D}_{t,1}, \mathbf{D}_{t,2}, \dots, \mathbf{D}_{t,M}), \\ \mathbf{D}_s &= [\mathbf{D}_{s,1}^T, \mathbf{D}_{s,2}^T, \dots, \mathbf{D}_{s,M}^T]^T, \\ \mathbf{D}_{t,i} &= [\rho_{\bar{\mathbf{t}}, \mathbf{u}} - \rho_{\bar{\mathbf{t}}, \bar{\mathbf{s}}_1}, \rho_{\bar{\mathbf{t}}, \mathbf{u}} - \rho_{\bar{\mathbf{t}}, \bar{\mathbf{s}}_2}, \dots, \rho_{\bar{\mathbf{t}}, \mathbf{u}} - \rho_{\bar{\mathbf{t}}, \bar{\mathbf{s}}_N}, -\mathbf{1}_{1 \times N} \otimes (\mathbf{A}_{\mathbf{u}, \bar{\mathbf{t}}_i} \hat{\mathbf{u}})]^T, \\ \mathbf{D}_{s,i} &= \begin{bmatrix} \operatorname{blkdiag}(\rho_{\bar{\mathbf{s}}_1, \mathbf{u}} + \rho_{\bar{\mathbf{t}}, \bar{\mathbf{s}}_1}, \rho_{\bar{\mathbf{s}}_2, \mathbf{u}} + \rho_{\bar{\mathbf{t}}, \bar{\mathbf{s}}_2}, \dots, \rho_{\bar{\mathbf{s}}_N, \mathbf{u}} + \rho_{\bar{\mathbf{t}}, \bar{\mathbf{s}}_N})^T \\ -\operatorname{blkdiag}(\mathbf{A}_{\mathbf{u}, \bar{\mathbf{s}}_1} \hat{\mathbf{u}}, \mathbf{A}_{\mathbf{u}, \bar{\mathbf{s}}_2} \hat{\mathbf{u}}, \dots, \mathbf{A}_{\mathbf{u}, \bar{\mathbf{s}}_N} \hat{\mathbf{u}})^T \end{bmatrix}, \end{aligned} \tag{A16}$$

for  $i = 1, 2, \dots, M$ .

By inspection of Equation (23) and Equation (24), the second-order error vector in the first stage  $\boldsymbol{\epsilon}_1^{(2)}$  can be expressed as

$$\boldsymbol{\epsilon}_1^{(2)} = [\boldsymbol{\epsilon}_{1,1}^{(2)T}, \boldsymbol{\epsilon}_{1,2}^{(2)T}, \dots, \boldsymbol{\epsilon}_{1,M}^{(2)T}]^T, \tag{A17}$$

where

$$\boldsymbol{\epsilon}_{1,i}^{(2)} = [(\boldsymbol{\epsilon}_{\tau,i} \circ \boldsymbol{\epsilon}_{\tau,i})^T, (\boldsymbol{\epsilon}_{\tau,i} \circ \boldsymbol{\epsilon}_{f,i})^T]^T,$$

for  $i = 1, 2, \dots, M$ .

### Appendix D. Linear Model for the Second Stage of Our Algorithm

$$\mathbf{h}_2 = \begin{bmatrix} \mathbf{0}_{2 \times 1} \\ \hat{\alpha}_1^2 - \|\hat{\mathbf{u}} - \bar{\mathbf{t}}_1\|^2 \\ \hat{\alpha}_2^2 - \|\hat{\mathbf{u}} - \bar{\mathbf{t}}_2\|^2 \\ \vdots \\ \hat{\alpha}_M^2 - \|\hat{\mathbf{u}} - \bar{\mathbf{t}}_M\|^2 \\ \mathbf{0}_{2 \times 1} \\ \hat{\alpha}_1 \hat{\beta}_1 - (\hat{\mathbf{u}} - \bar{\mathbf{t}}_1)^T \hat{\mathbf{u}} \\ \hat{\alpha}_2 \hat{\beta}_2 - (\hat{\mathbf{u}} - \bar{\mathbf{t}}_2)^T \hat{\mathbf{u}} \\ \vdots \\ \hat{\alpha}_M \hat{\beta}_M - (\hat{\mathbf{u}} - \bar{\mathbf{t}}_M)^T \hat{\mathbf{u}} \end{bmatrix}. \tag{A18}$$

$$\mathbf{G}_2 = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0}_{2 \times 2} \\ 2 [(\bar{\mathbf{t}}_1 - \hat{\mathbf{u}}), \dots, (\bar{\mathbf{t}}_M - \hat{\mathbf{u}})]^T & \mathbf{0}_{M \times 2} \\ \mathbf{0}_{2 \times 2} & \mathbf{I}_2 \\ \mathbf{1}_{M \times 1} \otimes (-\hat{\mathbf{u}}^T) & [(\bar{\mathbf{t}}_1 - \hat{\mathbf{u}}), \dots, (\bar{\mathbf{t}}_M - \hat{\mathbf{u}})]^T \end{bmatrix}. \tag{A19}$$

$$\mathbf{B}_2 = \begin{bmatrix} \text{diag}(-1, -1, 2\boldsymbol{\alpha}^T) & \mathbf{0}_{(M+2)\times(M+2)} \\ \text{diag}(0, 0, \boldsymbol{\beta}^T) & \text{diag}(-1, -1, \boldsymbol{\alpha}^T) \end{bmatrix}. \tag{A20}$$

Furthermore, Equation (37) and Equation (38) indicate that the second-order error vector in the second stage is

$$\boldsymbol{\epsilon}_2^{(2)} = \begin{bmatrix} \mathbf{0}_{2\times 1} \\ (\Delta\mathbf{u}^T\Delta\mathbf{u})\mathbf{1}_{M\times 1} + \Delta\boldsymbol{\alpha} \circ \Delta\boldsymbol{\alpha} \\ \mathbf{0}_{2\times 1} \\ (\Delta\mathbf{u}^T\Delta\dot{\mathbf{u}})\mathbf{1}_{M\times 1} + \Delta\boldsymbol{\alpha} \circ \Delta\boldsymbol{\beta} \end{bmatrix}. \tag{A21}$$

### Appendix E. Formulas for Computing Bias Vector of Our Estimator

#### Appendix E.1. Formulas for Computing Bias in the First Stage

$$\Delta\mathbf{G}_1 = [\Delta\mathbf{G}_{1,1}^T, \Delta\mathbf{G}_{1,2}^T, \dots, \Delta\mathbf{G}_{1,M}^T]^T,$$

where for  $i = 1, 2, \dots, M$ ,

$$\begin{aligned} \Delta\mathbf{G}_{1,i} &= c(2\mathbf{E}_{1,i}\boldsymbol{\delta}\mathbf{e}_{1,i} + \mathbf{E}_{3,i}\boldsymbol{\delta}\mathbf{e}_{1,i} + \mathbf{E}_{2,i}\boldsymbol{\delta}\mathbf{e}_{2,i}), \\ \mathbf{e}_{1,i} &= [\mathbf{0}_{1\times(i+1)}, \mathbf{1}, \mathbf{0}_{1\times(2M+2-i)}], \\ \mathbf{e}_{2,i} &= [\mathbf{0}_{1\times(M+i+3)}, \mathbf{1}, \mathbf{0}_{1\times(M-i)}], \\ \mathbf{E}_{1,i} &= [\mathbf{0}_{2N\times 2(M+iN)}, [\mathbf{I}_N, \mathbf{0}_N]^T, \mathbf{0}_{2N\times[2N(M-i)+N]}], \\ \mathbf{E}_{2,i} &= [\mathbf{0}_{2N\times 2(M+iN)}, [\mathbf{0}_N, \mathbf{I}_N]^T, \mathbf{0}_{2N\times[2N(M-i)+N]}], \\ \mathbf{E}_{3,i} &= [\mathbf{0}_{2N\times[2(M+iN)+N]}, [\mathbf{0}_N, \mathbf{I}_N]^T, \mathbf{0}_{2N\times 2N(M-i)}]. \end{aligned}$$

With some well-known formulas in multivariate statistics and Equation (A15), we list the related expected values as follows, where

$$\mathbf{Q} = \text{blkdiag}(\mathbf{Q}_z, \mathbf{Q}_m). \tag{A22}$$

- Let  $\mathbf{K}_1\{i\} = \mathbf{K}_1(2N(i-1) + 1 : 2Ni, :)$  for  $i = 1, 2, \dots, M$ . Then

$$\begin{aligned} &E[\Delta\mathbf{G}_1^T\mathbf{K}_1\mathbf{B}_1\boldsymbol{\epsilon}_1^{(1)}] \\ &= \sum_{i=1}^M E\left(\Delta\mathbf{G}_{1,i}^T\mathbf{K}_1\{i\}\mathbf{B}_1\boldsymbol{\epsilon}_1^{(1)}\right) \\ &= 2c \sum_{i=1}^M \mathbf{e}_{1,i}^T \text{tr}(\mathbf{E}_{1,i}^T\mathbf{K}_1\{i\}\mathbf{B}_1[\mathbf{D}_z, c\mathbf{I}]\mathbf{Q}) \\ &\quad + c \sum_{i=1}^M \mathbf{e}_{1,i}^T \text{tr}(\mathbf{E}_{3,i}^T\mathbf{K}_1\{i\}\mathbf{B}_1[\mathbf{D}_z, c\mathbf{I}]\mathbf{Q}) \\ &\quad + c \sum_{i=1}^M \mathbf{e}_{2,i}^T \text{tr}(\mathbf{E}_{2,i}^T\mathbf{K}_1\{i\}\mathbf{B}_1[\mathbf{D}_z, c\mathbf{I}]\mathbf{Q}). \end{aligned}$$

- 

$$\begin{aligned} &E[\Delta\mathbf{G}_1\mathbf{H}_1^0\mathbf{B}_1\boldsymbol{\epsilon}_1^{(1)}] \\ &= [E[\Delta\mathbf{G}_{1,1}\mathbf{H}_1^0\mathbf{B}_1\boldsymbol{\epsilon}_1^{(1)}]^T, \dots, E[\Delta\mathbf{G}_{1,M}\mathbf{H}_1^0\mathbf{B}_1\boldsymbol{\epsilon}_1^{(1)}]^T]^T, \end{aligned}$$

where

$$\begin{aligned} & E[\Delta \mathbf{G}_{1,i} \mathbf{H}_1^0 \mathbf{B}_1 \boldsymbol{\epsilon}_1^{(1)}] \\ &= 2c \mathbf{E}_{1,i} \mathbf{Q} (\mathbf{e}_{1,i} \mathbf{H}_1^0 \mathbf{B}_1 [\mathbf{D}_z, c \mathbf{I}])^T \\ &+ c \mathbf{E}_{3,i} \mathbf{Q} (\mathbf{e}_{1,i} \mathbf{H}_1^0 \mathbf{B}_1 [\mathbf{D}_z, c \mathbf{I}])^T \\ &+ c \mathbf{E}_{2,i} \mathbf{Q} (\mathbf{e}_{2,i} \mathbf{H}_1^0 \mathbf{B}_1 [\mathbf{D}_z, c \mathbf{I}])^T, \end{aligned}$$

for  $i = 1, 2, \dots, M$ .

Appendix E.2. Formulas for Computing Bias in the Second Stage

$$\begin{aligned} \Delta \mathbf{G}_2 &= \begin{bmatrix} \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} \\ \mathbf{1}_{M \times 1} \otimes (-2\Delta \mathbf{u}^T) & \mathbf{0}_{M \times 2} \\ \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} \\ \mathbf{1}_{M \times 1} \otimes (-\Delta \dot{\mathbf{u}}^T) & \mathbf{1}_{M \times 1} \otimes (-\Delta \mathbf{u}^T) \end{bmatrix}, \\ \Delta \mathbf{B}_2 &= \begin{bmatrix} \text{diag}([0, 0, 2\Delta \boldsymbol{\alpha}^T]) & \mathbf{0}_{(M+2) \times (M+2)} \\ \text{diag}([0, 0, \Delta \boldsymbol{\beta}^T]) & \text{diag}([0, 0, \Delta \boldsymbol{\alpha}^T]) \end{bmatrix}. \end{aligned}$$

All the expected values required for calculating  $\mu_2$  are listed as follows, where  $\mathbf{S}$  is the mean squared error matrix of  $\Delta \boldsymbol{\phi}_1$ , i.e.,

$$\mathbf{S} = \mathbf{P}_1^{o-1} + \boldsymbol{\mu}_1 \boldsymbol{\mu}_1^T.$$

1.

$$\begin{aligned} E[(\Delta \alpha_i)^2] &\approx \mathbf{S}(i+2, i+2), i = 1, 2, \dots, M, \\ E[\Delta \alpha_i \Delta \beta_i] &\approx \mathbf{S}(i+2, i+M+4), i = 1, 2, \dots, M, \\ E[\Delta \mathbf{u}^T \Delta \mathbf{u}] &\approx \text{tr}(\mathbf{S}(1:2, 1:2)), \\ E[\Delta \mathbf{u}^T \Delta \dot{\mathbf{u}}] &\approx \text{tr}(\mathbf{S}(1:2, M+3:M+4)). \end{aligned}$$

2.

$$\begin{aligned} & \Delta \mathbf{G}_2 \mathbf{H}_2^0 \\ &= \begin{bmatrix} \mathbf{0}_{2 \times 1} \\ \mathbf{1}_{M \times 1} \otimes (-2\Delta \mathbf{u}^T) \mathbf{H}_2^0(1:2, :) \\ \mathbf{0}_{2 \times 1} \\ \mathbf{1}_{M \times 1} \otimes (-\Delta \dot{\mathbf{u}}^T) \mathbf{H}_2^0(1:2, :) + \mathbf{1}_{M \times 1} \otimes (-\Delta \mathbf{u}^T) \mathbf{H}_2^0(3:4, :) \end{bmatrix}, \end{aligned}$$

where

$$\begin{aligned} E[\Delta \mathbf{u}^T \mathbf{H}_2^0(1:2, :) \mathbf{B}_2 \boldsymbol{\epsilon}_2^{(1)}] &= \text{tr}(\mathbf{H}_2^0(1:2, :) \mathbf{B}_2 \mathbf{S}(:, 1:2)), \\ E[\Delta \dot{\mathbf{u}}^T \mathbf{H}_2^0(1:2, :) \mathbf{B}_2 \boldsymbol{\epsilon}_2^{(1)}] &= \text{tr}(\mathbf{H}_2^0(1:2, :) \mathbf{B}_2 \mathbf{S}(:, M+3:M+4)), \\ E[\Delta \mathbf{u}^T \mathbf{H}_2^0(3:4, :) \mathbf{B}_2 \boldsymbol{\epsilon}_2^{(1)}] &= \text{tr}(\mathbf{H}_2^0(3:4, :) \mathbf{B}_2 \mathbf{S}(:, 1:2)). \end{aligned}$$

3.

$$\Delta \mathbf{B}_2 \mathbf{B}_2^{-1} = \begin{bmatrix} \mathbf{0}_{2 \times (2M+4)} \\ 2\Delta\alpha_1 \mathbf{B}_2^{-1}(3, :) \\ 2\Delta\alpha_2 \mathbf{B}_2^{-1}(4, :) \\ \vdots \\ 2\Delta\alpha_M \mathbf{B}_2^{-1}(M+2, :) \\ \mathbf{0}_{2 \times (2M+4)} \\ \Delta\alpha_1 \mathbf{B}_2^{-1}(M+5, :) + \Delta\beta_1 \mathbf{B}_2^{-1}(3, :) \\ \Delta\alpha_2 \mathbf{B}_2^{-1}(M+6, :) + \Delta\beta_2 \mathbf{B}_2^{-1}(4, :) \\ \vdots \\ \Delta\alpha_M \mathbf{B}_2^{-1}(2M+4, :) + \Delta\beta_M \mathbf{B}_2^{-1}(M+2, :) \end{bmatrix},$$

where

$$\begin{aligned} E[\Delta\alpha_i \mathbf{B}_2^{-1}(i+2, :) \mathbf{U} \mathbf{B}_2 \epsilon_2^{(1)}] &= \mathbf{S}(i+2, :) \mathbf{B}_2^T \mathbf{U}^T \mathbf{B}_2^{-1}(i+2, :)^T, \\ E[\Delta\alpha_i \mathbf{B}_2^{-1}(i+M+4, :) \mathbf{U} \mathbf{B}_2 \epsilon_2^{(1)}] &= \mathbf{S}(i+2, :) \mathbf{B}_2^T \mathbf{U}^T \mathbf{B}_2^{-1}(i+M+4, :)^T, \\ E[\Delta\beta_i \mathbf{B}_2^{-1}(i+2, :) \mathbf{U} \mathbf{B}_2 \epsilon_2^{(1)}] &= \mathbf{S}(i+M+4, :) \mathbf{B}_2^T \mathbf{U}^T \mathbf{B}_2^{-1}(i+2, :)^T, \end{aligned}$$

for  $i = 1, 2, \dots, M$ .

4.

$$\Delta \mathbf{G}_2^T \mathbf{K}_2 = \begin{bmatrix} -2\Delta \mathbf{u} \sum_{i=1}^M \mathbf{K}_2(i+2, :) - \Delta \mathbf{u} \sum_{i=1}^M \mathbf{K}_2(i+M+4, :) \\ -\Delta \mathbf{u} \sum_{i=1}^M \mathbf{K}_2(i+M+4, :) \end{bmatrix},$$

where

$$\begin{aligned} E[\Delta \mathbf{u} \sum_{i=1}^M \mathbf{K}_2(i+2, :) \mathbf{B}_2 \epsilon_2^{(1)}] &= \mathbf{S}(1:2, :) \mathbf{B}_2^T \sum_{i=1}^M \mathbf{K}_2(i+2, :)^T, \\ E[(\Delta \mathbf{u} \sum_{i=1}^M \mathbf{K}_2(i+M+4, :) \mathbf{B}_2 \epsilon_2^{(1)})] &= \mathbf{S}(M+3:M+4, :) \mathbf{B}_2^T \sum_{i=1}^M \mathbf{K}_2(i+M+4, :)^T, \\ E[\Delta \mathbf{u} \sum_{i=1}^M \mathbf{K}_2(i+M+4, :) \mathbf{B}_2 \epsilon_2^{(1)}] &= \mathbf{S}(1:2, :) \mathbf{B}_2^T \sum_{i=1}^M \mathbf{K}_2(i+M+4, :)^T, \end{aligned}$$

for  $i = 1, 2, \dots, M$ .

5.

$$\Delta \mathbf{B}_2^T \mathbf{K}_2 = \begin{bmatrix} \mathbf{0}_{2 \times (2M+4)} \\ 2\Delta\alpha_1 \mathbf{K}_2(3, :) + \Delta\beta_1 \mathbf{K}_2(M+5, :) \\ 2\Delta\alpha_2 \mathbf{K}_2(4, :) + \Delta\beta_2 \mathbf{K}_2(M+6, :) \\ \vdots \\ 2\Delta\alpha_M \mathbf{K}_2(M+2, :) + \Delta\beta_M \mathbf{K}_2(2M+4, :) \\ \mathbf{0}_{2 \times (2M+4)} \\ \Delta\alpha_1 \mathbf{K}_2(M+5, :) \\ \Delta\alpha_2 \mathbf{K}_2(M+6, :) \\ \vdots \\ \Delta\alpha_M \mathbf{K}_2(2M+4, :) \end{bmatrix},$$

where

$$\begin{aligned} E[\Delta\alpha_i \mathbf{K}_2(i+2, :) \mathbf{B}_2 \epsilon_2^{(1)}] &= \mathbf{S}(i+2, :) \mathbf{B}_2^T \mathbf{K}_2(i+2, :)^T, \\ E[\Delta\alpha_i \mathbf{K}_2(i+M+4, :) \mathbf{B}_2 \epsilon_2^{(1)}] &= \mathbf{S}(i+2, :) \mathbf{B}_2^T \mathbf{K}_2(i+M+4, :)^T, \\ E[\Delta\beta_i \mathbf{K}_2(i+M+4, :) \mathbf{B}_2 \epsilon_2^{(1)}] &= \mathbf{S}(i+M+4, :) \mathbf{B}_2^T \mathbf{K}_2(i+M+4, :)^T, \end{aligned}$$

for  $i = 1, 2, \dots, M$ .

**Appendix F. Some Formula for Proving Proposition 2**

By Equation (73),

$$\mathbf{G}_3 = [\mathbf{G}_{3,1}^T, \mathbf{G}_{3,2}^T, \dots, \mathbf{G}_{3,M}^T]^T, \tag{A23}$$

where

$$\mathbf{G}_{3,i} = \mathbf{B}_{1,i}^{-1} \mathbf{G}_{1,i}^o \cdot \mathbf{B}_2^{-1} \mathbf{G}_2^o. \tag{A24}$$

By performing algebraic manipulations, we have

$$\mathbf{B}_{1,i}^{-1} \mathbf{G}_{1,i}^o = \begin{bmatrix} \mathbf{Y}_i & \mathbf{0}_{N \times (M+2)} \\ \mathbf{Z}_i & \mathbf{Y}_i \end{bmatrix}, \quad \mathbf{B}_2^{-1} \mathbf{G}_2^o = \begin{bmatrix} -\mathbf{I}_2 & \mathbf{0}_{2 \times 2} \\ \mathbf{Y} & \mathbf{0}_{M \times 2} \\ \mathbf{0}_{2 \times 2} & -\mathbf{I}_2 \\ \mathbf{Z} & \mathbf{Y} \end{bmatrix}, \tag{A25}$$

where

$$\begin{aligned} \mathbf{Y}_i &= \begin{bmatrix} \frac{(\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_1)^T}{\|\mathbf{u} - \bar{\mathbf{s}}_1\|} & \mathbf{0}_{1 \times (i-1)} & \frac{c \tau_{i,1} + \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_1\|}{\|\mathbf{u} - \bar{\mathbf{s}}_1\|} & \mathbf{0}_{1 \times (M-i)} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{(\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_N)^T}{\|\mathbf{u} - \bar{\mathbf{s}}_N\|} & \mathbf{0}_{1 \times (i-1)} & \frac{c \tau_{i,1} + \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_N\|}{\|\mathbf{u} - \bar{\mathbf{s}}_N\|} & \mathbf{0}_{1 \times (M-i)} \end{bmatrix}, \\ \mathbf{Z}_i &= \begin{bmatrix} -\frac{(\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_1)^T \rho_{\mathbf{u}, \bar{\mathbf{s}}_1}^T \dot{\mathbf{u}}}{\|\mathbf{u} - \bar{\mathbf{s}}_1\|^2} & \mathbf{0}_{1 \times (i-1)} & -\frac{(c \tau_{i,1} + \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_1\|) \rho_{\mathbf{u}, \bar{\mathbf{s}}_1}^T \dot{\mathbf{u}}}{\|\mathbf{u} - \bar{\mathbf{s}}_1\|^2} + \frac{c f_{i,1}}{\|\mathbf{u} - \bar{\mathbf{s}}_1\|} & \mathbf{0}_{1 \times (M-i)} \\ \vdots & \vdots & \vdots & \vdots \\ -\frac{(\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_N)^T \rho_{\mathbf{u}, \bar{\mathbf{s}}_N}^T \dot{\mathbf{u}}}{\|\mathbf{u} - \bar{\mathbf{s}}_N\|^2} & \mathbf{0}_{1 \times (i-1)} & -\frac{(c \tau_{i,1} + \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_N\|) \rho_{\mathbf{u}, \bar{\mathbf{s}}_N}^T \dot{\mathbf{u}}}{\|\mathbf{u} - \bar{\mathbf{s}}_N\|^2} + \frac{c f_{i,1}}{\|\mathbf{u} - \bar{\mathbf{s}}_N\|} & \mathbf{0}_{1 \times (M-i)} \end{bmatrix}, \end{aligned}$$

$$\mathbf{Y} = \begin{bmatrix} -\rho_{\mathbf{u},\bar{\mathbf{t}}_1}^T \\ -\rho_{\mathbf{u},\bar{\mathbf{t}}_2}^T \\ \vdots \\ -\rho_{\mathbf{u},\bar{\mathbf{t}}_M}^T \end{bmatrix}, \quad \mathbf{Z} = \begin{bmatrix} -\dot{\mathbf{u}}^T \mathbf{A}_{\mathbf{u},\bar{\mathbf{t}}_1}^T \\ -\dot{\mathbf{u}}^T \mathbf{A}_{\mathbf{u},\bar{\mathbf{t}}_2}^T \\ \vdots \\ -\dot{\mathbf{u}}^T \mathbf{A}_{\mathbf{u},\bar{\mathbf{t}}_M}^T \end{bmatrix}.$$

Expanding Equation (A24) with Equation (A25), the details of  $\mathbf{G}_{3,i}(i = 1, 2, \dots, M)$  is as follows, where  $j = 1, 2, \dots, N$ .

$$\mathbf{G}_{3,i}(j, 1 : 2) = \frac{-1}{\|\mathbf{u} - \bar{\mathbf{s}}_j\|} \left[ (c \tau_{i,j} + \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_j\|) \rho_{\mathbf{u},\bar{\mathbf{t}}_i} + (\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_j) \right]^T, \tag{A26}$$

$$\mathbf{G}_{3,i}(N + j, 1 : 2) = \frac{(\rho_{\mathbf{u},\bar{\mathbf{s}}_j}^T \dot{\mathbf{u}}) (\rho_{\mathbf{u},\bar{\mathbf{t}}_i} + \rho_{\mathbf{u},\bar{\mathbf{s}}_j})}{\|\mathbf{u} - \bar{\mathbf{s}}_j\|} - \frac{c f_{i,j} \rho_{\mathbf{u},\bar{\mathbf{t}}_i}}{\|\mathbf{u} - \bar{\mathbf{s}}_j\|} - \frac{c \tau_{i,j} + \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_j\|}{\|\mathbf{u} - \bar{\mathbf{s}}_j\|} \mathbf{A}_{\mathbf{u},\bar{\mathbf{t}}_i} \dot{\mathbf{u}}^T, \tag{A27}$$

$$\mathbf{G}_{3,i}(j, 3 : 4) = \mathbf{0}_{1 \times 2}, \tag{A28}$$

$$\mathbf{G}_{3,i}(N + j, 3 : 4) = \mathbf{G}_{3,i}(j, 1 : 2). \tag{A29}$$

When the small error conditions in Conditions 1 through 6 are satisfied, by Equation (19) and Equation (21), we have

$$c \tau_{i,j} + \|\bar{\mathbf{t}}_i - \bar{\mathbf{s}}_j\| \approx \|\mathbf{u} - \bar{\mathbf{t}}_i\| + \|\mathbf{u} - \bar{\mathbf{s}}_j\|, \tag{A30}$$

$$c f_{i,j} \approx \rho_{\mathbf{u},\bar{\mathbf{t}}_i}^T \dot{\mathbf{u}} + \rho_{\mathbf{u},\bar{\mathbf{s}}_j}^T \dot{\mathbf{u}}. \tag{A31}$$

We also note that

$$\mathbf{A}_{\mathbf{u},\bar{\mathbf{t}}_i} \dot{\mathbf{u}} = \frac{\beta_i}{\alpha_i^2} (\bar{\mathbf{t}}_i - \mathbf{u}) + \frac{1}{\alpha_i} \dot{\mathbf{u}}, \tag{A32}$$

$$\mathbf{A}_{\mathbf{x},\mathbf{y}} = \frac{1}{\|\mathbf{x} - \mathbf{y}\|} \left( \mathbf{I} - \rho_{\mathbf{x},\mathbf{y}} \rho_{\mathbf{x},\mathbf{y}}^T \right). \tag{A33}$$

Combining the above formulas,

$$\mathbf{G}_{3,i}(N + j, 3 : 4) = \mathbf{G}_{3,i}(j, 1 : 2) \approx -(\rho_{\mathbf{u},\bar{\mathbf{t}}_i} + \rho_{\mathbf{u},\bar{\mathbf{s}}_j}), \tag{A34}$$

$$\mathbf{G}_{3,i}(N + j, 1 : 2) \approx -(\mathbf{A}_{\mathbf{u},\bar{\mathbf{t}}_i} + \mathbf{A}_{\mathbf{u},\bar{\mathbf{s}}_j}) \dot{\mathbf{u}}. \tag{A35}$$

## References

1. Zhang, Y.; Ho, K.C. Multistatic localization in the absence of transmitter position. *IEEE Trans. Signal Process.* **2019**, *67*, 4745–4760. [CrossRef]
2. He, C.; Wang, Y.; Yu, W.; Song, L. Underwater target localization and synchronization for a distributed SIMO sonar with an isogradient SSP and uncertainties in receiver locations. *Sensors* **2019**, *19*, 1976. [CrossRef] [PubMed]
3. Liang, J.; Chen, Y.; So, H.; Jing, Y. Circular/hyperbolic/elliptic localization via Euclidean norm elimination. *Signal Process.* **2018**, *148*, 102–113. [CrossRef]
4. Peters, D.J. A Bayesian method for localization by multistatic active sonar. *IEEE J. Ocean. Eng.* **2017**, *42*, 135–142. [CrossRef]
5. Yang, L.; Yang, L.; Ho, K.C. Moving target localization in multistatic sonar by differential delays and Doppler shifts. *IEEE Signal Process. Lett.* **2016**, *23*, 1160–1164. [CrossRef]
6. Rui, L.; Ho, K.C. Efficient closed-form estimators for multistatic sonar localization. *IEEE Trans. Aerosp. Electron. Syst.* **2015**, *51*, 600–614. [CrossRef]

7. Rui, L.; Ho, K.C. Elliptic localization: Performance study and optimum receiver placement. *IEEE Trans. Signal Process.* **2014**, *62*, 4673–4688. [[CrossRef](#)]
8. Ehlers, F.; Ricci, G.; Orlando, D. Batch tracking algorithm for multistatic sonars. *IET Radar Sonar Navig.* **2012**, *6*, 746–752. [[CrossRef](#)]
9. Daun, M.; Ehlers, F. Tracking algorithms for multistatic sonar systems. *EURASIP J. Adv. Signal Process.* **2010**, *2010*, 461538. [[CrossRef](#)]
10. Simakov, S. Localization in airborne multistatic sonars. *IEEE J. Ocean. Eng.* **2008**, *33*, 278–288. [[CrossRef](#)]
11. Coraluppi, S. Multistatic sonar localization. *IEEE J. Ocean. Eng.* **2006**, *31*, 964–974. [[CrossRef](#)]
12. Coraluppi, S.; Carthel, C. Distributed tracking in multistatic sonar. *IEEE Trans. Aerosp. Electron. Syst.* **2005**, *41*, 1138–1147. [[CrossRef](#)]
13. Sandys-Wunsch, M.; Hazen, M.G. Multistatic localization error due to receiver positioning errors. *IEEE J. Ocean. Eng.* **2002**, *27*, 328–334. [[CrossRef](#)]
14. Shin, H.; Chung, W. Target localization using double-sided bistatic range measurements in distributed MIMO radar systems. *Sensors* **2019**, *19*, 2524. [[CrossRef](#)]
15. Amiri, R.; Behnia, F.; Noroozi, A. Efficient algebraic solution for elliptic target localisation and antenna position refinement in multiple-input–multiple-output radars. *IET Radar Sonar Navig.* **2019**, *13*, 2046–2054. [[CrossRef](#)]
16. Amiri, R.; Behnia, F.; Noroozi, A. Efficient joint moving target and antenna localization in distributed MIMO radars. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 4425–4435. [[CrossRef](#)]
17. Amiri, R.; Behnia, F.; Zamani, H. Asymptotically efficient target localization from bistatic range measurements in distributed MIMO radars. *IEEE Signal Process. Lett.* **2017**, *24*, 299–303. [[CrossRef](#)]
18. Einemo, M.; So, H.C. Weighted least squares algorithm for target localization in distributed MIMO radar. *Signal Process.* **2015**, *115*, 144–150. [[CrossRef](#)]
19. Dianat, M.; Taban, M.R.; Dianat, J.; Sedighi, V. Target localization using least squares estimation for MIMO radars with widely separated antennas. *IEEE Trans. Aerosp. Electron. Syst.* **2013**, *49*, 2730–2741. [[CrossRef](#)]
20. Godrich, H.; Haimovich, A.M.; Blum, R.S. Target localization accuracy gain in MIMO radar-based systems. *IEEE Trans. Inf. Theory* **2010**, *56*, 2783–2803. [[CrossRef](#)]
21. Zhao, Y.; Hu, D.; Zhao, Y.; Liu, Z.; Zhao, C. Refining inaccurate transmitter and receiver positions using calibration targets for target localization in multi-static passive radar. *Sensors* **2019**, *19*, 3365. [[CrossRef](#)] [[PubMed](#)]
22. Wang, J.; Qin, Z.; Wei, S.; Sun, Z.; Xiang, H. Effects of nuisance variables selection on target localisation accuracy in multistatic passive radar. *Electron. Lett.* **2018**, *54*, 1139–1141. [[CrossRef](#)]
23. Chalise, B.K.; Zhang, Y.D.; Amin, M.G.; Himed, B. Target localization in a multi-static passive radar system through convex optimization. *Signal Process.* **2014**, *102*, 207–215. [[CrossRef](#)]
24. Gorji, A.A.; Tharmarasa, R.; Kirubarajan, T. Widely separated MIMO versus multistatic radars for target localization and tracking. *IEEE Trans. Aerosp. Electron. Syst.* **2013**, *49*, 2179–2194. taes.2013.6621809. [[CrossRef](#)]
25. Malanowski, M.; Kulpa, K. Two methods for target localization in multistatic passive radar. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 572–580. [[CrossRef](#)]
26. Yin, Z.; Jiang, X.; Yang, Z.; Zhao, N.; Chen, Y. WUB-IP: A high-precision UWB positioning scheme for indoor multiuser applications. *IEEE Syst. J.* **2019**, *13*, 279–288. [[CrossRef](#)]
27. Zhou, Y.; Law, C.L.; Guan, Y.L.; Chin, F. Indoor elliptical localization based on asynchronous UWB range measurement. *IEEE Trans. Instrum. Meas.* **2011**, *60*, 248–257. [[CrossRef](#)]
28. Smith, J.O.; Abel, J.S. Closed-form least-squares source location estimation from range-difference measurements. *IEEE Trans. Acoust. Speech Signal Process.* **1987**, *35*, 1661–1669. [[CrossRef](#)]
29. Schau, H.; Robinson, A. Passive source localization employing intersecting spherical surfaces from time-of-arrival differences. *IEEE Trans. Acoust. Speech Signal Process.* **1987**, *35*, 1223–1225. [[CrossRef](#)]
30. Amiri, R.; Behnia, F.; Sadr, M.A.M. Exact solution for elliptic localization in distributed MIMO radar systems. *IEEE Trans. Veh. Technol.* **2018**, *67*, 1075–1086. [[CrossRef](#)]
31. Chan, Y.T.; Ho, K.C. A simple and efficient estimator for hyperbolic location. *IEEE Trans. Signal Process.* **1994**, *42*, 1905–1915. [[CrossRef](#)]
32. Zheng, B.; Yang, Z. Perturbation analysis for mixed least squares–total least squares problems. *Numer. Linear Algebra Appl.* **2019**, *26*, e2239. [[CrossRef](#)]

33. Buranay, S.C.; Iyikal, O.C. A predictor-corrector iterative method for solving linear least squares problems and perturbation error analysis. *J. Inequal. Appl.* **2019**, *2019*, 203. [[CrossRef](#)]
34. Xie, P.; Xiang, H.; Wei, Y. A contribution to perturbation analysis for total least squares problems. *Numer. Algorithms* **2017**, *75*, 381–395. [[CrossRef](#)]
35. Harville, D.A. *Linear Models and the Relevant Distributions and Matrix Algebra*; CRC Press: Boca Raton, FL, USA, 2018.
36. Bar, S.; Tabrikian, J. The risk-unbiased Cramér–Rao bound for non-Bayesian multivariate parameter estimation. *IEEE Trans. Signal Process.* **2018**, *66*, 4920–4934. [[CrossRef](#)]
37. Bergel, I.; Noam, Y. Lower bound on the localization error in infinite networks with random sensor locations. *IEEE Trans. Signal Process.* **2018**, *66*, 1228–1241. [[CrossRef](#)]
38. Messer, H. The hybrid Cramér–Rao lower bound—From practice to theory. In Proceedings of the 4th IEEE Sensor Array and Multichannel Signal Processing Workshop, Waltham, MA, USA, 12–14 July 2006; pp. 304–307. [[CrossRef](#)]
39. Noam, Y.; Messer, H. The hybrid Cramér–Rao bound and the generalized Gaussian linear estimation problem. In Proceedings of the 5th IEEE Sensor Array and Multichannel Signal Processing Workshop, Darmstadt, Germany, 21–23 July 2008; pp. 395–399. [[CrossRef](#)]
40. Van Trees, H.L.; Bell, K.L.; Tian, Z. *Detection, Estimation, and Modulation Theory Part I: Detection, Estimation, and Filtering Theory*; John Wiley & Sons: Hoboken, NJ, USA, 2013.
41. Rockah, Y.; Schultheiss, P. Array shape calibration using sources in unknown locations—Part I: far-field sources. *IEEE Trans. Acoust. Speech Signal Process.* **1987**, *35*, 286–299. [[CrossRef](#)]
42. Magnus, J.R.; Neudecker, H. *Matrix Differential Calculus with Applications in Statistics and Econometrics*; John Wiley & Sons: Hoboken, NJ, USA, 2007.
43. Chui, C.K.; Chen, G. *Kalman Filtering with Real-Time Applications*; Springer International Publishing: Berlin/Heidelberg, Germany, 2017.
44. Li, X.; Wang, S.; Cai, Y. Tutorial: Complexity analysis of Singular Value Decomposition and its variants. *arXiv* **2019**, arXiv:1906.12085.
45. Foy, W.H. Position-location solutions by Taylor-series estimation. *IEEE Trans. Aerosp. Electron. Syst.* **1976**, *12*, 187–194. [[CrossRef](#)]
46. Horn, R.A.; Johnson, C.R. *Matrix Analysis*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2013.



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).