Bias Estimation and Sensor Registration for Target Tracking

BIAS ESTIMATION AND SENSOR REGISTRATION FOR TARGET TRACKING

BY

EHSAN TAGHAVI, M.Sc. (Electrical–Digital Communication Engineering),

Chalmers University of Technology, Gothenburg, Sweden

A THESIS

SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2016) McMaster University

(School of Computational Science and Engineering) Hamilton, Ontario, Canada

TITLE:	Bias Estimation and Sensor Registration for Target Tracking
AUTHOR:	Ehsan Taghavi M.Sc. (Electrical–Digital Communication Engineering), Chalmers University of Technology, Gothenburg, Sweden
SUPERVISOR:	Prof. T. Kirubarajan
NUMBER OF PAGES:	xvii, 150

To my family

Abstract

The main idea of this thesis is to define and formulate the role of bias estimation in multitarget–multisensor scenarios as a general framework for various measurement types. After a brief introduction of the work that has been done in this thesis, three main contributions are explained in detail, which exercise the novel ideas.

Starting with radar measurements, a new bias estimation method that can estimate offset and scaling biases in large network of radars is proposed. Further, Cramér–Rao Lower Bound is calculated for the bias estimation algorithm to show the theoretical accuracy that can be achieved by the proposed method. In practice, communication loss is also part of the distributed systems, which sometimes can not be avoided. A novel technique is also developed to accompany the proposed bias estimation method in this thesis to compensate for communication loss at different rates by the use of tracklets.

Next, bearing–only measurements are considered. Biases in this type of measurement can be difficult to tackle because the measurement noise and systematic biases are normally larger than in radar measurements. In addition, target observability is sensitive to sensor–target alignment and can vary over time. In a multitarget–multisensor bearing–only scenario with biases, a new model is proposed for the biases

that is decoupled form the bearing–only measurements. These decoupled bias measurements then are used in a maximum likelihood batch estimator to estimate the biases and then be used for compensation.

The thesis is then expanded by applying bias estimation algorithms into video sensor measurements. Video sensor measurements are increasingly implemented in distributed systems because of their economical benefits. However, geo–location and geo–registration of the targets must be considered in such systems. In last part of the thesis, a new approach proposed for modeling and estimation of biases in a two video sensor platform which can be used as a standalone algorithm. The proposed algorithm can estimate the gimbal elevation and azimuth biases effectively.

It is worth noting that in all parts of the thesis, simulation results of various scenarios with different parameter settings are presented to support the ideas, the accuracy, mathematical modelings and proposed algorithms. These results show that the bias estimation methods that have been conducted in this thesis are viable and can handle larger biases and measurement errors than previously proposed methods. Finally, the thesis conclude with suggestions for future research in three main directions.

Acknowledgements

I would like to take the opportunity to thank some very special people; without their help this thesis would never have been written. First and foremost, I would like to thank my supervisor Professor T. Kirubarajan, for accepting me as a Ph.D. student, for believing in me, and for all the support that he gave me during these four years.

Next, I would like to thank Dr. R. Tharmarasa, for much needed guidance in both theoretical and practical knowledge of field of study. I would also like to thank the chair of McMaster School of Computational Science and Engineering, Professor Bartosz Protas, for all his kind support. I would as well wish to express my appreciation to the School of Graduate Studies for their generous awards, which all provided me with sources of support and motivation throughout my Ph.D. studies at McMaster.

I would like to show my gratitude to my closest friends, Peyman, Mehdi, Ali, Maryam, Azadeh, Mina, Atta, Mahsa, Foad and Sina whose presence has always been priceless. Being with such a gifted group of friends has tremendously helped with passing the somehow gloomy days, which are an inevitable part and parcel of an immigrant life.

Last but by no means least, I would like to thank my father and my mother for their unconditional love and consistent support through all the years, without which all my achievements have simply been impossible. And here goes my special thanks

to my brother Hamidreza. He has been my source of motivation in every academic step I took from the very first day of school up until this moment that I am finishing my Ph.D. studies. I could not wish for a better sibling and I hope I did well for all his kindness.

Abbreviations

AMR	Associated Measurement Report
CRLB	Cramér–Rao Lower Bound
CWPA	Continuous Wiener Process Acceleration
DCWNA	Discretized Continuous White Noise Acceleration
FBEA	Fused Bias Estimation Algorithm
ESM	Electronic Support Measures
GA	Genetic Algorithm
GPS	Global Positioning System
IEEE	Institute of Electrical and Electronics Engineers
IRST	Infra–red Search and Track
IMM	Interactive Multiple Model
IMU	Inertial Measurement Unit
ML	Maximum Likelihood
MMSE	Minimum Mean Square Error
MSE	Mean Squared Error
NCA	Nearly Constant Acceleration
NCV	Nearly Constant Velocity
OMBE	Optimal MMSE Bias Estimation

RLS	Recursive Least Square
RLSBE	Recursive Least Square Bias Estimation
RMSE	Root-mean Squared Error
SFA	Sequential Fusion Algorithm
UAV	Unmanned Aerial Vehicle

Contents

List of Figures

Declaration of Academic Achievement

This research presents analytical and computational work carried out solely by Ehsan Taghavi, herein referred to as "the author", with advice and guidance provided by the academic supervisor Prof. T. Kirubarajan. Information that is presented from outside sources which has been used towards analysis or discussion, has been cited when appropriate, all other materials are the sole work of the author.

Chapter 1

Introduction

1.1 Bias modeling and estimation: A brief review

Any system that processes sensory data deals with errors generated by the sensor. There are different types of errors, such as residual error, offset and scaling biases, clock bias and uncertainties in the sensor positions to name a few. This thesis, focuses on the modeling and estimation of offset and scaling biases in different distributed systems.

Because of the huge impact of biases in the final output of the measurements, bias modeling, estimation and compensation are essential steps in distributed tracking systems. The main goal of this thesis is to model offset and scale biases in radar [2], bearing–only [1] and video [4] measurements for a general multisensor–multitarget scenario [3]. Bias modeling is a fundamental step in defining a generalized method for estimating biases of different sensory data and further compensate for the biases.

Previously proposed methods for radar bias estimation did not address the multisensor problem and mostly were fixed to work with two sensors only [8, 10, 9]. In

addition, traditional methods involve measurement parameters rather than the tracks only.

In Chapter 2, a unified algorithm that not only can estimate the offset and scaling biases of multiple radar sensors and targets and handle low communication rates by using tracklets [6, 5], but also only requires tracks that have been sent back to the fusion center is proposed. After reconstructing the biased measurements from tracks, the main idea behind the method is to first try decoupling the biases from radar measurements in the same way as it is done in [9], but with no limitation on the number of sensors, targets or communication rates.

This is followed by methods to fuse all the information from decoupled bias measurements in order to efficiently estimate the biases. This is the core block of this thesis, to exercise decoupling the bias errors form the measurements of any measurement type and then treat them as new measurements, here pseudo–measurements, and estimate the biases separately. For radar measurements, it has been shown, through mathematical models, that both offset and scaling biases can be estimated with reasonable accuracy, comparable with the Cramér–Rao Lower Bounds (CRLB), and then correct the tracks, accordingly.

In order to examine the idea of generalizing the bias modeling and estimation via decoupling in Chapter 2, distributed systems with bearing–only measurements are considered as the next system in Chapter 3. Bearing–only measurements have many interesting characteristics such as larger residual noise that can scale up when triangulating [7] to measure the position of the targets, severe problem of observability according to their relative alignments and larger offset biases. These issues are considered in Chapter 3 for a multisensor–multitarget case with access to associated

measurement reports, local bearing–only tracks or bearing–only measurements.

The start point to understand how the biases alter the bearing–only measurements is to understand how a position estimate can be configured when different number of bearing–only sensors are available in a surveillance region. In Chapter 3, three and four sensor cases have been considered to show how the proposed method can work when there are odd and even number of sensors. After triangulation, it is tried to model the effect of biases and through reasonable approximations, decouple them from the position measurements. Next, a likelihood function of the bias measurements is formed to use in the bias estimation step. It has been shown both mathematically and through simulations in Chapter 3, that the proposed method can estimate the biases, correct the measurements and gives significantly better tracking results.

Video sensors are now more commonly used in different distributed systems. This is the reason behind including the geo–registration modeling of video sensors as the last bit of this thesis. Modeling and estimating the biases for a video sensor helps defining a general approach to bias modeling and estimation by giving not only the results for different sensors, but the methodology behind modeling of the biases by exercising through different sensor models.

Video sensors collect target information in a totally different measurement space from radar and bearing–only. Pixel representation of the targets, however informative, cannot be of direct use for target tracking. It is the geo–location of the targets that is valuable to distributed systems. Calculating the geo–location of the targets based on their pixel positions come at the cost of including various types of residual and systematic errors. Residual errors can be filtered out by convenient methods but the systematic errors must be taken care of with different methodology.

In Chapter 4, based on the video measurement modeling of Unmanned Air Vehicles (UAVs) in [4], a model is suggested for decoupling the measurements from biases. The mathematical procedure to formulate such a model is given in detail in Chapter 4. The main idea behind bias modeling is to consider the two most affected parameters in the video measurements by biases, namely, gimbal azimuth and gimbal elevation biases, and try to separate their bias effect from other parameters. Through reasonable approximation, the effect of biases can be decoupled form video geo–locations, hence, a pseudo–measurement can be defined to be used for bias estimation. As final step, it has been shown through simulation results with comparisons to lower bounds that the proposed method can effectively estimate the biases and degrade the effect of biases in the corrected geo–locations.

The idea of bias estimation have been considered in the literature from different perspectives [3, 11]. In this thesis a more unified methodology for bias estimation and sensor registration is presented. In the three articles presented here, the author formulates the biases in different distributed systems based on the core concept of decoupling the biases from sensor measurements in multisensor–multitarget scenarios. This is followed by the correction of the tracks and/or measurements based on the application of the measurements. Finally, for all the methods and applications proposed in this thesis, simulation results have been conducted to support the mathematical models defined in the articles.

1.2 Theme and Objectives of Dissertation

In compliance with the terms and regulations of McMaster University, this dissertation has been assembled into a *sandwich thesis* format comprised of three journal

articles. These articles represent the independent work of the author of this dissertation, Ehsan Taghavi, henceforth referred to as "the author".

The articles in the dissertation follow a cohesive theme with a flow aimed at defining and expanding upon the current knowledge of multisensor–multitarget bias estimation and its practical applications. The general theme is based on the following:

i) To provide an overview of multisensor–multitarget bias estimation with radar and bearing–only measurements (Papers I and II).

ii) To introduce basic concepts and a rationale definition of radar and bearing–only biases, which matches practical needs (Papers I and II).

iii) To mathematically formulate the algorithmic processes of multisensor–multitarget bias estimation with desirable practical properties (Papers I and II).

iv) To formulate the biases in target geo–locations reported from processed image frames (Paper III).

v) To efficiently implement the proposed method for geo–registration and comparisons with benchmarks (Paper III).

1.3 Summary of Enclosed Articles

The papers enclosed in this thesis are listed as follows:

1.3.1 Paper I (Chapter II)

Ehsan Taghavi, R. Tharmarasa and T. Kirubarajan, Yaakov Bar-Shalom, Mike McDonald

A Practical Bias Estimation Algorithm for Multisensor–Multitarget Tracking, *IEEE Transactions on Aerospace and Electronics Systems*, To be appear in vol. 51, no. 4, October 2015.

Preface: This paper expands on the definition of multisensor–multitarget bias estimation by the use of local tracks, solely. In previously proposed algorithms, the estimators rely on local measurements in centralized systems or tracks in distributed systems, along with additional information like covariances, filter gains or targets of opportunity at every sampling time. From practical point of view, it is only the local track that is reported to the fusion center and not at every sampling time. These issues are considered in this paper and a practical algorithm proposed with sufficient mathematical background to support the efficiency of the algorithm that can estimate the biases in a computationally efficient manner for a multisensor–multitarget scenario.

1.3.2 Paper II (Chapter III)

Ehsan Taghavi, R. Tharmarasa and T. Kirubarajan, Mike McDonald

Multisensor–Multitarget Bearing–Only Sensor Registration, *IEEE Transactions on Aerospace and Electronics Systems*, To be appear in vol. 52, no. 4, August 2016.

Preface: This paper expands on limiting the measurements to bearing–only measurement and formulates a pseudo–measurement that can observe the biases in the measurements. If multisensor scenarios considered, the observability issue can be handled and biases can be estimated with the modeled proposed in this paper. The efficiency of the the algorithm tested over many simulation scenarios prove the efficiency and show that the algorithm can perform well in different practical situations.

Moreover, CRLB results show that the gain from correcting the biases in the original measurements is meaningful and can be considered as effective.

1.3.3 Paper III (Chapter IV)

Ehsan Taghavi, R. Tharmarasa and T. Kirubarajan, Mike McDonald
Geo–registration and Geo–location Using Two Airborne Video Sensors, To be Submitted to *IEEE Transactions on Aerospace and Electronics Systems*, April 2016.
Preface: This paper expands on the idea of bringing the bias estimation techniques to image geo–location and geo–registration. This starts with modeling the geo–location process as an algorithm with all the bias parameters included. Modeling the biases follows the introduction of a method for geo–registration, which includes, first, estimating the biases and then correcting the geo–location of the targets based on the biases. In order to model the biases, there must be at least two cameras available to record the field of interests from different angles. If this data is available the biases can be estimated with the proposed method. Finally, comparisons to lower bounds, which are presented in the simulation results, show that the estimated biases can effectively reduce the error when used for bias compensation.

Bibliography

[1] V. J. Aidala. "Kalman filter behavior in bearings–only tracking applications". *IEEE Transactions on Aerospace and Electronic Systems*, vol. 1, pp. 29–39, 1979.

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software.* Wiley, NY, 2001.

[3] Y. Bar-Shalom, P. K. Willett, and X. Tian. *Tracking and Data Fusion: A Handbook of Algorithms*. YBS Publishing, Storrs, CT, 2011.

[4] D. B. Barber, J. D. Redding, T. W. McLain, R. W. Beard, and C. N. Taylor. "Vision-based target geo–location using a fixed–wing miniature air vehicle". *Journal of Intelligent and Robotic Systems*, vol. 47, no. 4, pp. 361–382, 2006.

[5] O. E. Drummond. "Tracklets and a hybrid fusion with process noise". *International Society for Optics and Photonics*, pp. 512–524, 1997.

[6] O. E. Drummond. "A hybrid sensor fusion algorithm architecture and tracklets". *Proc. SPIE, Signal and Data Processing of Small Targets*, vol. 3163, 1997.

[7] G. Hendeby, R. Karlsson, F. Gustafsson, and N. Gordon. "Recursive triangulation using bearings–only sensors". *IEE Seminar on Target Tracking: Algorithms and Applications*, pp. 3–10, Birmingham, England, March 2006.

[8] D. Huang, H. Leung, and E. Bosse. "A pseudo-measurement approach to simultaneous registration and track fusion". *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 2315–2331, 2012.

[9] X. Lin, Y. Bar-Shalom, and T. Kirubarajan. "Multisensor-multitarget bias estimation of asynchronous sensors". *Proc. SPIE*, vol. 5429, pp. 105–116, 2004.

[10] X. Lin, Y. Bar-Shalom, and T. Kirubarajan. "Exact multisensor dynamic bias estimation with local tracks". *IEEE Transactions on Aerospace and Electronic Systems*, vol. 1, no. 40, pp. 576–590, 2004.

[11] N. Okello and S. Challa. "Joint sensor registration and track-to-track fusion for

distributed trackers". *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 3, pp. 808–823, 2004.

The following chapter is a reproduction of an IEEE copyrighted, published paper:

Ehsan Taghavi, R. Tharmarasa and T. Kirubarajan, Yaakov Bar-Shalom, Mike McDonald
IEEE Transactions on Aerospace and Electronics Systems, To be appear in vol. 51, no. 4, October 2015.

Chapter 2

A Practical Bias Estimation Algorithm for Multisensor–Multitarget Tracking

2.1 Abstract

Bias estimation or sensor registration is an essential step in ensuring the accuracy of global tracks in multisensor-multitarget tracking. Most previously proposed algorithms for bias estimation rely on local measurements in centralized systems or tracks in distributed systems, along with additional information like covariances, filter gains or targets of opportunity. In addition, it is generally assumed that such data are made available to the fusion center at every sampling time. In practical distributed multisensor tracking systems, where each platform sends local tracks to the fusion center, only state estimates and, perhaps, their covariances are sent to the fusion center at non-consecutive sampling instants or scans. That is, not all the

information required for exact bias estimation at the fusion center is available in practical distributed tracking systems. In this paper, a new algorithm that is capable of accurately estimating the biases even in the absence of filter gain information from local platforms is proposed for distributed tracking systems with intermittent track transmission. Through the calculation of the Posterior Cramér–Rao lower bound and various simulation results, it is shown that the performance of the new algorithm, which uses the tracklet idea and does not require track transmission at every sampling time or exchange of filter gains, can approach the performance of the exact bias estimation algorithm that requires local filter gains.

2.2 Introduction

Bias estimation and compensation are essential steps in distributed tracking systems. The objective of sensor registration is to estimate the biases in sensor measurements, such as scaling and offset biases in range and azimuth measurements of a radar, clock bias and/or uncertainties in sensor positions [5]. In a distributed multisensor tracking scenario, each local tracker provides its own estimates of target states for fusion. Local filters can be, e.g., Kalman filters or Interacting Multiple Model (IMM) estimators with different motion models. These local tracks, i.e., state estimate vectors and associated covariance matrices, are sent to the fusion center for further processing. Next, the fusion center carries out track–to–track fusion. The fusion is done sequentially subsequent to the estimation of biases based on common targets that are tracked by various sensors in different locations.

Usually, bias estimation is considered as a two–sensor problem [13, 8] where a stacked vector is assumed with all unknown biases and states target. A drawback of

this approach is the computational burden due to increased dimension of the stacked vector. In addition, most of the algorithms proposed for estimating the biases operate on the measurements directly [23]. That is, such methods perform filtering on the measurements received from sensors, which also include the biases. In many practical tracking systems, access to measurements before tracking at the sensor level is not always feasible. That is, sensors may provide only processed tracks to the user for further processing [5]. Thus, methods that can simultaneously handle track–to–track fusion and bias estimation are needed.

Although there are many different methods in the literature for bias estimation and compensation, there is still a need for a method that requires only the local track estimates and associated covariance matrices for bias estimation. In [25] and [24] a joint track-to-track bias estimation and fusion algorithm based on equivalent measurements of the local tracks was proposed. In [16], another approach based on pseudo-measurement along with the Expectation-Maximization (EM) algorithm to perform joint fusion and registration was proposed. A different method that uses a multistart local search to handle the joint track-to-track association and bias estimation problem was introduced in [26]. The concept of pseudo-measurement was used in [20] for *exact* bias estimation with further extensions in [19] and [21]. In order to achieve exact bias estimation, the algorithms in [20, 19, 21] require the Kalman gains from local trackers, which are not normally sent to the fusion center in practical systems [11]. Moreover, the previously mentioned algorithms assumed that the fusion center receives the local tracks from all sensors at every time step, which is not realistic in systems with bandwidth limitations [7]. In addition, these methods require perfect knowledge about each local filter and its dynamic model. Also, as the number

of sensors increases, the bias estimation problem suffers from the curse of dimensionality because of the commonly used stacked bias vector implementation [13]. Finally, as the number of sensors changes over time, the algorithms in [20, 19, 21] require appropriate pseudo–measurement to be defined for the specific number of sensors.

In this paper, these issues are addressed and a practical solution, which is mathematically sound and computationally feasible, is presented. The new approach is based on reconstructing the Kalman gains of the local trackers at the fusion center. In this approach, the tracklet method [11, 9, 10] along with sequential update as a fusion method is used to provide a low computational cost algorithm for bias estimation. Also, some of the constraints that were discussed above are relaxed in the proposed algorithm. The main contributions of the new algorithm are: a) reconstruction of Kalman gains at the fusion center, b) relaxing the constraints on receiving local tracks at every time step, c) correcting local tracks at the fusion center and d) providing a fused track with low computational cost.

The paper is structured as follows: The bias model and the assumptions for bias estimation are discussed in Section 2.3. In Section 2.4, a review of the exact bias estimation method [20, 19, 21] is given. The new approach and its mathematical developments are given in Section 2.5. Section 2.6 presents the calculation of the Cramér–Rao Lower Bound (CRLB) for proposed algorithm. Section 2.7 demonstrates the performance of the new algorithm for synchronous sensors and compares it with that of the method in [20] and shows comparisons with the CRLB. Conclusions are discussed in Section 2.8.

2.3 Problem Formulation

Assume that there are M sensors reporting range and azimuth measurements[1] in polar coordinates of N targets in the common surveillance region. Note that N is not exactly known to the algorithm and that it could be time varying. That is, bias estimation is carried out based on time–varying and possibly erroneous numbers of tracks reported by the local trackers. The model for the measurements originating from a target with biases at time k in polar coordinates (denoted by superscript p) for sensor s is [20, 19, 21]

$$z_s^p(k) = \begin{bmatrix} r_s^p(k) \\ \theta_s^p(k) \end{bmatrix} = \begin{bmatrix} [1+\epsilon_s^r(k)]\, r_s(k) + b_s^r(k) + w_s^r(k) \\ \left[1+\epsilon_s^\theta(k)\right] \theta_s(k) + b_s^\theta(k) + w_s^\theta(k) \end{bmatrix} \qquad s = 1, \ldots, M \quad (2.1)$$

where $r_s(k)$ and $\theta_s(k)$ are the true range and azimuth, respectively, $b_s^r(k)$ and $b_s^\theta(k)$ are the offset biases in the range and azimuth, respectively, $\epsilon_s^r(k)$ and $\epsilon_s^\theta(k)$ are the scale biases in the range and azimuth, respectively. The measurement noises $w_s^r(k)$ and $w_s^\theta(k)$ in range and bearing are zero-mean with corresponding variances σ_r^2 and σ_θ^2, respectively, and are assumed mutually independent.

The bias vector $\beta_s(k) = \begin{bmatrix} b_s^r(k) & b_s^\theta(k) & \epsilon_s^r(k) & \epsilon_s^\theta(k) \end{bmatrix}^{\mathrm{T}}$ can be modeled as an unknown constant over a certain window of scans (non–random variable). Consequently, the maximum likelihood (ML) estimator [2] or the weighted least squares (LS) estimator [1] can be used for bias estimation. On the other hand, a Gauss-Markov random model [27] can also be used, in which case a Kalman filter can be

[1] While this assumes 2–D radars, the extension to 3–D radars is straightforward.

adopted for bias estimation. We model the measurement as

$$z_s^p(k) = \begin{bmatrix} r_s(k) \\ \theta_s(k) \end{bmatrix} + C_s(k)\beta_s(k) + \begin{bmatrix} w_s^r(k) \\ w_s^\theta(k) \end{bmatrix} \tag{2.2}$$

where

$$C_s(k) \triangleq \begin{bmatrix} 1 & 0 & r_s(k) & 0 \\ 0 & 1 & 0 & \theta_s(k) \end{bmatrix} \tag{2.3}$$

Here, the measured azimuth $\theta_s^m(k)$ and range $r_s^m(k)$ can be utilized in (2.3) without any significant loss of performance [20, 19, 21].

Estimating the bias vector $\beta_s(k)$ for all the sensors is the main objective of this paper. After bias estimation, all the biases can be compensated for in the state estimates at the fusion center.

Since target motion is better modeled and most trackers operate in Cartesian coordinates, the polar measurements are converted into Cartesian coordinates. It is assumed that this does not introduce biases [4]; this is verified in the simulations. Then, sensor s has the measurement equation (with the same $H_s(k) = H(k)$ for all s)

$$z_s(k) \;=\; H(k)\mathbf{x}(k) + B_s(k)C_s(k)\beta_s(k) + w_s(k) \tag{2.4}$$

where the state vector $\mathbf{x}(k) = \begin{bmatrix} x(k) & \dot{x}(k) & y(k) & \dot{y}(k) \end{bmatrix}^{\mathrm{T}}$ and $H(k)$ is the measurement matrix given by

$$H(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \triangleq H \tag{2.5}$$

Since distributed tracking systems may cover a large geographical area, the earth can no longer be assumed to be flat and coordinate transformations need to include an earth curvature model like WGS-84 [18, 32].

The matrix $B_s(k)$ is a nonlinear function with the true range and azimuth as its arguments. A constant $B_s(k)C_s(k)$ also results in incomplete observability as discussed in [21]. Using the measured azimuth $\theta_s^m(k)$ and range $r_s^m(k)$ from sensor s, $B_s(k)$ can be written as [4]

$$B_s(k) = \begin{bmatrix} \cos\theta_s^m(k) & -r_s^m(k)\sin\theta_s^m(k) \\ \sin\theta_s^m(k) & r_s^m(k)\cos\theta_s^m(k) \end{bmatrix} \tag{2.6}$$

Finally, the new covariance matrix of the measurement in Cartesian coordinates (omitting index k in the measurements for clarity) is given by

$$R_s(k) = \begin{pmatrix} r_s^2\sigma_\theta^2\sin^2\theta_s + \sigma_r^2\cos^2\theta_s & (\sigma_r^2 - r_s^2\sigma_\theta^2)\sin\theta_s\cos\theta_s \\ (\sigma_r^2 - r_s^2\sigma_\theta^2)\sin\theta_s\cos\theta_s & r_s^2\sigma_\theta^2\cos^2\theta_s + \sigma_r^2\sin^2\theta_s \end{pmatrix} \tag{2.7}$$

where one can use the observed range and azimuth as well.

2.4 Review of Synchronous Sensor Registration

In this section, the bias estimation method introduced in [20, 21, 19] for synchronous sensors with known sensor locations is reviewed. Further, the methods in our previous work [29] are examined in more detail and are extended in this paper with various simulations and the calculation of the lower bounds for bias estimation in multisensor–multitarget scenarios.

Consider a multisensor tracking system with the decentralized architecture [5]. In this case, each local tracker runs its own filtering algorithm and obtains a local state estimate using only its own measurements. Then, all local trackers send their estimates to the fusion center where bias estimation is addressed. Only after bias estimation can the fusion center fuse local estimates correctly to obtain accurate global estimates.

The dynamic equation for the target state is

$$\mathbf{x}(k+1) \quad = \quad F(k)\mathbf{x}(k) + v(k) \tag{2.8}$$

where $F(k)$ is the transition matrix, and $v(k)$ is a zero-mean additive white Gaussian noise with covariance $Q(k)$.

Because the local trackers are not able to estimate the biases on their own, they yield inaccurate estimates of tracks by assuming no bias in their measurements. Hence, the state space model considered by local trackers for a specific target t and

sensor s is

$$\mathbf{x}^t(k+1) = F(k)\mathbf{x}^t(k) + v(k) \tag{2.9}$$

$$z_s^t(k) = H(k)\mathbf{x}^t(k) + w_s(k) \tag{2.10}$$

The difference between (2.1) and (2.10) is that the latter has no bias term and, as a result, the local tracks are bias-ignorant [20, 21, 19]. Note that this mismatch should be compensated for.

2.4.1 The pseudo-measurement of the bias vector

In this subsection, a brief discussion on how to find an informative pseudo-measurement by using the local tracks for the case $M = 2$ synchronized sensors is presented, based on the method given in [20, 21, 19]. As in these previous works, it is assumed that the local platforms run a Kalman filter-based tracker, although this assumption may not always be valid. However, as shown in the sequel, multiple-model based trackers can be handled within the proposed framework with some extensions.

In [20, 21, 19] it was assumed that one has access to the filter gain $W_1(k+1)$ and the residual $\nu_1(k+1)$ from the Kalman filter of local tracker 1 [31]. Then, one can

write

$$\begin{aligned}\hat{\mathbf{x}}_1(k+1 \mid k+1) &= F(k)\hat{\mathbf{x}}_1(k \mid k) + W_1(k+1)\nu_1(k+1) \\ &= F(k)\hat{\mathbf{x}}_1(k \mid k) + W_1(k+1)\left[z_1(k+1) - \hat{z}_1^0(k+1 \mid k)\right] \\ &= \left[I - W_1(k+1)H(k+1)\right]F(k)\hat{\mathbf{x}}_1(k \mid k) + W_1(k+1)\left[H(k+1)\right. \\ &\quad F(k)\mathbf{x}_1(k) + H(k+1)v(k) + B_1(k+1)C_1(k+1)\beta_1(k+1) \\ &\quad \left.+w_1(k+1)\right] \end{aligned} \tag{2.11}$$

Note that the predicted measurement $\hat{z}_1^0(k+1 \mid k)$ is based on the measurement in which no bias is assumed by local tracker 1, i.e., tracker 1 used a bias-ignorant measurement model. Therefore, there is no term related to biases in the predicted measurement.

Hence, if the local state estimate is moved to the left–hand side of (2.11), and left-multiplied by the left pseudo-inverse [15] of the gain, one has

$$\begin{aligned} z_b^1(k+1) &\triangleq W_1^{\dagger}(k+1)\left[\hat{\mathbf{x}}_1(k+1 \mid k+1) - \left(I - W_1(k+1)H(k+1)\right)F(k)\hat{\mathbf{x}}_1(k \mid k)\right] \\ &= H(k+1)F(k)\mathbf{x}(k) + H(k+1)v(k) + B_1(k+1)C_1(k+1)\beta_1(k+1) \\ &\quad +w_1(k+1) \end{aligned} \tag{2.12}$$

where the pseudo-inverse of the gain is

$$W_s^{\dagger} \triangleq \left(W_s^{\mathrm{T}}W_s\right)^{-1}W_s^{\mathrm{T}} \tag{2.13}$$

Similarly, one can define

$$\begin{aligned} z_b^2(k+1) &\triangleq W_2^{\dagger}(k+1)\left[\hat{\mathbf{x}}_2(k+1 \mid k+1) - (I - W_2(k+1)H(k+1))F(k)\hat{\mathbf{x}}_2(k \mid k)\right] \\ &= H(k+1)F(k)\mathbf{x}(k) + H(k+1)v(k) + B_2(k+1)C_2(k+1)\beta_2(k+1) \\ &\quad + w_2(k+1) \end{aligned} \tag{2.14}$$

It is worth mentioning that $\mathbf{x}(k)$ and $v(k)$ in (2.12) and (2.14) are the same. Thus, a pseudo-measurement of the bias vector, as in [20, 21, 19], can be defined as follows:

$$z_b(k+1) \triangleq z_b^1(k+1) - z_b^2(k+1) \tag{2.15}$$

for the case of using similar sensors. Then,

$$\begin{aligned} z_b(k+1) &= B_1(k+1)C_1(k+1)\beta_1(k+1) - B_2(k+1)C_2(k+1)\beta_2(k+1) \\ &\quad + w_1(k+1) - w_2(k+1) \end{aligned} \tag{2.16}$$

That is, one has the pseudo-measurement of the bias vector

$$z_b(k+1) = \mathcal{H}(k+1)\mathbf{b}(k+1) + \tilde{w}(k+1) \tag{2.17}$$

where the pseudo-measurement matrix $\mathcal{H}$, the bias parameter vector $\mathbf{b}$ and the pseudo-measurement noise $\tilde{w}(k+1)$ are defined as

$$\mathcal{H}(k+1) \triangleq [B_1(k+1)C_1(k+1),\ -B_2(k+1)C_2(k+1)] \tag{2.18}$$

$$\mathbf{b}(k+1) \triangleq \begin{bmatrix} \beta_1(k+1) \\ \beta_2(k+1) \end{bmatrix} \tag{2.19}$$

and

$$\tilde{w}(k+1) \triangleq w_1(k+1) - w_2(k+1) \tag{2.20}$$

The bias pseudo-measurement noises $\tilde{w}$ are additive white Gaussian with zero–mean, and their covariance is

$$\mathcal{R}(k+1) = R_1(k+1) + R_2(k+1) \tag{2.21}$$

The main property of (2.20) is its whiteness, which results in a bias estimate that is exact [20, 21, 19]. In this approach, there is no approximation in deriving (2.17)–(2.21) unlike the methods previously proposed in [17, 28, 30]. This was one of the main contributions of [20].

When the measurement matrices $H_s(k)$ are the same for different local trackers, but only the second sensor has a bias, the following simplifications result:

$$z_b(k+1) = z_b^1(k+1) - z_b^2(k+1) \tag{2.22}$$

$$\mathbf{b}(k+1) = \beta_2(k+1) \tag{2.23}$$

$$\mathcal{H}(k+1) = -B_2(k+1)C_2(k+1) \tag{2.24}$$

$$\tilde{w}(k+1) = w_1(k+1) - w_2(k+1) \tag{2.25}$$

$$\mathcal{R}(k+1) = R_1(k+1) + R_2(k+1). \tag{2.26}$$

2.4.2 The recursive least square bias estimator

If the biases are constant over a certain window of scans, one can construct a Recursive Least Square (RLS) estimator by using the pseudo-measurement equation (2.17) [20]. The recursion of the RLS estimator has two stages: the first is to update the bias estimate recursively for different targets and the second is to update it through different time scans.

Assume that at time k, one has access to the estimate of the bias vector and its associated covariance matrix up to time k as $\hat{\mathbf{b}}_{t-1}(k)$ and $\Sigma_{t-1}(k)$, form on the first $t-1$ targets and all previously updated estimates. Now, the RLS method can be carried out as in Figure 2.1 to update the bias estimation at time k for all targets [20, 21, 19].

Note that the covariance update equation in line 5 of Figure 2.1 may cause $\Sigma_t(k)$ to lose positive definiteness due to numerical errors. To avoid this problem, the Joseph's form of the covariance update is used [4] as

$$\Sigma_t(k) \;=\; [I - G_t(k)\mathcal{H}_t(k)]\,\Sigma_{t-1}(k)\,[I - G_t(k)\mathcal{H}_t(k)]^{\mathrm{T}} + G_t(k)G_t(k)^{\mathrm{T}} \quad (2.27)$$

2.4.3 Time-varying bias estimation: The optimal MMSE estimator

In the case of time-varying biases with the standard linear white–Gaussian assumptions one can implement the optimal MMSE estimator based on the pseudo-measurement equation (2.17) and the dynamic model of the bias [20]. f For the stacked bias vector, the dynamics model can be defines as

Input: $\hat{\mathbf{b}}_0(k)$, $\Sigma_0(k)$, $z^1_{b,t}(k)$, $z^2_{b,t}(k)$
Output: $\hat{\mathbf{b}}_0(k+1)$, $\Sigma_0(k+1)$

1: At time k
2: **for** $t = 1, ..., N$ **do**
3: Get the new pseudo-measurement using

$$z_{b,t}(k) \triangleq z^1_{b,t}(k) - H(k)H^{\dagger}(k)z^2_{b,t}(k)$$

4: Compute the bias update gain and the residual

$$\begin{aligned} G_t(k) &= \Sigma_{t-1}(k)\mathcal{H}_t(k)^{\mathrm{T}}\left[\mathcal{H}_t(k)\Sigma_{t-1}(k)\mathcal{H}_t(k)^{\mathrm{T}}\ \mathcal{R}_t(k)\right]^{-1} \\ r_t(k) &= z_{b,t}(k) - \mathcal{H}_t(k)\hat{\mathbf{b}}_{t-1}(k+1) \end{aligned}$$

5: Update the bias estimate and covariance

$$\begin{aligned} \hat{\mathbf{b}}_t(k) &= \hat{\mathbf{b}}_{t-1}(k+1) + G_t(k)r_t(k) \\ \Sigma_t(k) &= \Sigma_{t-1}(k) - \Sigma_{t-1}(k)\mathcal{H}_t(k)^{\mathrm{T}})^{\mathrm{T}}\left[\mathcal{H}_t(k)\ \Sigma_{t-1}(k)\mathcal{H}_t(k)^{\mathrm{T}} + \mathcal{R}_t(k)\right]^{-1} \\ &\quad \times\mathcal{H}_t(k)\Sigma_{t-1}(k) \end{aligned}$$

6: **end for**
7: **return**

$$\begin{aligned} \hat{\mathbf{b}}_0(k+1) &\triangleq \hat{\mathbf{b}}_N(k) \\ \Sigma_0(k+1) &\triangleq \Sigma_N(k) \end{aligned}$$

Figure 2.1: The Recursive Least Square Bias Estimation (RLSBE) algorithm [20].

$$\mathbf{b}(k+1) = F_b(k)\mathbf{b}(k) + v_b(k) \tag{2.28}$$

in which $F_b(k)$ is the transition matrix of the stacked bias vector $\mathbf{b}$, and $v_b(k)$ is the stacked process noise of the bias vector, zero-mean white with covariance $Q_b(k)$.

Assume that at time k one has access to the estimate of the bias vector and its

associated covariance matrix up to time k as $\hat{\mathbf{b}}_{t-1}(k \mid k)$ and $\Sigma_{t-1}(k \mid k)$, respectively. Now, a Kalman filter can be used as in Figure 2.2 to update the bias estimates at time k for all targets [20, 21, 19].

2.5 The new bias estimation algorithm

The algorithms given in Section 2.4, i.e., Recursive Least Square Bias Estimation (RLSBE, Figure 2.1) and Optimal MMSE Bias Estimation (OMBE, Figure 2.2), are dependent on the Kalman gains provided by the local trackers, in addition to the state estimates and the associated covariance matrices at every time step. Moreover, as the number of the sensors increases, the above algorithms face an increase in computational requirements cubic in M. This is because a stacked vector of bias parameters is used. In addition, for $M > 2$, it is challenging to extend (2.15) and (2.17). The extension of (2.15) and (2.17) can be done as in [3] by taking $M - 1$ differences. Moreover, these approaches do not address the joint fusion problem as well. With this motivation, in this section, a new approach to relax the requirement of the Kalman gain matrices availability from the local trackers is given. In addition, the new algorithm alleviates the problem of the dimensionality by taking advantage of (2.22)–(2.26) for a multisensor–multitarget scenario, and by solving the fusion problem as well. Finally, the algorithm is able to function properly with asynchronous local track updates.

In order to obtain the new algorithm, first, a simple approach to calculate tracklets based on [11] is discussed. This approach makes it possible to obtain approximate equivalent measurements of the local tracks directly and efficiently without any further processing and it supports updating the bias estimates whenever a new local

Input: $\hat{\mathbf{b}}_0(k \mid k)$, $\Sigma_0(k \mid k)$, $z^1_{b,t}(k)$, $z^2_{b,t}(k)$
Output: $\hat{\mathbf{b}}_0(k+1 \mid k+1)$, $\Sigma_0(k+1 \mid k+1)$

1: At time k
2: **for** $t = 1, ..., N$ **do**
3: Get the new pseudo-measurement using

$$z_{b,t}(k) \triangleq z^1_{b,t}(k) - H(k)H^{\dagger}(k)z^2_{b,t}(k)$$

4: Compute the bias update gain and the residual

$$\begin{aligned} G_t(k) &= \Sigma_{t-1}(k \mid k)\mathcal{H}_t(k)^{\mathrm{T}} [\mathcal{H}_t(k)\Sigma_{t-1}(k \mid k) \\ &\quad \mathcal{H}_t(k)^{\mathrm{T}} + \mathcal{R}_t(k)]^{-1} \\ r_t(k) &= z_{b,t}(k) - \mathcal{H}_t(k)\hat{\mathbf{b}}_{t-1}(k \mid k) \end{aligned}$$

5: Update the bias estimate and covariance

$$\begin{aligned} \hat{\mathbf{b}}_t(k \mid k) &= \hat{\mathbf{b}}_{t-1}(k \mid k) + G_t(k)r_t(k) \\ \Sigma_t(k \mid k) &= \Sigma_{t-1}(k \mid k) - \Sigma_{t-1}(k \mid k)\mathcal{H}_t(k)^{\mathrm{T}} \\ &\quad \left[\mathcal{H}_t(k)\Sigma_{t-1}(k \mid k)\mathcal{H}_t(k)^{\mathrm{T}} + \mathcal{R}_t(k)\right]^{-1} \mathcal{H}_t(k)\Sigma_{t-1}(k \mid k) \end{aligned}$$

6: **end for**
7: Update the bias estimate according to the model

$$\begin{aligned} \hat{\mathbf{b}}(k+1 \mid k) &\triangleq F_b(k)\hat{\mathbf{b}}_N(k \mid k) \\ \Sigma(k+1 \mid k) &\triangleq F_b(k)\Sigma_N(k \mid k)F_b(k)^{\mathrm{T}} + Q_b(k) \end{aligned}$$

8: **return**

$$\begin{aligned} \hat{\mathbf{b}}_0(k+1 \mid k+1) &= \hat{\mathbf{b}}(k+1 \mid k) \\ \Sigma_0(k+1 \mid k+1) &= \Sigma(k+1 \mid k) \end{aligned}$$

Figure 2.2: The optimal MMSE bias estimation algorithm [20] (OMBE).

track is available at the fusion center. In addition, it can handle asynchronous updates from different local trackers and map them to a common time [11, 10]. Then, a sequential update algorithm is proposed for the fusion step. Although it is not an optimal approach for fusing the local tracks, it is computationally cheaper than parallel update [5]. Finally, the complete algorithm based on these two approaches with additional steps is presented.

2.5.1 Equivalent measurement computation using the inverse Kalman filter method based tracklet

The main goal in this subsection is to construct a set of approximately uncorrelated equivalent measurements ("tracklets") from the local tracks and the associated covariance matrices for sequential update in the fusion step and also to reconstruct the local Kalman gains at the fusion center. It also relaxes the requirements of receiving the local tracks at every time step. To do so, the "inverse Kalman filter based" tracklet method from [11] is used (for a clear derivation and the reason for its sub–optimality, see [5, p. 577]). Based on this method, the equations relating to the equivalent measurement vector, $\mathbf{u}_s(k, k')$, for a local track from platform s at time frame k, given that the track data was previously sent to the global tracker for time frame $k' < k$, are as follows:

$$\mathbf{u}_s(k, k') = \hat{\mathbf{x}}_s(k \mid k') + \mathbf{A}_s(k \mid k') \left[\hat{\mathbf{x}}_s(k \mid k) - \hat{\mathbf{x}}_s(k \mid k')\right] \tag{2.29}$$

where

$$\begin{aligned} \mathbf{u}_s(k,k') &= \mathbf{x}(k) + \tilde{\mathbf{u}}_s(k \mid k) && (2.30)\\ \mathbb{E}\left[\tilde{\mathbf{u}}_s(k,k') \mid \mathbf{Z}^{k'}\right] &= 0 && (2.31)\\ \mathbf{A}_s(k,k') &= \mathbf{P}_s(k \mid k')\left[\mathbf{D}_s(k,k')\right]^{-1} && (2.32)\\ \mathbf{D}_s(k,k') &= \mathbf{P}_s(k \mid k') - \mathbf{P}_s(k \mid k) && (2.33) \end{aligned}$$

where $\mathbf{Z}_s^{k'} = \left\{\mathbf{z}_s^1, ..., \mathbf{z}_s^{k'}\right\}$, and

$$\begin{aligned} \mathbf{U}_s(k,k') &= \mathbb{E}\left[\tilde{\mathbf{u}}_s(k,k')\left(\tilde{\mathbf{u}}_s(k,k)'\right)^{\mathrm{T}} \mid \mathbf{Z}_s^{k'}\right]\\ &= \mathbf{A}_s(k,k')\mathbf{P}_s(k \mid k)\\ &= \left[\mathbf{A}_s(k,k') - I\right]\mathbf{P}_s(k \mid k') && (2.34) \end{aligned}$$

The information that the global tracker or the fusion center uses consists of the calculated equivalent measurement vector $\mathbf{u}_s(k,k')$ and its error covariance matrix $\mathbf{U}_s(k,k')$. Note that in order to calculate $\hat{\mathbf{x}}_s(k \mid k')$ and $\mathbf{P}_s(k \mid k')$ one needs the estimated target state $\hat{\mathbf{x}}_s(k' \mid k')$ and its covariance matrix $\mathbf{P}_s(k' \mid k')$, in addition to the dynamic models the local trackers used for filtering. Then one needs to compute $L = k - k'$ prediction steps without any new measurement data to find $\hat{\mathbf{x}}_s(k \mid k')$ and $\mathbf{P}_s(k \mid k')$. Here, it is necessary to consider the L-step prediction of transition and process noise covariance matrices as $F(k,k')$ and $Q(k,k')$, respectively. One can use the concept of missing observations in Kalman filter to find $F(k,k')$ and $Q(k,k')$ as in [12, pp. 110]. It should be mentioned that all these computations require that $\mathbf{P}_s(k' \mid k')$, $\mathbf{P}_s(k \mid k')$ and $[\mathbf{P}_s(k \mid k)^{-1} - \mathbf{P}_s(k \mid k')^{-1}]$ be non-singular. This method

was previously used in [24] for $k' = k - 1$ (for which the non-singularity requirement does not hold in general) and with a different approach for sensor registration. For the proof of (2.34) see Appendix A.

2.5.2 Sequential update as fusion method

After calculating the equivalent measurements of the state for each local track at the fusion center, they can be used as new measurements for the estimation of fused state and its covariance matrix. To do this recursively, it should be assumed that the fused state estimate and its covariance matrix at time k' as $\mathbf{x}_f(k' \mid k')$ and $\mathbf{P}_f(k' \mid k')$, respectively, are already computed. For $k' = 1$, the parameters $\mathbf{x}_f(k' \mid k')$ and $\mathbf{P}_f(k' \mid k')$ are initialized with $\mathbf{x}(k' \mid k')$ and $\mathbf{P}(k' \mid k')$, respectively. Then these two can be updated by following the steps in Figure 2.3. Although the sequential update is sub-optimal [5], it has the advantage of being computationally efficient to implement and, in addition, it is not dependent on the previous equivalent measurements at time k'.

2.5.3 Multisensor fusion and track-to-track bias estimation

The first step for implementing a general bias estimation algorithm for radar systems is to find the Kalman gains of each local track at the fusion center, by only using the state estimates and the associated covariance matrices. To do so, first, one must calculate the equivalent measurement and its covariance matrix as in (2.29) and (2.34) for sensor s, and at time frame k (the target index is omitted for simplicity). Since

Input: $\mathbf{x}_f(k' \mid k')$ and $\mathbf{P}_f(k' \mid k')$, $y_{s,k}$ and $R_{s,k}$ for $s = 1, ..., M$
Output: $\mathbf{x}_f(k \mid k)$ and $\mathbf{P}_f(k \mid k)$

1: **Compute** $\mathbf{x}_f(k \mid k')$ and $\mathbf{P}_f(k \mid k')$ according to their dynamic model (prediction step)
2: **Assign:**

$$\begin{aligned} \mathbf{x}_{\text{temp}} &= \mathbf{x}_f(k \mid k') \\ \mathbf{P}_{\text{temp}} &= \mathbf{P}_f(k \mid k'). \end{aligned}$$

3: **for** $s = 1, ..., M$ **do**
4: Update $\mathbf{x}_{\text{temp}}$ and $\mathbf{P}_{\text{temp}}$ with new measurement and its covariance matrix, i.e., $y_{s,k}$ and $R_{s,k}$ according to

$$\begin{aligned} \mathbf{x}_{\text{temp}} &= \mathbf{x}_{\text{temp}} + W_{\text{temp}}\tilde{\mathbf{y}} \\ W_{\text{temp}} &= \mathbf{P}_{\text{temp}}H^{\mathrm{T}}\left(H\mathbf{P}_{\text{temp}}H^{\mathrm{T}} + R_{s,k}\right)^{-1} \\ \tilde{\mathbf{y}} &= y_{s,k} - H\mathbf{x}_{\text{temp}} \\ \mathbf{P}_{\text{temp}} &= \left(I - W_{\text{temp}}H\right)\mathbf{P}_{\text{temp}} \\ H &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{aligned}$$

5: **end for**
6: **return**

$$\begin{aligned} \mathbf{x}_f(k \mid k) &= \mathbf{x}_{\text{temp}} \\ \mathbf{P}_f(k \mid k) &= \mathbf{P}_{\text{temp}} \end{aligned}$$

Figure 2.3: The Sequential Fusion Algorithm (SFA) with equivalent measurements

the measurement model here is linear, one has

$$R_{s,k} = H(k)\mathbf{U}_s(k,k')H(k)^{\mathrm{T}} \tag{2.35}$$

$$W_{s,k} = \mathbf{P}_s(k \mid k')H(k)^{\mathrm{T}}\left[H(k)\mathbf{P}_s(k \mid k')H(k)^{\mathrm{T}} + R_{s,k}\right]^{-1} \tag{2.36}$$

$$y_{s,k} = H(k)\mathbf{u}_s(k,k') \tag{2.37}$$

Note that the reason to keep the position information only is that further in the new bias estimation algorithm we use the corrected positions as new measurements for sequential fusion. Because the equivalent measurements are used for the fused track, the Kalman gain for it (denoted by subscript f) at time frame k can be recovered as

$$R_{f,k} = H(k)\left[\sum_{i=1}^{M}\left(\mathbf{U}_i(k,k')\right)^{-1}\right]^{-1}H(k)^{\mathrm{T}} \tag{2.38}$$

$$W_{f,k} = \mathbf{P}_f(k \mid k')H(k)^{\mathrm{T}}\left[H(k)\mathbf{P}_f(k \mid k')H(k)^{\mathrm{T}} + R_{f,k}\right]^{-1} \tag{2.39}$$

Note that the noises/errors in the equivalent measurements are not white so using a Kalman filter is not optimal. This amounts to the same approximation as in [5, p. 563]. Also in (2.39), a common coordinate system is used for equivalent measurements. As a result, the use of a common measurement matrix $H(k)$ for all the equivalent measurements is feasible.

The sensor registration method proposed here uses the simplified formulas, i.e., (2.22)–(2.26). To use these formulas, an approximately bias-compensated[2] fused track needs to be found at the fusion center for the set $\{\mathcal{S}\} \setminus s$, where $\mathcal{S} = 1, 2, ..., M$. The notation $\{\mathcal{S}\} \setminus s$ stands for the set that contains all those elements of $\mathcal{S}$ excluding

[2]Here, bias-compensated means a fused track, in which the bias-corrected equivalent measurements by using the latest estimated biases are used for fusion.

element s. Then, the bias estimation problem can be reduced to the case of two sensors. The first one is an equivalent sensor with fusion of bias-corrected measurements of the set $\{\mathcal{S}\} \setminus s$, and the second is sensor s which has bias. Then the bias in sensor s can be found with either the RLSBE (see Figure 2.1) or OMBE (see Figure 2.2) algorithm.

The next step in this approach is to correct the biases in the measurement domain of the sensors (except for sensor s) and then fuse them together. This can be done by going from the Cartesian coordinate of equivalent measurements to the polar coordinate of the radar and correct with the previously estimated biases. Then by correcting the covariance matrix of the new bias compensated measurements, they can be fused by sequentially updating the fused track (excluding the track from sensor s) by using the Sequential Fusion Algorithm (SFA). Then, the Kalman gain for the fused and the now-corrected track can be calculated using (2.39).

For the equivalent measurement, define

$$\mathbf{u}_s \triangleq \begin{bmatrix} u_x & \dot{u}_x & u_y & \dot{u}_y \end{bmatrix}^{\mathrm{T}} \tag{2.40}$$

where time and target indexes are omitted for simplicity. Then, to correct the biases in the measurement domain, assuming that the latest–estimated biases are $\hat{\epsilon}_\theta$, $\hat{\epsilon}_r$, $\hat{b}_\theta$ and $\hat{b}_r$, one has[3]

$$\theta_s^{\text{b-c}} = \frac{\arctan\left(\frac{u_y}{u_x}\right) - \hat{b}_\theta}{(1+\hat{\epsilon}_\theta)} \tag{2.41}$$

$$r_s^{\text{b-c}} = \frac{\sqrt{(u_x)^2 + (u_y)^2} - \hat{b}_r}{(1+\hat{\epsilon}_r)} \tag{2.42}$$

[3]Here, the superscript "b-c" is used to denote the bias-corrected bearing and range.

Now that the scale and offset biases are compensated for, one can go back to Cartesian coordinates as follows:

$$u_x^{\text{b-c}} = \lambda_\theta r_s^{\text{b-c}} \cos\left(\theta_s^{\text{b-c}}\right) \tag{2.43}$$

$$u_y^{\text{b-c}} = \lambda_\theta r_s^{\text{b-c}} \sin\left(\theta_s^{\text{b-c}}\right) \tag{2.44}$$

$$\mathcal{Y}^{\text{b-c}} = \begin{bmatrix} u_x^{\text{b-c}} & u_y^{\text{b-c}} \end{bmatrix}^{\text{T}} \tag{2.45}$$

where

$$\lambda_\theta = \exp\left(-\frac{\sigma_\theta^2}{2}\right) \tag{2.46}$$

is the compensation factor for the bias in coordinate conversion from polar to Cartesian [22]. The method from [22] is used here because of the fact that bias correction and compensation along with the changes in the covariance matrices and uncertainties may violate the assumptions made for the debaised conversion in Section 2.3. The next step is to update the covariance matrix of the corrected equivalent measurements. In addition to the term (2.35), the additional uncertainty in the bias estimates, i.e., their associated covariance matrix and the uncertainty in the model of the radar, both in Cartesian coordinates, must be accounted for. The final formula for the covariance matrix with proper conversion from polar to Cartesian coordinate is

$$\begin{aligned} R_s^{\text{b-c}} = {} & H(k)\mathbf{U}_s(k \mid k)H(k)^{\text{T}} + B_s^{\text{b-c}}(k) \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} B_s^{\text{b-c}}(k)^{\text{T}} \\ & + K_s^{\text{b-c}}(k)\mathbf{P}_{b,s}(k \mid k)K_s^{\text{b-c}}(k)^{\text{T}} \end{aligned} \tag{2.47}$$

where

$$K_s^{\text{b-c}}(k) \;=\; B_s^{\text{b-c}}(k) C_s^{\text{b-c}}(k) \tag{2.48}$$

and $\mathbf{P}_{b,s}(k \mid k)$ is the latest-updated bias covariance matrix at time k and for sensor s. Now that all the required formulations and variables are available, the new algorithm to find the bias estimates for all the sensors is given in Figure 2.4.

Input: inputs of SFA and RLSBE (defined within the corresponding algorithms).
Output: $\mathbf{b}_s(k)$ and $\Sigma_s(k)$ for $s = 1, ..., M$.

1: At time k
2: **for** $s = 1, ..., M$ **do**
3: **Compute** $W_{s,k}$ as in (2.36) and
4:

$$z_{b,t}^s(k) \;\triangleq\; W_{s,k}^{\dagger} \left[\hat{\mathbf{x}}(k \mid k) - (I - W_{s,k} H_s(k)) F(k, k-L) \hat{\mathbf{x}}(k-L \mid k-L)\right]$$

5: $\bar{s} \in \{1, ..., M\} \backslash s$
6: **Call** SFA with inputs $\mathbf{x}_f^s(k' \mid k')$ and $\mathbf{P}_f^s(k' \mid k')$, $\mathcal{Y}_{\bar{s}}^{\text{b-c}}(k \mid k)$ and $R_{\bar{s}}^{\text{b-c}}(k \mid k)$.
7: **Compute** $W_{f,k}^s$ as in (2.39) and

$$z_{b,f}^s(k) \;\triangleq\; \left(W_{f,k}^s\right)^{\dagger} \left[\hat{\mathbf{x}}_f^s(k \mid k) - (I - W_{f,k}^s H_f(k)) F(k, k-L) \hat{\mathbf{x}}_f^s(k-L \mid k-L)\right]$$

8: **Call** RLSBE with inputs $z_{b,t}^s(k)$, $z_{b,f}^s(k)$ and the last update of the estimated biases and their associated covariance matrix.
9: **return** $\mathbf{b}_s(k)$ and $\Sigma_s(k)$
10: **end for**

Figure 2.4: The Fused Bias Estimation algorithm (FBEA)

As shown in Figure 2.4, one only needs to call SFA and RLSBE with new input parameters. One of the advantages of this approach is that in each "**for loop**" only a low dimensional Kalman filter that is *independent* of the size of the stacked bias vector and number of the sensors is needed. In addition, the fusion of local tracks can be done by only adding one sequential update for the latest corrected measurement

of sensor s to the previously fused track. It is also important to note that the there is no constraint on the rate of receiving local tracks from the individual sensors. To show how well this new algorithm performs, in the next section, the simulation results on two different scenarios are used to compare its performance with those of the previously proposed algorithm in [20, 19, 21] for synchronous sensors.

To better illustrate how the new bias estimation algorithm (FBEA) works, a block–diagram representation of the method is shown in Figure 2.5 for a single time step estimation of the biases for the first sensor. In Figure 2.5, by receiving the local track estimates from all available sensors at time step k, the first step is to calculate the tracklet for all of them using (2.29)–(2.34). Then, the equivalent measurement of the first sensor is sent for Kalman gain recovery using (2.35) and (2.36). At the same time, the equivalent measurements of all other sensors are sent to bias correction to first remove the bias from the equivalent measurements by using the previously estimated biases at time step $k-1$ using (2.41)–(2.45) and the Kalman gain recovery in (2.38) and (2.39). The next step is to fuse the tracks by using SFA algorithm. Then, the fused and corrected estimate is sent to the pseudo–measurement calculation block for each individual sensor. At this point one has a two–sensor problem with only one sensor having biases in the measurement. The output is now sent to the RLSBE algorithm along with the previously estimated biases for the first sensor so that the bias estimates can be updated at time step k before proceeding to the next time step.

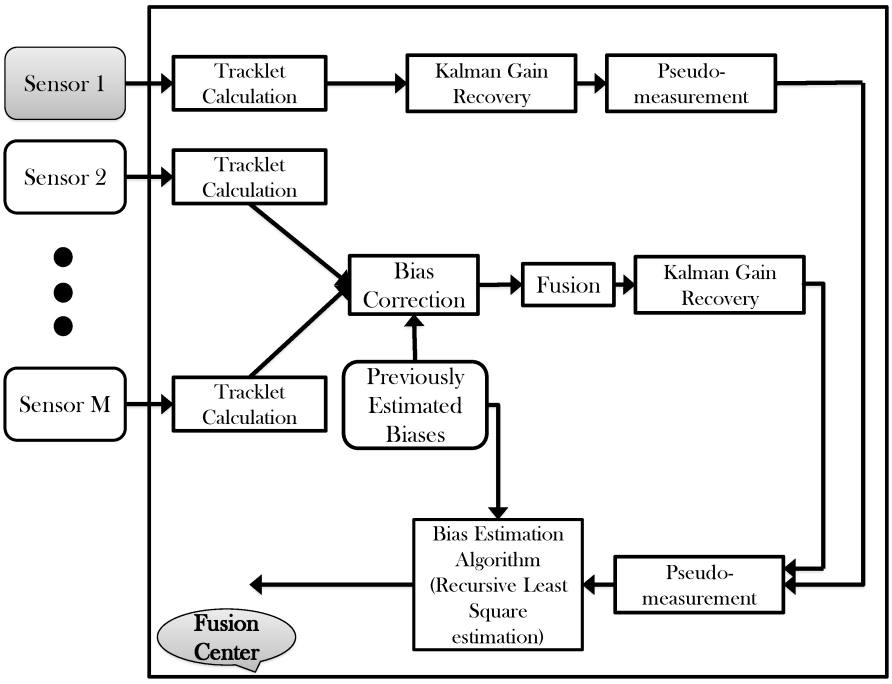

Figure 2.5: Block diagram of the new offset and scaling bias estimation algorithm.

2.6 Cramér-Rao Lower Bound for Sensor Registration

In this section, a step-by-step procedure is given for calculating the CRLB for sensor registration algorithms as the benchmark. Rewriting (2.4) for the case of two similar sensors, one has

$$z_1(k) - H(k)\mathbf{x}(k) = B_1(k)\beta_1(k) + w_1(k) \tag{2.49}$$

and

$$z_2(k) - H(k)\mathbf{x}(k) = B_2(k)\beta_2(k) + w_2(k) \tag{2.50}$$

If no biases exist, the sensors must point to the same position of the observed target. Consequently, one has

$$z_1(k) - B_1(k)\beta_1(k) - w_1(k) = z_2(k) - B_2(k)\beta_2(k) - w_2(k) \tag{2.51}$$

and by reordering the measurement terms and using the matrix form, one can rewrite it as

$$[z_1(k) - z_2(k)] = \begin{bmatrix} B_1(k) & -B_2(k) \end{bmatrix} \begin{bmatrix} \beta_1(k) \\ \beta_2(k) \end{bmatrix} + w_{1,2}(k) \tag{2.52}$$

Further, for future use, one can denote the terms in (2.52) as

$$Y(k) = B(k)\mathbf{b}(k) + w_{1,2}(k) \tag{2.53}$$

where

$$Y(k) = [z_1(k) - z_2(k)] \tag{2.54}$$

$$B(k) = \begin{bmatrix} B_1(k) & -B_2(k) \end{bmatrix} \tag{2.55}$$

$$\mathbf{b}(k) = \begin{bmatrix} \beta_1(k) \\ \beta_2(k) \end{bmatrix} \tag{2.56}$$

and $w_{1,2}(k)$ is additive white Gaussian noise with covariance matrix equal to $R_1(k) + R_2(k)$

2.6.1 Calculation of the CRLB

In the case of having two sensors and multiple targets, the CRLB can be calculated as a batch process. Taking all the (linearly independent) K pairs of measurements for N targets in the surveillance region, one can write the measurement equation as

$$\mathbf{Y} = \mathbf{gb} + \mathbf{u} \tag{2.57}$$

where $\mathbf{Y}$, $\mathbf{g}$ and $\mathbf{u}$ are stacked vectors given by

$$\mathbf{Y} = \begin{bmatrix} (Y^1(1))^{\mathrm{T}} & \cdots & \left(Y^N(1)\right)^{\mathrm{T}} & \cdots & (Y^1(K))^{\mathrm{T}} & \cdots & \left(Y^N(K)\right)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{2.58}$$

$$\mathbf{g} = \begin{bmatrix} (B^1(1))^{\mathrm{T}} & \cdots & \left(B^N(1)\right)^{\mathrm{T}} & \cdots & (B^1(K))^{\mathrm{T}} & \cdots & \left(B^N(K)\right)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{2.59}$$

and

$$\mathbf{u} = \left[\begin{array}{cccccccc} \left(w_{1,2}^1(1)\right)^{\mathrm{T}} & \cdots & \left(w_{1,2}^N(1)\right)^{\mathrm{T}} & \cdots & \left(w_{1,2}^1(K)\right)^{\mathrm{T}} & \cdots & \left(w_{1,2}^N(K)\right)^{\mathrm{T}} \end{array} \right]^{\mathrm{T}} \tag{2.60}$$

Further, the covariance matrix of the noise vector $\mathbf{u}$ is

$$\mathcal{R} = \mathrm{diag}\left(\left[\begin{array}{cccccccc} \mathcal{R}^1(1) & \cdots & \mathcal{R}^N(1) & \cdots & \mathcal{R}^1(K) & \cdots & \mathcal{R}^N(K) \end{array} \right] \right) \tag{2.61}$$

where $\mathcal{R}^i(k) = R_1^i(k) + R_2^i(k)$ and the lower index indicates a specific sensor. As stated in [4], the covariance matrix of an unbiased estimator $\hat{\mathbf{b}}$ is bounded from below as

$$\mathbb{E}\left\{ \left(\hat{\mathbf{b}} - \mathbf{b}\right) \left(\hat{\mathbf{b}} - \mathbf{b}\right)^{\mathrm{T}} \right\} \geq \mathbf{J}^{-1} \tag{2.62}$$

In the above, $\mathbf{J}$ is the Fisher Information Matrix (FIM) given by

$$\begin{aligned} \mathbf{J} &= \mathbb{E}\left\{ \left[\nabla_{\mathbf{b}} \ln p(\mathbf{Y} \mid \mathbf{b})\right] \left[\nabla_{\mathbf{b}} \ln p(\mathbf{Y} \mid \mathbf{b})\right]^{\mathrm{T}} \right\} |_{\mathbf{b}=\mathbf{b}_{\mathrm{true}}} \\ &= \mathbb{E}\left\{ \left[\nabla_{\mathbf{b}} \lambda\right] \left[\nabla_{\mathbf{b}} \lambda\right]^{\mathrm{T}} \right\} |_{\mathbf{b}=\mathbf{b}_{\mathrm{true}}} \end{aligned} \tag{2.63}$$

where $\mathbf{b}_{\mathrm{true}}$ is the true value of the bias vector $\mathbf{b}$, $p(\mathbf{Y} \mid \mathbf{b})$ is the likelihood function of $\mathbf{b}$, $\lambda = -\ln p(\mathbf{Y} \mid \mathbf{b})$ and ∇ is gradient operator. From (2.57), one has

$$p(Y \mid \mathbf{b}) = \frac{1}{(2\pi)^K \sqrt{|\mathcal{R}|}} \exp\left\{ -\frac{1}{2} \left[\mathbf{Y} - \mathbf{gb}\right]^{\mathrm{T}} \mathcal{R}^{-1} \left[\mathbf{Y} - \mathbf{gb}\right] \right\} \tag{2.64}$$

and therefore

$$\lambda = \mathrm{Const.} + \frac{1}{2} \left[\mathbf{Y} - \mathbf{gb}\right]^{\mathrm{T}} \mathcal{R}^{-1} \left[\mathbf{Y} - \mathbf{gb}\right] \tag{2.65}$$

Using the results from [14] to simplify the differentiations, $\nabla_{\mathbf{b}}\lambda$ can be written as

$$\nabla_{\mathbf{b}}\lambda = \mathbf{g}^{\mathrm{T}}\mathcal{R}^{-1}\left(\mathbf{Y} - \mathbf{g}\mathbf{b}\right) \tag{2.66}$$

which yields

$$\mathbf{J} = \mathbf{g}^{\mathrm{T}}\mathcal{R}^{-1}\mathbf{g} \tag{2.67}$$

Finally, when calculating the FIM, CRLB of desired elements (here the diagonal) will be

$$\mathrm{CRLB}\left\{[\mathbf{b}]_i\right\} = \left[\mathbf{J}^{-1}\right]_{ii} \tag{2.68}$$

for $i = 1, 2, 3, 4$.

2.6.2 Multitarget–multisensor CRLB

When number of the targets is greater than two, the same procedure as the proposed bias estimation algorithm to calculate the CRLB can be used. In this case, one should fuse all the measurements, except for sensor "i" to create a single pseudo-measurement for this "combined" sensor and then treat the problem as a two–sensor problem. Starting with the calculation of the combined measurement and covariance matrix of the set S_1 as in [5] yields

$$Z_{\mathrm{comb}} \;=\; \mathcal{R}_{\mathrm{comb}}\left(\sum_{j\in\{S\}\backslash i}\left(\mathcal{R}_j\right)^{-1} z_j\right) \tag{2.69}$$

and

$$\mathcal{R}_{\text{comb}} = \left(\sum_{j \in \{S\} \setminus i} \mathcal{R}_j^{-1} \right)^{-1} \tag{2.70}$$

As in the bias estimation algorithm, it is assumed that only sensor "i" has bias. This means that in the calculation of CRLB one should have access to the measurements both with and without bias. The next is to calculate $B_{\text{comb}}(k)$. Similar to the bias estimation algorithm, one should define $H(k) = -B_{\text{comb}}(k)$ and use the combined covariance and measurement matrices as a bias free measurement to find the CRLB of the bias estimation for sensor "i". Finally, $\mathcal{R}(k)$ can be calculated as

$$\mathcal{R}(k) = \mathcal{R}_{\text{comb}}(k) + \mathcal{R}_i(k) \tag{2.71}$$

By using (2.69), (2.70) and (2.71), the formulas for two sensors and multiple targets can be modified to calculate the CRLB for the case of multisensor–multitarget scenario.

2.7 Simulation results

2.7.1 Motion dynamics and measurement parameters

Here, a distributed tracking scenario with five sensors and sixteen targets is considered as shown in Figure 2.6. It is assumed that all sensors are synchronized. Without loss of generality and to easily compare the results of bias estimation for different sensors,

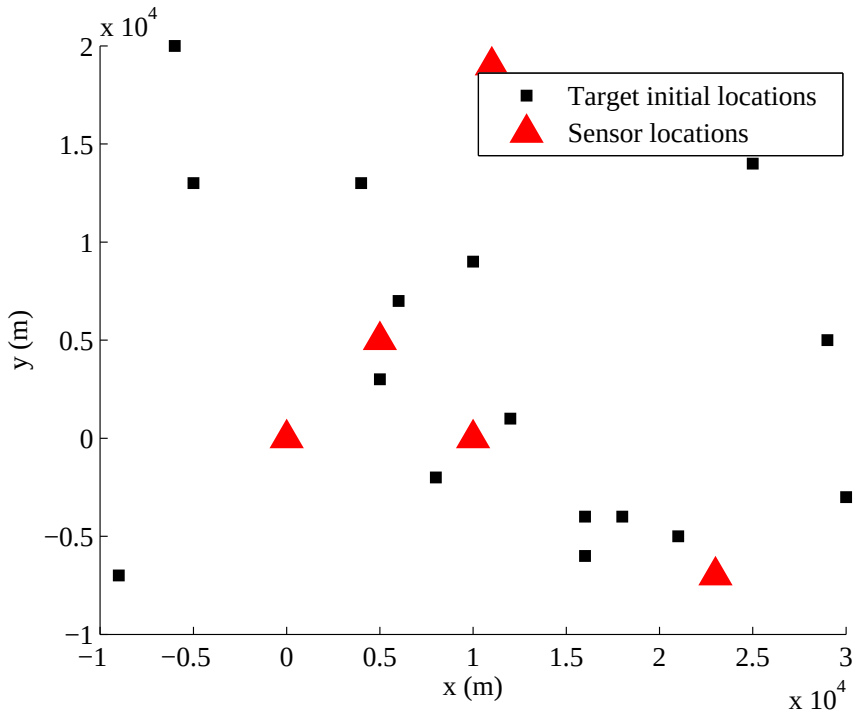

Figure 2.6: Initial locations of the targets and sensors

all the biases are assumed as

$$\beta_s = [\ 20\text{m} \quad 1\text{mrad}\ ]^{\mathrm{T}} \tag{2.72}$$

The standard deviations of measurement noises are $\sigma_r = 10$ m and $\sigma_\theta = 1$ mrad for target range and azimuth measurements, respectively. In this problem $r \simeq 20000$. According to [4], $\frac{r\sigma_\theta^2}{\sigma_r} = \frac{2\times10^4\times10^{-6}}{10} = 2 \times 10^{-3} \ll 0.4$ which is the threshold for polar to Cartesian conversion to be unbiased. Note that this condition holds for $r \simeq 200$km as well.

Scaling biases can be handled by the proposed method as well but ignored for simplicity. The true dynamics of the targets are modeled using the Discretized Continuous White Noise Acceleration (DCWNA) or nearly constant velocity (NCV) model with $q_x = q_y = 0.1\frac{\mathrm{m}^2}{\mathrm{s}^3}$ and constant turn rate model with rate $\omega = 0.1\frac{\mathrm{deg}}{\mathrm{s}}$. As for the filtering in local trackers, DCWNA and Continuous Wiener Process Acceleration (CWPA) are used with various settings to be able to create different scenarios for the simulation (for detail see [4, p. 268 and p. 467]).

2.7.2 A two-sensor problem

To compare the results of the new approach with those of the previously proposed algorithm [20, 19, 21], the case of two sensors (placed at $(0\text{m}, 0\text{m})$ and $(5000\text{m}, 0\text{m})$ in Cartesian coordinates) is considered with the same kinematics and filter settings as in [20]. The new algorithm that uses the reconstructed Kalman gains detailed in (2.35) and (2.36) is denoted as EXL while the previous algorithm in [20, 19, 21] is denoted as EX. Note that there is no fusion step in this case and the two methods

only differ in the Kalman gain calculation, which in the EXL [4] case is reconstructed approximately.

The settings of the variables are as follows. To handle the $L = 1$ and non–singularity problem, the "tracklet with decorrelated state estimate" [5] that can be used with only one measurement in the tracklet interval [11] must be selected instead of "tracklet computed using inverse Kalman filter" . The sampling intervals are $T = 1$s. The lag between each update at the fusion center is $L = k - k' = 1$. The initial state estimates are the converted measurements from polar coordinate to Cartesian coordinate with zero velocity and covariance matrix [20, 19, 21]

$$P_s(0 \mid 0) = \text{diag}\left[\begin{array}{cccc} (200\text{m})^2 & (20\text{m/s}^2)^2 & (200\text{m})^2 & (20\text{m/s}^2)^2 \end{array}\right] \tag{2.73}$$

Finally, the initial bias parameter estimate of all the sensors are zero with initial bias covariance

$$\Sigma_s(0 \mid 0) = \text{diag}\left[\begin{array}{cc} (20\text{m})^2 & (1\text{mrad})^2 \end{array}\right] \qquad s = 1, 2 \tag{2.74}$$

In the simulations, 100 Monte Carlo runs are used over 20 frames. The results of the Root Mean Squared Error (RMSE) in logarithmic scale for offset bias estimates are shown in Figures 2.7 and 2.8. From Figures 2.7 and 2.8, it can be seen that the performance of the EXL method, which recovers the Kalman gain at the fusion center by taking advantage of tracklet calculation, is very close to the accuracy of the EX method. The small variations in the results are due to the fact that the logarithmic

[4] The EXL algorithm has the luxury of operating without the local Kalman gains.

[5] This is equivalent to the information matrix fusion method, which, for $L = 1$, is algebraically equivalent to the Kalman filter (see [5, eq. (8.4.1-14)]).

scale is used to show the convergence rates clearly.

The CPU time for one iteration of bias estimation for a single target is 4.84×10^{-4} s for the EX method and 6.02×10^{-4} s for the EXL method, which represents a 20% increase in CPU time for the Kalman gain reconstruction. The CPU time over all iterations and targets for bias estimation with the above methods are 0.1526 and 0.1877 s, respectively. All simulations are done on a computer with Intel® Core™ i7-3720Qm 2.60GHz processor and 8GB RAM.

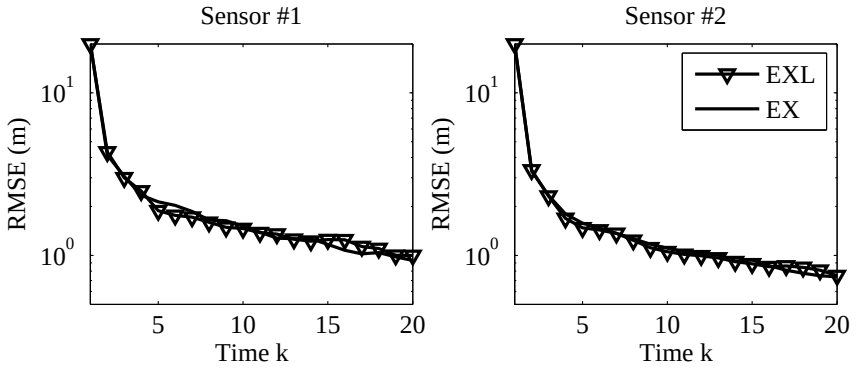

Figure 2.7: RMSE of the bias parameter b_r for sensors 1 and 2 in logarithmic scale (comparison between the previous (EX) and the proposed (EXL) algorithm)

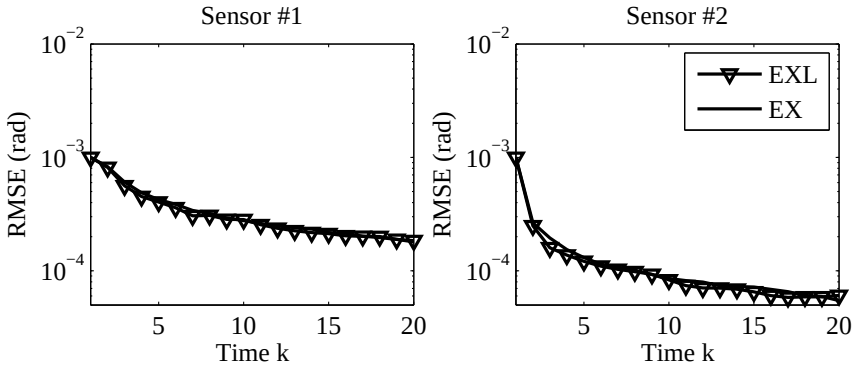

Figure 2.8: RMSE of the bias parameter b_θ for sensors 1 and 2 in logarithmic scale (comparison between the previous (EX) and the proposed (EXL) algorithm))

2.7.3 A five–sensor problem with nearly constant velocity (NCV) Kalman filter as local tracker

In the second case, a scenario with five sensors with the same sixteen targets, $L = k - k' = 10$ and $k = 1, ..., 100$ is simulated. The motivation for using 100 time steps is to have enough update steps to demonstrate the convergence results in terms of RMSE. Here the Fused Bias Estimation Algorithm (FBEA, Figure 2.4) is used to estimate the biases at the fusion center. The results of this simulation are shown in Figure 2.9.

From Figure 2.9, it can be seen that the proposed algorithm works well in a scenario with five sensors and with tracklet update at every $L = 10$ steps. Note that the FBEA is using only a two-dimensional state space for the bias estimation step for each sensor. For the same scenario, the previous EX algorithm would required a ten dimensional state space model. At its core, FBEA is a recursive least square (RLS) estimator. The computational complexity of RLS is of the order of $O(n^2)$, where n is number of parameters to be estimated. With $n = 10$ in the simulation, the computational complexity of the EX algorithm will be substantially higher.

To show the performance of the new algorithm in the fusion step, the results in terms of RMSE of the local track estimates for a specific sensor (sensor 1) and the fused estimates are presented in Figure 2.10 in logarithmic scale. To compare the results, the RMSE values of the local tracks and fused tracks with no biases are also included.

Figure 2.10 shows that the sequential fusion step used in the proposed bias estimation algorithm (FBEA) is a viable solution to the fusion problem. Clearly, the RMSE of the fused track with corrected measurements is between those of the local track

and the fused track of measurements with no biases. This observation indicates that the bias correction and fusion steps work well, which is another feature in the new algorithm, as this correction is done at the fusion center and not at the local trackers. In this case, there is no need for a feedback channel. This reduces the communication requirements.

In order to further evaluate the performance of the proposed algorithm, one can assume that all sensors have scaling and offset biases. Let the value of biases be

$$\beta_s = [\ 20\text{m} \quad 1\text{mrad} \quad 0.001 \quad 0.001\ ]^{\mathrm{T}} \tag{2.75}$$

The results in terms of the RMSE of the local track estimates for a specific sensor (sensor 1) and the fused estimates are shown in Figure 2.11 in logarithmic scale. Figure 2.11 shows that the proposed algorithm can handle offset and scaling biases at the same time and fuse the corrected tracks in order to achieve better estimates of the targets' state.

2.7.4 A five–sensor problem with a two–NCV IMM as local estimator

Previously, the noise levels of local tracker filters were assumed to be known. To demonstrate how well the proposed algorithm works when this information is not available, the IMM estimator is used in the next two examples as local tracker filters. To start with, an IMM estimator with two nearly constant velocity (NCV) or DCWNA Kalman filters with different noise intensities are used. The first filter uses $q_x = q_y = 10\frac{\mathrm{m}^2}{\mathrm{s}^3}$ while the second one uses $q_x = q_y = 2\frac{\mathrm{m}^2}{\mathrm{s}^3}$ as intensities in the east and north

directions, respectively. Note that the noise intensity parameters q_x and q_y have the same meaning as in [19], i.e., power spectral densities. To ensure accurate bias estimation, parameters of the RLSBE algorithm should be changed to handle the mismatch in the models between local trackers and fusion center which uses only an NCV model for data processing. Although there is no systematic way to select the intensities for the NCV model at the fusion center, q should be the value of the higher intensity in each coordinate. Figure 2.12 shows the RMSE results for the bias parameters that are estimated in the case of having NCV–NCV IMM estimators as local trackers and only one NCV model at the fusion center with inflated intensity level.

Figure 2.12 shows how well the bias estimation can be handled by EXL even when it is not possible to recover the exact Kalman gains that are used the at local trackers. The difference in convergence between the previous example (Figure 2.9) and Figure 2.12 is negligible, which shows the effectiveness of the new algorithm in different situations.

2.7.5 A five–sensor problem with nearly constant acceleration–nearly constant velocity (NCA–NCV) IMM as local estimators

Next, we demonstrate the effectiveness of the new algorithm in recovering the Kalman gain and estimating the biases and show how well the new algorithm works in the case of mismatch in the models at the fusion center and local trackers. In this example it is assumed that local trackers are using an IMM estimator with one nearly constant acceleration (NCA) and one NCV Kalman filter with $q_x = q_y = 10\frac{\mathrm{m}^2}{\mathrm{s}^3}$ for the NCA

model and $q_x = q_y = 2\frac{\mathrm{m}^2}{\mathrm{s}^3}$ for the NCV model. The issue in this case is that the local trackers send only the combined output from the IMM estimator without any information about the acceleration. Figure 2.13 shows the results on the scenario of this subsection with $q_x = q_y = 200\frac{\mathrm{m}^2}{\mathrm{s}^3}$ at the fusion center.

Simulation results show convergence as in the previous examples (Figures 2.9 and 2.12). This demonstrates the robustness of the new algorithm even in the presence of uncertainty about the local trackers.

2.7.6 Lower bound and convergence results

The calculation of the CRLB was discussed in Section 2.6. Three different examples are used to demonstrate the performance of the proposed algorithm with respect to the CRLB. The first example is the scenario implemented in Subsection 2.7.3. The comparison is between the square root of the diagonal elements of the CRLB ($\sqrt{\mathrm{CRLB}\left\{[\mathbf{b}]_i\right\}}$) , the square root of the diagonal elements of bias estimation covariance matrix ($\sqrt{\Sigma_{ii}}$) and the RMSE of the estimated biases.

Figure 2.14 shows that both $\sqrt{\Sigma_{ii}}$ and the RMSE follow the $\sqrt{\mathrm{CRLB}\left\{[\mathbf{b}]_i\right\}}$. The results for the examples in Subsections 2.7.4 and 2.7.5 are shown in Figure 2.15 and Figure 2.16, respectively. These are approximately within the 95% probability region around the CRLB [6]. Once again, the figures show that the estimation errors follow the CRLB.

2.7.7 Consistency of bias estimation algorithms

In this section, a brief analysis of consistency of the proposed bias estimation algorithms is given. The analysis is based on Normalized Estimation Error Squared

Table 2.1: Offset bias $\mathbf{b}_{r_1}$related RMSE, $\sqrt{\text{CRLB}\{[\mathbf{b}]_i\}}$ and $\sqrt{\Sigma_{ii}}$ at final time-step for three different local filters

	Kalman filter	NCV–NCV	NCA–NCV
RMSE	0.7621	1.002	1.105
$\sqrt{\Sigma_{ii}}$	1.018	1.025	1.029
$\sqrt{\text{CRLB}\{[\mathbf{b}]_i\}}$	0.8795	0.9122	0.9350
Upper 95% confidence interval	1.055	1.106	1.220
Lower 95% confidence interval	0.7036	0.7666	0.7481

(NEES) [4]. First, the results for EXL algorithm are shown in Figure 2.17 for 100 Monte–Carlo runs. The bounds are for the 95% probability interval which shows that the EXL algorithm is consistent at each time step.

To further examine the consistency of the proposed algorithm, the NEES for FBEA are computed and shown in Figure 2.18 for three different types of local estimators, i.e., Kalman filter, NCV–NCV IMM and NCA–NCV IMM for 100 Monte–Carlo runs. Here we used one–sided 95% probability interval. The results show that FBEA is a pessimistic filtering approach. This is mostly due to the fact that the use of the pseudo–measurement in a Kalman filter fashion is an approximation because its error and the state prediction error at the fusion center are correlated because of the common process noise.

Finally, in Tables 2.1 and 2.2 the RMSE, $\sqrt{\text{CRLB}\{[\mathbf{b}]_i\}}$ and $\sqrt{\Sigma_{ii}}$ are compared for both offset bias parameters at their last update for sensor 1. As can be seen from Tables 2.1 and 2.2, both RMSE and $\sqrt{\Sigma_{ii}}$ are within the 95% confidence region of $\sqrt{\text{CRLB}\{[\mathbf{b}]_i\}}$, which indicates that the parameter estimates are unbiased.

Table 2.2: Offset bias $\mathbf{b}_{\theta_1}$related RMSE, $\sqrt{\text{CRLB}\left\{[\mathbf{b}]_i\right\}}$ and $\sqrt{\Sigma_{ii}}$ at final time-step for three different local filters

	Kalman filter	NCV–NCV	NCA–NCV
RMSE	9.266×10^{-5}	7.950×10^{-5}	9.197×10^{-5}
$\sqrt{\Sigma_{ii}}$	9.322×10^{-5}	9.434×10^{-5}	9.198×10^{-5}
$\sqrt{\text{CRLB}\left\{[\mathbf{b}]_i\right\}}$	9.826×10^{-5}	9.714×10^{-5}	10.090×10^{-5}
Upper 95% confidence interval	11.79×10^{-5}	11.66×10^{-5}	12.10×10^{-5}
Lower 95% confidence interval	7.861×10^{-5}	7.771×10^{-5}	8.069×10^{-5}

2.8 Conclusions

In this paper, a new bias estimation algorithm that works with only the state estimates and their associated covariance matrices from synchronized local trackers at varying reporting rates was presented. The algorithm does not require the stacking of the bias vectors of all the sensors together, which is a problem for previous algorithms with a large number of sensors in the surveillance area. Also, the new algorithm works without the local filter gains, which are not available at the fusion center in practical systems. In addition, it gives a solution to the problem of joint fusion and bias estimation. The results from simulations show that the algorithm works reliably in different scenarios with various numbers of sensors. Furthermore, this algorithm can work with low data rates between the sensors and the data fusion center. Finally, the CRLB for multisensor–multitarget scenarios with bias estimation was presented and the RMSE results matched well with the CRLB. This demonstrates the statistical efficiency and the versatility of the new algorithm.

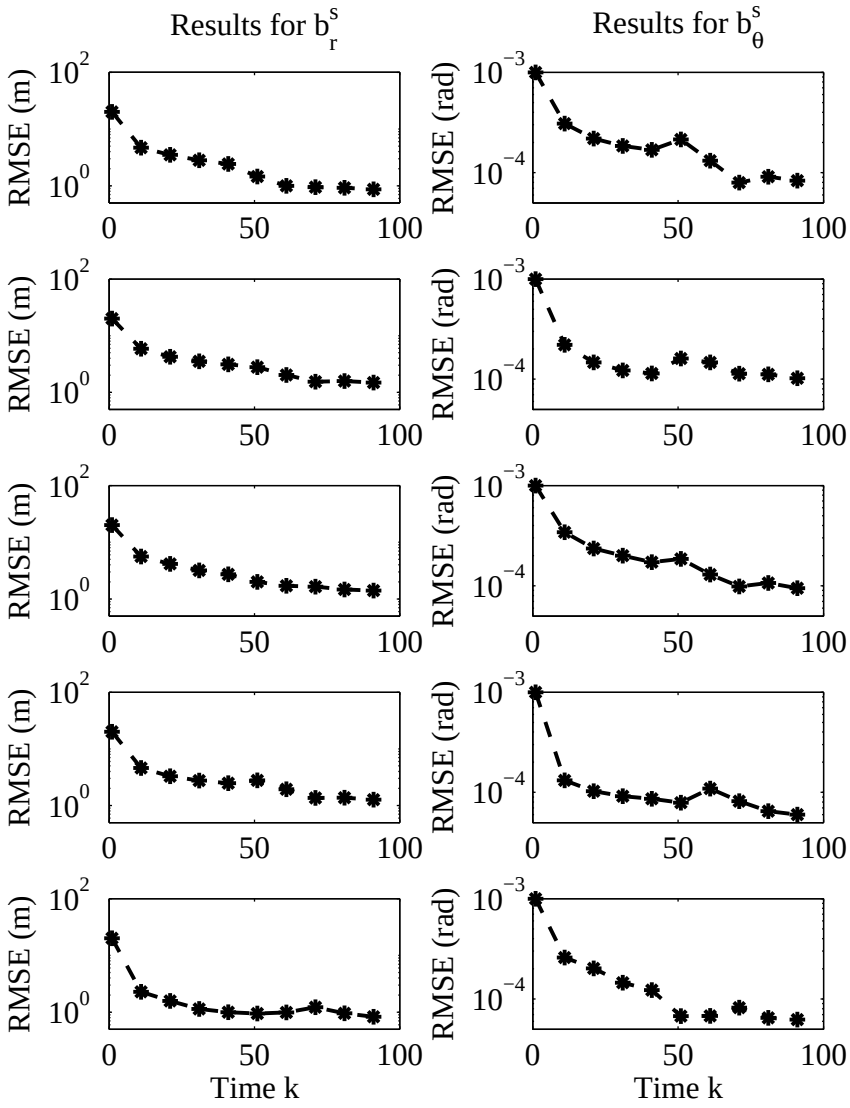

Figure 2.9: RMSE of the bias parameter b_r (left column) and b_θ (right column) for all 5 sensors from sensor 1 (top) to sensor 5 (bottom) in logarithmic scale. Note that residual bias RMSE is an order of magnitude below the measurement noise standard deviations, i.e., it becomes negligible.

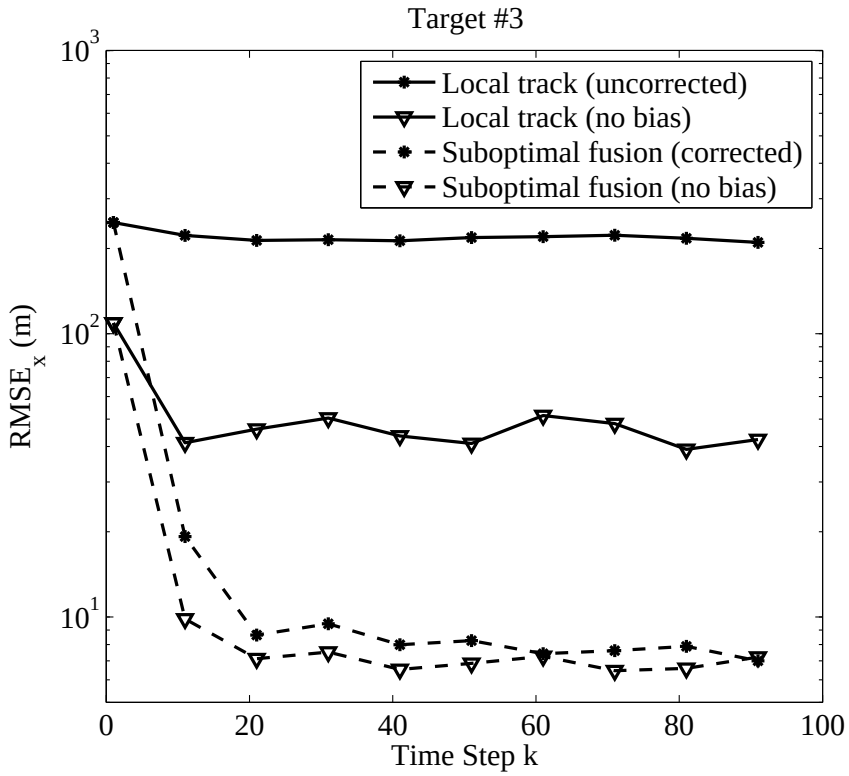

Figure 2.10: RMSE of local track (sensor 1) and the output of the fusion algorithm including offset biases for all sensors in logarithmic scale

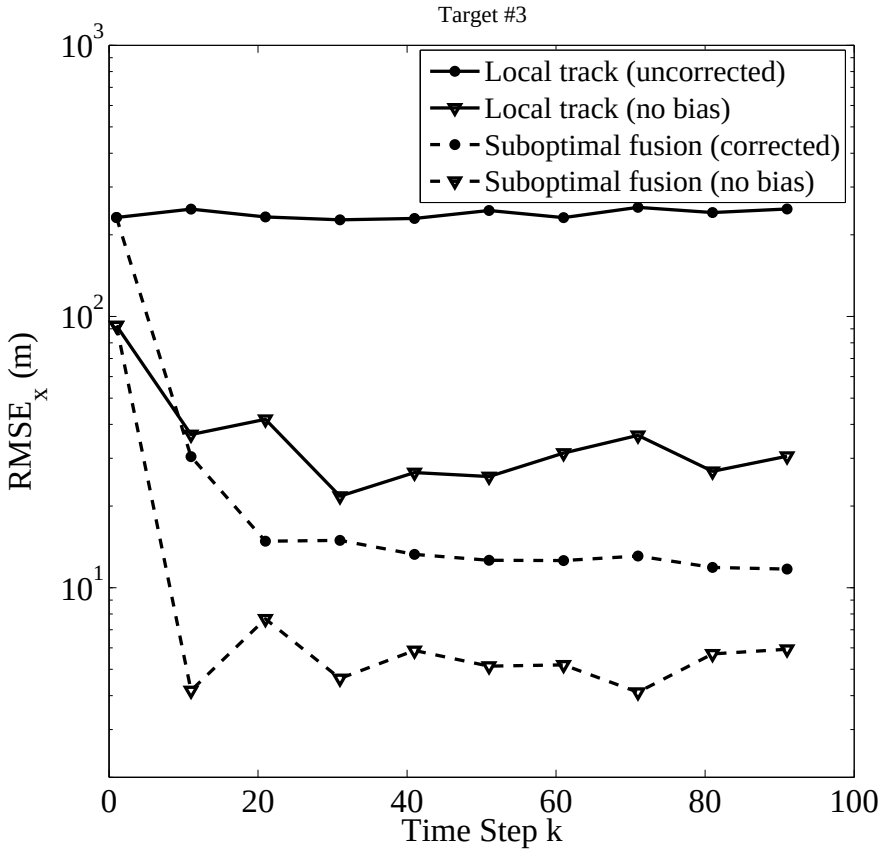

Figure 2.11: RMSE of local track (sensor 1) and the output of the fusion algorithm including scaling and offset biases for all sensors in logarithmic scale

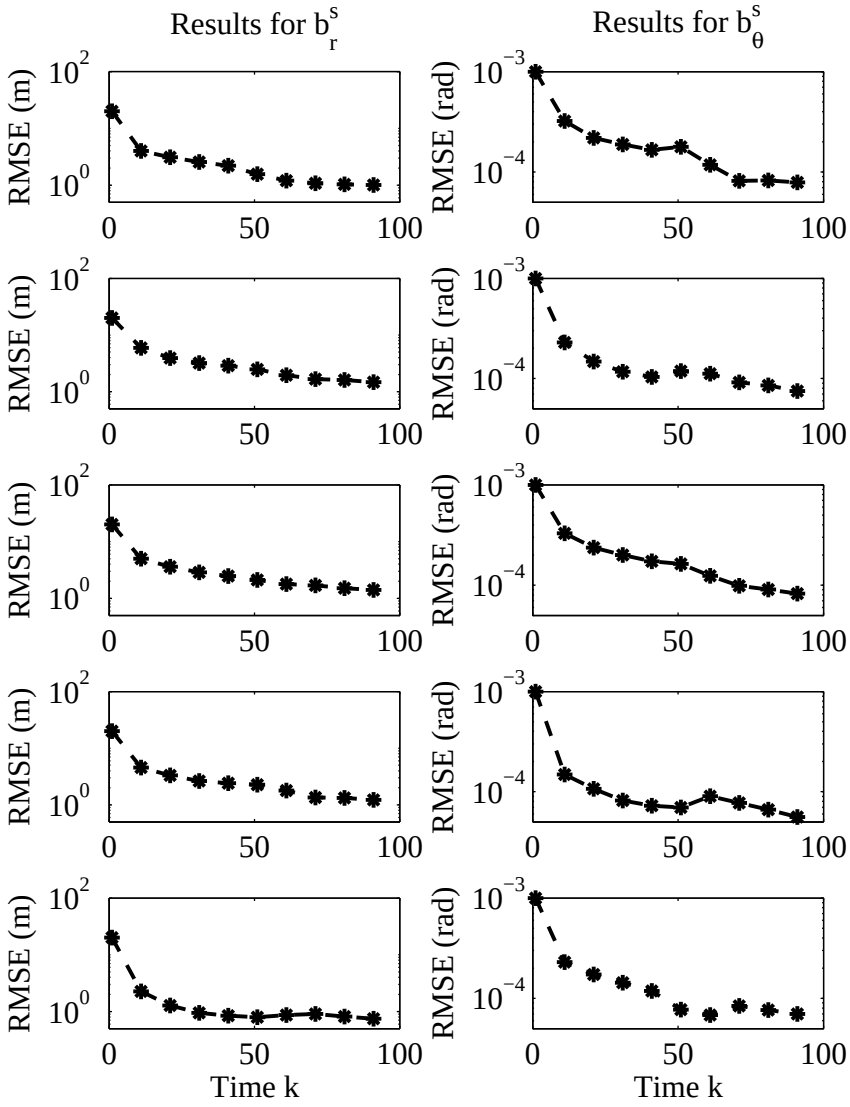

Figure 2.12: RMSE of the bias parameter b_r (left column) and b_θ (right column) for all 5 sensors from sensor 1 (top) to sensor 5 (bottom) in logarithmic scale. The local trackers use IMM estimators with NCV–NCV models.

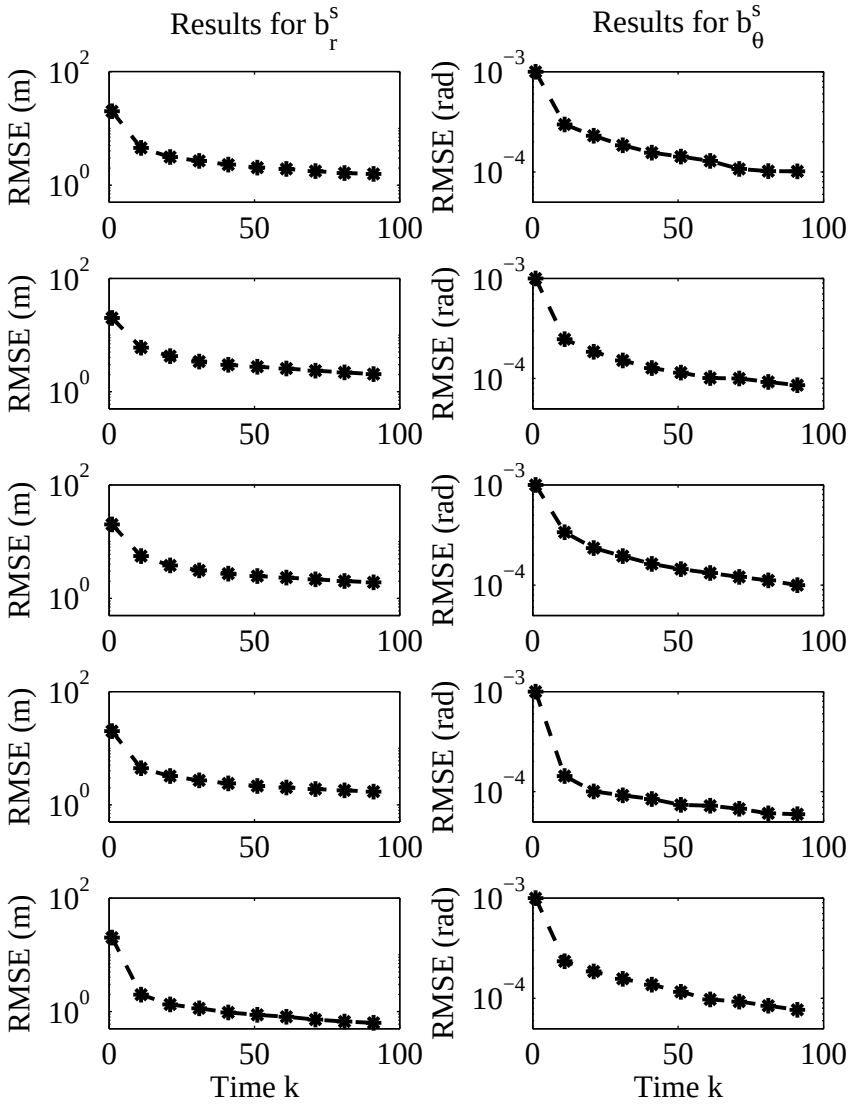

Figure 2.13: RMSE of the bias parameter b_r (left column) and b_θ (right column) for all 5 sensors from sensor 1 (top) to sensor 5 (bottom) in logarithmic scale. The local trackers use IMM estimators with NCA–NCV models.

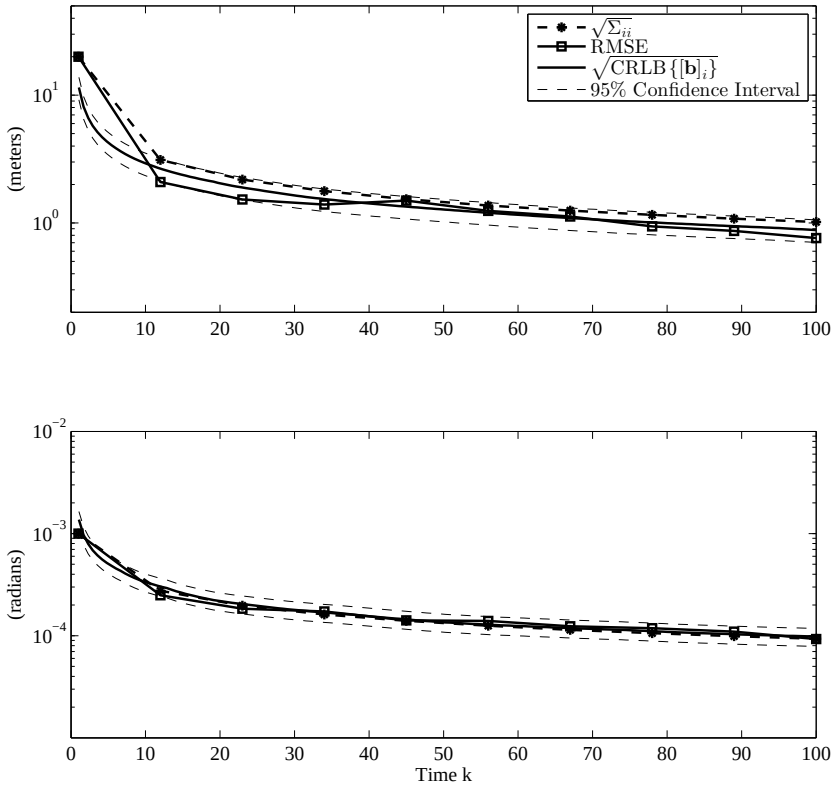

Figure 2.14: Comparison between the square root of diagonal elements of CRLB ($\sqrt{\text{CRLB}\{[\mathbf{b}]_i\}}$), square root of diagonal elements of the covariance matrix of bias estimation algorithm ($\sqrt{\Sigma_{ii}}$) and RMSE of the bias estimation for the case of 5 sensors with Kalman filter as local trackers (only the results for the first sensor are shown).

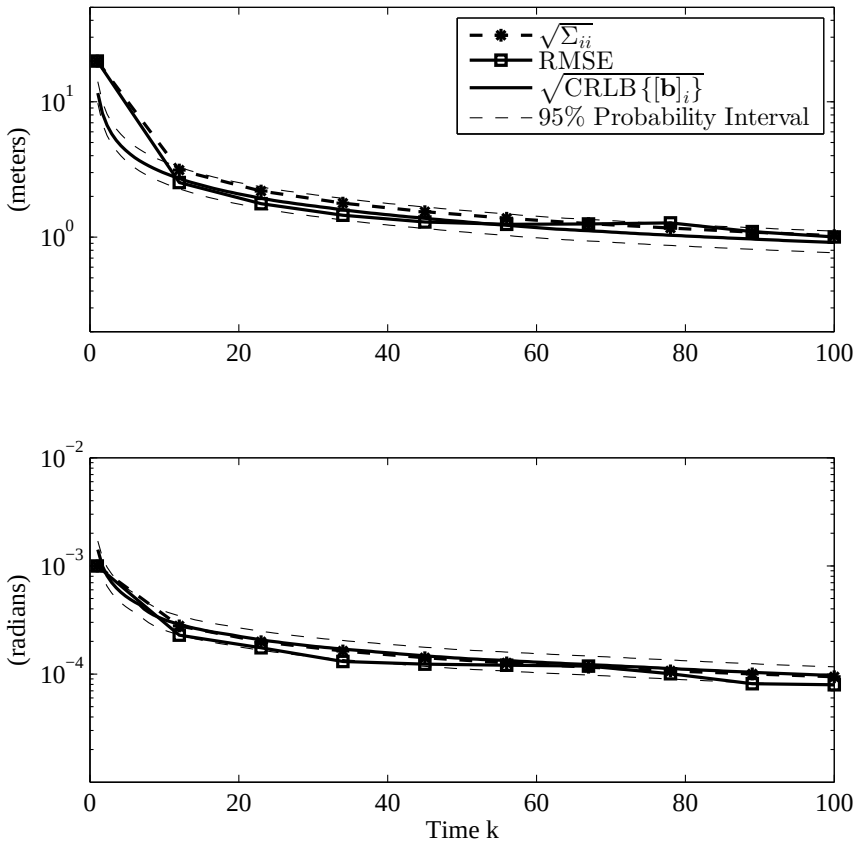

Figure 2.15: Comparison between the square root of diagonal elements of CRLB ($\sqrt{\text{CRLB}\{[\mathbf{b}]_i\}}$), square root of diagonal elements of the covariance matrix of bias estimation algorithm ($\sqrt{\Sigma_{ii}}$) and RMSE of the bias estimation for the case of 5 sensors with NCV–NCV IMM estimator as local trackers (only the results for the first sensor are shown).

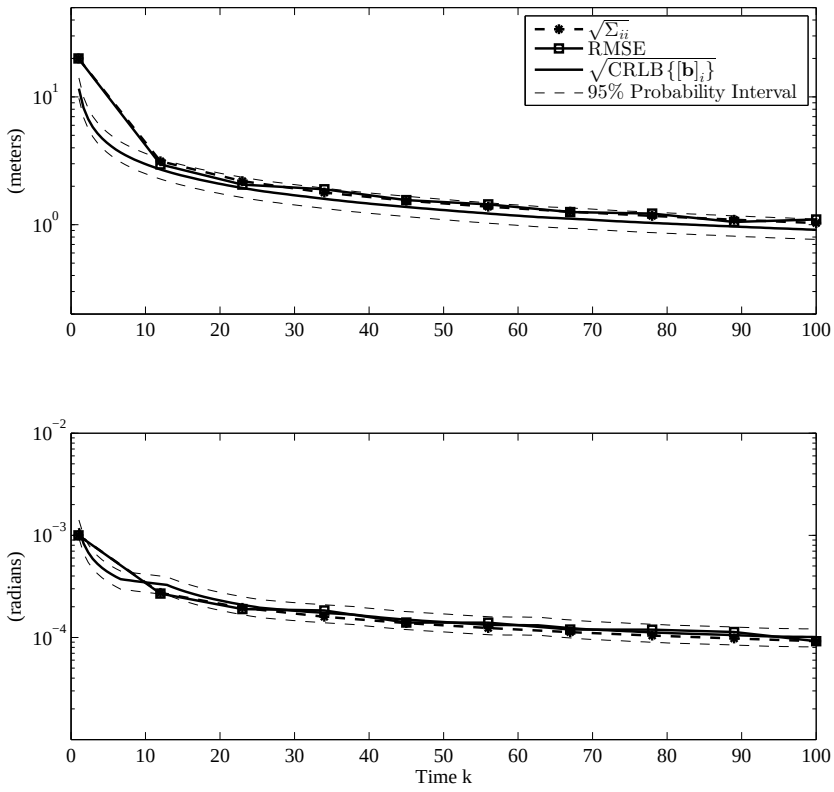

Figure 2.16: Comparison between the square root of diagonal elements of CRLB ($\sqrt{\text{CRLB}\{[\mathbf{b}]_i\}}$), square root of diagonal elements of the covariance matrix of bias estimation algorithm ($\sqrt{\Sigma_{ii}}$) and RMSE of the bias estimation for the case of 5 sensors with NCA–NCV IMM estimator as local trackers (only the results for the first sensor are shown).

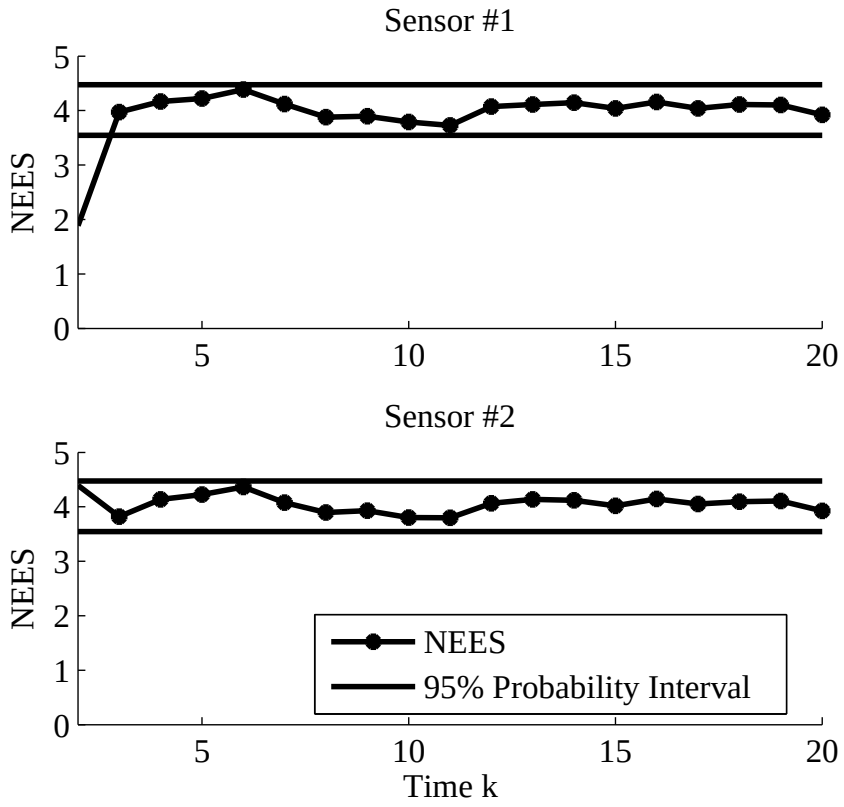

Figure 2.17: NEES for EXL algorithm with Kalman gain recovery instead of using original Kalman gains.

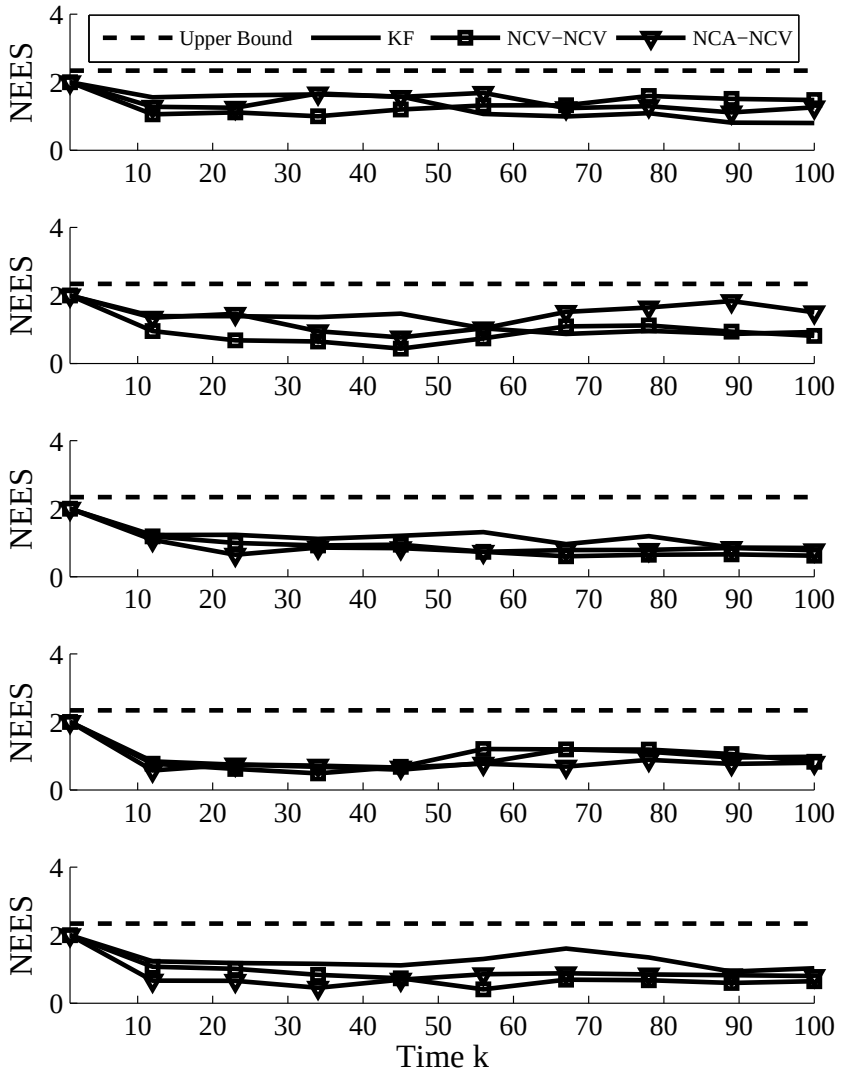

Figure 2.18: NEES for FBEA and three different local tracker estimators (Kalman filter, NCV–NCV IMM and NCA–NCV IMM) for sensor 1 (top) to sensor 5 (bottom) compared to the upper–bound of 95% probability interval.

Appendix A

Derivation of the equivalent measurement covariance (2.34)

Using the properties

$$\mathbb{E}\left[\tilde{\mathbf{u}}_s\left(k,k'\right) \mid \mathbf{Z}_s^{k'}\right] = 0 \tag{A.76}$$

$$\mathbb{E}\left[\tilde{\mathbf{u}}_s\left(k,k'\right) \mid \mathbf{Z}_s^{k}\right] \neq 0 \tag{A.77}$$

and

$$\begin{aligned}
\mathbb{E}\left[\tilde{\mathbf{x}}\left(k \mid k\right)\tilde{\mathbf{x}}\left(k \mid k'\right)^{\mathrm{T}} \mid \mathbf{Z}_s^{k'}\right] &= \mathbb{E}\left[\mathbb{E}\left[\left[\mathbf{x}_s(k) - \hat{\mathbf{x}}_s\left(k \mid k\right)\right]\right.\right. \\
&\quad \left.\left.\left[\mathbf{x}_s(k) - \hat{\mathbf{x}}_s\left(k \mid k\right) + \hat{\mathbf{x}}_s\left(k \mid k\right) - \hat{\mathbf{x}}_s\left(k \mid k'\right)\right]^{\mathrm{T}} \mid \mathbf{Z}_s^{k}\right] \mid \mathbf{Z}_s^{k'}\right] \qquad \text{(A.78)} \\
&= \mathbb{E}\left[\mathbb{E}\left[\left[\mathbf{x}_s(k) - \hat{\mathbf{x}}_s\left(k \mid k\right)\right]\left[\mathbf{x}_s(k) - \hat{\mathbf{x}}_s\left(k \mid k\right)\right]^{\mathrm{T}} \mid \mathbf{Z}_s^{k}\right] \mid \mathbf{Z}_s^{k'}\right] \\
&\quad + \mathbb{E}\left[\underbrace{\mathbb{E}\left[\left[\mathbf{x}_s(k) - \hat{\mathbf{x}}_s\left(k \mid k\right)\right] \mid \mathbf{Z}_s^{k}\right]}_{=0}\left[\hat{\mathbf{x}}_s\left(k \mid k\right) - \hat{\mathbf{x}}_s\left(k \mid k'\right)\right]^{\mathrm{T}} \mid \mathbf{Z}_s^{k'}\right] \\
&= \mathbf{P}_s\left(k \mid k\right) \qquad \text{(A.79)}
\end{aligned}$$

$\mathbf{U}_s\left(k,k'\right)$ can be expanded as

$$\begin{aligned}
\mathbf{U}_s\left(k,k'\right) &= \mathbb{E}\left[\tilde{\mathbf{u}}_s\left(k,k'\right)\tilde{\mathbf{u}}_s\left(k,k'\right)^{\mathrm{T}} \mid \mathbf{Z}_s^{k'}\right] \\
&= \mathbf{A}_s\mathbf{P}_s\left(k \mid k\right)\mathbf{A}_s^{\mathrm{T}} + \left[\mathbf{A}_s - I\right]\mathbf{P}_s\left(k \mid k'\right)\left[\mathbf{A}_s - I\right]^{\mathrm{T}} \\
&\quad -\mathbf{A}_s\mathbf{P}_s\left(k \mid k\right)\left[\mathbf{A}_s - I\right]^{\mathrm{T}} - \left[\mathbf{A}_s - I\right]\mathbf{P}_s\left(k \mid k\right)\mathbf{A}_s \\
&= \left[\mathbf{A}_s - I\right]\mathbf{P}_s\left(k \mid k'\right)\left[\mathbf{A}_s - I\right]^{\mathrm{T}} - \mathbf{A}_s\mathbf{P}_s\left(k \mid k\right)\mathbf{A}_s^{\mathrm{T}} + \mathbf{A}_s\mathbf{P}_s\left(k \mid k\right) \\
&\quad +\mathbf{P}_s\left(k \mid k\right)\mathbf{A}_s^{\mathrm{T}}
\end{aligned} \tag{A.80}$$

where the arguments of $\mathbf{A}_s(k,k')$ are dropped for clarity. By using the property

$$\begin{aligned}
\left[\mathbf{A}_s - I\right]\mathbf{P}_s\left(k \mid k'\right) &= \left[\mathbf{P}_s\left(k \mid k'\right)\left[\mathbf{P}_s\left(k \mid k'\right) - \mathbf{P}_s\left(k \mid k\right)\right]^{-1} - I\right]\mathbf{P}_s\left(k \mid k'\right) \\
&= \mathbf{P}_s\left(k \mid k'\right)\left[\left[\mathbf{P}_s\left(k \mid k'\right) - \mathbf{P}_s\left(k \mid k\right)\right]^{-1} - \mathbf{P}_s\left(k \mid k'\right)^{-1}\right]\mathbf{P}_s\left(k \mid k'\right) \\
&= \mathbf{P}_s\left(k \mid k'\right)\left[\mathbf{P}_s\left(k \mid k'\right) - \mathbf{P}_s\left(k \mid k\right)\right]^{-1} \\
&\quad \left[I - \left[\mathbf{P}_s\left(k \mid k'\right) - \mathbf{P}_s\left(k \mid k\right)\right]\mathbf{P}_s\left(k \mid k'\right)^{-1}\right]\mathbf{P}_s\left(k \mid k'\right) \\
&= \mathbf{A}_s\left[I - I + \mathbf{P}_s\left(k \mid k\right)\mathbf{P}_s\left(k \mid k'\right)^{-1}\right]\mathbf{P}_s\left(k \mid k'\right) \\
&= \mathbf{A}_s\mathbf{P}_s\left(k \mid k\right)
\end{aligned} \tag{A.81}$$

$\mathbf{U}_s\left(k,k'\right)$ can be further simplified as

$$\begin{aligned}
\mathbf{U}_s\left(k,k'\right) &= \mathbf{A}_s\mathbf{P}_s\left(k \mid k\right)\left[\mathbf{A}_s - I\right]^{\mathrm{T}} - \mathbf{A}_s\mathbf{P}_s\left(k \mid k\right)\mathbf{A}_s^{\mathrm{T}} + \mathbf{A}_s\mathbf{P}_s\left(k \mid k\right) + \mathbf{P}_s\left(k \mid k\right)\mathbf{A}_s^{\mathrm{T}} \\
&= \mathbf{A}_s\mathbf{P}_s\left(k \mid k\right)\mathbf{A}_s^{\mathrm{T}} - \mathbf{A}_s\mathbf{P}_s\left(k \mid k\right) - \mathbf{A}_s\mathbf{P}_s\left(k \mid k\right)\mathbf{A}_s^{\mathrm{T}} + \mathbf{A}_s\mathbf{P}_s\left(k \mid k\right) \\
&\quad +\mathbf{P}_s\left(k \mid k\right)\mathbf{A}_s^{\mathrm{T}} \\
&= \mathbf{P}_s\left(k \mid k\right)\mathbf{A}_s^{\mathrm{T}}
\end{aligned} \tag{A.82}$$

which yields (2.34). Note that (A.81) and (A.82) are transpose of each other, but, since $\mathbf{U}_s(k, k')$ is symmetric, they are equal to each other.

Bibliography

[1] T. Asparouhov and B. Muthén. "Weighted least squares estimation with missing data". *Available at: http://www.statmodel.com/download/ GstrucMissingRevision.pdf*, 2010.

[2] A. Balleri, A. Nehorai, and J. Wang. "Maximum likelihood estimation for compound-Gaussian clutter with inverse gamma texture. *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 2, pp. 775–779, 2007.

[3] Y. Bar-Shalom and H. Chen. "Multisensor track-to-track association for tracks with dependent errors. *Journal of Advances in Information Fusion*, 1(1), July 2006.

[4] Y. Bar-Shalom, X.R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software.* Wiley, New York, NY, 2001.

[5] Y. Bar-Shalom, P.K. Willett, and X. Tian. *Tracking and Data Fusion: A Handbook of Algorithms.* YBS Publishing, Storr, CT, 2011.

[6] D. Belfadel, R. W. Osborne, and Y. Bar-Shalom. "Bias estimation for optical sensor measurements with targets of opportunity". *International Conference on Information Fusion*, pp. 1805–1812, Istanbul, Turkey, July 2013.

[7] H. Chen and X. R. Li. "On track fusion with communication constraints". *International Conference on Information Fusion*, Quebec, QC, Canada, July 2007.

[8] M. P. Dana. "Registration: A prerequisite for multiple sensor tracking". *Multitarget-Multisensor Tracking: Advanced Applications*, vol. 1, pp. 155–185, Artech House, Norwood, MA, 1990.

[9] O. E. Drummond. "A hybrid sensor fusion algorithm architecture and tracklets". *Proc. SPIE, Signal and Data Processing of Small Targets*, vol. 3163, 1997.

[10] O. E. Drummond. "Tracklets and a hybrid fusion with process noise". *Proc. SPIE, Signal and Data Processing of Small Targets*, vol. 3163, pp. 512–524, 1997.

[11] O. E. Drummond. "Track and tracklet fusion filtering". *Proc. SPIE, Signal and Data Processing of Small Targets*, vol. 4728, pp. 176–195, 2002.

[12] J. Durbin and S. J. Koopman. *Time Series Analysis by State–Space Methods*. no. 38, Oxford University Press, Oxford, UK, 2012.

[13] B. Friedland. "Treatment of bias in recursive filtering". *IEEE Transactions on Automatic Control*, vol. 14, no. 4, pp. 359–367, 1969.

[14] F. A. Graybill. *Theory and Applications of The Linear Model*. Duxbury Press, Belmont, CA, 1976.

[15] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge university press, Cambridge, UK, 2012.

[16] D. Huang, H. Leung, and E. Bosse. "A pseudo-measurement approach to simultaneous registration and track fusion". *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 2315–2331, 2012.

[17] K. Kastella, B. Yeary, T. Zadra, R. Brouillard, and E. Frangione. "Bias modeling and estimation for GMTI applications". *International Conference on Information Fusion*, pp. TUC1–7, Paris, France, July 2000.

[18] I. Li and J. Georganas. "Multi-target multi-platform sensor registration in geodetic coordinates". *Proc. International Conference on Information Fusion*, vol. 1, pp. 366–373, Annapolis, MD, July 2002.

[19] X. Lin, Y. Bar-Shalom, and T. Kirubarajan. "Multisensor-multitarget bias estimation of asynchronous sensors". *Proc. SPIE*, vol. 5429, pp. 105–116, 2004.

[20] X. Lin, Y. Bar-Shalom, and T. Kirubarajan. "Exact multisensor dynamic bias estimation with local tracks". *IEEE Transactions on Aerospace and Electronic Systems*, vol. 1, no. 40, pp. 576–590, 2004.

[21] X. Lin, Y. Bar-Shalom, and T. Kirubarajan. "Multisensor multitarget bias estimation for general asynchronous sensors". *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 3, pp. 899–921, 2005.

[22] M. Longbin, S. Xiaoquan, Z. Yiyu, S. Z. Kang, and Y. Bar-Shalom. "Unbiased converted measurements for tracking". *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 3, pp. 1023–1027, 1998.

[23] N. Nabaa and R. H. Bishop. "Solution to a multisensor tracking problem with

sensor registration errors". *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 1, pp. 354–363, 1999.

[24] N. Okello and S. Challa. "Joint sensor registration and track-to-track fusion for distributed trackers". *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 3, pp. 808–823, 2004.

[25] N. Okello and B. Ristic. "Maximum likelihood registration for multiple dissimilar sensors". *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 3, pp. 1074–1083, 2003.

[26] D. J. Papageorgiou and J. D Sergi. "Simultaneous track-to-track association and bias removal using multistart local search". *Proc. IEEE Aerospace Conference*, BigSky, MT, March 2008.

[27] H. Rue and L. Held. *Gaussian Markov Random Fields: Theory and Applications*. CRC Press, Boca Raton, FL, 2005.

[28] P. J. Shea, T. Zadra, D. Klamer, E. Frangione, R. Brouillard, and K. Kastella. "Precision tracking of ground targets". *Proc. IEEE Aerospace Conference*, pp. 473–482, BigSky, MT, March 2000.

[29] E. Taghavi, R. Tharmarasa, T. Kirubarajan, and Y. Bar-Shalom. "Bias estimation for practical distributed multiradar-multitarget tracking systems". *International Conference on Information Fusion*, pp. 1304–1311, Istanbul, Turkey, July 2013.

[30] B. A. van Doorn and H.A.P. Blom. "Systematic error estimation in multisensor

fusion systems". *Proc. SPIE Conference on Aerospace Sensing*, vol. 1954, pp. 450–461, 1993.

[31] G. Welch and G. Bishop. *An Introduction to The Kalman Filter*. University of North Carolina: Chapel Hill, North Carolina, 1995.

[32] M. Yeddanapudi, Y. Bar-Shalom, and K. Pattipati. "IMM estimation for multitarget-multisensor air traffic surveillance". *Proc. IEEE*, vol. 85, no. 1, pp. 80-96, 1997.

The following chapter is a reproduction of an IEEE copyrighted, accpeted for publication paper:

Ehsan Taghavi, R. Tharmarasa and T. Kirubarajan, Mike McDonald Multisensor–Multitarget Bearing–Only Sensor Registration, *IEEE Transactions on Aerospace and Electronics Systems*, To be appear in vol. 52, no. 4, August 2016.

Chapter 3

Multisensor–Multitarget Bearing–Only Sensor Registration

3.1 Abstract

Bearing–only estimation is one of the fundamental and challenging problems in target tracking. As in the case of radar tracking, the presence of offset or position biases can exacerbate the challenges in bearing–only estimation. Modeling various sensor biases is not a trivial task and not much has been done in the literature specifically for bearing–only tracking. This paper addresses the modeling of offset biases in bearing–only sensors and the ensuing multitarget tracking with bias compensation. Bias estimation is handled at the fusion node to which individual sensors report their local tracks in the form of associated measurement reports (AMR) or angle-only tracks. The modeling is based on a multisensor approach that can effectively handle a time–varying number of targets in the surveillance region. The proposed

algorithm leads to a maximum likelihood bias estimator. The corresponding Cramér–Rao Lower Bound to quantify the theoretical accuracy that can be achieved by the proposed method or any other algorithm is also derived. Finally, simulation results on different distributed tracking scenarios are presented to demonstrate the capabilities of the proposed approach. In order to show that the proposed method can work even with false alarms and missed detections, simulation results on a centralized tracking scenario where the local sensors send all their measurements (not AMRs or local tracks) are also presented.

3.2 Introduction

Multisensor–multitarget bearing–only tracking is a challenging problem with many applications [4, 5, 27, 13]. Some applications of bearing–only tracking are in maritime surveillance using sonobuoys, underwater target tracking using sonar and passive ground target tracking using Electronic Support Measures (ESM) or Infra–red Search and Track (IRST) sensors. In such applications, it is of interest to find the target position as well as any biases that may affect estimation performance. From the early works in [1, 2, 36] to the more recent works in [29] and the references therein, the focus has been only on tracking the targets based on measurements from a bearing–only sensor. However, due to the limitations of single sensor bearing–only tracking, i.e., due to the need for own–ship maneuvers for the observability of state parameters [28], the issue of biases in passive single sensor tracking has not been addressed in the literature. The main focus of this paper is multisensor bearing–only tracking in the presence of biases. In multisensor bearing-only tracking, observability is no longer a major issue. However, in the case of port–starboard ambiguity, the problem

of observability was discussed in detail in [10]. Besides, the presence of sensor biases that are often unaccounted for can degrade the estimation results significantly. Most of the works on bias estimation have been about radar tracking (see [42, 43, 20] and the references therein) or using other measurements besides bearing information [6, 7]. For example, when the elevation information is available, one can estimate the offset biases as in [7, 14].

With the objective of providing a combined bias estimation and target tracking algorithm that is both theoretically sound and practical, the problem of multisensor bearing-only multitarget tracking is considered in this paper. Having more than one passive sensor in the surveillance region ensures the observability of the state parameters, i.e., position and velocity of the target, without the need for maneuvers [33, 44, 35]. One of the issues that can complicate bearing–only tracking is the bias in the sensors. For example, in maritime surveillance using sonobuoys, which are usually dropped from an aircraft or thrown from a ship close to an area of interest, the exact locations of the sonobuoys are not known. This leads to position biases [25]. This is also an isuue in modern systems such as autonomous underwater vehicles (AUV) [9]. In addition, the impact with the water surface and the waves can result in systemic offset biases [6]. In wide area surveillance using airborne IRST sensors, uncertain platform motion can contribute to biases as well. Offset bias can be modeled as an additive constant term affecting the measurement equation and the sensor position uncertainty can be modeled using a random walk [31].

Negligible biases can be treated as residual errors. This residual error can be used in the form of additional uncertainty in the measurements later in the filtering step. However, if offset biases are larger than the noise standard deviation of the

bearing–only measurements, a mirror of the target's bearing is sent to fusion center instead of its actual value, which will result in totally erroneous estimates. Further, these erroneous measurements can worsen state estimation results when fused with measurements from similarly biased sensors. In order to benefit from the information available from multiple bearing–only sensors with offset bias, one needs to model, estimate and then correct the biases. This is precisely the motivation for this paper.

The focus of this paper is offset bias estimation. In order to model offset biases, one can transform the measurement space of sensor data to Cartesian coordinates followed by covariance matrix transformation. This transformation will make it possible to find an exact model for the biases that can be used in bias estimation and correction. However, the full position information is not available in a single bearing–only measurement. The process of measurement transformation is done by paring measurements from different sensors in the surveillance region. That is, this transformation is done through triangulation [21]. With the new pseudo–measurement in Cartesian coordinates, position and velocity of the targets can be estimated over time as new measurements are generated from paired sensors at subsequent times [35, 19, 3, 24]. Assuming that these state estimates also carry the effects of offset biases, it is possible to find such biases, if any, and correct them. In the case of bistatic passive sensors, one can use the methods in [44, 37]. However, previous work on multisensor bearing–only bias estimation is still limited. The method presented here gives a comprehensive analysis of offset bias modeling in multisensor passive bearing–only sensors.

The proposed method gives an exact model for bearing–only biases in Cartesian coordinates. In addition, the formulation of an appropriate likelihood function enables

the use of maximum likelihood estimators to find the biases. Also, the proposed model is robust against large sensor noise standard deviation. Finally, as shown through simulation results, large bearing biases can be estimated accurately, which leads to correspondingly accurate target state estimation results.

The goal of this paper is to present a step–by–step approach for designing a multisensor–multitarget tracking system based on biased bearing–only measurements and give a practical solution to the problem of bias estimation. In Section 3.3, a detailed model of the multisensor bearing–only estimation problem with bias is given. Section 3.4 is devoted to modeling the offset biases in Cartesian coordinates. In Section 3.5, a practical solution for bias estimation is proposed. Section 3.6 presents the derivation of Cramér–Rao lower bounds. Simulation results are shown in Section 3.7 along with discussions on different scenarios. Finally, Section 3.8 ends the paper with conclusions.

3.3 Bearing–Only Estimation Problem

Bearing–only sensors with operating ranges of hundreds of meters to a few kilometers are one of the most crucial sensors in maritime or ground surveillance applications. These sensors can actively or passively detect the directions of arrival of signals emitted by the targets of interest. While underwater surveillance is the common application of bearing–only tracking, it is also used in surface and air target tracking. For example, ESM and IRST sensors also use bearing–only sensors for tracking. As shown in Figure 3.1, bearing–only sensors can be on the own–ship or deployed separately in the surveillance region. Moreover, they can operate under different environmental conditions as shown in Figure 3.1 [8, 26].

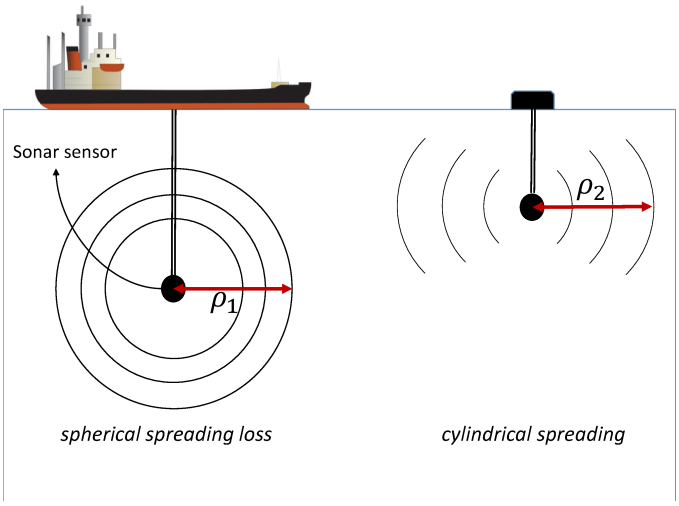

Figure 3.1: Bearing–only sensors and signal propagation types [26]

The bearing–only measurement from passive sensors is written as

$$z_s(k) \quad = \quad \theta_s(k) + b_s + w_s(k), \tag{3.1}$$

where $z_s(k)$ is the direction of arrival at sensor s, θ_s is the true bearing of the target, b_s is the constant bias in the measurements of sensor s and w_s is an additive zero–mean white Gaussian noise with variance σ_θ^2. It is assumed that there are S bearing–only sensors in the surveillance region at positions (x_s, y_s) for $s = 1, 2, ..., S$, and, they record targets' bearings at time instants $k = 1, ..., K$. Note that there is no index to denote target ID, but wherever such clarification is needed, it will be included.

In this paper, bias estimation is handled at the fusion node to which individual sensors report their local tracks in the form of associated measurement reports (AMR) [46, 16] or angle-only tracks. That is, a distributed tracking system is considered as in the case of [31, 32, 42]. However, the difference is that in these earlier works local full-state tracks were sent to the fusion node for bias estimation whereas AMRs or

bearing-only tracks are sent in the present case where the local sensors do not have full observability due to bearing-only measurements. As in [31, 32, 42], working with tracks or AMRs obviates the need at the fusion center to address false alarms and missed detections that are inevitable at the local sensors although ghost tracks may be present among local tracks. However, in order to show that the proposed method can work even with false alarms and missed detections in a centralized tracking system, in one experiment in Section 3.7 we assume that the local sensors send all their measurements (including false alarms and missed detections instead of AMRs or local bearing-only tracks) and evaluate the performance of the proposed method. In [37] and [44], where bias estimation at measurement level (rather than at track level as in [31, 32, 42]) is considered, false alarms and missed detections are not addressed at all. In [30], a joint data association and bias estimation method was proposed for linear measurement models, which is not applicable for bearing-only systems. A general case of multistatic passive radar system with false alarms and missed detections was considered in [17] and [15], but, the bias problem was not addressed.

The goal of bearing–only tracking is to find the bias in each sensor and estimate each target's position as accurately as possible based on the model given in (3.1), either as decoupled parameters or jointly [11, 45]. Due to the computational burdens of joint tracking and parameter estimation methods [22], a decoupled bias and state estimation is presented in this paper.

It is assumed that each target is following the Discretized Continuous White Noise Acceleration (DCWNA) or the nearly constant velocity (NCV) model [5]. As a result,

a target's state vector in 2D Cartesian coordinate is given by

$$\mathbf{x}(k) = \begin{bmatrix} x(k) & v_x(k) & y(k) & v_y(k) \end{bmatrix}^{\mathrm{T}} \tag{3.2}$$

with $(x(k), y(k))$ being the position and $(v_x(k), v_y(k))$ being the velocity. The motion model can further be defined as

$$\mathbf{x}(k+1) = F(k)\mathbf{x}(k) + \nu(k) \tag{3.3}$$

where the state transition matrix is

$$F(k) \triangleq \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

and the covariance matrix of $\nu(k)$ which is an additive zero–mean white Gaussian noise vector is

$$Q = \begin{bmatrix} Q_x & 0 \\ 0 & Q_y \end{bmatrix} \tag{3.5}$$

in which Q_x and Q_y are defined as

$$Q_x = \begin{bmatrix} \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ \frac{1}{2}T^2 & T \end{bmatrix} \bar{q}_x \tag{3.6}$$

$$Q_y = \begin{bmatrix} \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ \frac{1}{2}T^2 & T \end{bmatrix} \bar{q}_y \tag{3.7}$$

where $\bar{q}_x$ and $\bar{q}_y$ are noise intensities with dimension $\mathrm{m}^2/\mathrm{s}^3$ [4].

In order to transform the measurements from polar to Cartesian coordinates, it is initially assumed that S is even and that the sensors are paired into $\frac{S}{2}$ one–to–one pairs. Note that this constraint is relaxed in Section 3.5. For a pair of sensors with a single target in the common field of view, the best estimate for the location of the target, independent of its previous location, can be obtained through triangulation [14, 38, 21]. The triangulated estimates of the target position at time k using sensor pair (i, j), ignoring measurement noise, are given by

$$\begin{aligned} \hat{x}_{ij}(k) = & \frac{y_j - y_i + x_i \tan\left(\theta_i(k) + b_i\right)}{\tan\left(\theta_i(k) + b_i\right) - \tan\left(\theta_j(k) + b_j\right)} \\ & - \frac{x_j \tan\left(\theta_j(k) + b_j\right)}{\tan\left(\theta_i(k) + b_i\right) - \tan\left(\theta_j(k) + b_j\right)} \end{aligned} \tag{3.8}$$

$$\begin{aligned} \hat{y}_{ij}(k) = & \frac{y_j \tan\left(\theta_i(k) + b_i\right) - y_i \tan\left(\theta_j(k) + b_j\right)}{\tan\left(\theta_i(k) + b_i\right) - \tan\left(\theta_j(k) + b_j\right)} \\ & + \frac{(x_i - x_j) \tan\left(\theta_i(k) + b_i\right) \tan\left(\theta_j(k) + b_j\right)}{\tan\left(\theta_i(k) + b_i\right) - \tan\left(\theta_j(k) + b_j\right)} \end{aligned} \tag{3.9}$$

where $\hat{x}_{ij}(k)$ and $\hat{y}_{ij}(k)$ are the X and Y Cartesian estimates, respectively.

In addition, by defining the stacked covariance matrix in the bearing–only coordinate for the stacked measurement $\begin{bmatrix} \theta_i(k) & \theta_j(k) \end{bmatrix}^{\mathrm{T}}$ as

$$R_{ij}(k) = \begin{bmatrix} \sigma^2_{\theta_i(k)} & 0 \\ 0 & \sigma^2_{\theta_j(k)} \end{bmatrix} \tag{3.10}$$

one can calculate the covariance matrix of the transformed vector $\begin{bmatrix} x_{ij}(k) & y_{ij}(k) \end{bmatrix}^{\mathrm{T}}$ as

$$R^{xy}_{ij}(k) = \left(J(k)^{\mathrm{T}} \left(R_{ij}(k)\right)^{-1} J(k)\right)^{-1} \tag{3.11}$$

where $J(k)$ is the Jacobian matrix with respect to $\begin{bmatrix} x_{ij}(k) & y_{ij}(k) \end{bmatrix}^{\mathrm{T}}$ and

$$J(k) = \begin{bmatrix} J_i(k) & J_j(k) \end{bmatrix}^{\mathrm{T}} \tag{3.12}$$

Further, the elements of $J(k)$ can be written as

$$J_i(k) = \left[\frac{-\left(y_{ij}(k) - y_i\right)}{\left(x_{ij}(k) - x_i\right)^2 + \left(y_{ij}(k) - y_i\right)^2} , \frac{\left(x_{ij}(k) - x_i\right)}{\left(x_{ij}(k) - x_i\right)^2 + \left(y_{ij}(k) - y_i\right)^2} \right] \tag{3.13}$$

and

$$J_j(k) = \left[\frac{-\left(y_{ij}(k) - y_j\right)}{\left(x_{ij}(k) - x_j\right)^2 + \left(y_{ij}(k) - y_j\right)^2} , \frac{\left(x_{ij}(k) - x_j\right)}{\left(x_{ij}(k) - x_j\right)^2 + \left(y_{ij}(k) - y_j\right)^2} \right] \tag{3.14}$$

If no bias estimation is needed, these pseudo–measurements in Cartesian coordinates can be used in a Kalman filter with their associated covariance matrices to recursively estimate the target's position [12].

3.4 Bias Modeling in Cartesian Coordinates

As discussed in Section 3.3, bearing–only biases are in the form of the additive constants. Although additive constant biases have already been dealt with in the case of radar measurements [42] and [43], it is not possible to generate similar pseudo–measurements with bearing–only data directly. One way to formulate pseudo–measurements with biases is to model in Cartesian coordinates. In this section, a step–by–step procedure to model the biases and to separate them from original track positions in Cartesian coordinates is given. In Section 3.5, the pseudo–measurement generation is discussed in detail.

In Section 3.3, the process of mapping from bearing–only measurements to Cartesian was given. In order to model the biases, one can start with separating the bias terms in (3.8) and (3.9) from the original track position in Cartesian coordinates. This separation of bias terms provides the necessary information to create a pseudo–measurement that properly addresses the biases as in the case of radar measurements. Once the pseudo–measurements are generated, it is possible to estimate the biases and remove them. The process of finding the bias terms that contribute to (3.8) and (3.9) starts with expanding the $\tan(\cdot)$ function as

$$\tan(\alpha+\beta) = \frac{\tan(\alpha)+\tan(\beta)}{1-\tan(\alpha)\tan(\beta)} \tag{3.15}$$

Applying (3.15) to (3.8) and (3.9), and separating the terms related to the bias from those related to the target position will give a new set of equations to define the position of the target in Cartesian coordinates. To make the parameter separation easier to follow, the common terms are defined and named first. For the common terms in $\hat{x}_{ij}(k)$ and $\hat{y}_{ij}(k)$, one can define

$$D = \tan(\theta_i(k)) - \tan(\theta_j(k)) \tag{3.16}$$

$$B = 1 + \tan(\theta_i(k))\tan(\theta_j(k)) \tag{3.17}$$

Further, define the following for the terms in $\hat{x}_{ij}(k)$:

$$D_x = x_i \tan(\theta_i(k)) - x_j \tan(\theta_j(k)) + (y_j - y_i) \tag{3.18}$$

$$B_x^i = -(y_j - y_i)\tan(\theta_i(k)) + x_j \tan(\theta_i(k))\tan(\theta_j(k)) + x_i \tag{3.19}$$

$$B_x^j = -(y_j - y_i)\tan(\theta_j(k)) - x_i \tan(\theta_i(k))\tan(\theta_j(k)) + x_j \tag{3.20}$$

$$B_x^{ij} = x_j \tan(\theta_i(k)) - x_i \tan(\theta_j(k)) + (y_j - y_i)\tan(\theta_i(k))\tan(\theta_j(k)) \tag{3.21}$$

Similarly, for the terms in $\hat{y}_{ij}(k)$, define

$$D_y = y_j \tan(\theta_i(k)) - y_i \tan(\theta_j(k)) + (x_j - x_i)\tan(\theta_i(k))\tan(\theta_j(k)) \tag{3.22}$$

$$B_y^i = (x_j - x_i)\tan(\theta_j(k)) + y_i \tan(\theta_i(k))\tan(\theta_j(k)) + y_j \tag{3.23}$$

$$B_y^j = (x_j - x_i)\tan(\theta_i(k)) - y_j \tan(\theta_i(k))\tan(\theta_j(k)) - y_i \tag{3.24}$$

$$B_y^{ij} = y_i \tan(\theta_i(k)) - y_j \tan(\theta_j(k)) + (x_j - x_i) \tag{3.25}$$

With these factorizations, bias terms can be separated from the target state values

in Cartesian coordinates. It can be seen that the vector $\begin{bmatrix} \hat{x}_{ij}(k) & \hat{y}_{ij}(k) \end{bmatrix}^{\mathrm{T}}$ can be written as[1]

$$\begin{bmatrix} \hat{x}_{ij}(k) \\ \hat{y}_{ij}(k) \end{bmatrix} = \begin{bmatrix} x_{ij}^{\mathrm{u}}(k) + \beta_x\left(\theta_i(k), \theta_j(k), b_i, b_j\right) \\ y_{ij}^{\mathrm{u}}(k) + \beta_y\left(\theta_i(k), \theta_j(k), b_i, b_j\right) \end{bmatrix} \tag{3.26}$$

where

$$x_{ij}^{\mathrm{u}}(k) = \frac{D_x}{D} \tag{3.27}$$

$$y_{ij}^{\mathrm{u}}(k) = \frac{D_y}{D} \tag{3.28}$$

and the bias terms can be written in as

$$\begin{aligned} \beta_x\left(\theta_i(k), \theta_j(k), b_i, b_j\right) = & \frac{B_x^i \tan(b_i) + B_x^j \tan(b_j) + B_x^{ij} \tan(b_i)\tan(b_j)}{D + D\tan(b_i) + B\tan(b_j) + D\tan(b_i)\tan(b_j)} \\ & - \frac{D_x B \tan(b_i) - D_x B \tan(b_j) + D_x D \tan(b_i)\tan(b_j)}{D^2 + BD\tan(b_i) - BD\tan(b_j) + D^2\tan(b_i)\tan(b_j)} \end{aligned} \tag{3.29}$$

$$\begin{aligned} \beta_y\left(\theta_i(k), \theta_j(k), b_i, b_j\right) = & \frac{B_y^i \tan(b_i) + B_y^j \tan(b_j) + B_y^{ij} \tan(b_i)\tan(b_j)}{D + D\tan(b_i) + B\tan(b_j) + D\tan(b_i)\tan(b_j)} \\ & - \frac{D_y B \tan(b_i) - D_y B \tan(b_j) + D_y D \tan(b_i)\tan(b_j)}{D^2 + BD\tan(b_i) - BD\tan(b_j) + D^2\tan(b_i)\tan(b_j)} \end{aligned} \tag{3.30}$$

[1]The superscript "u" indicates that the parameter is unbiased.

Note that in the above formulations it is assumed that the biases and the true bearings are available. In practice, only the noisy or estimated values are available. Assuming small values for the bias and noise terms, one can use the above formulation without significant loss in accuracy. Similar assumptions has been made in the previous works on bias estimation [42, 31, 32]. A technical discussion on the range of bias and noise values for which the above formulation is valid is given in Section 3.7.

3.5 Bearing–Only Tracking and Registration

Bearing–only sensor registration is a challenging problem in target tracking that has been addressed in [41, 34, 37]. In order to find the biases and correct the measurements, one should first look into the observability of the bias variables. Note that in (3.26), provided that the target is not on the line that connects the two sensors used in the triangulation or in the vicinity of one of the sensors, the state parameters are observable [4]. In addition, if there are two pairs of sensors tracking the same target, the biases become observable as it is shown in 3.5.1. To estimate the biases decoupled from the state vector, a pseudo–measurement that can address the bias vector directly must be defined. In this section, a new formulation is proposed to create a pseudo–measurement that can be used for bias estimation with bearing–only data. The key requirement of this method in order to ensure observability of all parameters is to have at least two sensor pairs in the surveillance region. In the following, two practical scenarios that can be expanded to a more general formulation to handle varying number of sensors and targets are discussed in detail.

3.5.1 Pseudo–measurement of bearing–only measurements

To handle practical bearing–only scenarios, two different cases are analyzed here. In each case, a separate pseudo–measurement model is proposed along with its associated covariance matrix. The main idea is to use two different position approximations to create a pseudo–measurement as discussed below.

Four–sensor (or any even number of sensors) case

Assuming that there are two pairs of sensors in the surveillance region, a vector of nonlinear pseudo–measurements can be defined by subtracting the target positions based on the pairs (i, j) and (m, n) as

$$z_b(k) = \begin{bmatrix} \hat{x}_{ij}(k) - \hat{x}_{mn}(k) \\ \hat{y}_{ij}(k) - \hat{y}_{mn}(k) \end{bmatrix} + w(k) \tag{3.31}$$

where $w(k)$ is the additive zero–mean white Gaussian noise associated with the pseudo–measurement and its covariance matrix is defined as

$$R_w(k) = R_{ij}^{xy}(k) + R_{mn}^{xy}(k) \tag{3.32}$$

Using the fact that in the absence of bias and noise terms, measurements from any two sensors point to the same target location regardless of the sensor locations, the pseudo–measurement $z_b(k)$ can be written as

$$z_b(k) = \begin{bmatrix} \beta_x(b_i, b_j) - \beta_x(b_m, b_n) \\ \beta_y(b_i, b_j) - \beta_x(b_m, b_n) \end{bmatrix} + w(k) \tag{3.33}$$

for two uncorrelated pairs of sensors. This can be applied to any even number of sensors.

Three–sensor (or any odd number of sensors) case

In this case, one must create two position approximations from triangulation to be able to create a pseudo–measurement for biases. Since there are only three sensors, the possible pairs are (i, j) and (i, m). Then, the pseudo–measurement can be approximated by

$$z_b(k) = \begin{bmatrix} \hat{x}_{ij}(k) - \hat{x}_{im}(k) \\ \hat{y}_{ij}(k) - \hat{y}_{im}(k) \end{bmatrix} + w(k) \tag{3.34}$$

where $w(k)$ is approximately additive zero–mean white Gaussian noise associated with the pseudo–measurement and its covariance matrix is defined as

$$R_w(k) = R_{ij}^{xy}(k) + R_{im}^{xy}(k) \tag{3.35}$$

Because of the correlation between the two tracks from three sensors in Cartesian coordinates, the noise is not white anymore and this formulation is only an approximation.

As in the case for four sensors, the pseudo–measurement $z_b(k)$ can be written as

$$z_b(k) = \begin{bmatrix} \beta_x(b_i, b_j) - \beta_x(b_i, b_m) \\ \beta_y(b_i, b_j) - \beta_x(b_i, b_m) \end{bmatrix} + w(k) \tag{3.36}$$

Note that for simplicity, the arguments $\theta_i(k)$, $\theta_j(k)$, $\theta_m(k)$ and $\theta_n(k)$ have been

dropped from (3.33) and (3.36). With an arbitrary odd number of sensors, one sensor need to be paired with two other, resulting in in a similar approximation. As for the more general case of n arbitrary sensors, methods similar to [42] can be adopted to handle the situation.

3.5.2 Batch maximum–likelihood estimator

To apply a batch estimator for bias estimation, one needs to form a likelihood function. Following (3.35) and (3.33), and assuming that the noise is white, zero–mean and Gaussian, the likelihood function of the bias parameters given two pairs of sensors is

$$\begin{aligned} p\left(z_b(k) \mid \mathbf{b}\right) &= \frac{1}{2\pi \left|R_w(k)\right|^{-\frac{1}{2}}} \exp\Bigg(-\frac{1}{2}\left[z_b(k) - h(\mathbf{b})^{\mathrm{T}}\right] \\ &\quad \left(R_w(k)\right)^{-1}\left[z_b(k) - h(\mathbf{b})\right]\Bigg) \end{aligned} \tag{3.37}$$

where the nonlinear function $h(\mathbf{b})$ of the bias vector is given by

$$h(\mathbf{b}) = \begin{bmatrix} \beta_x\left(b_i, b_j\right) - \beta_x\left(b_m, b_n\right) \\ \beta_y\left(b_i, b_j\right) - \beta_x\left(b_m, b_n\right) \end{bmatrix} \tag{3.38}$$

Assuming independence over time, one can write the likelihood function over $k = 1, ..., K$ as

$$p\left(Z_b \mid \mathbf{b}\right) = \prod_{k=1}^{K} p\left(z_b(k) \mid \mathbf{b}\right) \tag{3.39}$$

where

$$Z_b = \{z_b(1), z_b(2), \ldots, z_b(K)\} \tag{3.40}$$

Finally, the vector $\hat{\mathbf{b}}$ that maximizes the likelihood function can be written as

$$\hat{\mathbf{b}} = \arg \max_{\mathbf{b}} p\left(Z_b \mid \mathbf{b}\right) \tag{3.41}$$

The above assumes that there is only one target, but it can be extended to the multitarget case using the stacked parameter estimator.

3.6 Cramér–Rao Lower bound for Bearing only bias estimation

This section is devoted to the calculation of the Cramér–Rao Lower Bound (CRLB) on the estimation accuracy of bias parameters by using the pseudo–measurements introduced in Subsection 3.5.1. Note that based on (3.38), the measurement equation for target t at time k is

$$h^t_{\mathbf{b}}(k) = \begin{bmatrix} \beta^t_x\left(b_i, b_j\right) - \beta^t_x\left(b_m, b_n\right) \\ \beta^t_y\left(b_i, b_j\right) - \beta^t_x\left(b_m, b_n\right) \end{bmatrix} + w^t(k) \tag{3.42}$$

Assuming that there are N targets in the surveillance region and that the bias parameters are constant over time $k = 1, \ldots K$, the stacked measurement vector can be

written as

$$\mathbf{Z} = \mathbf{h}(\mathbf{b}) + \mathbf{w} \tag{3.43}$$

where

$$\mathbf{Z} = \left[\begin{array}{cccccccc} z_b^1(1)^{\mathrm{T}} & \cdots & z_b^N(1)^{\mathrm{T}} & \cdots & z_b^1(K)^{\mathrm{T}} & \cdots & z_b^N(K)^{\mathrm{T}} \end{array} \right]^{\mathrm{T}} \tag{3.44}$$

$$\mathbf{h}(\mathbf{b}) = \left[\begin{array}{cccccccc} h_{\mathbf{b}}^1(1)^{\mathrm{T}} & \cdots & h_{\mathbf{b}}^N(1)^{\mathrm{T}} & \cdots & h_{\mathbf{b}}^1(K)^{\mathrm{T}} & \cdots & h_{\mathbf{b}}^N(K)^{\mathrm{T}} \end{array} \right]^{\mathrm{T}} \tag{3.45}$$

$$\mathbf{w} = \left[\begin{array}{cccccccc} w^1(1)^{\mathrm{T}} & \cdots & w^N(1)^{\mathrm{T}} & \cdots & w^1(K)^{\mathrm{T}} & \cdots & w^N(K)^{\mathrm{T}} \end{array} \right]^{\mathrm{T}} \tag{3.46}$$

Further, the covariance matrix of the noise vector $\mathbf{w}$ is

$$R = \mathrm{diag}\left(\left[\begin{array}{cccccccc} R^1(1) & \cdots & R^N(1) & \cdots & R^1(K) & \cdots & R^N(K) \end{array} \right] \right) \tag{3.47}$$

The covariance matrix of an unbiased estimator $\hat{\mathbf{b}}$ is bounded from below by [4]

$$\mathbb{E}\left\{ \left(\hat{\mathbf{b}} - \mathbf{b} \right) \left(\hat{\mathbf{b}} - \mathbf{b} \right)^{\mathrm{T}} \right\} \geq \mathbf{J}^{-1} \tag{3.48}$$

where $\mathbf{J}$ is the Fisher Information Matrix (FIM) given by

$$\begin{aligned} \mathbf{J} &= \mathbb{E}\left\{ \left[\nabla_{\mathbf{b}} \log p(\mathbf{Y} \mid \mathbf{b}) \right] \left[\nabla_{\mathbf{b}} \log p(\mathbf{Y} \mid \mathbf{b}) \right]^{\mathrm{T}} \right\} |_{\mathbf{b}=\tilde{\mathbf{b}}} \\ &= \mathbb{E}\left\{ \left[\nabla_{\mathbf{b}} \lambda \right] \left[\nabla_{\mathbf{b}} \lambda \right]^{\mathrm{T}} \right\} |_{\mathbf{b}=\tilde{\mathbf{b}}} \end{aligned} \tag{3.49}$$

in which $\tilde{\mathbf{b}}$ is the true value of the bias vector $\mathbf{b}$, $p(\mathbf{Z} \mid \mathbf{b})$ is the likelihood function of $\mathbf{b}$, $\lambda = -\ln p(\mathbf{Y} \mid \mathbf{b})$ and ∇ is gradient operator. Based on the model for the stacked

measurement vector in (3.43), one can define the Jacobian matrix of $h^t_{\mathbf{b}}(k)$ evaluated at the true value $\mathbf{b}$ [39] as

$$\tilde{H}^t_{\mathbf{b}}(k) \;=\; \left[\bigtriangledown_{\mathbf{b}} h^t_{\mathbf{b}}(k)^{\mathrm{T}}\right]^{\mathrm{T}} \tag{3.50}$$

Then, defining

$$\tilde{\mathbf{H}}_{\mathbf{b}} \;=\; \left[\begin{array}{cccccccc} \tilde{H}^1_{\mathbf{b}}(1)^{\mathrm{T}} & \cdots & \tilde{H}^N_{\mathbf{b}}(1)^{\mathrm{T}} & \cdots & \tilde{H}^1_{\mathbf{b}}(K)^{\mathrm{T}} & \cdots & \tilde{H}^N_{\mathbf{b}}(K)^{\mathrm{T}} \end{array}\right]^{\mathrm{T}} \tag{3.51}$$

one can write

$$J \;=\; \mathbb{E}\left\{\tilde{\mathbf{H}}^{\mathrm{T}}_{\mathbf{b}} R^{-1} \tilde{\mathbf{H}}_{\mathbf{b}}\right\} \tag{3.52}$$

3.7 Simulation Results

To evaluate how the proposed method performs in practical scenarios, a series of simulations is presented in this section. The implementation details on bias estimation, filtering and fusion are also discussed. Simulation results on different scenarios are given with discussions on the advantages and disadvantages of the proposed bias estimator.

3.7.1 Motion models and measurement generation

A tracking scenario with four bearing–only sensors and sixteen targets is considered as shown in Figure 3.2. It is assumed that all sensors are synchronized and that the bias ranges are between -0.05 rad and 0.05 rad. The standard deviation of measurement

noise is $\sigma_\theta = 0.0261$ rad $= 1.5^\circ$ for target bearing measurements, which is higher than what was previously assumed in the literature [6]. The true motion of the targets is

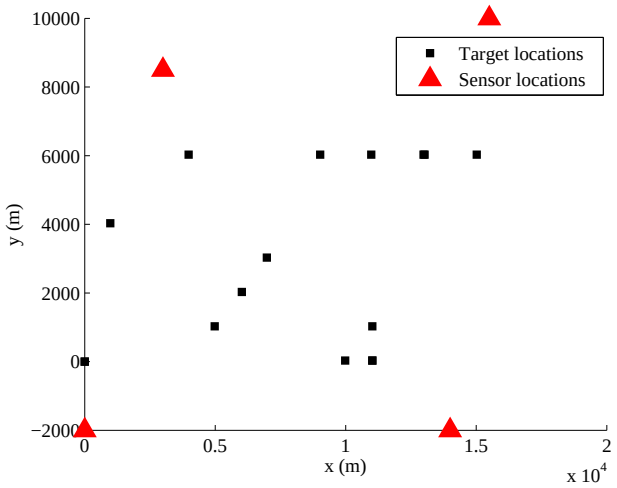

Figure 3.2: Initial locations of the targets and sensors

modeled using the DCWNA or the NCV model [4] with $q_x = q_y = 0.001\text{m}^2/\text{s}^3$. In the local trackers, DCWNA and Continuous Wiener Process Acceleration (CWPA) are used with $q_x = q_y = 0.1\text{m}^2/\text{s}^3$ to create different scenarios for the simulation. In all scenarios, the sampling time is $T = 1$s. To validate the proposed bias model and to quantify its performance at different bias values, the biases are set to both positive and negative values in different ranges as follows:

$$\mathbf{b}_{\text{test}_1} = \begin{bmatrix} 0.04\,\text{rad} & -0.02\,\text{rad} & 0.03\,\text{rad} & -0.02\,\text{rad} \end{bmatrix}^{\text{T}} \tag{3.53}$$

$$\mathbf{b}_{\text{test}_2} = \begin{bmatrix} -0.04\,\text{rad} & -0.02\,\text{rad} & -0.03\,\text{rad} & -0.02\,\text{rad} \end{bmatrix}^{\text{T}} \tag{3.54}$$

$$\mathbf{b}_{\text{test}_3} = \begin{bmatrix} 0.04\,\text{rad} & 0.02\,\text{rad} & 0.03\,\text{rad} & 0.02\,\text{rad} \end{bmatrix}^{\text{T}} \tag{3.55}$$

Table 3.1: Parameter settings for the Genetic Algorithm

Parameter	Value
Lower bound	-0.05 rad
Upper bound	0.05 rad
Number of generations	100
Tolerance value	1×10^{-15}

3.7.2 Bias estimation and the Genetic Algorithm

In this paper, the Genetic Algorithm (GA) [18] is used to solve the optimization problem in (3.41). The Genetic Algorithm is an efficient optimization algorithm for highly nonlinear objective functions [40] that is widely used in different applications [18]. Note that although the GA is a batch ML estimator, the length of the window can be varied depending on user criteria to meet the real–time requirements. The parameters used in the simulations are shown in Table 3.1. The algorithms were implemented on a computer with Intel® Core™ i7-3720Qm 2.60GHz processor and 8GB RAM.

3.7.3 Bias estimation: Four–sensor distributed problem

In this scenario, all four sensors defined earlier are used to implement the GA. Four out of sixteen targets are used for performance evaluation. AMRs or local bearing–only tracks are collected over 100 time steps and the GA is applied to the whole data in batch mode. The GA is run with the settings in Table 3.1 and the final results are gathered after the termination of the GA. Then, the estimated bias vector is used over 100 Monte Carlo runs to calculate the Root Mean Square Error (RMSE) for comparison. As the benchmark, the CRLB is also calculated based on the derivations in Section 3.6. The RMSE values and $\sqrt{\text{CRLB}\left\{[\mathbf{b}]_i\right\}}$ of the ML estimates with the

Table 3.2: Comparison of $\sqrt{\text{CRLB}\{[\mathbf{b}]_i\}}$ and RMSE of GA output for bias estimation of $\mathbf{b}_{\text{test}_1}$ in a distributed system

Bias parameter	$\sqrt{\text{CRLB}\{[\mathbf{b}]_i\}}$ (rad)	RMSE (rad) of the GA bias estimate
b_i	0.071×10^{-3}	3.32×10^{-3}
b_j	0.071×10^{-3}	3.28×10^{-3}
b_m	0.537×10^{-3}	3.92×10^{-3}
b_n	0.462×10^{-3}	1.45×10^{-3}

Table 3.3: Comparison of $\sqrt{\text{CRLB}\{[\mathbf{b}]_i\}}$ and RMSE of the GA output for bias estimation of $\mathbf{b}_{\text{test}_2}$ in a distributed system

Bias parameter	$\sqrt{\text{CRLB}\{[\mathbf{b}]_i\}}$ (rad)	RMSE (rad) of the GA bias estimate
b_i	0.0904×10^{-3}	1.72×10^{-3}
b_j	0.0941×10^{-3}	1.81×10^{-3}
b_m	0.4243×10^{-3}	2.95×10^{-3}
b_n	0.3787×10^{-3}	1.73×10^{-3}

three different sets of bias parameters are shown in Tables 3.2, 3.3 and 3.4.

Although there is a difference between the RMSE and $\sqrt{\text{CRLB}\{[\mathbf{b}]_i\}}$, the RMSE results are nearly an order of magnitude smaller than the bias values, which indicates that any correction made based on the estimated biases will result in better position estimates. Note that the CRLB in bearing–only tracking problems can be overly optimistic and may even approach zero (i.e., perfect estimates) in a network of bearing–only sensors [23]. Thus, the difference between the theoretical CRLB and

Table 3.4: Comparison of $\sqrt{\text{CRLB}\{[\mathbf{b}]_i\}}$ and RMSE of GA output for bias estimation of $\mathbf{b}_{\text{test}_3}$ in a distributed system

Bias parameter	$\sqrt{\text{CRLB}\{[\mathbf{b}]_i\}}$ (rad)	RMSE (rad) of the GA bias estimate
b_i	0.0973×10^{-3}	2.73×10^{-3}
b_j	0.0954×10^{-3}	3.75×10^{-3}
b_m	0.5347×10^{-3}	2.43×10^{-3}
b_n	0.4880×10^{-3}	2.66×10^{-3}

the empirical RMSE is not of major concern.

To show how much the proposed bias estimation method can help in correcting the target tracks, another simulation is conducted. In this simulation, it is assumed that the tracker has access to the final estimated bias vector (output of the GA) and then a Kalman filter is run with the bias estimates in hand. To use the estimated bias parameters, one should, first, correct the bearing–only measurements with the estimated values. The correction must be done both in the measurement vector and its associated covariance matrix. Since the estimated biases do not have the covariance information, a scaled version of the the calculated CRLB of the bias parameters is used instead. The scaling factor can be determined through experiments. Then, the tracker can be run with these corrected measurements to find the position and velocity estimates of all targets in the surveillance region. The position RMSE of the original tracks before correction, the RMSE of the corrected estimates and the Cramér–Rao lower bounds are shown in Figures 3.3 and 3.4.

As shown in Figures 3.3 and 3.4, for case 1, Figures 3.5 and 3.6 for case 2 and Figures 3.7 and 3.8 for case 3, the position error is reduced significantly in terms of the position RMSE. This demonstrates the effectiveness of the bias model proposed and the ML estimation algorithm, i.e., the GA, that is applied to the data. Although, according to the results, the correction factor varies based on the sensor–target orientation, the corrected track and its associated covariance do follow the bias–free values accurately, which demonstrates the capability of the proposed algorithm in estimating the biases. To process a batch of data with $K = 100$, the computational time is 44.1s in MATLAB.

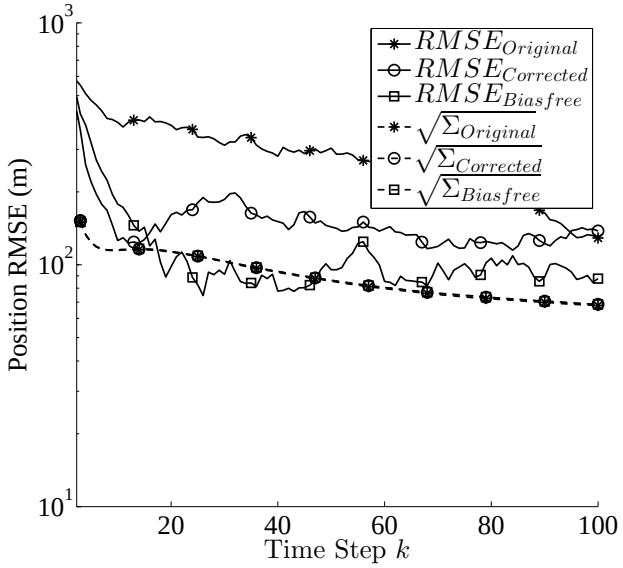

Figure 3.3: Position RMSE with distributed tracking for corrected and original tracks of target 2 (set 1)

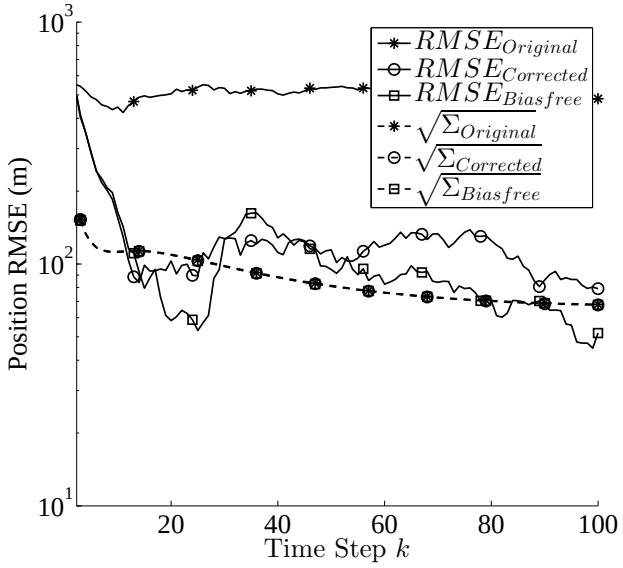

Figure 3.4: Position RMSE with distributed tracking for corrected and original tracks of target 3 (set 1)

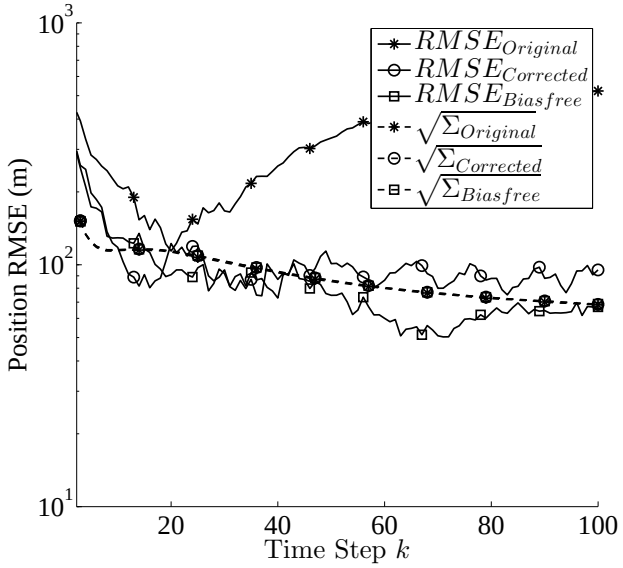

Figure 3.5: Position RMSE with distributed tracking for corrected and original tracks of target 2 (set 2)

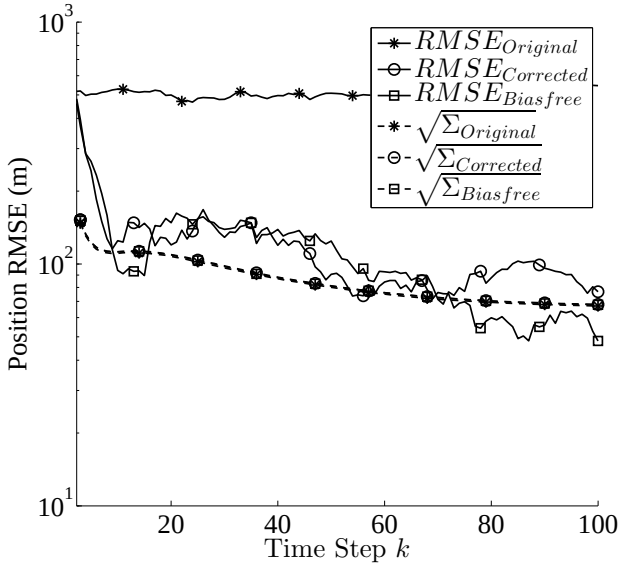

Figure 3.6: Position RMSE with distributed tracking for corrected and original tracks of target 3 (set 2)

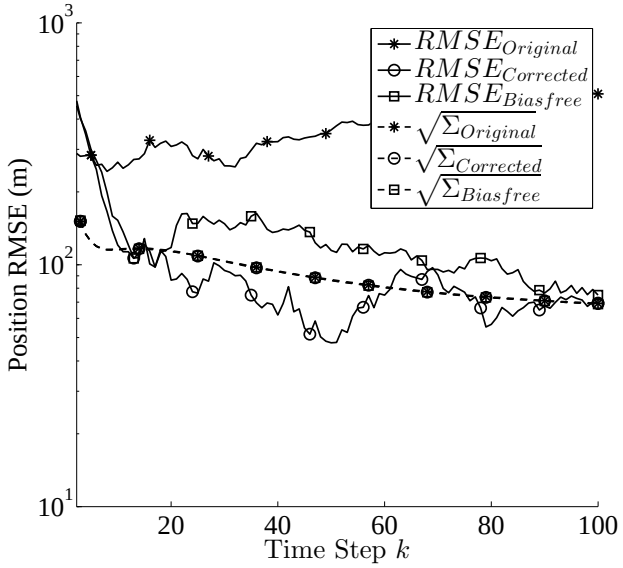

Figure 3.7: Position RMSE with distributed tracking for corrected and original tracks of target 2 (set 3)

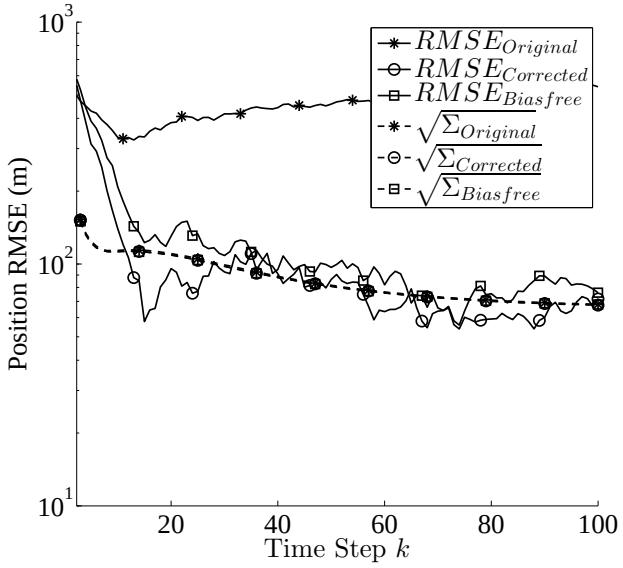

Figure 3.8: Position RMSE with distributed tracking for corrected and original tracks of target 3 (set 3)

3.7.4 Bias estimation: Three–sensor distributed problem

To show the accuracy of bias estimation when there are only three sensors in the surveillance region, the same Genetic Algorithm is used to solve the ML problem in (3.41). The primary issue with three sensors is that one of them, here sensor i, is used to pair with both sensors j and m. This leads to correlation between the two tracks over time. Assuming that the correlation is negligible, the same method is applied to estimate the biases. Further, for the case of three sensors, the bias values are set to

$$\mathbf{b}_{\text{test}} = \begin{bmatrix} 0.04\,\text{rad} & 0.02\,\text{rad} & 0.03\,\text{rad} \end{bmatrix}^{\text{T}} \tag{3.56}$$

to be able to distinguish between the original and corrected tracks. The constraints on the lower and upper bounds of the optimization algorithm are set to -0.05 rad and 0.05 rad, respectively. The tolerance value for the GA is also set to 1×10^{-15}. Figures 3.9 and 3.10 show the result for position estimates in Cartesian coordinates.

The corrected tracks are sill better in terms of RMSE, which means that the estimated biases are accurate enough in spite of the correlation.

3.7.5 Real–time window–based Genetic Algorithm

It is important for the proposed method to be able to work in real–time. For this purpose, the GA can be set to run, in each iteration, for a specific window size or duration. The settings of the GA for real time scenarios are given in Table 3.5.

The final estimates and the population matrix in one window can be used as the initial conditions for the next window. Thus, the estimates of the biases can be used to correct the measurements at the end of processing each window of data. The

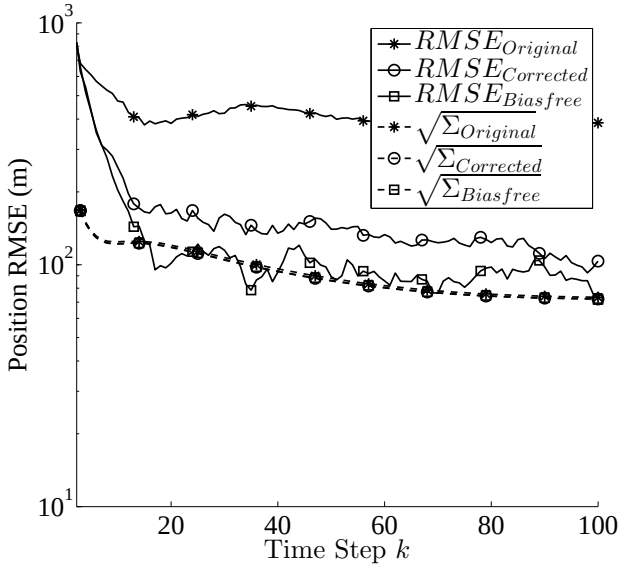

Figure 3.9: Position RMSE of corrected and original tracks for the three–sensor distributed tracking case (Target 2)

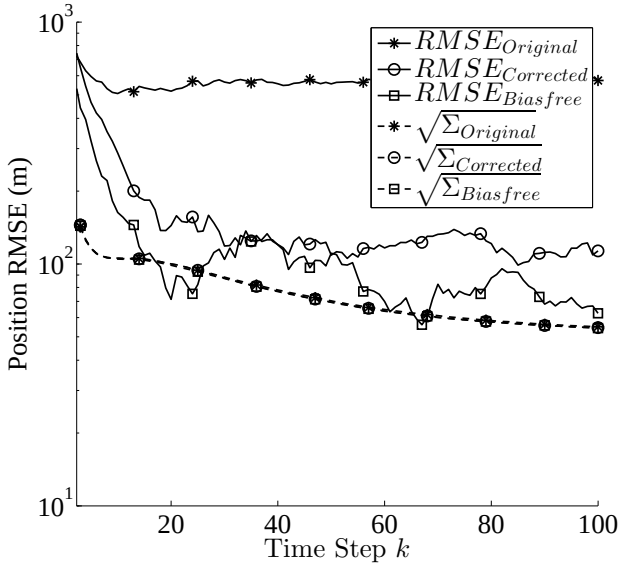

Figure 3.10: Position RMSE of corrected and original tracks for the three–sensor distributed tracking case (Target 3)

Table 3.5: Parameter settings for real time genetic algorithm

Parameter name	Value
Lower bound	-0.05 rad
Upper bound	0.05 rad
Number of generations	50
Tolerance value	1×10^{-15}
Window size	10

results of the simulations using this approach for the case of four sensors (test 1) are given in Figures 3.11 and 3.12.

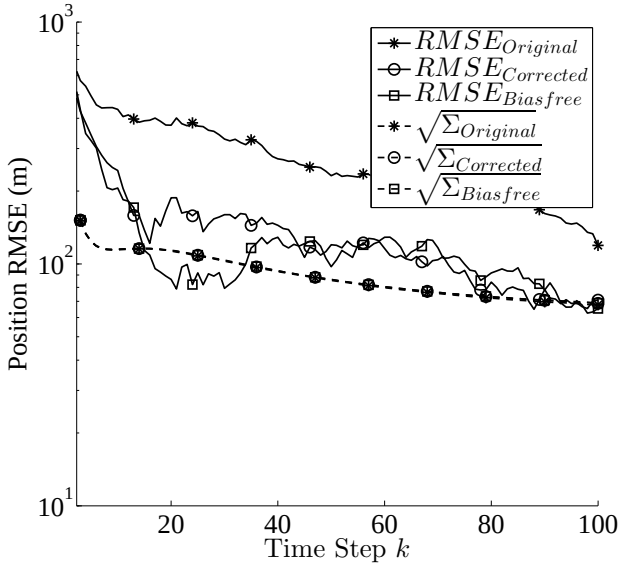

Figure 3.11: Position RMSE for corrected and original tracks for the four–sensor distributed tracking case and window size of 10 (Target 2)

As shown in Figures 3.11 and 3.12, the GA is still able to find the biases with a smaller window size and fewer generations. Note that updating the biases with smaller window sizes enables the use of methods similar to [43, 42] for arbitrary number of sensors in the surveillance region. With this setting, biases can be updated every 9.25 s in MATLAB. This simple example shows that even when the processing time

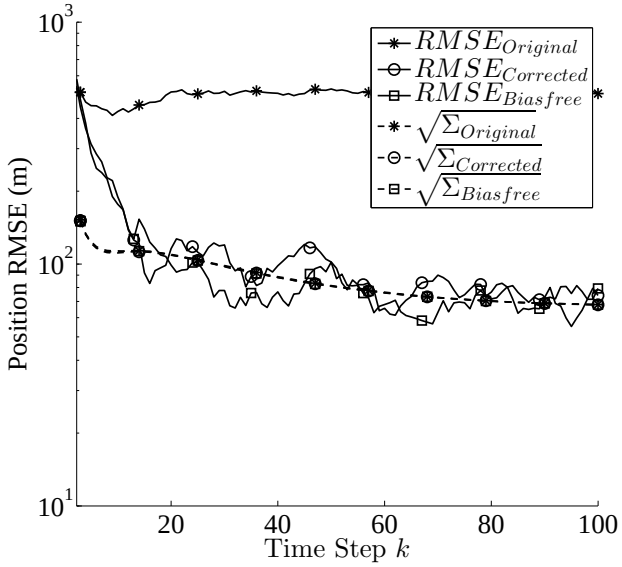

Figure 3.12: Position RMSE for corrected and original tracks for the four–sensor distributed tracking case and window size of 10 (Target 3)

is a crucial parameter in the design, one can still handle bias estimation in real–time using the proposed method.

3.7.6 Bias estimation with false measurements: Four–sensor centralized problem

To demonstrate how the proposed method performs in the presence of false alarms and missed associations in a centralized fusion framework where local sensors send all their measurements instead of AMRs or bearing–only tracks, a simulation is presented in this section. The probability mass function of the number of false alarms or clutter points in surveillance volume V as a function of their spatial density λ is defined as

$$\mu_{\text{FA}}(m) = e^{-\lambda V}\frac{(\lambda V)^m}{m!} \tag{3.57}$$

where m is the number of false alarms [5]. With bearing–only measurements, the volume is $V = 2\pi$ (rad). It is assumed that the average number of false alarms per unit volume in a scan, i.e., λ, is 0.5. Also, it is assumed that $P_D = 0.7$ for each sensor. Note that in this centralized case, the local sensors send all measurements (rather than AMRs or local bearing–only tracks) to the fusion node. Because the proposed method is a batch estimator and uses measurements from both sensor pairs to create a pseudo–measurement for bias estimation, false tracks are often removed prior to generating the pseudo–measurement vector, which then is sent to the bias estimator. Typically, false tracks do not exist for more than a few time steps as they are dependent on all four sensors creating false alarms at the same time steps, in the same region, and for a reasonably long interval of time.

To show the accuracy of bias estimation in a centralized system, the same Genetic Algorithm is used to solve the ML problem in (3.41). The RMSE values and $\sqrt{\text{CRLB}\{[\mathbf{b}]_i\}}$ of the ML estimates with the bias parameters as defined in (3.53) are shown in Table 3.6. Note that the CRLB values are optimistic because they do not factor in the false alarms or the missed detections and that the ML estimator does not factor in the false alarms or the missed detections explicitly. A comprehensive centralized bias estimator is under development. The focus of this paper is the decentralized one.

Table 3.6: Comparison of CRLB and GA output for bias estimation of $\mathbf{b}_{\text{test}_1}$ with measurement origin uncertainty in a centralized system ($\lambda = 0.5$ and $P_D = 0.7$)

Bias parameter	CRLB	GA bias estimate
b_i	0.2123×10^{-3}	7.419×10^{-3}
b_j	0.2123×10^{-3}	12.85×10^{-3}
b_m	0.7674×10^{-3}	6.143×10^{-3}
b_n	0.5779×10^{-3}	4.352×10^{-3}

3.8 Conclusions

In this paper, a new mathematical model for bearing–only bias estimation in distributed tracking systems was proposed. This model was based on triangulation using the associated measurement reports or local bearing–only tracks from different sensor pairs. It was shown that the proposed bias model has the advantage of being practical in scenarios with multiple sensors. In particular, the proposed algorithm is effective when the sensor noise level and bias values are high. In addition, previously proposed algorithms were dependent on target–sensor maneuvers and/or limited to certain noise levels. The new bias model can handle any type of target–sensor motion and it is effective against 5° of offset bias in each sensor and uncertainty levels up to 1.5° of noise standard deviation, which is higher than what was assumed in the literature previously. Also, the proposed method can handle false alarms and missed detections in a centralized architecture. That is, the proposed algorithm is practical in scenarios with realistic sensor parameter values. Finally, a batch ML estimator was proposed to solve the bias estimation problem along with simulation results. A comprehensive centralized bias estimation algorithm with data association for bearing–only sensors is in progress.

Bibliography

[1] V. J. Aidala. "Kalman filter behavior in bearings–only tracking applications". *IEEE Transactions on Aerospace and Electronic Systems*, vol. 1, pp. 29–39, 1979.

[2] V. J. Aidala and S. E. Hammel. "Utilization of modified polar coordinates for

bearings–only tracking". *IEEE Transactions on Automatic Control*, vol. 28, no. 3, pp. 283–294, 1983.

[3] M. S. Arulampalam, B. Ristic, N. Gordon, and T. Mansell. "Bearings–only tracking of manoeuvring targets using particle filters". *EURASIP Journal on Applied Signal Processing*, pp. 2351–2365, 2004.

[4] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. Wiley, NY, 2001.

[5] Y. Bar-Shalom, P. K. Willett, and X. Tian. *Tracking and Data Fusion: A Handbook of Algorithms*. YBS Publishing, Storrs, CT, 2011.

[6] D. Belfadel, R. W. Osborne, and Y. Bar-Shalom. "Bias estimation for optical sensor measurements with targets of opportunity". *16th International Conference on Information Fusion*, pp. 1805–1812, Istanbul, Turkey, July 2013.

[7] D. Belfadel, R. W. Osborne, and Y. Bar-Shalom. "A minimalist approach to bias estimation for passive sensor measurements with targets of opportunity". *Proc. SPIE Conference on Signal and Data Processing of Small Targets*, Vol. 8857, 2013.

[8] D. A. Blank and A. E. Bock. *Introduction to Naval Engineering*. Naval Institute Press, Annapolis, MD, U.S.A, 2005.

[9] P. Braca, R. Goldhahn, G. Ferri, and K. LePage. "Distributed information fusion in multistatic sensor networks for underwater surveillance". *IEEE Sensors Journal*, 2015.

[10] P. Braca, P. Willett, K. Lepage, S. Marano, and V. Matta. "Bayesian tracking in underwater wireless sensor networks with port-starboard ambiguity". *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1864–1878, April 2014.

[11] M. Bugallo, T. Lu, and P. Djuric. "Bearings–only tracking with biased measurements". *2nd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, pp. 265–268, 2007.

[12] D. E. Catlin. *Estimation, Control, and the Discrete Kalman Filter*. vol. 71, Springer Science & Business Media, 2012.

[13] Y. T. Chan and S. W. Rudnicki. "Bearings–only and Doppler–bearing tracking using instrumental variables". *IEEE Transactions on Aerospace and Electronic Systems*, vol. 28, no. 4, pp. 1076–1083, 1992.

[14] H. Chen and F. Lian. "Bias estimation for multiple passive sensors". *International Conference on Measurement, Information and Control (MIC)*, vol. 2, pp. 1081–1084, May 2012.

[15] S. Choi, D. F. Crouse, P. Willett, and S. Zhou. "Approaches to cartesian data association passive radar tracking in a DAB/DVB network". *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 1, pp. 649–663, 2014.

[16] R. L. Cooperman. "Tactical ballistic missile tracking using the interacting multiple model algorithm". *Fifth International Conference on Information Fusion*, vol. 2, pp. 824–831, Annapolis, Maryland, USA, 2002.

[17] M. Daun, U. Nickel, and W. Koch. "Tracking in multistatic passive radar systems

using DAB/DVB-T illumination". *Signal Processing*, vol. 92, no. 6, pp. 1365–1386, 2012.

[18] L. Davis. *Handbook of Genetic Algorithms.* vol. 115, Van Nostrand Reinhold, New York, 1991.

[19] A. Farina. "Target tracking with bearings–only measurements". *Signal Processing*, vol. 78, no. 1, pp. 61–78, 1999.

[20] S. Fortunati, A. Farina, F. Gini, A. Graziano, M. S. Greco, and S. Giompapa. "Least squares estimation and cramér–rao type lower bounds for relative sensor registration process". *IEEE Transactions on Signal Processing*, vol. 59, no. 3, pp. 1075–1087, 2011.

[21] G. Hendeby, R. Karlsson, F. Gustafsson, and N. Gordon. "Recursive triangulation using bearings–only sensors". *IEE Seminar on Target Tracking: Algorithms and Applications*, pp. 3–10, Birmingham, England, March 2006.

[22] M. A. Hopkins and H. F. Vanlandingham. "Pseudo-linear identification: Optimal joint parameter and state estimation of linear stochastic MIMO systems". *IEEE American Control Conference*, pp. 1301–1306, 1988.

[23] P. R. Horridge and M. L. Hernandez. "Performance bounds for angle–only filtering with application to sensor network management". *6th International Conference on Information Fusion*, pp. 695–703, Cairns, Queensland Australia, July 2003.

[24] R. A. Iltis and K. L. Anderson. "A consistent estimation criterion for multi-sensor bearings–only tracking". *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 1, pp. 108–120, 1996.

[25] K. Johansson, K. Jöred, and P. Svensson. "Submarine tracking using multi–sensor fusion and reactive planning for the positioning of passive sonobuoys". *Hydroakustik*, vol. 97, 1997.

[26] N. Z. Kolev. *Sonar Systems*. InTech, Croatia, 2011.

[27] T. R. Kronhamn. "Bearings–only target motion analysis based on a multihypothesis Kalman filter and adaptive own–ship motion control". *IEE Proceedings–Radar, Sonar and Navigation*, vol. 145, no. 4, pp. 247–252, 1998.

[28] B. La Scala and M. Morelande. "An analysis of the single sensor bearings–only tracking problem". *11th International Conference on Information Fusion*, pp. 1–6, Cologne, Germany, June 2008.

[29] P. H. Leong, S. Arulampalam, T. A. Lamahewa, and T. D. Abhayapala. "A Gaussian–sum based cubature Kalman filter for bearings–only tracking". *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 1161–1176, 2013.

[30] Z. Li, S. Chen, H. Leung, and E. Bosse. "Joint data association, registration, and fusion using EM–KF". *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 2, pp. 496–507, 2010.

[31] X. Lin, Y. Bar-Shalom, and T. Kirubarajan. "Exact multisensor dynamic bias

estimation with local tracks". *IEEE Transactions on Aerospace and Electronic Systems*, vol. 1, no. 40, pp. 576–590, 2004.

[32] X. Lin, Y. Bar-Shalom, and T. Kirubarajan. "Multisensor multitarget bias estimation for general asynchronous sensors". *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 3, pp. 899–921, 2005.

[33] M. Mallick and T. Kirubarajan. "Multi-sensor single target bearing–only tracking in clutter". *NASA STI/Recon Technical Report N*, vol. 3, pp. 15779, 2001.

[34] D. W. McMichael and N. N. Okello. "Maximum likelihood registration of dissimilar sensors". *The First Australian Data Fusion Symposium*, pp. 31–34, November 1996.

[35] D. Mušicki. "Bearings–only multi-sensor maneuvering target tracking". *Systems & Control Letters*, vol. 57, no. 3, pp. 216–221, 2008.

[36] S. C. Nardone, A. G. Lindgren, and K. F. Gong. "Fundamental properties and performance of conventional bearings–only target motion analysis". *IEEE Transactions on Automatic Control*, vol. 29, pp. 775–787, 1984.

[37] N. Okello and B. Ristic. "Maximum likelihood registration for multiple dissimilar sensors". *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 3, pp. 1074–1083, July 2003.

[38] Y. Qi, Z. Jing, and S. Hu. "Modified maximum likelihood registration based on information fusion". *Chinese Optics Letters*, vol. 5, no. 11, pp. 639–641, 2007.

[39] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. vol. 685, Artech House, Boston, MA, 2004.

[40] K. C. Sharman and G. D. McClurkin. "Genetic algorithms for maximum likelihood parameter estimation". *International Conference on Acoustics, Speech, and Signal Processing*, pp. 2716–2719, Glasgow, Scotland, May 1989.

[41] Q. Song and Y. He. "A real–time registration algorithm for passive sensors with TOA and angle biases". *3rd International Congress on Image and Signal Processing*, vol. 9, pp. 4170–4173, October 2010.

[42] E. Taghavi, R. Tharmarasa, T. Kirubarajan, and Y. Bar-Shalom. "A practical bias estimation algorithm for multisensor–multitarget tracking". *IEEE Transaction on Aerospace and Electronic Systems*, February 2016.

[43] E. Taghavi, R. Tharmarasa, T. Kirubarajan, and Y. Bar-Shalom. "Bias estimation for practical distributed multiradar-multitarget tracking systems". *15th International Conference on Information Fusion*, pp. 1304–1311, Istanbul, Turkey, July 2013.

[44] B. Xu and Z. Wang. "Biased bearings–only parameter estimation for bistatic system". *Journal of Electronics (China)*, vol. 24, no. 3, pp. 326–331, 2007.

[45] B. Xu, Z. Wu, and Z. Wang. "On the Cramér–Rao lower bound for biased bearings–only maneuvering target tracking". *Signal Processing*, vol. 87, no. 12, pp. 3175–3189, 2007.

[46] J. Yosinski, N. Coult, and R. Paffenroth. "Network-centric angle only tracking". *SPIE Optical Engineering+ Applications*, pp. 74450O–74450O, International Society for Optics and Photonics, 2009.

The following chapter is a reproduction of a ready to submit paper to IEEE Transactions on Aerospace and Electronics Systems:

Ehsan Taghavi, R. Tharmarasa, T. Kirubarajan and Mike McDonald
Geo–registration and Geo–location Using Two Airborne Video Sensors, To be submitted to *IEEE Transactions on Aerospace and Electronics Systems*, April 2016.

Chapter 4

Geo–registration and Geo–location Using Two Airborne Video Sensors

4.1 Abstract

Geo–registration and geo–location of data collected by video sensors such as electro–optical and infrared cameras are two fundamental steps in the airborne surveillance of ground targets. With the availability of high resolution imaging sensors and detailed mapping or terrain data sources, video data plays an increasingly important role in modern surveillance platforms like unmanned aerial vehicles and airborne, ground or maritime surveillance systems. Surveillance systems without any compensation for the inevitable sensor registration errors, i.e., biases, may make geo–location erroneous and renders the surveillance platform less effective for precision targeting. This paper deals with the modeling of sensor biases in geo–location and proposes a method to estimate them. The proposed method leads to a minimization problem with a nonlinear cost function. Detailed derivation of the bias model is given along with

an algorithm to find the bias parameters. The achievable lower–bounds for debiased geo–location are provided and simulations are used to demonstrate the validity of the proposed method.

4.2 Introduction

The process of assessing a target's location in world coordinates [10], i.e., geo–location, has many real–world applications. Airborne surveillance platforms, e.g., unmanned aerial vehicles (UAV), are commonly used to locate ground or maritime targets. Such platforms are used for mapping, intelligence gathering and reconnaissance [4, 21, 13, 25]. The effectiveness of these tasks depends on the accuracy of the location estimates of ground–based objects. These systems use aerial images from one or more video sensors mounted on one or more airborne platforms to find the locations of the targets of interest. Some of the algorithms used in airborne video surveillance solve a localization problem with imaging data from a stationary platform, e.g., a blimp or rotorcraft [6, 29]. Gimbaled cameras on–board a small, fixed–wing UAV are commonly used to locate ground targets and determine their locations in the world or inertial frame [2]. Tracking in world frame of reference facilitates the tracking of multiple targets distributed over a large area using sensors that are widely separated.

Most previous works in the literature on target geo–location involve unmanned ground vehicles and controlled laboratory settings [29, 23] or stationary air vehicles [6, 29]. In order to have a precise geo–location of a target, geo–registration must also be addressed. Geo–registration corresponds to identifying systematic errors in the measurements and removing them from the measurements for further processing to improve geo–location. Geo–registration methods that use pre–existing geo–referenced

imagery for geo–registration are proposed in [14, 15]. In [18], cross-modality of multiple images is used fro geo–registration. In [2], a minimum variance approach is used to address the problem of geo–registration. Creating a geo–referenced mosaic from aerial videos is considered in [24] with rather stringent assumptions such as highly accurate Inertial Measurement Unit (IMU) data.

In this paper, the geo–registration problem is addressed in a systematic manner similar to those approaches typically used in radar and bearing–only sensor registration [26, 27]. Starting with target geo–location, a matrix based formulation is proposed in this paper to derive a novel geo–registration algorithm. Although similar matrix based formulations were attempted in [2, 8, 11, 30] for geo–location, neither the problem of geo–registration was not addressed and no particular model was given for the source of the error in geo–registration. That is, biases contributing to the errors in geo–location were not addressed properly. The primary focus of this paper is to explicitly model the biases and decouple them from the (ideal) bias–free geo–location model in a scenario with two video sensors that surveil common targets in their respective fields of view.

To model the biases, first, a mathematical representation of target geo–location is provided to incorporate the effect of bias parameters. Next, using some approximations that are valid in real scenarios, the biases are decoupled from the true target geo–location. This is a crucial step in estimating biases from geo–registration. Unless the biases are decoupled from other parameters, a stacked vector consisting of target geo–location parameters and bias must be considered, which increases computational complexity. After decoupling, the next step is to create a pseudo–measurement to combine

the information from the two video sensors and reconstruct the biases. Using the information from the common targets in both image planes, a new pseudo–measurement that is mathematically sound and practically implementable is proposed. The proposed method can estimate the biases directly from pseudo–measurements without the need to estimate the parameters in the geo–location formulation. The new method is more efficient in the sense that it estimates the actual values of the biases in a decoupled manner rather than minimizing the error in the final target geo–location in a stacked vector formulation. The decoupled formulation is also more computationally efficient.

The proposed approach provides a comprehensive geo–registration and geo–location solution with all the details necessary to implement it in a real airborne surveillance system. Further, through various simulations, it is shown that the proposed method can handle large biases regardless of bias parameter intensity. Moreover, the debiased geo–locations obtained with the proposed method are shown to be as good as bias–free target geo–locations in terms of distance errors.

The paper is organized as follows: The problem of target geo–location for a single target is defined in Section 4.3 along with discussions on different coordinate frames that are used in the paper. Section 4.4 is devoted to modeling the biases and decoupling from target geo–location. In Section 4.5, a new method is proposed for geo–registration based on the models given in Section 4.4. Simulation results are presented in Section 4.6. Conclusions are discussed in Section 4.7.

4.3 Basics of Geo–location

To locate a target in world coordinates (i.e., to geo–locate) and to estimate and correct the biases (i.e., to geo–register), one needs to transform the pixels on the image plane to positions in world coordinates and vice versa. The approach presented in [2] is discussed in detail as the necessary background to propose the geo–registration method. Starting with a discussion on the orientation of an airborne platform, this section is devoted to defining the transformations used in the formulation of the new geo–location algorithm.

4.3.1 Coordinate Systems

Assuming that the position of the airborne platform is defined as $\begin{bmatrix} x_{\text{UAV}} & y_{\text{UAV}} & z_{\text{UAV}} \end{bmatrix}^{\text{T}}$ in inertial frame (X_I, Y_I, Z_I), in which X_I points to the North, Y_I points to the East, and Z_I points to the center of the earth, other related coordinates can be defined as follows and are shown in Figures 4.1 and 4.2:

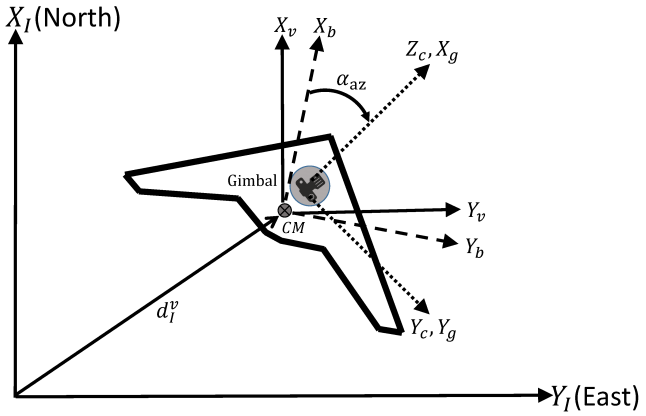

Figure 4.1: Lateral view of coordinate frames [2]

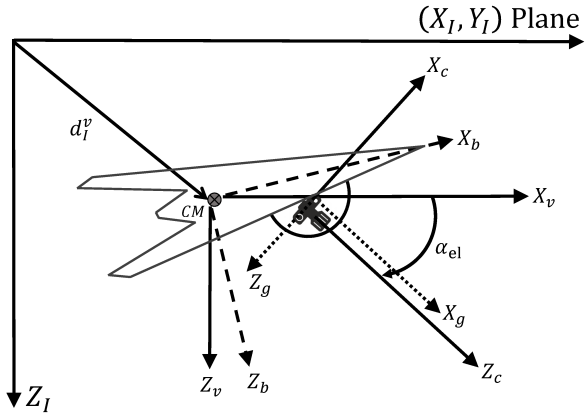

Figure 4.2: Longitudinal view of coordinate frames [2]

- **Vehicle frame** (X_v, Y_v, Z_v): originates at the platform's center of mass with the same orientation as the inertial frame.

- **Body frame** (X_b, Y_b, Z_b): originates at the platform's center of mass with X_b directed toward the nose, Y_b directed out the right wing and Z_b directed towards the body of the platform.

- **Gimbal frame** (X_g, Y_g, Z_g): originates at the gimbal's rotation center with X_g pointing along the optical axis, Z_g pointing down in image plane and Y_g pointing right in the image plane.

- **Camera frame** (X_c, Y_c, Z_c): originates at the camera's optical center with X_c pointing up in the image, Y_c pointing right in the image plane and Z_c being directed along the optical axis.

4.3.2 Homogeneous Transformation

To address translational and rotational transformations from one frame to another, a unified mathematical framework is needed. Translational transformation [1] can be described by a homogeneous transformation matrix H as

$$H = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

so that the final transformed vector v, which is a translational displacement of vector q by a distance vector d, is obtained as

$$v = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x+a \\ y+b \\ z+c \\ 1 \end{bmatrix} \tag{4.2}$$

where $q = \begin{bmatrix} x & y & z & 1 \end{bmatrix}^{\mathrm{T}}$ and $d = \begin{bmatrix} a & b & c \end{bmatrix}$. To define the rotational displacement in a right–handed rectangular coordinate frame, one can split the transformation into $R(x, \alpha)$, $R(y, \beta)$ and $R(z, \gamma)$, where $R(\cdot, \cdot)$ is a homogeneous transformation. Further, the first argument of $R(\cdot, \cdot)$ refers to the axis while the second one refers to the angle of rotation around the axis in the first argument, respectively (see Figure 4.3).

Defining the matrices such that the first three rows of the transformation matrix correspond to x, y and z axes of the reference frame, respectively, and that the

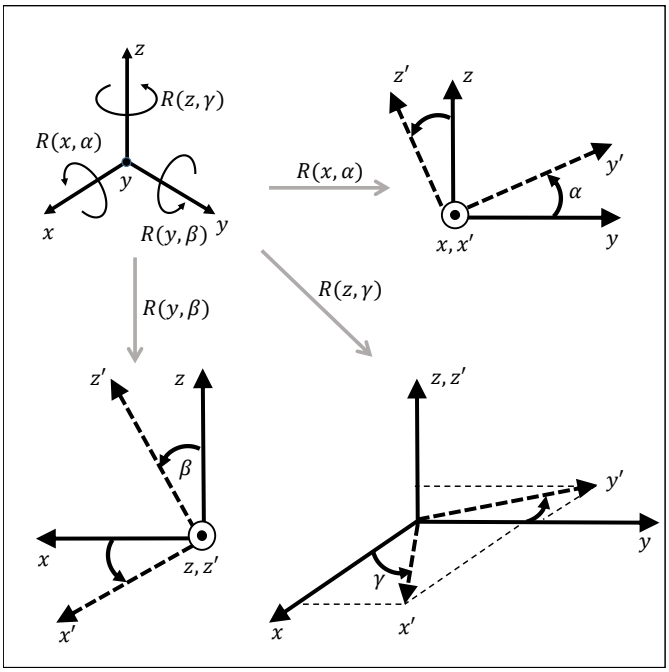

Figure 4.3: Rotational transformations

first three columns refer to x', y' and z' axes of the rotated frame, respectively, the transformation matrices can be defined as

$$
\begin{aligned}
R(x,\alpha) &= \begin{bmatrix} \cos(0) & \cos\left(\frac{\pi}{2}\right) & \cos\left(\frac{\pi}{2}\right) & 0 \\ \cos\left(\frac{\pi}{2}\right) & \cos(\alpha) & \cos\left(\frac{\pi}{2}+\alpha\right) & 0 \\ \cos\left(\frac{\pi}{2}\right) & \cos\left(\frac{\pi}{2}-\alpha\right) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{4.3}
$$

$$R(y,\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.4}$$

and

$$R(z,\gamma) = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.5}$$

Finally, one can write the transformation matrix T_i^j as

$$T_i^j = \begin{bmatrix} R_i^j & -d_i^j \\ \underline{\mathbf{0}} & 1 \end{bmatrix} \tag{4.6}$$

where $\underline{\mathbf{0}} \in \mathbb{R}^3$ is a row vector of zeros, d_i^j is the displacement vector resolved in the j^{th} coordinate and

$$R_i^j = R_i^j(x,\alpha)R_i^j(y,\beta)R_i^j(z,\gamma) \tag{4.7}$$

where $R_i^j(\cdot,\cdot)$ is the 3×3 upper–left matrix in $R(\cdot,\cdot)$. The inverse transformation is

then given by

$$T_j^i \triangleq \left(T_i^j\right)^{-1} = \begin{bmatrix} \left(R_i^j\right)^{\mathrm{T}} & \left(R_i^j\right)^{\mathrm{T}} d_i^j \\ \underline{\mathbf{0}} & 1 \end{bmatrix} \tag{4.8}$$

Now that a well–defined transformation is in hand, one can write the frame to frame transformations as follows:

Inertial to vehicle frame transformation

The transformation from the inertial frame to the vehicle frame is only a translational one from the origin of the inertial frame to the new frame of the vehicle. Thus, T_I^v can be defined as

$$T_I^v = \begin{bmatrix} I_{3\times 3} & -d_I^v \\ \underline{\mathbf{0}} & 1 \end{bmatrix} \tag{4.9}$$

where $I_{n\times n}$ is an n by n identity matrix and

$$d_I^v = \begin{bmatrix} x_{\mathrm{p}} & y_{\mathrm{p}} & -h_{\mathrm{p}} \end{bmatrix}^{\mathrm{T}} \tag{4.10}$$

Note that x_{p} and y_{p} represent the North and East location of the platform that come from a Global Positioning System (GPS) [19] sensor on the vehicle, and h_{p} is the platform's altitude that can be measured by a barometric pressure sensor or an altimeter radar. Note that GPS and altitude measurements are perturbed by sensor noise.

Vehicle to body frame transformation

Since the origins of the vehicle frame and body frame are the same, this is only a rotational transformation based on the measured Euler angles. Representing platform's roll, pitch and heading angles as ϕ, θ and ψ, respectively, the vehicle to body frame transformation can be defined as

$$T_v^b = \begin{bmatrix} R_v^b & 0 \\ \underline{\mathbf{0}} & 1 \end{bmatrix} \tag{4.11}$$

where

$$R_v^b = \begin{bmatrix} \cos(\theta)\cos(\psi) & \cos(\theta)\sin(\psi) & -\sin(\theta) \\ \sin(\phi)\sin(\theta)\cos(\psi) - \cos(\phi)\sin(\psi) & \sin(\phi)\sin(\theta)\sin(\psi) + \cos(\phi)\cos(\psi) & \sin(\phi)\cos(\theta) \\ \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) & \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi) & \cos(\phi)cos(\theta) \end{bmatrix} \tag{4.12}$$

The Euler angles can be estimated by a two–stage Kalman filter [9, 2] using rate gyros for the propagation model and accelerometers for the measurement update.

Body to gimbal frame transformation

In order to go from body to gimbal frame, both translational and rotational displacements should be considered. The translational displacement, which is resolved in gimbal frame, is dependent on the location of the platform's center of mass with respect to gimbal's rotation center and is denoted by d_b^g. The rotational displacement

is denoted by R_b^g. To model R_b^g correctly, the azimuth and elevation measurements of the camera are needed. Let α_{az} denote the azimuth angle of the rotation about Z_g, and α_{el} the elevation angle of rotation about Y_g after the former rotation of α_{el}. The transformation is given by

$$T_b^g = \begin{bmatrix} R_b^g & -d_b^g \\ \underline{\mathbf{0}} & 1 \end{bmatrix} \tag{4.13}$$

where

$$\begin{aligned} R_b^g &= R_b^g(y, \alpha_{\text{el}})\, R_b^g(x, \alpha_{\text{az}}) \\ &= \begin{bmatrix} \cos(\alpha_{\text{el}}) & 0 & \sin(\alpha_{\text{el}}) \\ 0 & 1 & 0 \\ -\sin(\alpha_{\text{el}}) & 0 & \cos(\alpha_{\text{el}}) \end{bmatrix} \begin{bmatrix} \cos(\alpha_{\text{az}}) & \sin(\alpha_{\text{az}}) & 0 \\ -\sin(\alpha_{\text{az}}) & \cos(\alpha_{\text{az}}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos(\alpha_{\text{el}})\cos(\alpha_{\text{az}}) & \cos(\alpha_{\text{el}})\sin(\alpha_{\text{az}}) & \sin(\alpha_{\text{el}}) \\ -\sin(\alpha_{\text{el}}) & \cos(\alpha_{\text{az}}) & 0 \\ -\sin(\alpha_{\text{el}})\cos(\alpha_{\text{az}}) & -\sin(\alpha_{\text{el}})\sin(\alpha_{\text{az}}) & \cos(\alpha_{\text{el}}) \end{bmatrix} \end{aligned} \tag{4.14}$$

Gimbal to camera frame transformation

Going from gimbal to camera frame depends on the translational displacement d_g^c, which describes the location of the gimbal's rotation center relative to the camera's optical center and is resolved in the camera's coordinate frame. Moreover, the transformation is dependent on rotational displacement R_g^c, which aligns the camera's coordinate frame with that of the gimbal. Note that in the coordinate frames introduced in Section 4.3.1, $X_c = -Z_g$ and $Z_c = X_g$. As result, the transformation can be

defined as

$$T_g^c = \begin{bmatrix} R_g^c & -d_g^c \\ \underline{\mathbf{0}} & 1 \end{bmatrix} \tag{4.15}$$

where

$$R_g^c = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{4.16}$$

4.3.3 Camera model

To better understand how a camera works in the framework given in this work, a simple camera projection model is shown in Figure 4.4, where the origin of the coordinate frame (X_c, Y_c, Z_c) is set at the camera center with elements measure in meters. Furthermore, the frame $(X_{im}, Y_{im}, Z_c - f)$ which its units are in meters, is set at the image plane center. Finally, the frame (X_{ip}, Y_{ip}) which is a coordinate in pixel unit, has its origin at the upper–left hand corner of the image.

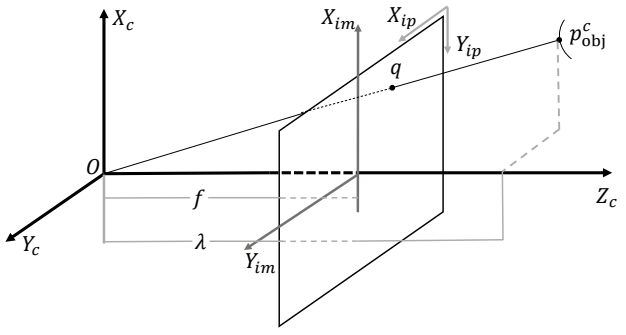

Figure 4.4: Camera image plane and its associated parameters [2]

Starting with defining point $q = \begin{bmatrix} x_{ip} & y_{ip} & 1 & 1 \end{bmatrix}^{\mathrm{T}}$ as the homogeneous projection of point $p^c_{\mathrm{obj}} = \begin{bmatrix} p_x & p_y & p_z & 1 \end{bmatrix}^{\mathrm{T}}$ onto the image plane in pixels, one can go from pixels to meters in the image plane as [28]

$$x_{im} = (-y_{ip} + \mathrm{O}_y)\, S_y \tag{4.17}$$

$$y_{im} = (x_{ip} - \mathrm{O}_x)\, S_x \tag{4.18}$$

where the units of the pairs (x_{ip}, y_{ip}) and (x_{im}, y_{im}) are pixels and meters, respectively. The offsets to the center of the frame are measured by O_x and O_y for x and y, respectively and, S_x and S_y denote the corresponding conversion factors from pixels to meters, respectively. Using the similar triangles in Figure 4.4, the relationship between p^c_{obj} and (x_{im}, y_{im}) is given by

$$\frac{x_{im}}{f} = \frac{p_x}{p_z} \tag{4.19}$$

$$\frac{y_{im}}{f} = \frac{p_y}{p_z} \tag{4.20}$$

where f is the focal length of the camera. The relationship between p^c_{obj} and q in (X_c, Y_c, Z_c) and (X_{ip}, Y_{ip}) frames is given by

$$\Lambda q = C p^c_{\mathrm{obj}} \tag{4.21}$$

where the calibration matrix C is

$$C = \begin{bmatrix} 0 & f_x & \mathrm{O}_x & 0 \\ -f_y & 0 & \mathrm{O}_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.22}$$

and matrix Λ is defined as

$$\Lambda = \begin{bmatrix} \lambda I_{3\times 3} & 0 \\ \underline{\mathbf{0}} & 1 \end{bmatrix} \tag{4.23}$$

while the remainder of the parameters in (4.22) and (4.23) are defined as

$$f_x \triangleq \frac{f}{S_x} \tag{4.24}$$

$$f_y \triangleq \frac{f}{S_y} \tag{4.25}$$

$$\lambda \triangleq p_z \tag{4.26}$$

4.3.4 Image depth

To use (4.21) and calculate the geo–location of the target, a method to calculate the image depth λ which refers to the distance along the camera's optical axis to the object of interest in the image was suggested in [17] and [2]. Note that a similar technique can be derived if terrain data is available to the user. In the case of having a non–flat earth, target's latitude and longitude can be obtained using the algorithm in [30, 8]. By using the latitude and longitude of the target, target altitude can be

calculated using the terrain data. Then, platform's altitude above target can be found and image depth can be obtained using this information. Note that if terrain data is not available, the length of target can be used to obtain the platform's altitude above target [31].

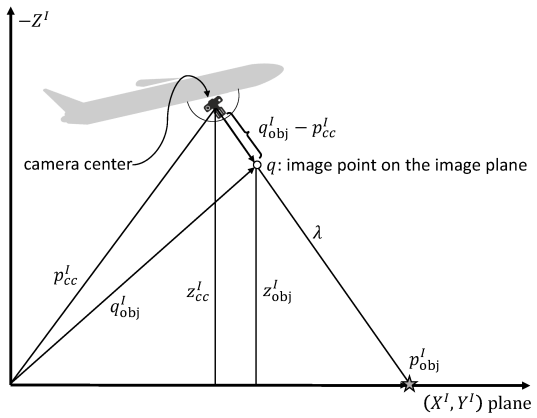

Figure 4.5: Camera image plane and related parameters (X_{ip}, Y_{ip})

Assume p_{cc} is the location of the camera's optical center. Further, if p_{cc} is resolved in the camera frame, one has

$$p^c_{cc} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^{\mathrm{T}} \tag{4.27}$$

Thus, resolving p^c_{cc} in the inertial frame gives

$$p^I_{cc} = \begin{bmatrix} x^I_{cc} \\ y^I_{cc} \\ z^I_{cc} \\ 1 \end{bmatrix} = \left(T^c_g T^g_b T^b_v T^v_I\right)^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.28}$$

Based on flat earth model and the assumption of using homogeneous transformations (Figure 4.5), q^I_{obj} and p^I_{cc} are related by their z–components as

$$0 = z^I_{cc} + \lambda \left(z^I_{\text{obj}} - z^I_{cc} \right) \tag{4.29}$$

In the case of using high–altitude airborne platforms or satellite SAR imagery [16], one should consider the standards related to modeling curvature of the earth such as WGS–84 [32]. Note that from Figures 4.4 and 4.5, based on the homogeneous transformation assumption, the projection of $\left(q^I_{\text{obj}} - p^I_{cc} \right)$ onto $\left(-Z^I, \left(X^I, Y^I \right) \right)$ plane is equal to the unit vector.

4.3.5 Target location

In the final step of transforming a point on the image plane to its real location on the inertial frame (geo–location), one can write

$$\Lambda q = C p^c_{\text{obj}} = C T^c_g T^g_b T^b_v T^v_I p^I_{\text{obj}} \tag{4.30}$$

Solving (4.30) for p^I_{obj} gives

$$\begin{aligned} p^I_{\text{obj}} &= \left(C T^c_g T^g_b T^b_v T^v_I \right)^{-1} \Lambda q \\ &= T^I_v T^v_b T^b_g T^g_c C^{-1} \Lambda q \end{aligned} \tag{4.31}$$

In a more computationally efficient form, (4.31) can be written as

$$\bar{p}^I_{\text{obj}} = \bar{p}^I_{cc} + \lambda \left(\bar{q}^I_{\text{obj}} - \bar{p}^I_{cc} \right) \tag{4.32}$$

where $\bar{p}$ represents the first three elements of p.

With above equations, it is possible to estimate the geo–location of the desired target if the user receives telemetry information from the platform. However, this is only possible when there is no noise or bias in measuring the parameters such as the Euler angles and gimbal azimuth and elevation. This paper primarily focuses on estimating the constant biases in azimuth and elevation of the platform and on correcting them in the original measurements to be able to locate the targets more accurately.

4.4 Bias Modeling of Video Sensor Measurements

Zero–mean noise and constant biases can result from sensor noise and geometric uncertainties. As discussed in [2], Recursive Least Square (RLS) methods can effectively remove the zero–mean noise. However, biases cannot be removed with RLS methods. As such, a separate algorithm must be adopted to estimate and correct the effect of biases in platform measurements.

According to (4.31), the bias parameters vector is given by

$$\mathbf{b} = \begin{bmatrix} \bar{\alpha}_{\text{az}} & \bar{\alpha}_{\text{el}} & \bar{\phi} & \bar{\theta} & \bar{\psi} & \bar{z} \end{bmatrix}^{\text{T}} \tag{4.33}$$

where $\bar{\alpha}_{\text{az}}$, $\bar{\alpha}_{\text{el}}$, $\bar{\phi}$, $\bar{\theta}$, $\bar{\psi}$, and $\bar{z}$ are the biases associated with the measurements of gimbal azimuth, gimbal elevation, roll, pitch, yaw and altitude, respectively.

For the airborne platform model given in this paper, the center of mass and the gimbal center are located close to each other. Thus, according to Figure 4.1, the rotation axes for heading, denoted as ψ, and gimbal azimuth angle, denoted as

α_{az}, are closely aligned. This alignment between these two parameters makes the related biases numerically indistinguishable. Assuming that the platform's path is in circular shape, the gimbal azimuth angle will be close to 90°, which makes the airframe roll axis and the gimbal elevation axis to be nearly aligned. This makes the biases in ϕ and α_{el} almost indistinguishable. It is worth noting that even when the flight path is not circular, if the body pitch angle is close to zero, biases in roll and heading measurements are indistinguishable from biases in gimbal elevation and azimuth measurement, respectively [2]. Therefore, the bias vector under consideration can be reduced to

$$\mathbf{b} = \begin{bmatrix} \bar{\alpha}_{\text{az}} & \bar{\alpha}_{\text{el}} \end{bmatrix}^{\text{T}} \tag{4.34}$$

To find the above remaining biases, i.e., biases in gimbal azimuth and elevation, with the assumption of having two platforms looking with overlapping fields of view, a novel approach is depicted here to decouple the biases from target geo–location as it is discussed in this section. Starting with the geo–location of a target and the assumption of having biases in gimbal elevation and azimuth only, one has

$$\begin{aligned} p^I_{\text{obj}} &= (T^v_I)^{-1} \left(T^b_v\right)^{-1} (T^g_b)^{-1} \left(T^c_g\right)^{-1} C^{-1} \Lambda q \\ &= (T^v_I)^{-1} \left(T^b_v\right)^{-1} \left(T^g_{b,\text{free}} + T^g_{b,\text{bias}}\right)^{-1} \left(T^c_g\right)^{-1} C^{-1} \Lambda q \end{aligned} \tag{4.35}$$

where

$$T^g_b = T^g_{b,\text{free}} + T^g_{b,\text{bias}} \tag{4.36}$$

in which $T^g_{b,\text{free}}$ is the transformation without any contribution from the biases and $T^g_{b,\text{bias}}$ is the transformation with contribution only from gimbal azimuth and elevation biases. Because of the nonlinearity in the contribution of biases in T^g_b, approximations must be made in decoupling $T^g_{b,\text{free}}$ and $T^g_{b,\text{bias}}$. Starting from the rotational transformation from body to gimbal

$$R^g_b = \begin{bmatrix} \cos(\alpha_{\text{el}}+\bar{\alpha}_{\text{el}})\cos(\alpha_{\text{az}}+\bar{\alpha}_{\text{az}}) & \cos(\alpha_{\text{el}}+\bar{\alpha}_{\text{el}})\sin(\alpha_{\text{az}}+\bar{\alpha}_{\text{az}}) & \sin(\alpha_{\text{el}}+\bar{\alpha}_{\text{el}}) \\ -\sin(\alpha_{\text{el}}+\bar{\alpha}_{\text{el}}) & \cos(\alpha_{\text{az}}+\bar{\alpha}_{\text{az}}) & 0 \\ -\sin(\alpha_{\text{el}}+\bar{\alpha}_{\text{el}})\cos(\alpha_{\text{az}}+\bar{\alpha}_{\text{az}}) & -\sin(\alpha_{\text{el}}+\bar{\alpha}_{\text{el}})\sin(\alpha_{\text{az}}+\bar{\alpha}_{\text{az}}) & \cos(\alpha_{\text{el}}+\bar{\alpha}_{\text{el}}) \end{bmatrix} \tag{4.37}$$

one can further expand each term in R^g_b as

$$\begin{aligned} R^g_b(1,1) = {} & \sin(\alpha_{\text{el}})\sin(\alpha_{\text{az}})\sin(\bar{\alpha}_{\text{el}})\sin(\bar{\alpha}_{\text{az}}) + \cos(\alpha_{\text{el}})\cos(\alpha_{\text{az}})\cos(\bar{\alpha}_{\text{el}})\cos(\bar{\alpha}_{\text{az}}) \\ & - \sin(\alpha_{\text{el}})\cos(\alpha_{\text{az}})\sin(\bar{\alpha}_{\text{el}})\cos(\bar{\alpha}_{\text{az}}) - \cos(\alpha_{\text{el}})\sin(\alpha_{\text{az}})\cos(\bar{\alpha}_{\text{el}})\sin(\bar{\alpha}_{\text{az}}) \end{aligned} \tag{4.38}$$

$$\begin{aligned} R^g_b(1,2) = {} & \cos(\alpha_{\text{el}})\cos(\alpha_{\text{az}})\cos(\bar{\alpha}_{\text{el}})\sin(\bar{\alpha}_{\text{az}}) - \sin(\alpha_{\text{el}})\cos(\alpha_{\text{az}})\sin(\bar{\alpha}_{\text{el}})\sin(\bar{\alpha}_{\text{az}}) \\ & + \cos(\alpha_{\text{el}})\sin(\alpha_{\text{az}})\cos(\bar{\alpha}_{\text{el}})\cos(\bar{\alpha}_{\text{az}}) - \sin(\alpha_{\text{el}})\sin(\alpha_{\text{az}})\sin(\bar{\alpha}_{\text{el}})\cos(\bar{\alpha}_{\text{az}}) \end{aligned} \tag{4.39}$$

$$R_b^g(1,3) = \cos(\alpha_{\text{el}})\sin(\bar{\alpha}_{\text{el}}) + \sin(\alpha_{\text{el}})\cos(\bar{\alpha}_{\text{el}}) \tag{4.40}$$

$$R_b^g(2,1) = -\cos(\alpha_{\text{el}})\sin(\bar{\alpha}_{\text{el}}) - \sin(\alpha_{\text{el}})\cos(\bar{\alpha}_{\text{el}}) \tag{4.41}$$

$$R_b^g(2,2) = \cos(\alpha_{\text{az}})\cos(\bar{\alpha}_{\text{az}}) - \sin(\alpha_{\text{az}})\sin(\bar{\alpha}_{\text{az}}) \tag{4.42}$$

$$\begin{aligned} R_b^g(3,1) = & -\cos(\alpha_{\text{el}})\cos(\alpha_{\text{az}})\sin(\bar{\alpha}_{\text{el}})\cos(\bar{\alpha}_{\text{az}}) + \cos(\alpha_{\text{el}})\sin(\alpha_{\text{az}})\sin(\bar{\alpha}_{\text{el}})\sin(\bar{\alpha}_{\text{az}}) \\ & -\sin(\alpha_{\text{el}})\cos(\alpha_{\text{az}})\cos(\bar{\alpha}_{\text{el}})\cos(\bar{\alpha}_{\text{az}}) + \sin(\alpha_{\text{el}})\sin(\alpha_{\text{az}})\cos(\bar{\alpha}_{\text{el}})\sin(\bar{\alpha}_{\text{az}}) \end{aligned} \tag{4.43}$$

$$\begin{aligned} R_b^g(3,2) = & -\cos(\alpha_{\text{el}})\cos(\alpha_{\text{az}})\sin(\bar{\alpha}_{\text{el}})\sin(\bar{\alpha}_{\text{az}}) - \cos(\alpha_{\text{el}})\sin(\alpha_{\text{az}})\sin(\bar{\alpha}_{\text{el}})\cos(\bar{\alpha}_{\text{az}}) \\ & -\sin(\alpha_{\text{el}})\cos(\alpha_{\text{az}})\cos(\bar{\alpha}_{\text{el}})\sin(\bar{\alpha}_{\text{az}}) - \sin(\alpha_{\text{el}})\sin(\alpha_{\text{az}})\cos(\bar{\alpha}_{\text{el}})\cos(\bar{\alpha}_{\text{az}}) \end{aligned} \tag{4.44}$$

and

$$R_b^g(3,3) = \cos(\alpha_{\text{el}})\cos(\bar{\alpha}_{\text{el}}) - \sin(\alpha_{\text{el}})\sin(\bar{\alpha}_{\text{el}}) \tag{4.45}$$

Here, some approximations are applied to (4.38)–(4.45). Assuming the biases in gimbal azimuth and elevation are distributed near zero, one can apply the following

approximation:

$$\sin(\bar{\alpha}_{\text{el}})\sin(\bar{\alpha}_{\text{az}}) \approx 0 \tag{4.46}$$

$$\cos(\bar{\alpha}_{\text{el}})\cos(\bar{\alpha}_{\text{az}}) \approx 1 \tag{4.47}$$

$$\cos(\bar{\alpha}_{\text{el}}) \approx 1 \tag{4.48}$$

$$\cos(\bar{\alpha}_{\text{az}}) \approx 1 \tag{4.49}$$

With the approximations given in (4.46)–(4.49), one can rewrite the terms in R_b^g as

$$\begin{aligned} R_b^g(1,1) \approx\ & (\cos(\alpha_{\text{el}})\cos(\alpha_{\text{az}})) \\ & + (-\sin(\alpha_{\text{el}})\cos(\alpha_{\text{az}})\sin(\bar{\alpha}_{\text{el}}) - \cos(\alpha_{\text{el}})\sin(\alpha_{\text{az}})\sin(\bar{\alpha}_{\text{az}})) \end{aligned} \tag{4.50}$$

$$\begin{aligned} R_b^g(1,2) \approx\ & (\cos(\alpha_{\text{el}})\sin(\alpha_{\text{az}})) \\ & + (\cos(\alpha_{\text{el}})\cos(\alpha_{\text{az}})\sin(\bar{\alpha}_{\text{az}}) - \sin(\alpha_{\text{el}})\sin(\alpha_{\text{az}})\sin(\bar{\alpha}_{\text{el}})) \end{aligned} \tag{4.51}$$

$$R_b^g(1,3) \approx (\sin(\alpha_{\text{el}})) + (\cos(\alpha_{\text{el}})\sin(\bar{\alpha}_{\text{el}})) \tag{4.52}$$

$$R_b^g(2,1) \approx (-\sin(\alpha_{\text{el}})) + (-\cos(\alpha_{\text{el}})\sin(\bar{\alpha}_{\text{el}})) \tag{4.53}$$

$$R_b^g(2,2) \approx (\cos(\alpha_{\text{az}})) + (-\sin(\alpha_{\text{az}})\sin(\bar{\alpha}_{\text{az}})) \tag{4.54}$$

$$\begin{aligned} R_b^g(3,1) &\approx (-\sin(\alpha_{\rm el})\cos(\alpha_{\rm az})) \\ &\quad + (-\cos(\alpha_{\rm el})\cos(\alpha_{\rm az})\sin(\bar{\alpha}_{\rm el}) + \sin(\alpha_{\rm el})\sin(\alpha_{\rm az})\sin(\bar{\alpha}_{\rm az})) \end{aligned} \quad (4.55)$$

$$\begin{aligned} R_b^g(3,2) &\approx (-\sin(\alpha_{\rm el})\sin(\alpha_{\rm az})) \\ &\quad + (-\cos(\alpha_{\rm el})\sin(\alpha_{\rm az})\sin(\bar{\alpha}_{\rm el}) - \sin(\alpha_{\rm el})\cos(\alpha_{\rm az})\sin(\bar{\alpha}_{\rm az})) \end{aligned} \quad (4.56)$$

and

$$R_b^g(3,3) \approx (\cos(\alpha_{\rm el})) + (-\sin(\alpha_{\rm el})\sin(\bar{\alpha}_{\rm el})) \quad (4.57)$$

In (4.50)–(4.57), the first expression inside the parentheses quantifies the contribution of $T^g_{b,\rm free}$ while that inside the second parentheses quantifies the contribution of $T^g_{b,\rm bias}$, respectively. Thus, the contribution of the biases can be decoupled from bias–free part as

$$R_b^g = R^g_{b,\rm free} + R^g_{b,\rm bias} \quad (4.58)$$

Using (4.58), one can reform the transformation matrix T_b^g as

$$T_b^g = \begin{bmatrix} R^g_{b,\rm free} & -d_b^g \\ \underline{\mathbf{0}} & 1 \end{bmatrix} + \begin{bmatrix} R^g_{b,\rm bias} & (\underline{\mathbf{0}})^{\rm T} \\ \underline{\mathbf{0}} & 0 \end{bmatrix} \quad (4.59)$$

Assuming $\left(T^g_{b,\rm free} + T^g_{b,\rm bias}\right)$ and $T^g_{b,\rm free}$ are both invertible and $T^g_{b,\rm bias}$ has rank $0 < r < 4$,

if one let

$$T^g_{b,\text{bias}} = T^g_{b,\text{bias},1} + \cdots + T^g_{b,\text{bias},r} \tag{4.60}$$

where $T^g_{b,\text{bias},i}$ has rank 1, and each of

$$D^{-1}_{k+1} = D^{-1}_k - g_k D^{-1}_k T^g_{b,\text{bias},k} D^{-1}_k \tag{4.61}$$

is non singular, then by starting with $D_1 = T^g_{b,\text{free}}$, one has [12]

$$\left(T^g_{b,\text{free}} + T^g_{b,\text{bias}}\right)^{-1} = D^{-1}_r - g_r D^{-1}_r T^g_{b,\text{bias},r} D^{-1}_r \tag{4.62}$$

where

$$g_k = \frac{1}{1 + \text{trace}\left(D^{-1}_k T^g_{b,\text{bias},k}\right)} \tag{4.63}$$

Finally, the decoupled geo–location measurement can be written as

$$\begin{aligned} p^I_{\text{obj}} &= (T^v_I)^{-1} \left(T^b_v\right)^{-1} \left(D^{-1}_r - g_r D^{-1}_r T^g_{b,\text{bias},r} D^{-1}_r\right) \left(T^c_g\right)^{-1} C^{-1} \Lambda q \\ &= \left[(T^v_I)^{-1} \left(T^b_v\right)^{-1} \left(D^{-1}_r\right) \left(T^c_g\right)^{-1} C^{-1} \Lambda q\right] \\ &\quad - \left[(T^v_I)^{-1} \left(T^b_v\right)^{-1} \left(g_r D^{-1}_r T^g_{b,\text{bias},r} D^{-1}_r\right) \left(T^c_g\right)^{-1} C^{-1} \Lambda q\right] \end{aligned} \tag{4.64}$$

in which the first term in the brackets on the right–hand side is solely the contribution from the bias–free measurements and the rest is a function of gimbal elevation and azimuth biases. This gives the opportunity to propose a method to only estimate the biases and then use it to correct the measurement.

4.5 Geo–registration of Two Video Sensors

Starting with the geo–location of multiple targets at one time instant or a single target over a period of time from two video sensors, one can write

$$p^I_{\text{obj},i} = \left(C_i T^c_{g,i} T^g_{b,i} T^b_{v,i} T^v_{I,i}\right)^{-1} \Lambda_i q_i \qquad i = 1, 2 \tag{4.65}$$

where extra index i is for video sensor number. If both video sensors point to the same targets in a single frame or a specific target over a period of time, one can, derive a pseudo–measurement that can address the effect of biases as

$$z_b(k) = p^I_{\text{obj},1}(k) - p^I_{\text{obj},2}(k) \tag{4.66}$$

where

$$\begin{aligned} z_b(k) = & \left(T^v_{I,2}\right)^{-1} \left(T^b_{v,2}\right)^{-1} \left(g_{r,2} D^{-1}_{r,2} T^g_{b,\text{bias},r,2} D^{-1}_{r,2}\right) \left(T^c_{g,2}\right)^{-1} C_2^{-1} \Lambda_2 q_2 \\ & - \left(T^v_{I,1}\right)^{-1} \left(T^b_{v,1}\right)^{-1} \left(g_{r,1} D^{-1}_{r,1} T^g_{b,\text{bias},r,1} D^{-1}_{r,1}\right) \left(T^c_{g,1}\right)^{-1} C_1^{-1} \Lambda_1 q_1 \end{aligned} \tag{4.67}$$

in which the contribution of the zero–mean noises are neglected and time index is omitted for simplicity.

To estimate the biases, one needs to form a cost function for a batch estimator. By neglecting the covariance of the pseudo–measurements due to its severe non–linearity, the Euclidean norm can be used to form a cost function as

$$Z_{\mathbf{b}}(k) = \parallel z_b(k) - h(k, \mathbf{b}) \parallel \tag{4.68}$$

where

$$h(k, \mathbf{b}) \;=\; p^{I}_{\text{obj},1}(k, \bar{\alpha}_{\text{az}}, \bar{\alpha}_{\text{el}}) - p^{I}_{\text{obj},2}(k, \bar{\alpha}_{\text{az}}, \bar{\alpha}_{\text{el}}) \tag{4.69}$$

The Euclidean norm is a non–negative cost function with $\mathcal{O}(K)$ that is widely used as a model fit problem to non–linear functions to adjust the desired parameters, here $\mathbf{b}$, so that the model fits the pseudo–measurement [3, 5, 22].

Assuming that there are $k = 1, ..., K$ independent measurements, the cost function can be modified as

$$Z_{\mathbf{b}} \;=\; \sum_{k=1}^{K} Z_{\mathbf{b}}(k) \tag{4.70}$$

Finally, the vector $\hat{\mathbf{b}}$ that minimizes the function $Z_{\mathbf{b}}$ can be written as

$$\hat{\mathbf{b}} \;=\; \arg\min_{\mathbf{b}} \; Z_{\mathbf{b}} \tag{4.71}$$

4.6 Simulation Results

To evaluate the performance of the proposed method, a series of simulations are presented in this section. The targets in the fields of view of the two sensors are on the surface of the Earth (inertial frame) and measurements corresponding to video detections are generated assuming that the center of the target is detected by an image processing algorithm. Results on real data are not possible since video data from multiple airborne platforms are not available in the open literature. Note that the proposed approach can be applied to multiple frames from one target over a

certain amount of time.

The assumption here is that residual errors and biases are present in all the parameters in addition to gimbal elevation and azimuth. Although the biases in the Euler angles and altitude are not observable, including them makes the simulations more realistic. The cameras' intrinsic parameters, their extrinsic parameters and associated noise standard deviations are given in Tables 4.1, 4.2 and 4.3, respectively.

Table 4.1: The camera's intrinsic parameters

Parameter	Camera 1	Camera 2
f_x	548	548
f_y	556	556
O_x	316.4	520
O_y	223.0	498

Table 4.2: The camera's extrinsic parameters

Parameter	Camera 1	Camera 2
x_{UAV}	-50m	100m
y_{UAV}	-100m	25m
h_{UAV}	350m	350m
θ	4°	3°
ψ	2°	3°
ϕ	3°	3°
α_{az}	81°	79°
α_{el}	85°	82°
$d_b^g(x)$	0.05m	0.05m
$d_b^g(y)$	0.05m	0.05m
$d_b^g(z)$	0.02m	0.02m
$d_g^c(x)$	0.01m	0.01m
$d_g^c(y)$	-0.01m	-0.01m
$d_g^c(z)$	-0.05m	-0.05m

For the unobservable biases, that are related to the Euler angles, it is assumed

Table 4.3: The camera parameters' standard deviations

Standard deviation	Camera 1	Camera 2
$\sigma_{x_{\text{UAV}}}$	2m	2m
$\sigma_{y_{\text{UAV}}}$	2m	2m
$\sigma_{h_{\text{UAV}}}$	2m	2m
σ_{θ}	0.1°	0.1°
σ_{ψ}	0.1°	0.1°
σ_{ϕ}	0.1°	0.1°
$\sigma_{\alpha_{\text{az}}}$	1°	1°
$\sigma_{\alpha_{\text{el}}}$	1°	1°

that the bias in each parameter is 0.5° for both sensors. The bias in the altitude measurement, $\bar{z}$, is omitted in the simulations. It is worth noting that however in the simulations it is assumed that cameras are mounted on two different platforms, the proposed method is not constrained to this setting and same results can be found by mounting both cameras on a single platform.

There are four parameters that need to be estimated with the proposed method, namely,

$$\mathbf{b} = \begin{bmatrix} \bar{\alpha}_{\text{az},1} & \bar{\alpha}_{\text{el},1} & \bar{\alpha}_{\text{az},2} & \bar{\alpha}_{\text{el},2} \end{bmatrix} \tag{4.72}$$

where the second indices correspond the sensor number. To evaluate the performance of the proposed method, a wide range of values (eight sets) for $\mathbf{b}$, as listed in Table 4.4, are used in the simulations. First, an optimization algorithm is applied to solve (4.71) for eight sets of parameter given in Table 4.4 over 100 Monte Carlo runs. Further, the Root Mean Squared Errors (RMSE) are calculated for the bias parameter estimates to show how accurate the estimation is based on the models given in Section 4.5. In this paper, the Genetic Algorithm (GA) [7, 20] is used to solve the optimization

Table 4.4: Test sets for gimbal azimuth and elevation biases

Test set	$\bar{\alpha}_{\text{az},1}$	$\bar{\alpha}_{\text{el},1}$	$\bar{\alpha}_{\text{az},2}$	$\bar{\alpha}_{\text{el},2}$
1	$-2°$	$4.5°$	$2°$	$-3°$
2	$-3°$	$-3°$	$-3°$	$-3°$
3	$3°$	$3°$	$3°$	$3°$
4	$0°$	$0°$	$0°$	$0°$
5	$4°$	$0°$	$0°$	$0°$
6	$0°$	$4°$	$0°$	$0°$
7	$0°$	$0°$	$4°$	$0°$
8	$0°$	$0°$	$0°$	$4°$

Table 4.5: Parameters for the Genetic Algorithm

Parameter	Value
Lower bound	$-5°$
Upper bound	$5°$
Number of generations	100
Tolerance value	1×10^{-15}

problem in (4.71). Note that, although the GA is a batch ML estimator, the length of the window can be varied to meet the real–time requirements. The parameters used in the simulations are given in Table 4.5.

The RMSE of the estimated biases for the eight test sets are given in Table 4.6.

Table 4.6: RMSE of gimbal azimuth and elevation bias estimates

Test set	$\text{RMSE}_{\bar{\alpha}_{\text{az},1}}$	$\text{RMSE}_{\bar{\alpha}_{\text{el},1}}$	$\text{RMSE}_{\bar{\alpha}_{\text{az},2}}$	$\text{RMSE}_{\bar{\alpha}_{\text{el},2}}$
1	$0.32°$	$0.71°$	$0.24°$	$0.50°$
2	$0.81°$	$0.72°$	$0.62°$	$0.62°$
3	$0.76°$	$0.67°$	$0.51°$	$0.66°$
4	$0.71°$	$0.63°$	$0.56°$	$0.52°$
5	$0.73°$	$0.94°$	$0.67°$	$0.89°$
6	$0.48°$	$0.76°$	$0.33°$	$0.72°$
7	$0.99°$	$0.65°$	$0.65°$	$0.56°$
8	$0.98°$	$0.74°$	$0.73°$	$0.67°$

As the numbers in Table 4.6 suggest, for a wide range of biases in gimbal azimuth

and elevation, the proposed method can estimate the parameters accurately. To demonstrate the significance of the proposed method and its ability to handle bias estimation, the estimated biases are used to correct the target geo–location. Then, the corrected (debiased) target geo–locations are compared with the biased geo–locations as well as the absolute lower bound that can be achieved, i.e., the geo–location with the perfect knowledge of gimbal azimuth and elevation biases but not that of the Euler angle biases. Figures 4.6 and 4.7 show the geo–location errors for test sets 1 and 2 for biased, debiased and ideal bias–free measurements.

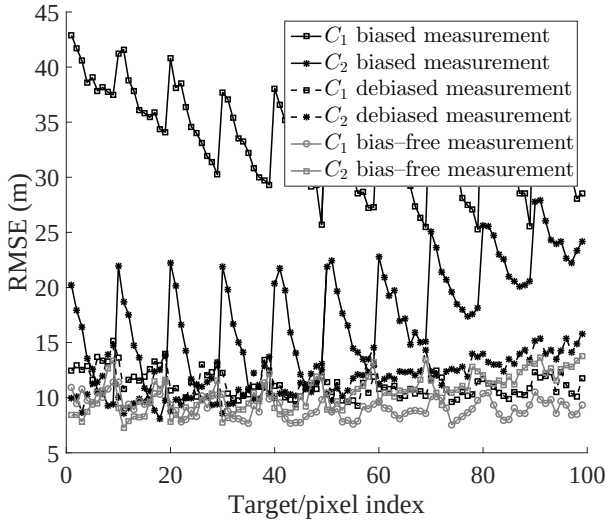

Figure 4.6: RMSE of geo–location estimates of common targets in test set 1 (C_i: Camera i)

Figures 4.6 and 4.7 show the improvement in target geo–location that can be achieved by debiasing. It is worth noting that, the debiased geo–locations match the ideal bias–free geo–locations and do not have the large ripples in RMSE that are observed when biased geo–locations are used. The results show significant advantage of using debiased target geo–locations obtained using the proposed method over biased

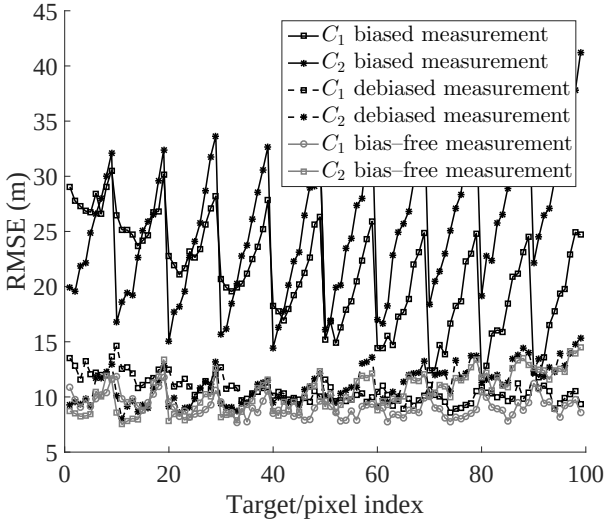

Figure 4.7: RMSE of geo–location estimates of common targets in test set 2 (C_i: Camera i)

ones. The advantage will be even more significant when debiased measurements from multiple sensors are used.

Finally, to better understand the effects of correction in all the test sets, the average RMSE for all the common targets are shown in Tables 4.7 and 4.8 for biased, debiased and ideal bias–free geo–locations. Tables 4.7 and 4.8 show that for all common targets on the image plane, the results of the debiased geo–location estimates are similar to the idea bias–free ones in terms of RMSE, which indicates bias correction has a positive and significant effect on the final geo–location estimates.

4.7 Conclusions

Modeling of biases in data collected from video sensors, especially those on airborne platforms, and compensating them is crucial for accurate geo–location. In this paper,

Table 4.7: Average RMSE of biased, udebiased and bias–free geo–location estimates for Camera 1

Test sets	Biased	Debiased	Bias–free
1	33.50m	11.39m	9.20m
2	20.82m	9.82m	9.13m
3	24.45m	10.10m	8.98m
4	9.21m	10.58m	9.36m
5	14.32m	10.35m	9.52m
6	30.16m	11.42m	9.15m
7	9.31m	11.10m	9.03m
8	8.86m	11.05m	8.82m

Table 4.8: Average RMSE of biased, udebiased and bias–free geo–location estimates for Camera 2

Test sets	Biased	Debaised	Bias–free
1	17.01m	11.84m	10.21m
2	25.27m	10.26m	10.25m
3	33.74m	10.25m	10.16m
4	10.33m	10.79m	10.25m
5	10.27m	10.78m	10.18m
6	10.20m	12.34m	10.33m
7	20.26m	10.91m	10.51m
8	20.17m	11.12m	10.23m

a comprehensive algorithm for modeling the biases in the Euler angles, altitude, and gimbal azimuth and elevation and removing them was presented. Further, the effects of gimbal azimuth and elevation biases was discussed along with the modeling and decoupling of the biases from geo–location formulation for two video sensors. A minimization problem was used to estimate the biases for practical scenarios in Section 4.6 and the results were compared with ideal bias–free results to demonstrate the accuracy of the proposed method. The nature of the presented algorithms lends itself for computationally efficient parallel implementations using graphics processing units (GPU).

Bibliography

[1] T. Bajd. *Robotics*. vol. 43, Springer Science & Business Media, Berlin, Germany,2010.

[2] D. B. Barber, J. D. Redding, T. W. McLain, R. W. Beard, and C. N. Taylor. "Vision-based target geo–location using a fixed–wing miniature air vehicle". *Journal of Intelligent and Robotic Systems*, vol. 47, no. 4, pp. 361–382, 2006.

[3] D. P. Bertsekas. *Nonlinear Programming*. Athena scientific, Belmont, MA, USA, 1999.

[4] E. Bone and C. Bolkcom. "Unmanned aerial vehicles: Background and issues for congress". DTIC Document, 2003.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, United Kingdom, 2004.

[6] L. Chaimowicz, B. Grocholsky, J. E. Keller, V. Kumar, and C. Taylor. "Experiments in multirobot air–ground coordination". *International Conference on Robotics and Automation*, vol. 4, pp. 4053–4058, New Orleans, Louisiana, April 2004.

[7] L. Davis. *Handbook of Genetic Algorithms*. vol. 115, Van Nostrand Reinhold, New York, 1991.

[8] V. N. Dobrokhodov, I. Kaminer, K. D. Jones, and R. Ghabcheloo. "Vision–based tracking and motion estimation for moving targets using small UAVs". *American Control Conference*, pp. 1428–1433, Minneapolis, Minnesota, USA, June 2006.

[9] A. M. Eldredge. "Improved state estimation for miniature air vehicles". Master's Thesis, Brigham Young University, 2006.

[10] N. Farmani, L. Sun, and D. Pack. "An optimal sensor management technique for unmanned aerial vehicles tracking multiple mobile ground targets". *International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 570–576, Orlando, Florida, USA, May 2014.

[11] D. Gibbins, P. Roberts, and L. Swierkowski. "A video geo–location and image enhancement tool for small unmanned air vehicles (UAVs)". *Intelligent Sensors, Sensor Networks and Information Processing Conference*, pp. 469–473, Melbourne, Australia, December 2004.

[12] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, UK, 2012.

[13] H. R. Hosseinpoor, F. Samadzadegan, and F. Javan. "Pricise target geo–location based on integeration of thermal video imagery and RTK GPS in UAVs". *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, no. 1, pp. 333–338, Copernicus GmbH, November, 2015.

[14] R. Kumar, S. Samarasekera, S. Hsu, and K. Hanna. "Registration of highly–oblique and zoomed in aerial video to reference imagery". *International Conference on Pattern Recognition*, vol. 4, pp. 303–307, Barcelona, Spain, September 2000.

[15] R. Kumar, H. S. Sawhney, J. C. Asmuth, A. Pope, and S. Hsu. "Registration of video to geo–referenced imagery". *International Conference on Pattern Recognition*, vol. 2, pp. 1393–1400, Brisbane, Australia, August 1998.

[16] H. Liu, Z. Zhao, and K. C. Jezek. "Correction of positional errors and geometric distortions in topographic maps and DEMs using a rigorous SAR simulation technique". *Photogrammetric Engineering & Remote Sensing*, vol. 70, no. 9, pp. 1031–1042, September 2004.

[17] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. *An Invitation to 3D Vision: From Images to Geometric Models.* vol. 26, Springer Science & Business Media, Berlin, Germany, 2012.

[18] T. Mckay and H. Hirsch. "A fast, accurate, cross–modality image geo-registration and target/object detection algorithm. *Proc. SPIE, Defense, Security, and Sensing*, pp. 874708–874708, Washington, USA, May 2013.

[19] P. Misra and P. Enge. *Global Positioning System: Signals, Measurements and Performance*. Ganga-Jamuna Press, Lincoln, MA, USA, 2006.

[20] P. Pospichal, J. Jaros, and J. Schwarz. "Parallel genetic algorithm on the CUDA architecture". *Applications of Evolutionary Computation*.

[21] J. D. Redding, T. W. McLain, R. W. Beard, and C. N. Taylor. "Vision-based target localization from a fixed-wing miniature air vehicle". *American Control Conference*, pp. 2862–2867, Minneapolis, Minnesota, USA, June 2006.

[22] Ø. Ryan, G. Dahl, and K. Mørken. Nonlinear optimization lecture notes for the course MAT-INF2360. University of Oslo, February 2014.

[23] P. Saeedi, D. G. Lowe, and P. D. Lawrence. "3D localization and tracking in unknown environments", vol. 1, pp. 1297–1303, Taipel, Taiwan, September 2003.

[24] H. J. Schultz, A. R. Hanson, E. M. Riseman, F. Stolle, Z. Zhu, C. D. Hayward, and D. Slaymaker. "System for real–time generation of geo–referenced terrain models". *SPIE Enabling Technologies for Law Enforcement*, pp. 29–36, Boston, Massachusetts, November, 2000.

[25] R. Sharma, J. Yoder, H. Kwon, and D. Pack. "Vision based mobile target geo–localization and target discrimination using Bayes detection theory. *The 11th International Symposium on Distributed Autonomous Robotic Systems*, pp. 59–71, Springer, Heidelberg, Germany, 2014.

[26] E. Taghavi, R. Tharmarasa, T. Kirubarajan, Y. Bar-Shalom, and M. McDonald. "A practical bias estimation algorithm for multisensor–multitarget tracking".

IEEE Transaction on Aerospace and Electronic Systems, vol. 51, no. 4, October 2015.

[27] E. Taghavi, R. Tharmarasa, T. Kirubarajan, and M. McDonald. "Multisensor–multitarget bearing–only sensor registration". *IEEE Transaction on Aerospace and Electronic Systems*, To be appear in vol. 52, no. 4, August 2016.

[28] E. Trucco and A. Verri. *Introductory Techniques for 3D Computer Vision.* vol. 201, Prentice Hall Englewood Cliffs, N.J., USA, 1998.

[29] R. Vidal and S. Sastry. "Vision–based detection of autonomous vehicles for pursuit–evasion games". *IFAC World Congress on Automatic Control*, Barcelona, Spain, July 2002.

[30] I. H. Wang, V. N. Dobrokhodov, I. I. Kaminer, and K. D. Jones. "On vision–based target tracking and range estimation for small UAVs". *AIAA Guidance, Navigation and Control Conference*, San Francisco, California, USA, August 2005.

[31] R. Zhang and H. Liu. "Vision–based relative altitude estimation of small unmanned aerial vehicles in target localization". *American Control Conference*, pp. 4622–4627, San Francisco, California, USA, June 2011.

[32] H. Zhao, B. Zhang, C. Wu, Z. Zuo, and Z. Chen. "Development of a coordinate transformation method for direct geo–referencing in map projection frames". *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 77, pp. 94–103, 2013.

Chapter 5

Conclusion

5.1 Research Summary

5.1.1 List of contributions

The contributions of this thesis are listed as follows:

1. Developing a computationally efficient multisensor–multitarget bias estimation algorithm for radar measurements.

2. Developing a method that can handle large network of sensors as well as communication loss with different rates by the use of tracklet methodology.

3. Mathematically formulating the pseudo–measurement for biases in bearing–only sensors and designing a batch estimator to estimate the biases.

4. Calculating the CRLB for both radar and bearing–only bias estimation algorithms.

5. Formulating a multisensor video registration problem.

6. Formulating the decoupled bias effect for video sensor geo–location report and calculating the pseudo–measurement that addresses the biases directly.

7. Developing an algorithm to solve the unknown parameters in the geo–registration measurements.

8. Delivering object–oriented code, with proper design for further research and industrial developments.

5.2 Future Research

There are few open problems that can be considered for future research as extension to the work that has been done in this thesis. Most related to the articles presented here, motion compensation and tracking for ships can be considered as the next step. Furthermore, there are two interesting problem that were raised in geo–registration of video sensors. First, calculation of covariance matrix for the geo–location estimates can be considered as a separate topic. Only after calculating the covariance matrix can once design an optimal estimator based on the biased measurements. Finally, computational aspects of implementing the geo–registration algorithm that was proposed in Chapter 4 can be investigated in more detail with an aim for parallel implementation.

This page is left blank intentionally.