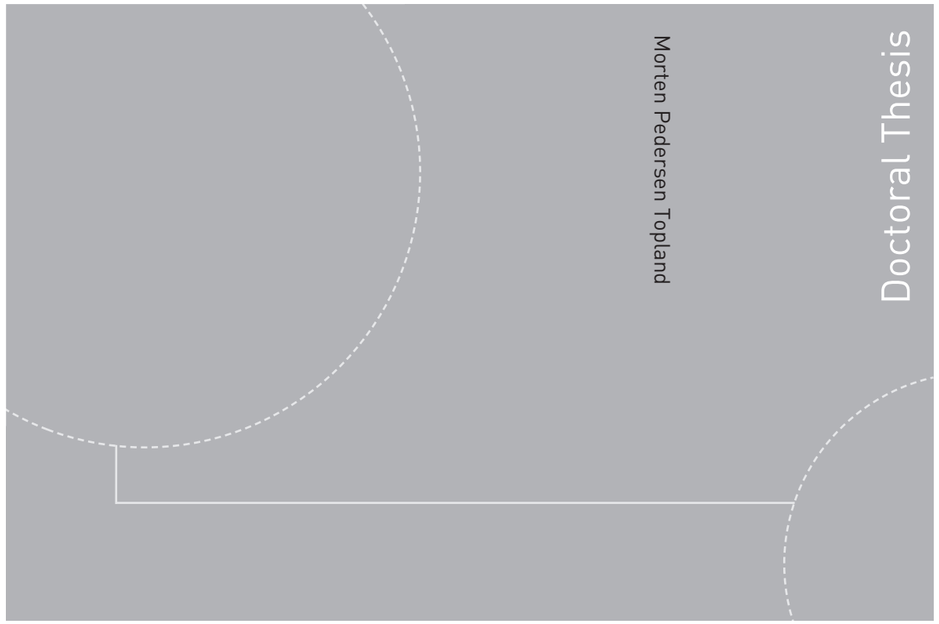
ISBN 978-82-326-1412-7 (printed version)
ISBN 978-82-326-1413-4 (electronic version)
ISSN 1503-8181

NTNU
Norwegian University of
Science and Technology




Doctoral theses at NTNU, 2016:36

Morten Pedersen Topland

# Joint Multisensor Fusion and Tracking Using Distributed Radars

Doctoral theses at NTNU, 2016:36

NTNU

**NTNU**
Norwegian University of
Science and Technology
Faculty of Information Technology,
Mathematics and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Morten Pedersen Topland

# Joint Multisensor Fusion and Tracking Using Distributed Radars

Thesis for the degree of Philosophiae Doctor

Trondheim, February 2016

Norwegian University of Science and Technology
Faculty of Information Technology,
Mathematics and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

**NTNU**
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology,
Mathematics and Electrical Engineering
Department of Engineering Cybernetics



ISBN 978-82-326-1412-7 (printed version)
ISBN 978-82-326-1413-4 (electronic version)
ISSN 1503-8181

ITK-report: 2016-4-W

Doctoral theses at NTNU, 2016:36

Printed by Skipnes Kommunikasjon as

# Preface

This thesis is submitted in fulfillment of the requirements for the degree Philosophiae Doctor (PhD) at the Norwegian University of Science and Technology (NTNU). It treats the problem of tracking a target, such as an aircraft, with multiple sensors. Target tracking with a single sensor, such as a radar, is a mature field of study where publications date back to the sixties. The extension to multiple sensors has received more recent attention. In target tracking the position measurements from all sensors are combined (fused) to create a track consisting of a target ID, position and velocity. In theory multiple sensors improve accuracy, reliability and coverage, but with regards to accuracy there are some challenges. These stem from the fact that the target position measurements may differ substantially between sensors, causing problems in the measurement fusion process. This problem is addressed herein.

I have a Master's degree in Engineering Cybernetics from the Norwegian University of Science and Technology (NTNU) in Trondheim, Norway. I started this work in 2005, and it has been performed at University Graduate Center (UNIK) at Kjeller in Norway under the supervision of Adjunct Professor Oddvar Hallingstad (NTNU). I was part of the Sea Air and Land Surveillance (SEALS) research project. In 2006 I spent half a year in Canada at McMaster University in Hamilton, Ontario. During that time I was part of the research group of Professor Thiagalingam Kirubarajan, a group which specializes in multisensor-multitarget tracking and information fusion.

I would like to thank the parties providing the funding for this work as part

of the SEALS project, which was the Research Council of Norway, the Norwegian Defence Research Establishment (FFI), Kongsberg Defence & Aerospace (KDA) and Indra Navia AS (IN). Furthermore I would like to thank my research colleagues at UNIK for fruitful discussions and exciting table tennis matches, Morten Stakkeland, Edmund Brekke, Anders Rodningsby, Kjetil Aanonsen, Are Willumsen, Oyvind Hegrenas and Kjell Magne Fauske. From my time at McMaster University I would like to thank Dr Abhijit Sinha for valuable discussions. From the SEALS partners I would like to thank Tore Smestad (FFI) and Svein Fagerlund (KDA) for valuable feedback, and I would like to thank previous IN employees Borge Midtgaard and Tom Borge, in addition to current employee Siv Brendbakken, for supporting my work during my time at UNIK. I started working for IN in 2008, and would like to thank Dr Hans Christian Guren and Terje Dalen for their support, and for giving me time to work on my thesis. Special thanks to Dr Thiagalingam Kirubarajan for letting me participate in his research group at McMaster University, and to my supervisor Dr Oddvar Hallingstad for supporting me all this time. Lastly I thank my wife and kids for love, support and time to finish this thesis.

# Summary

This thesis deals with multisensor fusion in the presence of systematic errors in the context of target tracking. A typical target tracking problem consists of multiple sensors producing position measurements of multiple targets, for instance aircraft. The goal is to establish tracks on all targets that are observed by the sensors. A track usually consists of an id, such as a track number, target position, and target velocity. The systematic errors are modeled as measurement biases. If unaccounted for these biases may lead to inaccurate estimates of the target state (position and in particular velocity) and a single target may appear as several targets if the biases are large enough (ghost tracks). Furthermore, if all sensors are biased it is challenging to find an unbiased estimate of target state with respect to a coordinate system independent of the sensors.

In this thesis the sensors are radars producing measurements in 3D. The systematic errors (biases) are called alignment bias, location bias and sensor bias. The first two are related to sensor deployment, as they describe errors in orientation (misalignment) and sensor placement (location). The sensor bias addresses errors caused by sensor imperfections. These biases are estimated relative to a sensor independent coordinate system and relative to a sensor of reference (master sensor). A novel distinction is made in this context, where a universal bias estimator (UBE) is used relative to sensor independent coordinates, while an absolute bias estimator (ABE) is used relative to a master sensor. The estimability of the biases is investigated using a novel estimability index, which quantifies whether a bias can be estimated more accurately with the available measurements. The estimability index is based on the Cramer-Rao Lower Bound.

The study of estimability is used to determine a multisensor-multitarget scenario where several bias estimators are compared with respect to performance using a Monte Carlo simulation. The simulation includes alignment, location and sensor biases, and all sensors are affected. The estimators are evaluated in sensor independent coordinates and master sensor coordinates. Two Kalman Filter (KF) based estimators are used as references. A lower bound is represented by a KF where the bias values are known, while an upper bound is represented by a KF where the measurement noise is increased to reflect the biases present. The alignment, location and sensor biases contain three elements each, to a total of nine bias values to estimate per sensor. The UBE performs well (below the upper bound) in sensor independent coordinates when one of the sensor bias values are removed from the simulation, estimating eight bias values per sensor. Performance is close to the lower bound when the location bias only is removed, yielding six bias values per sensor. In master sensor coordinates the ABE has the best performance. However a simplified version has almost identical performance. It is called the Relative Bias Estimator (RBE), and it neglects the biases of the master sensor. This is a popular assumption in the literature, and this study confirms that this simplification should be preferred in an implementation.

Possible extensions of this work are explored. First curved target motion is explored by letting the target move at constant altitude above the Earth. The curvature of the trajectory results in increased bias estimability. However, observing this curvature requires observing the target for a long time with high accuracy. This is challenging in practice, and therefore this path was not explored further.

Second, extending the application to Air Traffic Control (ATC) is considered. At airports radars typically produce 2D measurements, so to extend the developed 3D bias estimators it is necessary to incorporate altitude measurements from the aircraft Mode C transponders with these 2D measurements. The altitude measurements are quantized and received with a coarse resolution which may have a negative impact on bias estimator performance since the vertical velocity estimate becomes unstable. Several estimators are developed to estimate altitude and vertical velocity, and these are tested on real measurement data for a performance comparison. The main contribution is the use of the Interacting Multiple Model (IMM) and Unscented Kalman Filter (UKF) based estimators on quantized real world measurements. The UKF produces the best performance for long term altitude predictions, meaning that its vertical velocity estimate is the most stable.

# Contents

# Nomenclature

## Acronyms

**ABE** Absolute Bias Estimator

**AEE** Average Euclidean Error

**ATC** Air Traffic Control

**BLUE** Best Linear Unbiased Estimator

**cpdf** conditional probability density function

**CRLB** Cramer-Rao Lower Bound

**CTF** Coordinate Transformation Function

**CTM** Coordinate Transformation Matrix

**ECEF** Earth Centered Earth Fixed

**EM** Expectation Maximization

**EKF** Extended Kalman Filter

**FIM** Fisher Information Matrix

**IMM** Interacting Multiple Model

**KF** Kalman Filter

**KFBU** bias ignorant Kalman Filter which tracks in $\{u\}$

**KFBS** bias ignorant Kalman Filter which tracks in $\{s_0\}$

**KFU** unbiased Kalman Filter

**LS** Least Squares

**MC** Monte Carlo

**ML** Maximum Likelihood

**MMSE** Minimum Mean Square Error

**NED** North East Down

**NIS** Normalized Innovation Squared

| | |
|---|---|
| **PDA** | Probabilistic Data Association |
| **pdf** | probability density function |
| **PF** | Particle Filter |
| **PSR** | Primary Surveillance Radar |
| **RBE** | Relative Bias Estimator |
| **RMSE** | Root Mean Square Error |
| **SABE** | Standard Absolute Bias Estimator |
| **SBTM** | Sensor Bias Transformation Matrix |
| **SIR** | Sequential Importance Resampling |
| **SUBE** | Standard Universal Bias Estimator |
| **SMR** | Surface Movement Radar |
| **SSR** | Secondary Surveillance Radar |
| **UBE** | Universal Bias Estimator |
| **UKF** | Unscented Kalman Filter |

# Notation

| | |
|---|---|
| $x$ | Scalar. |
| $\vec{d}_{AB}$ | Geometric (not coordinatized) displacement vector from point $A$ to point $B$. |
| $\{a\}$ | A Cartesian coordinate system given by the origin and coordinate axes along base vectors. |
| $\{u\}$ | A universal (sensor independent) Cartesian coordinate system. |
| $\{s_i\}$ | The local Cartesian coordinate system of Sensor $i$. |
| $\underline{x}^a$ | Column vector coordinatized (written) in $\{a\}$ with dimension $n_x$. |
| $X$ | Matrix. |
| $[X]_{ij} = x_{ij}$ | The element at index $ij$ of matrix $X$. |
| $X^a$ | Matrix coordinatized in $\{a\}$. |
| $T_a^b$ | The Coordinate Transformation Matrix which transforms vectors from $\{a\}$ to $\{b\}$. |
| $\underline{d}_{ab}^a$ | Displacement vector from the origin of $\{a\}$ to the origin of $\{b\}$, coordinatized in $\{a\}$. Note that $\underline{d}_{ab}^a = -\underline{d}_{ba}^a$. |
| $\underline{p}^a = [p_x\ p_y\ p_z]^T$ | Target position in Cartesian coordinates in $\{a\}$. |
| $\underline{\varrho}^a = [\rho\ \theta\ \phi]^T$ | Target position in spherical coordinates in $\{a\}$. |
| $\rho, \theta, \phi$ | The spherical coordinates range, azimuth and elevation respectively. |

| | |
|---|---|
| $\underline{\varrho}^a = \underline{h}_s\left(\underline{p}^a\right)$ | The coordinate transformation function $\underline{h}_s(\cdot)$ converts Cartesian coordinates to spherical coordinates. |
| $\underline{p}^a = \underline{h}_c\left(\underline{\varrho}^a\right)$ | The coordinate transformation function $\underline{h}_c(\cdot)$ converts spherical coordinates to Cartesian coordinates. |
| $C^a$ | The Coordinate Transformation Matrix for $\underline{h}_c(\cdot)$ obtained through linearization, which transforms spherical coordinates into Cartesian coordinates in $\{a\}$. |
| $I$ | The identity matrix. |
| $S\left(\underline{x}^a\right)$ | The skew symmetric form (matrix) of $\underline{x}^a$. |
| $E\left\{\underline{x}^a\right\}$ | The expected (mean) value of $\underline{x}^a$. |
| $\underline{x}^a(t)$ | Vector at continuous time $t$. |
| $\underline{x}_k^a$ | Vector at discrete time index $k$. |
| $\dot{\underline{x}^a}$ | Time differentiation of $\underline{x}^a$. |
| $\delta_{kl}$ | The Kronecker delta $\delta_{kl} = \begin{cases} 0 \text{ if } k \neq l \\ 1 \text{ if } k = l \end{cases}$. |
| $\delta(t-\tau)$ | The Dirac delta function which satisfies $f(t) = \int_{-\infty}^{\infty} f(\tau)\,\delta(t-\tau)\,d\tau$. |
| $\underline{v}_k^a \sim \mathcal{N}\left(\underline{m}^a, P^a\delta_{kl}\right)$ | The vector $\underline{v}_k^a$ is stochastic and time variant with a normal (Gaussian) distribution of mean $\underline{m}^a$ and covariance matrix $P^a$ in discrete time. |
| $\underline{v}^a(t) \sim \mathcal{N}\left(\underline{m}^a, \tilde{Q}^a\delta(t-\tau)\right)$ | The vector $\underline{v}^a(t)$ is stochastic and time variant with a normal (Gaussian) distribution of mean $\underline{m}^a$ and spectral density matrix $\tilde{Q}^a$ in continouos time. |
| $\alpha, \beta, \gamma$ | The Euler angles roll, pitch and yaw respectively. |
| $\underline{b}_a^b = [b_\alpha\ b_\beta\ b_\gamma]^T$ | A vector containing an alignment bias in $\alpha, \beta, \gamma$. |
| $\underline{b}_{ab}^a = [b_x\ b_y\ b_z]^T$ | A vector containing a location bias in $x, y, z$. |
| $\underline{b} = [b_\rho\ b_\theta\ b_\phi]^T$ | A vector containing a sensor bias in $\rho, \theta, \phi$. |
| $\underline{z}^a$ | A measurement in $\{a\}$. |
| $p\left(\underline{x}^a\right)$ | The probability density function (pdf) of the stochastic state vector $\underline{x}^a$. |
| $p\left(\underline{x}^a \mid \underline{z}^a\right)$ | The pdf of $\underline{x}^a$, given the measurement $\underline{z}^a$ and a prior pdf for $\underline{x}^a$. |
| $p\left(\underline{x}^a : \underline{z}^a\right)$ | The pdf of $\underline{x}^a$, given the measurement $\underline{z}^a$ with no prior pdf for $\underline{x}^a$. |
| $Z_k = \{\underline{z}_1, \ldots, \underline{z}_k\}$ | All measurements from discrete times 1 to $k$. |
| $\underline{z}_{i,k}^{s_i} = \underline{z}_k^{s_i}$ | Measurement from Sensor $i$ at time $k$ in coordinate system $\{s_i\}$ associated to Sensor $i$. |
| $\Delta t$ | Sampling interval. |
| $\sigma_x$ | Standard deviation of scalar stochastic variable $x$. |

| | |
|---|---|
| $\sigma_x^2$ | Variance of scalar stochastic variable $x$. |
| $\sigma_{xy}$ | Cross covariance of scalar stochastic variables $x$ and $y$. |
| $J$ | The Fisher Information Matrix. |
| $\eta$ | The estimability index. |

# 1

# Introduction

Surveillance of aircraft is performed at airports all over the world, and has a long history in defense applications. In recent years more and more effort has been put into combining (fusing) information and measurements from different sources and sensors to improve performance of aircraft surveillance systems. Combining measurements from several sensors is called *multisensor fusion*. A major goal for researchers is to find algorithms which ensure that more sensors mean better surveillance. This work addresses two of the challenges in that regard, systematic errors (measurement biases) for radars, and quantized altitude measurements for aircraft.

## 1.1 Motivation

This section presents the motivation for the work in this thesis. First we present target tracking, and second we treat multisensor fusion.

### 1.1.1 Target Tracking

Aircraft surveillance is an important part of the field of study called *target tracking*. A typical target tracking problem consists of multiple sensors producing position measurements of multiple targets, for instance aircraft. The goal is to establish *tracks* on all targets that are observed by the sensors. A track usually consists of an id, such as a track number, target position, and target velocity. More information could be included, such as target acceleration and target type.

A process producing tracks is usually called a *tracker*. The target tracking problem has several challenges:

*Data association* Which measurement belongs to which target?

*Track initiation* Is this measurement from a new target? How is the track initialized?

*Track maintenance* How does one predict where the target is heading? How accurate is the track? How accurate are the measurements?

*Track termination* When should a track be deleted?

This work focuses on handling systematic measurement errors, called measurement *biases*, during track maintenance. Data association is not treated. Track maintenance is typically done through recursive *filtering*. A filter provides estimates of the *target state*, which usually consists of position and velocity. Estimate accuracy is included through *covariance* estimates, or alternatively through the probability density function (pdf) of the estimate.

The most famous filtering algorithm is the Kalman Filter. The estimate at time step $k$ is modeled as a state vector, denoted $\underline{\hat{x}}_k$, which is an estimate of the true state vector $\underline{x}_k$. The filtering process starts with a prior estimate $\underline{\bar{x}}_k$ of the target state vector consisting of position and velocity. Using a mathematical filter model, the estimate can be propagated (predicted) to any time step. If a measurement of the target state is received at time $k+1$, $\underline{\hat{x}}_k$ is propagated to $k+1$ and updated with the measurement, yielding the posterior estimate $\underline{\hat{x}}_{k+1}$. The measurement will typically be a position measurement, and its accuracy is included through covariance information or the associated pdf. The posterior estimate is obtained by giving probability weights to the prediction based on the prior estimate, and the measurement. This weighting differs across filtering algorithms, and the ones used in this work are discussed later. Note that this type of estimation is called Bayesian estimation, as opposed to Fisher estimation where the state vector is considered an unknown constant and no prior information is available. Only the measurement is modeled stochastically.

## 1.1.2 Multisensor Fusion and Systematic Errors

### 1.1.2.1 Benefits and Challenges of Multisensor Fusion

In the early stages of aircraft surveillance, aircraft were monitored using a single radar. A radar is a sensor which emits electromagnetic waves which are reflected

by the aircraft body. The arrival time of the received reflected waves are used to calculate the aircraft position. Presently many aircraft surveillance systems use several sensors to monitor aircraft. There are several reasons for doing so:

1. Some radars produce three-dimensional measurements of aircraft position, but two-dimensional measurements are more common. In the latter case more sensors are needed to get aircraft position in 3D.

2. More sensors of different types provide more information about the aircraft.

3. More sensors ensure redundancy. If a sensor fails, another can take over.

4. More sensors achieve a better coverage of the surrounding airspace.

5. More sensors give more measurements which can be used to calculate aircraft position, which should yield improved accuracy when calculating aircraft position.

However, there are some problems associated with multisensor fusion, which originate from errors in the measurements used:

1. *Sensor registration errors* are errors caused by misplacement and misalignment of the sensor, and these lead to systematic errors (biases) in position measurements of aircraft when comparing measurements from one sensor to measurements from another.

2. Imperfections in the sensors themselves can lead to systematic errors as in item 1.

3. Measurements can be offset in time, for instance if the sensors are not synchronized or the measurements are not time stamped. Again this leads to systematic errors as in item 1.

Sensor registration errors are errors associated with deploying the sensor. First it is placed at the appropriate location, given by for instance a map or GPS (Global Positioning System). This location is uncertain, as both maps and GPS have an associated level of accuracy. Thus the sensor will get a systematic location error (bias), which is present in all position measurements from the sensor. If measurements of a given aircraft from a sensor (Sensor $i$) are compared to measurements of the same aircraft from another sensor (Sensor $j$), one will observe that the measurements from Sensor $i$ are biased with respect to Sensor $j$. If this bias

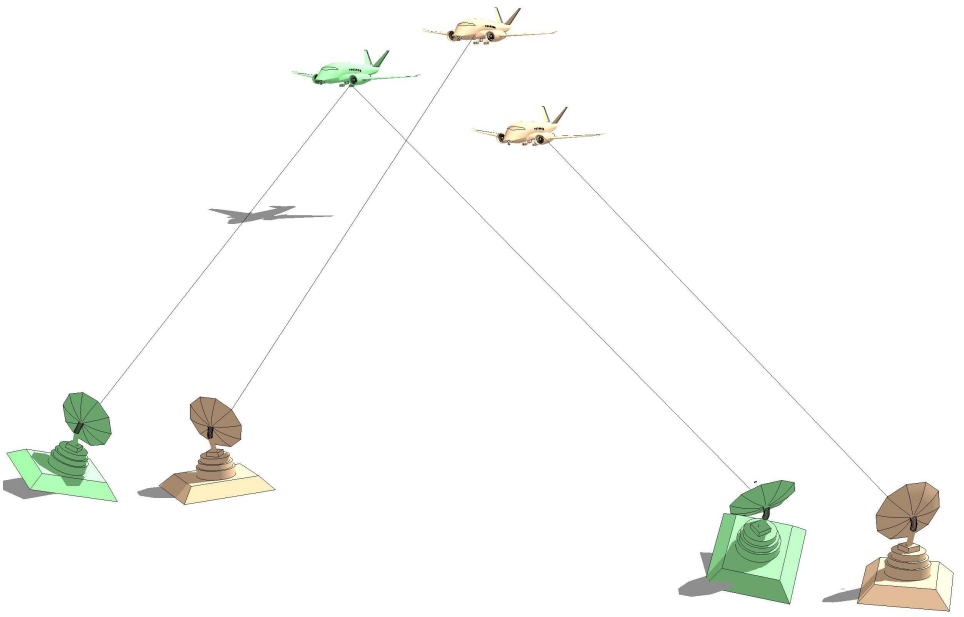


***Figure 1.1:*** *An aircraft is seen as two aircraft because of sensor registration errors. The green items represent the truth, and the red items represent the assumed sensor alignment and location, and the resulting erroneous measurements of aircraft position.*

originates from the systematic location error, it is called a *location bias* herein.

When deploying the sensor, it is aligned to match the local North, East and Down directions. Aligning it perfectly is not feasible, so when comparing measurements of an aircraft from two different sensors, the alignment error will cause a systematic error called an *alignment bias*. Other sources of systematic errors are sensor imperfections, a *sensor bias*, and unsynchronized measurement time stamps from different sensors. The latter leads to measurements being offset in time, a *time bias*, which can be caused by unsynchronized clocks or network delays when reporting the measurements. Thus we have the following measurement biases:

*Alignment Bias* Sensor registration error caused by misalignment.

*Location Bias* Sensor registration error caused by misplacement.

*Sensor Bias* Systematic measurement error due to sensor imperfections.

*Time Bias* Systematic error due to time offsets.

Measurement biases lead to several problems when using multisensor fusion to perform target tracking:

1. Biases can lead to a situation where position measurements of a specific target are so far apart that the target is identified as several targets, as in Figure 1.1.

2. Biased measurements may lead to inaccurate estimates of the target state.

3. If all sensors are biased, it is challenging to find an unbiased estimate of target state with respect to a coordinate system independent of the sensors.

For these issues to become problems in a multisensor target tracking scenario, the biases need to be large enough to have a noticeable effect on target measurements. This means that they are large enough not to be negligible compared to the measurement noise. Compensating for the existence of measurement biases can be done in several ways. A popular approach in defense applications has been to use a local tracker for each sensor, and only use the best track for each target across sensors. This concept is called Reporting Responsibility, where every target has one sensor which is responsible for tracking it. Since multiple sensors are used, both redundancy and better coverage of the airspace are ensured. However this is not multisensor fusion, since only measurements from one sensor is used to update a specific target. Thus problem 1. above is not applicable, and ignoring the measurement biases appears as a more viable option. Problem 2. above is reduced to only apply when there is a switch of responsible sensor, i.e. when the sensor with the best track for a given target changes. If this happens often, problem 2. may still be significant. Another downside of this approach is that the measurements used to update a track which is not the best for a given target may never be of use. The concept is throwing away information. Furthermore, problem 3. above is not addressed.

Multisensor fusion has the potential to improve track quality (accuracy) considerably since it allows for processing of all measurements to update the tracks. However, in the presence of measurement biases the performance of a multisensor fusion tracker may be inferior to a single sensor tracker, because of the problems noted previously. Compensating for measurement biases by estimating them is one way to gain the benefits of multisensor fusion. This technique has the potential to solve all of the problems mentioned, provided that all biases can be estimated successfully. However, perfect bias estimation is generally not feasible. The measurement noise, the measurement biases and the target motion need to be modeled mathematically. Perfect modeling is generally infeasible,

thus errors are introduced. Another important aspect is *observability* (Maybeck, 1994) and *estimability*, which apply for deterministic and stochastic systems respectively. Criterions for observability and estimability answer the following question. Given the models, measurements and the geometry of the scenario, is it possible to observe or estimate the states of all targets and the measurement biases?

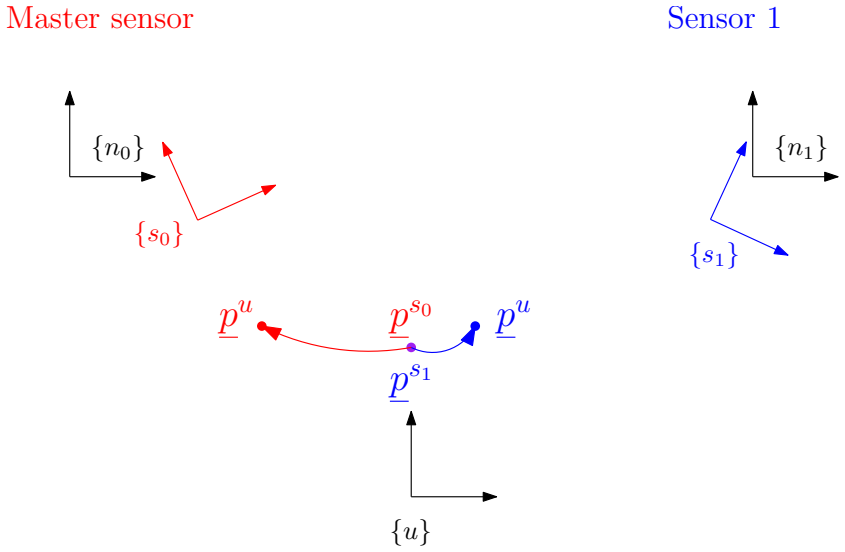



***Figure 1.2:*** *Universal Bias Estimation. The black coordinate systems $\{u\}$, $\{n_0\}$ and $\{n_1\}$ are known. The coordinate systems $\{s_0\}$ and $\{s_1\}$ have alignment biases, which are observed as rotations about the origin, and location biases, which are observed as a displacement of the origin. The curved arrows indicate coordinate transformations of target position from one coordinate system to another. When the two sensors observe the same target, its position is correct in $\{s_0\}$ and $\{s_1\}$, denoted $\underline{p}^{s_0}$ and $\underline{p}^{s_1}$ respectively. However when the target observations from Sensor 0 and Sensor 1 are transformed from $\{s_0\}$ and $\{s_1\}$ respectively to $\{u\}$, the resulting positions are different. This position error is caused by the alignment and location biases.*

#### 1.1.2.2 Universal, Absolute and Relative Bias Estimation

There are different ways of modeling and estimating measurement biases. In *universal* bias estimation the goal is to estimate all biases of all sensors with

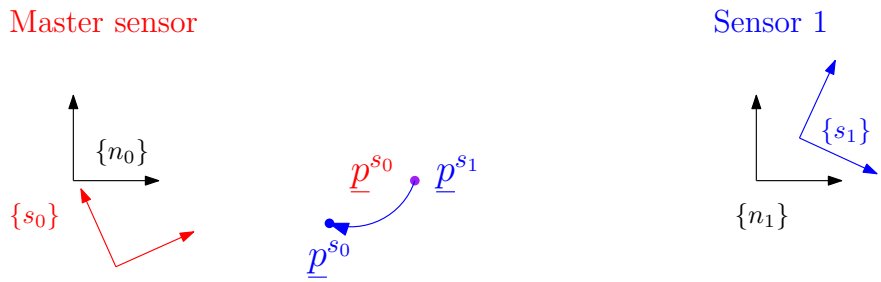



***Figure 1.3:*** *Absolute Bias Estimation. The black coordinate systems $\{n_0\}$ and $\{n_1\}$ are known. The coordinate systems $\{s_0\}$ and $\{s_1\}$ have alignment biases, which are observed as rotations about the origin, and location biases, which are observed as a displacement of the origin. The curved arrows indicate coordinate transformations of target position from one coordinate system to another. When the two sensors observe the same target, its position is correct in $\{s_0\}$ and $\{s_1\}$, denoted $\underline{p}^{s_0}$ and $\underline{p}^{s_1}$ respectively. However when the target observation from Sensor 1 is transformed from $\{s_0\}$ to $\{s_1\}$, the resulting position is different from the position observed by Sensor 0. This position error is caused by the alignment and location biases.*

respect to a sensor independent universal coordinate system. In Figure 1.2 this coordinate system is denoted $\{u\}$. If done successfully, target tracks will be unbiased in the universal coordinate system. This is a challenging problem where for instance the location bias is unestimable if no sensor locations are known exactly.

Measurement biases are usually estimated relative to a *master* sensor, in a master sensor coordinate system $\{s_0\}$ as shown in Figure 1.3. There are two typical ways of doing this, and it is usually possible to reduce the number of biases to estimate with respect to universal bias estimation, thereby increasing estimability. The first method is to reduce the amount of biases, but retaining the mathematical correctness of the model. The location bias for instance can be expressed relative to a master sensor without making simplifying assumptions leading to an inferior mathematical model. This method leads to an estimator performing *absolute* bias estimation. If done successfully, target tracks will be unbiased in the master sensor coordinate system. However the tracks will be biased with respect to a universal coordinate system.

The second method is to not estimate some or all of the measurement biases

of the master sensor, which is called *relative* bias estimation herein. In this case there are fewer biases to estimate with respect to universal and absolute bias estimation. If done successfully relative bias estimation leads to tracks that are approximately unbiased with respect to the master sensor. However, not estimating the master sensor biases may lead to simplifying assumptions in the mathematical model, making it inferior to a model used for absolute bias estimation. Note that the tracks are biased with respect to a universal coordinate system.

Although universal bias estimation may seem to be the overall best method, it may not be necessary in practice. A common military application for multi-sensor bias estimation is an air defense system, where a surface-to-air missile launcher could be connected to each radar. If the launcher and radar use the same coordinate system, the only thing that matters is that the target coordinates are correct in that system. Universal, absolute and relative bias estimation may all solve this problem, and in such cases relative bias estimation may be preferable since it will usually include less biases to estimate and better estimability.

It is sometimes necessary to know the target position in reference to a sensor independent coordinate system. In an air defense system this could mean that the launcher is not connected to a radar, but has its own sensor independent coordinates. Another case would be an Air Traffic Control (ATC) scenario where target positions with respect to a universal coordinate system tied to an airport is of interest.

The main strength of universal bias estimation is that the estimated biases can be used to compute correct target position in any known coordinate system. When the biases are estimated successfully, the sensor coordinate systems are known with respect to a known sensor independent coordinate system. The target position can then be transformed to any coordinate system known with respect to these coordinate systems.

### 1.1.3 Aircraft Altitude Determination

As stated in Section 1.1.2 the radars used for aircraft surveillance typically return measurements in two dimensions, range and bearing. Thus the aircraft altitude is missing. The altitude of an aircraft in civilian airspace is determined by Mode C transponder transmissions using Secondary Surveillance Radars (SSRs). This measurement has a resolution of 100 ft due to bandwidth constraints, resulting in measurements forming a staircase as shown in Figure 1.4. The coarse reso-

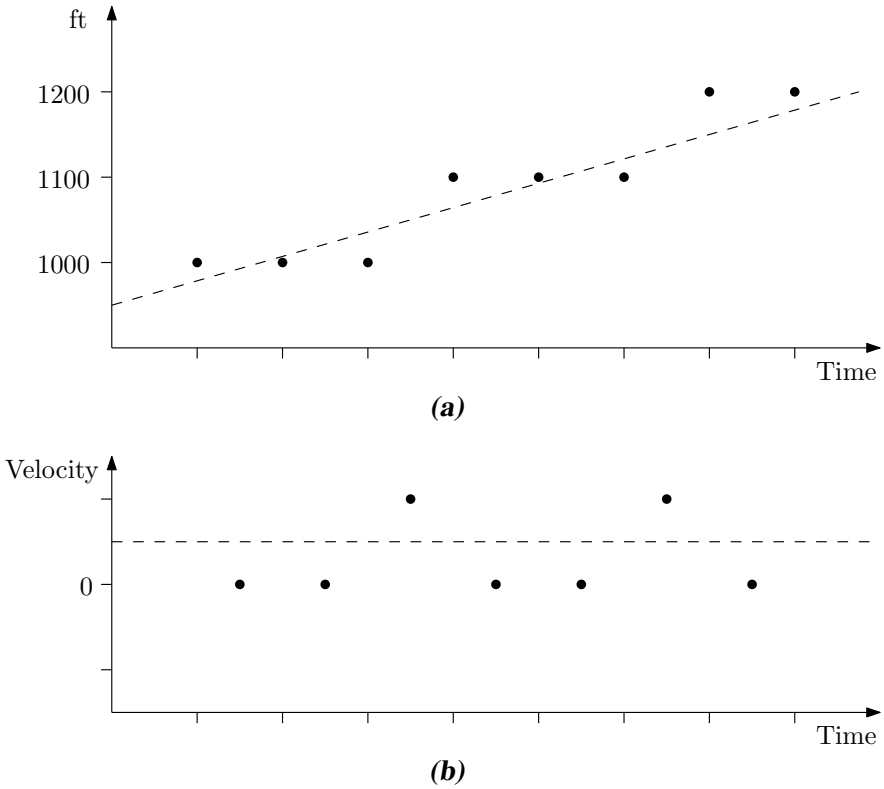



***Figure 1.4:*** *The Mode C measurements (dots) form a staircase in (a) where the dashed line is the true trajectory. The measurements lead to a velocity as shown in (b) where the dashed line is the true velocity.*

lution creates difficulties in estimating the vertical velocity of the aircraft which influences altitude prediction. The latter is used by ATC surveillance systems to alert controllers of possible dangerous situations.

## 1.2 Previous Work

This section presents relevant references for this thesis.

### 1.2.1 Multisensor Bias Estimation

Unless otherwise specified, the sensors broadcast their estimates as track data, not the measurements they receive (plots). This is the most common approach, as it has a lower impact on the sensor network communication load (Drummond,

2002). Table 1.1 shows relevant references concerning multisensor bias estimation, and unless otherwise stated the sensor measurements are from active (as opposed to passive) radars, and bias estimation is performed by tracking targets that are observed by all sensors. In general the sensors are *distributed*, as opposed to *collocated* where the distance between sensors is negligible compared to the distance from the sensors to the targets. When the latter is the case, it will be specified.

***Table 1.1:*** *Literature on multisensor bias estimation. The table shows reference, bias modeling, measurement dimension (Dim.), and if the alignment (Align.), location (Loc.) or sensor (Sen.) bias is estimated. In addition the algorithm used to perform bias estimation is specified.*

| Reference | Modeling | Dim. | Align. | Loc. | Sen. | Algorithm |
|---|---|---|---|---|---|---|
| (Dela Cruz et al., 1992) | Relative | 3D | | | x | KF |
| (Helmick and Rice, 1993) | Relative | 3D | x | | x | KF |
| (Helmick et al., 1994) | Relative | 3D | x | | x | KF |
| (Blackman and Banh, 1994) | Relative | 3D | x | | | Error Median |
| (Nabaa and Bishop, 1999) | Relative | 3D | x | x | | EKF |
| (Rhodes, 2002) | Relative | 2D | x | x | | Hough Transform |
| (Okada et al., 2004) | Relative | 3D | | x | | EKF |
| (Matsuzaki et al., 2010) | Relative | 2D | | | x | KF and grid search |
| (Fortunati et al., 2011) | Relative | 3D | x | x | x | LS |
| (Easthope, 1999) | Absolute | 3D | x | x | x | LS |
| (Lin et al., 2004a,b, 2005) | Absolute | 2D | | x | x | KF |
| (Novoselov et al., 2005) | Absolute | 3D | x | | x | KF |
| (Rafati et al., 2006, 2007) | Absolute | 2D | | x | x | KF |
| (Qi et al., 2008a) | Absolute | 2D | | x | x | KF |
| (Ying et al., 2010) | Absolute | 3D | | x | x | KF |
| (Okello and Pulford, 1996) | Universal | 2D | | | x | KF using PDA |
| (Zhou et al., 1997) | Universal | 2D | | | x | ML |
| (Zhou et al., 1999) | Universal | 3D | | | x | LS |
| (Watson et al., 1999) | Universal | 3D | | | x | EKF |
| (Easthope, 2000) | Universal | 3D | | | x | LS |

***Table 1.1:*** *(continued)*

| Reference | Modeling | Dim. | Align. | Loc. | Sen. | Algorithm |
|---|---|---|---|---|---|---|
| (Sviestins, 2000) | Universal | 2D | x | x | x | Modified LS |
| (Bar-Shalom, 2000) | Universal | 2D | | x | | KF |
| (Kosuge and Okada, 2000) | Universal | 3D | | | x | KF |
| (Watson and Rice, 2001, 2002a,b) | Universal | 3D | x | | x | EKF |
| (Besada Portas et al., 2002) | Universal | 2D | | | x | LS and BLUE |
| (Gade, 2004) | Universal | 3D | x | | | EKF |
| (Okello and Challa, 2004) | Universal | 2D | | | x | EKF |
| (Besada Portas et al., 2004) | Universal | 2D | | | x | BLUE |
| (Besada et al., 2005) | Universal | 2D | | | x | BLUE |
| (Li and Leung, 2006) | Universal | 2D | | | x | EM |
| (Herman and Poore, 2006) | Universal | 3D | x | x | x | Nonlinear LS |
| (Scott Danford and Poore, 2007a,b) | Universal | 3D | x | x | x | Nonlinear LS |
| (Kragel et al., 2007) | Universal | 3D | x | | | Nonlinear LS |
| (Qi et al., 2008b) | Universal | 2D | | | x | Gaussian Mean Shift |
| (Davey et al., 2008) | Universal | 2D | | | x | EM |
| (Pan et al., 2008) | Universal | 3D | x | | x | LS |
| (Fortunati et al., 2012a) | Universal | 3D | x | x | x | LS |

#### 1.2.1.1 The Sensor Bias

Studying Table 1.1 it is clear that the sensor bias has been treated extensively for both 2D and 3D radars. Observability and estimability for the sensor bias has been investigated, and it has been found to be universally estimable (Blackman and Popoli, 1999; Easthope, 2000; Kosuge and Okada, 2000, page 289-299). In Kosuge and Okada (2000) an index of radar bias estimation is presented which is based on the minimum singular value of the observation matrix. The index is used to determine if a bias estimation problem is estimable. An alternative observability index using the inverse condition number for the observation matrix is presented in Arrichiello et al. (2013). The sensor bias for a radar is usually modeled as an additive bias in range, azimuth and elevation, but some references (Lin et al., 2004a,b, 2005; Qi et al., 2008a; Rafati et al., 2006, 2007; Ying et al., 2010) include multiplicative scale biases as well. Note that when this is the only bias estimated, any unmodeled alignment or location biases present will be partially included in the sensor bias. This means that an estimator estimating the sensor bias may partially compensate for the unmodeled biases as well. This applies to the alignment bias in particular since it is usually modeled using the three Euler angles (roll, pitch and yaw). The sensor bias includes angular biases in azimuth and elevation which can be assumed to compensate partially for the alignment bias. In fact, if the problem is modeled in 2D, with 2D target motion and 2D sensor measurements, the alignment bias will only include a yaw bias which will be the same angle as the sensor azimuth bias. Thus the alignment bias will be included in the sensor bias. This makes the 2D bias estimation problem easier to solve than the 3D counterpart.

#### 1.2.1.2 Bias Estimation using Reference Data

Applied to ATC, Besada Portas et al. (2002) proposes a solution to estimate sensor biases relative to GPS measurements. A more general approach is presented in Besada Portas et al. (2004). In a series of papers Watson and Rice (2001, 2002a,b); Watson et al. (1999) uses a Kalman Filter (KF) to find alignment and sensor biases, but the targets are assumed cooperative and truth data is available from them. A similar method is proposed by Novoselov et al. (2005) which accounts for cross-correlation between target state and residual bias. A nonlinear Least Squares (LS) method has been used to find alignment, location and sensor biases in Herman and Poore (2006) and Scott Danford and Poore (2007a,b) with the use of truth data in 3D.

#### 1.2.1.3 Relative Alignment, Location and Sensor Biases

The papers by Helmick and Rice (1993); Helmick et al. (1994) deal with all three biases treated herein with 3D measurements, although the location bias is not estimated, and perform estimation based on targets observed by all sensors. They assume that the sensors are collocated, which means that any location bias can be neglected since the distance between sensors is negligible compared to the distance from the sensors to the targets. In a recent paper by Fortunati et al. (2011) all the biases are estimated using a linear LS estimator. The hybrid Cramer-Rao Lower Bound (CRLB) for the estimation problem is provided.

#### 1.2.1.4 Absolute Alignment, Location and Sensor Biases

Using a LS method Easthope (1999) estimates biases for two distributed radars. The sensor bias is found to be absolutely estimable, while alignment, location and time biases are not.

#### 1.2.1.5 Universal Alignment, Location and Sensor Biases

The observability (estimability) of the location bias has been investigated by Bar-Shalom (2000), who finds that the location bias can be estimated by using fixed (static) targets of opportunity. The location biases are modeled as slowly varying, and if they vary differently as a function of time across sensors they can be separated and estimated universally. If the same bias model is used for all sensors, the biases are relatively estimable. Universal alignment bias estimation is performed by Gade (2004) and Kragel et al. (2007). The paper by Sviestins (2000) uses a modified LS method including a priori data to find universal alignment, location, sensor and time biases in a quasi-recursive manner. The bias estimation is done centrally, using targets of opportunity. Results from three experiments are presented, and it is the use of a priori data which makes absolute bias estimation possible. The first experiment estimates a combination of sensor, alignment and time biases. The second and third experiments estimate location and alignment biases respectively. A LS method is also used in Pan et al. (2008) to find alignment and sensor biases, but not all angle biases are found successfully for two sensors. In a recent paper by Fortunati et al. (2012a) all the biases are estimated using a linear LS estimator. The hybrid CRLB for the estimation problem is provided.

#### 1.2.1.6 Topics for Further Investigation

From the previous discussion it is clear that multisensor bias estimation has already been well treated in the literature. However, there is still room for improvement. Regarding universal bias estimation, an attempt to recursively estimate as many biases as possible using 3D measurements from common targets should be made. At the time of writing no papers present such results when truth data is unavailable. Additionally a comparison of relative, absolute and universal bias estimators should be made to see when the added complexity of universal bias estimation is needed. These topics are addressed in this thesis.

### 1.2.2 Aircraft Altitude Prediction and Estimation

Altitude prediction is important in conflict alarm systems which monitor separation distances between aircraft. Examples are Short Term Conflict Alert (STCA), Minimum Safe Altitude Warning (MSAW) and Area Proximity Warning (APW). These systems rely on altitude prediction of the aircraft at most 2 minutes ahead in time (Technical Staff, 1999). Poor altitude prediction, which is caused by a poor vertical velocity estimate, results in unnecessary false alarms. These difficulties arise since the Mode C altitude measurements are quantized because of bandwidth constraints.

The effect of quantization of data is discussed by Vardeman and Lee (2005). This problem has been treated before (Sviestins and Wigren, 2001), where an estimator derived from the solution of the Fokker-Planck equation was used to filter the quantized Mode C measurements. The aircraft motion was modeled as deterministic with constant velocity, and it outperformed the Extended Kalman Filter (EKF) on this problem. The latter was based on a stochastic model with colored noise. A numerical algorithm for approximate Minimum Mean Square Error (MMSE) is developed in Duan et al. (2008a,b) for quantized measurements. A Maximum Likelihood (ML) method is presented by Kim (2011). A simple approach to handle quantized measurements is presented in Hodel and Hung (2003) where a quantizer is included in a deterministic observer for state estimation.

## 1.3 Thesis Outline and Contribution

Target tracking and bias estimation are mature fields of study, and many problems are already well treated in the literature. Mathematical background material

concerning the CRLB and coordinate systems can be found in Chapter 2. Chapter 3, Chapter 4 and Chapter 5 contain the main contributions of this thesis, and a discussion of results and conclusions are found in Chapter 6.

Chapter 3 contains the derivation of formulas for converting a 3D measurement in spherical coordinates to Cartesian coordinates. The derivation is analogous to the 2D case in Bar-Shalom and Li (1995). Furthermore the fundamental measurement models for universal, absolute and relative bias estimation are derived. In the Universal Bias Estimator (UBE), tracking is performed in sensor independent coordinates, while the Absolute Bias Estimator (ABE) and the Relative Bias Estimator (RBE) perform tracking in the coordinates of a master sensor. The mathematical model used in the ABE is mathematically equivalent to the model in the UBE, but tracking in sensor coordinates allows some simplifications. It is shown that the ABE model holds for small alignment biases even if the coordinate transformation from a biased sensor to the master sensor involves large angles. The measurement biases of the master sensor are not estimated in the RBE, and a method to compensate for this uncertainty through added measurement noise is presented. The author has not seen this clear distinction in the literature previously, although universal, absolute and relative bias estimation have all been treated.

Chapter 4 deals with estimation of measurement biases. The main contributions of this chapter are the study of universal, absolute and relative estimability using the novel CRLB based estimability index of alignment, location and sensor biases, and a performance comparison of the UBE with the ABE, RBE and standard filters from the literature. Estimability is studied when the system and filter models are the same, but the performance of the estimators is evaluated in scenarios where system and filter models are different. This is important, as it is common in the literature that system and filter models are equal. These results can serve as a guideline for estimator design in the sense that we can see which unmodeled biases have the strongest effect on estimator performance. This research complements the work in Helmick and Rice (1993), Easthope (1999), Kosuge and Okada (2000), Gade (2004), Kragel et al. (2007) and Fortunati et al. (2011, 2012a,b,c). Some of the results have been published in the peer reviewed conference paper Topland and Hallingstad (2008).

Furthermore, a novel demonstration of increased bias estimability for a target moving at constant altitude above the Earth contributes to bias estimation research. These results are published in the peer reviewed conference paper Top-

land et al. (2007). All results are derived in three dimensions assuming three-dimensional radar measurements.

Aircraft altitude prediction is investigated in Chapter 5 and several Bayesian estimators are tuned and tested on real measurement data for a performance comparison. It is shown that a single model filter with maneuver handling using the Unscented Kalman Filter (UKF) is superior to Interacting Multiple Model (IMM) estimators and KFs for long term predictions. The major contribution of this chapter is the use of the IMM estimators and the UKF algorithm on quantized measurements using measurements from the real world. These measurements are used both for evaluation (testing), and for LS and ML based estimator tuning. This work has been published in the peer reviewed conference paper Topland and Hallingstad (2007), and complements the research presented in Section 1.2.2.

# 2

# Mathematical Background

This chapter presents mathematical background material used in this dissertation. Section 2.1 presents the Cramer-Rao Lower Bound (CRLB), and Section 2.2 discusses relevant coordinate systems.

## 2.1 The Cramer-Rao Lower Bound

The CRLB is often used as a measure of filtering performance. It yields the minimum variance possible for an *unbiased* estimator given a deterministic state vector (Fisher estimation). This minimum variance can be computed by inverting the Fisher Information Matrix (FIM). In Fisher estimation the state vector $\underline{x}$ is assumed to be a deterministic unknown constant, and no prior information is available. This is opposed to Bayesian estimation where $\underline{x}$ is a stochastic vector, and prior information exists. The CRLB can be calculated for a Bayesian system as well, which is discussed in Section 2.1.2. The state vector $\underline{x}$ is in this section assumed to be coordinatized in an appropriate coordinate system, omitting the name of the coordinate system for simplicity.

### 2.1.1 The CRLB for a Static System

**Theorem 2.1**
*When estimating the deterministic state vector $\underline{x}$ from the measurement $\underline{z}$ with*

*an unbiased estimator, the covariance is bounded from below*

$$E\left\{(\underline{x}-\underline{\hat{x}})(\underline{x}-\underline{\hat{x}})^T|\underline{z}\right\} \geq J^{-1} \tag{2.1}$$

*where $\underline{\hat{x}}$ is the estimate, $J^{-1}$ is the CRLB, and $J$ is the FIM which is a positive definite symmetric matrix (Papoulis and Pillai, 2002)*

$$[J_{ij}] = \left[E\left\{\frac{\partial \ln p(\underline{z}:\underline{x})}{\partial x_i}\frac{\partial \ln p(\underline{z}:\underline{x})}{\partial x_j}\right\}\right] \tag{2.2a}$$

$$= \left[-E\left\{\frac{\partial^2 \ln p(\underline{z}:\underline{x})}{\partial x_i \partial x_j}\right\}\right] \tag{2.2b}$$

*or in vector form*

$$J = E\left\{\frac{\partial \ln p(\underline{z}:\underline{x})}{\partial \underline{x}}\frac{\partial \ln p(\underline{z}:\underline{x})}{\partial \underline{x}^T}\right\} \tag{2.3a}$$

$$= -E\left\{\frac{\partial^2 \ln p(\underline{z}:\underline{x})}{\partial \underline{x}\partial \underline{x}^T}\right\} \tag{2.3b}$$

*where $\underline{z}$ is a vector containing all measurements, and the state vector $\underline{x}$ is an unknown constant with dimension $n_x$. The parameters $x_i$ and $x_j$ are elements of $\underline{x}$ where $i, j \in \{1, 2, \ldots, n_x\}$.*

Note that a colon is used to show dependence in (2.2) instead of a $|$. This is to emphasize the absence of statistical information about $\underline{x}$, thus separating Fisher and Bayesian notation. For proof of Theorem 2.1 see Kay (1993) or Appendix A.1.

**Example 2.1**

Given a measurement $\underline{z}$ of the unknown constant $\underline{x}$

$$\underline{z} = H\underline{x} + \underline{w} \tag{2.4a}$$

$$\underline{w} \sim \mathcal{N}(\underline{0}, R) \tag{2.4b}$$

the CRLB covariance matrix $P$ is given by

$$P^{-1} = J = H^T R^{-1} H \tag{2.5}$$

where $H$ is the measurement matrix and $R$ is the measurement covariance matrix. A static system is a system which is time invariant. Equation (2.5) can also be used for nonlinear systems through linearization. The following assumptions are made in (2.5):

1. The state vector $\underline{x}$ is an unknown constant vector.
2. The measurement equation is known.
3. The measurement noise is additive with known mean and covariance.

Note that although there is no prior information in Fisher estimation, such information can still be included as additional measurements of the state vector. A proof of (2.5) follows. Given (2.4) the probability density function (pdf) is

$$p\left(\underline{z}:\underline{x}\right)=\frac{e^{-\frac{1}{2}\left(\underline{z}-H\underline{x}\right)^{T}R^{-1}\left(\underline{z}-H\underline{x}\right)}}{\left(2\pi\right)^{n_z/2}\left|\left|R\right|\right|^{1/2}} \tag{2.6}$$

since $\underline{z}-H\underline{x}=\underline{w}$. Its logarithm is

$$\ln p\left(\underline{z}:\underline{x}\right)=-\ln\left(\left(2\pi\right)^{n_z/2}\left|\left|R\right|\right|^{1/2}\right)-\frac{1}{2}\left(\underline{z}-H\underline{x}\right)^{T}R^{-1}\left(\underline{z}-H\underline{x}\right) \tag{2.7a}$$

$$=-\ln\left(\left(2\pi\right)^{n_z/2}\left|\left|R\right|\right|^{1/2}\right)-\frac{1}{2}\left(\underline{z}^{T}R^{-1}\underline{z}-\underline{x}^{T}H^{T}R^{-1}\underline{z}\right.$$
$$\left.-\underline{z}^{T}R^{-1}H\underline{x}+\underline{x}^{T}H^{T}R^{-1}H\underline{x}\right) \tag{2.7b}$$

Using (2.7a) and (2.3a) and recalling that $R$ is symmetric

$$\frac{\partial\ln p\left(\underline{z}:\underline{x}\right)}{\partial\underline{x}}\frac{\partial\ln p\left(\underline{z}:\underline{x}\right)}{\partial\underline{x}^{T}}=H^{T}R^{-1}\left(\underline{z}-H\underline{x}\right)\left(\underline{z}-H\underline{x}\right)^{T}R^{-1}H$$
$$E\left\{\frac{\partial\ln p\left(\underline{z}:\underline{x}\right)}{\partial\underline{x}}\frac{\partial\ln p\left(\underline{z}:\underline{x}\right)}{\partial\underline{x}^{T}}\right\}=H^{T}R^{-1}H. \tag{2.8}$$

Using (2.7b) and (2.3b)

$$\frac{\partial^{2}\ln p\left(\underline{z}:\underline{x}\right)}{\partial\underline{x}\partial\underline{x}^{T}}=-H^{T}R^{-1}H$$
$$-E\left\{\frac{\partial^{2}\ln p\left(\underline{z}:\underline{x}\right)}{\partial\underline{x}\partial\underline{x}^{T}}\right\}=H^{T}R^{-1}H. \tag{2.9}$$

Thus (2.3a) and (2.3b) yields the same expected result

$$J=H^{T}R^{-1}H. \tag{2.10}$$

**Example 2.2**

It is possible to obtain an analogous expression to (2.5) for a nonlinear stochastic system with additive noise. Consider the system

$$\underline{z} = \underline{h}(\underline{x}) + \underline{w} \tag{2.11a}$$
$$\underline{w} \sim \mathcal{N}(\underline{0}, R) \tag{2.11b}$$

and define

$$\underline{f}(\underline{x}, \underline{z}) = \underline{z} - \underline{h}(\underline{x}) = \underline{w}. \tag{2.12}$$

The pdf is

$$p(\underline{z} : \underline{x}) = \frac{e^{-\frac{1}{2}\underline{f}(\underline{x},\underline{z})^T R^{-1} \underline{f}(\underline{x},\underline{z})}}{(2\pi)^{n_z/2} ||R|\,|^{1/2}} \tag{2.13}$$

and its logarithm is

$$\ln p(\underline{z} : \underline{x}) = -\ln\left((2\pi)^{n_z/2} ||R|\,|^{1/2}\right) - \frac{1}{2}\underline{f}(\underline{x}, \underline{z})^T R^{-1} \underline{f}(\underline{x}, \underline{z}). \tag{2.14}$$

Using (2.14), the fact that $R$ is symmetric,

$$\frac{\partial \ln p(\underline{z} : \underline{x})}{\partial \underline{x}} = -\left[\frac{\partial \underline{f}(\underline{x}, \underline{z})}{\partial \underline{x}}\right]^T R^{-1} \underline{f}(\underline{x}, \underline{z}), \tag{2.15}$$

and (2.3a) leads to

$$\frac{\partial \ln p(\underline{z} : \underline{x})}{\partial \underline{x}} \frac{\partial \ln p(\underline{z} : \underline{x})}{\partial \underline{x}^T} = \left[\frac{\partial \underline{f}(\underline{x}, \underline{z})}{\partial \underline{x}}\right]^T R^{-1} \underline{f}(\underline{x}, \underline{z}) \underline{f}(\underline{x}, \underline{z})^T R^{-1} \left[\frac{\partial \underline{f}(\underline{x}, \underline{z})}{\partial \underline{x}}\right]$$
$$E\left\{\frac{\partial \ln p(\underline{z} : \underline{x})}{\partial \underline{x}} \frac{\partial \ln p(\underline{z} : \underline{x})}{\partial \underline{x}^T}\right\} = \left[\frac{\partial \underline{f}(\underline{x}, \underline{z})}{\partial \underline{x}}\right]^T R^{-1} \left[\frac{\partial \underline{f}(\underline{x}, \underline{z})}{\partial \underline{x}}\right]. \tag{2.16}$$

because $E\left\{\underline{f}(\underline{x}, \underline{z}) \underline{f}(\underline{x}, \underline{z})^T\right\} = R$. Alternatively (2.3b) yields

$$\frac{\partial^2 \ln p(\underline{z} : \underline{x})}{\partial \underline{x} \partial \underline{x}^T} = -\left[\frac{\partial \underline{f}(\underline{x}, \underline{z})}{\partial \underline{x}}\right]^T R^{-1} \left[\frac{\partial \underline{f}(\underline{x}, \underline{z})}{\partial \underline{x}}\right] - \underline{f}(\underline{x}, \underline{z})^T R^{-1} \left[\frac{\partial^2 \underline{f}(\underline{x}, \underline{z})}{\partial \underline{x} \partial \underline{x}^T}\right]$$
$$-E\left\{\frac{\partial^2 \ln p(\underline{z} : \underline{x})}{\partial \underline{x} \partial \underline{x}^T}\right\} = \left[\frac{\partial \underline{f}(\underline{x}, \underline{z})}{\partial \underline{x}}\right]^T R^{-1} \left[\frac{\partial \underline{f}(\underline{x}, \underline{z})}{\partial \underline{x}}\right] \tag{2.17}$$

since derivatives of $\underline{f}(\underline{x}, \underline{z})$ with respect to $\underline{x}$ are deterministic and $E\left\{\underline{f}(\underline{x}, \underline{z})\right\} = \underline{0}$. Thus

$$J = \left[\frac{\partial \underline{f}(\underline{x}, \underline{z})}{\partial \underline{x}}\right]^T R^{-1} \left[\frac{\partial \underline{f}(\underline{x}, \underline{z})}{\partial \underline{x}}\right] \tag{2.18}$$

which we rewrite to an expression analogous to (2.5):

$$\boxed{J(\underline{x}) = H(\underline{x})^T R^{-1} H(\underline{x})} \tag{2.19a}$$

$$H(\underline{x}) = \frac{\partial \underline{h}(\underline{x})}{\partial \underline{x}^T} \tag{2.19b}$$

## 2.1.2 The CRLB for a Dynamic System

The CRLB can be calculated for a dynamic system with no process noise using a Bayesian model with state vector $\underline{x}_k$. The CRLB for the model

$$\underline{x}_{k+1} = \underline{f}_k(\underline{x}_k) \tag{2.20a}$$

$$\underline{x}_0 \sim \mathcal{N}(\underline{x}_0, P_0) \tag{2.20b}$$

$$\underline{z}_k = \underline{h}_k(\underline{x}_k) + \underline{w}_k \tag{2.20c}$$

$$\underline{w}_k \sim \mathcal{N}(\underline{0}, R_k) \tag{2.20d}$$

is calculated using the Extended Kalman Filter (EKF) covariance propagation equations linearized about the true system states in Taylor (1979).

In the Bayesian case where the state vector is dynamic and stochastic with prior information, the Posterior CRLB can be calculated. Using the Bayesian model

$$\underline{x}_{k+1} = \underline{f}(\underline{x}_k, \underline{v}_k) \tag{2.21a}$$

$$\underline{x}_0 \sim p_0(\underline{x}_0) \tag{2.21b}$$

$$\underline{v}_k \sim p_v(\underline{v}_k) \tag{2.21c}$$

$$\underline{z}_k = \underline{h}(\underline{x}_k, \underline{w}_k) \tag{2.21d}$$

$$\underline{w}_k \sim p_w(\underline{w}_k) \tag{2.21e}$$

where $\underline{v}_k$ and $\underline{w}_k$ are white noise processes, and $\underline{x}_0$, $\underline{v}_k$ and $\underline{w}_k$ are independent, a general way of computing the Posterior CRLB is found in Tichavsky et al. (1998). The special case with additive $\underline{v}_k$ and $\underline{w}_k$ is also treated.

## 2.2 Coordinate Systems

When modeling a multisensor target tracking problem, *coordinate systems* and transformations between them are mandatory. A coordinate system is usually attached to a physical body, for instance the Earth or a sensor gathering measurements. To clarify how a coordinate system is defined mathematically, we start with the physical body. A physical body can be viewed as a collection of points, which can be expanded in space. This expanded set of points is in physics referred to as a *reference frame*. To work with this set of points it is necessary to add a vector space. The mathematical operations of an affine space are well suited for this, and when adding these operations we get a *frame* which we can use for mathematical modeling. In the literature an affine frame is a chosen origin and a set of base vectors. The base vectors make it possible to define orientation, and orthonormal base vectors are chosen for mathematical simplicity. Lastly the frame is coordinatized, which corresponds to choosing a function to transform points in the affine frame to $\mathbb{R}^n$ (labeling). The Cartesian coordinate system is the most common, but in addition we will use spherical and geodetic coordinates. For more on coordinate systems and frames see Zipfel (2007). The coordinate systems used in this thesis are described next.

### 2.2.1 Earth Centered Earth Fixed Coordinate System

The *Earth Centered Earth Fixed (ECEF)* coordinate system has its origin at the center of the Earth, its $z$-axis $z^e$ points to the North Pole, and its $x$-axis $x^e$ points to the prime meridian. The $y$-axis $y^e$ is chosen to make $x^e y^e z^e$ a right-handed Cartesian coordinate system. In the following the ECEF coordinate system will be denoted as coordinate system $\{e\}$, and vectors written in this system will have the superscript $^e$.

### 2.2.2 Geodetic Coordinate System

The *geodetic* coordinate system is the system describing a location in reference to the Earth using latitude $\mu$, longitude $l$ and altitude $h$. The Earth is modeled as the rotation symmetrical WGS-84 ellipsoid. The geodetic coordinate system is denoted $\{g\}$, and vectors written in this system will have the superscript $^g$.

### 2.2.3 Geographic Coordinate System

The standard local coordinates for a geographic location are given in the *geographic* coordinate system, which is a right-handed *North East Down (NED)*

system. The origin is defined in reference to the Earth, and in this work the origin is typically chosen to coincide with the sensor location. The $x^n$, $y^n$ and $z^n$-axes point respectively towards north, east and down. The $x^n y^n$ plane is tangent to the Earth's surface. A geographic coordinate system is denoted $\{n\}$, and vectors written in this system will have the superscript $^n$.

### 2.2.4 Universal Coordinate System

The universal coordinate system $\{u\}$ is a coordinate system independent of the sensors where we wish to track targets. It can be an ECEF, geodetic, or geographic coordinate system. This depends on the scenario treated.

### 2.2.5 Sensor Coordinate System

In this work sensors are used to get measurements of target positions. The sensor locations are described in $\{u\}$, and each sensor, Sensor $i$, has an associated nominal geographic coordinate system denoted $\{n_i\}$. The term nominal means that $\{n_i\}$ is an assumed known coordinate system. When Sensor $i$ is deployed it is aligned and located to coincide with $\{n_i\}$. If aligned and located perfectly, the transformation of a measurement to any known coordinate system is known. This enables target tracking in any known coordinate system. However, perfect alignment and location is usually infeasible in the real world. Therefore we introduce a sensor coordinate system $\{s_i\}$ which is a NED coordinate system whose $x^{s_i} y^{s_i}$ plane may not coincide with $x^{n_i} y^{n_i}$, and whose origin may not coincide with the origin of $\{n_i\}$. In other words $\{s_i\}$ is a perturbation of $\{n_i\}$. The coordinates of $\{s_i\}$ will in the following be referred to as the local coordinates of the sensor.

### 2.2.6 Cartesian Coordinate Transformations

The transformation of the point coordinates from one Cartesian coordinate system $\{n\}$ to another $\{s\}$ is in general given by

$$\underline{p}^s = T_n^s \left( \underline{p}^n + \underline{d}_{ns}^n \right) = T_n^s \underline{p}^n + \underline{d}_{ns}^s \tag{2.22}$$

where $\underline{p}^n$ and $\underline{p}^s$ are coordinate vectors represented in $\{n\}$ and $\{s\}$ respectively, $T_n^s$ is a Coordinate Transformation Matrix (CTM) and $\underline{d}_{ns}^n$ and $\underline{d}_{ns}^s$ are *displacement vectors* from the origin of $\{n\}$ to the origin of $\{s\}$, represented in $\{n\}$ and $\{s\}$ respectively. A composite coordinate transformation is represented by the product of two CTMs $T_e^s = T_n^s T_e^n$. Note that $(T_e^s)^T = T_s^e$ because $T_e^s$ is

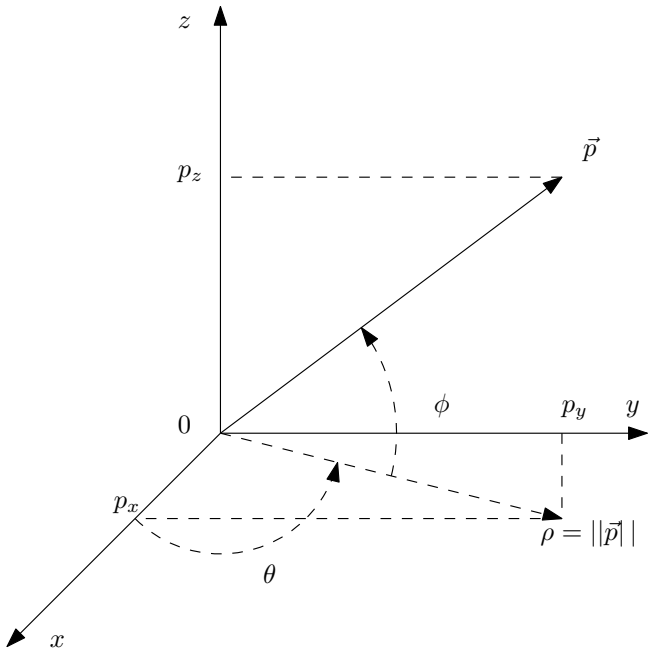



***Figure 2.1:*** *The geometric position vector $\vec{p}$ is in Cartesian coordinates equal to $\underline{p} = [p_x\ p_y\ p_z]^T$, and in spherical coordinates equal to $\underline{\varrho} = [\rho\ \theta\ \phi]^T$.*

orthonormal. In fact a CTM $T$ is a member of the special orthogonal group of order 3,

$$SO(3) = \left\{T \mid T \in \mathbb{R}^{3\times 3}, T^T T = I, \det T = 1\right\}. \tag{2.23}$$

### 2.2.7 Conversion Between Cartesian and Spherical Coordinates

Let a point in space in Cartesian coordinates be given by

$$\underline{p} = \begin{bmatrix} x & y & z \end{bmatrix}^T. \tag{2.24}$$

Then the same point in spherical coordinates is

$$\underline{\varrho} = \begin{bmatrix} \rho & \theta & \phi \end{bmatrix}^T \tag{2.25}$$

where range, azimuth and elevation are denoted $\rho$, $\theta$ and $\phi$ respectively. Figure 2.1 shows Cartesian and spherical coordinates. The Coordinate Transformation Function (CTF) from spherical to Cartesian coordinates is defined by

$$\underline{p} = \underline{h}_c\left(\underline{\varrho}\right) \tag{2.26}$$

where the elements of $\underline{h}_c(\cdot)$ transform the coordinates in the following nonlinear way:

$$x = \rho \cos\theta \cos\phi \tag{2.27a}$$
$$y = \rho \sin\theta \cos\phi \tag{2.27b}$$
$$z = \rho \sin\phi \tag{2.27c}$$

The inverse transformation $\underline{h}_s(\cdot)$ is:

$$\rho = \sqrt{x^2 + y^2 + z^2} \tag{2.28a}$$
$$\theta = \arctan\frac{y}{x} \tag{2.28b}$$
$$\phi = \arcsin\frac{z}{\sqrt{x^2 + y^2 + z^2}} \tag{2.28c}$$

# 3

# Measurement Modeling and Estimability

This chapter deals with measurement models and estimability of biases. Section 3.1 explains the difference between system and filter models, Section 3.2 treats sensors and measurement models concerning bias estimation, and Section 3.4 discusses estimability for static targets using the presented measurement models.

## 3.1 System and Filter Models

When evaluating estimators or filters in a simulation it is important to separate system and filter models.

*System Model* An accurate mathematical description of the truth. The model is used to generate measurements, and is also called *simulation model.*

*Filter Model* The model used to design an estimator or filter.

The system model is in general nonlinear and computationally demanding to use. When this is the case the filter model is usually a simplified version of the system model. This can be done through linearization or omitting of system variables. In many cases the system model is the most accurate description sought for a real world scenario. Because of its complexity it may be unusable in a realtime system, hence a filter model is used in the design of the estimator instead.

## 3.2 Modeling of Measurements and Biases for 3D Radars

In this work we focus on modeling of measurements from 3D radars with errors that can be modeled as unknown constants. Registration errors and measurement noise which is not zero-mean fall into this category. These constants are called biases in the following. This section explains the modeling of the measurement biases, and presents several models for the sensor measurement. There are two reference models. One where the measurement biases are known, and one where they are neglected. These will serve as references when the other models are used to estimate the biases. A good choice of filter model should yield tracking performance between these two reference models. The final part of this section derives measurement models for universal, absolute and relative bias estimation.

### 3.2.1 The Bias Model

**Definition 3.1.** A bias is a random constant, thus the bias $\underline{b}^a$ in coordinate system $\{a\}$ obeys

$$\dot{\underline{b}}^a(t) = 0 \tag{3.1a}$$

$$\underline{b}^a(t_0) \sim p_b(\underline{b}_0^a) \tag{3.1b}$$

in continuous time and

$$\boxed{\begin{aligned} \underline{b}_{k+1}^a &= \underline{b}_k^a \\ \underline{b}_0^a &\sim p_b(\underline{b}_0^a) \end{aligned}} \tag{3.2a, 3.2b}$$

in discrete time where $k$ is the discrete time index. The probability density function (pdf) $p_b(\cdot)$ is arbitrary. In this work we assume that the covariance matrix $E\left\{\underline{b}_0^a \underline{b}_0^{aT}\right\}$ is diagonal and

$$E\{\underline{b}_0^a\} = \underline{0}. \tag{3.3}$$

Note that using Definition 3.1 for bias modeling is correct for constant biases and approximate for biases that are slowly varying. In the system model (3.1) or (3.2) are used to model biases. In Section 3.4 and Section 4.2 both the system and filter models use (3.2) for bias modeling, but in the Monte Carlo (MC) simulations in Section 4.1 we assume slowly varying biases in the filter models and add Gaussian white noise to avoid divergence of bias estimates. In these simulations divergence may occur because the filter models are not equal to the

system model. Furthermore the pdf $p_b(\cdot)$ is assumed to be Gaussian. Thus the bias model becomes

$$\dot{\underline{b}}^a(t) = \underline{n}^a(t) \tag{3.4a}$$
$$\underline{b}^a(t_0) \sim \mathcal{N}(\underline{0}, B^a) \tag{3.4b}$$
$$\underline{n}^a(t) \sim \mathcal{N}\left(\underline{0}, \tilde{Q}^a \delta(t-\tau)\right) \tag{3.4c}$$

in continuous time, and

$$\boxed{\begin{aligned} \underline{b}^a_{k+1} &= \underline{b}^a_k + \underline{n}^a_k && (3.5a)\\ \underline{b}^a_0 &\sim \mathcal{N}(\underline{0}, B^a_0) && (3.5b)\\ \underline{n}^a_k &\sim \mathcal{N}(\underline{0}, Q^a \delta_{kl}) && (3.5c) \end{aligned}}$$

in discrete time where $\underline{n}^a(t)$ and $\underline{n}^a_k$ are Gaussian noise vectors. The spectral density matrix $\tilde{Q}^a$ and the covariance matrices $B^a$ and $Q^a$ are assumed constant.

### 3.2.2 The Spherical Measurement Model with Spherical State Vector

The use of radars is common for tracking of aircraft. Radar is an acronym for RAdio Detection And Ranging. Although the results herein may be applicable to several types of sensors, the modeling is done under the assumption that the sensor is a pulsed radar system which measures the position of targets in 3D. Radars use electromagnetic waves to obtain these measurements. Pulses are emitted, and when they hit a target they are reflected and scattered in several directions. Some pulses come back to the emitting radar, and the position is measured based on the travel time of the pulse. Radars typically receive measurements in 2D or 3D. 3D radars are the primary focus of this work, and unless clearly stated otherwise the word *sensor* refers to a 3D radar. A 3D target position measurement $\underline{z}$ from a sensor at discrete time $k$ is assumed to be given by

$$\underline{z}_k = \underline{\varrho}_k + \underline{b} + \underline{w}_k \tag{3.6a}$$
$$\underline{w}_k \sim \mathcal{N}(\underline{0}, R\delta_{kl}) \tag{3.6b}$$
$$\underline{b} \sim \mathcal{N}(\underline{0}, B) \tag{3.6c}$$

where $\underline{\varrho}_k$ is the target position in local spherical sensor coordinates, which consist of range $\rho$, azimuth $\theta$ and elevation $\phi$. The measurement noise is given

by $\underline{w}_k$ which is assumed Gaussian and zero-mean with covariance $R$. The Kronecker delta $\delta_{kl}$ means that the noise is white. The covariance matrix $R$ is diagonal and constant

$$R = \begin{bmatrix} \sigma_\rho^2 & 0 & 0 \\ 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & \sigma_\phi^2 \end{bmatrix}. \tag{3.7}$$

Because of imperfections in the sensor itself we assume that it is subject to a *sensor bias* $\underline{b}$ which include constant biases in range, azimuth and elevation

$$\underline{b} = \begin{bmatrix} b_\rho & b_\theta & b_\phi \end{bmatrix}^T \tag{3.8}$$

with covariance matrix

$$B = \begin{bmatrix} \sigma_{b_\rho}^2 & 0 & 0 \\ 0 & \sigma_{b_\theta}^2 & 0 \\ 0 & 0 & \sigma_{b_\phi}^2 \end{bmatrix} \tag{3.9}$$

where the sensor bias variances are found as elements of the diagonal. In Barton (1988) several sources of radar measurement errors are presented:

*Refraction* The emitted electromagnetic waves are bent in the atmosphere, which results in a range bias.

*Local Oscillator* The radar system uses an oscillator to clock the range counter. Errors result in a range bias.

*Receiver Delay* A time delay in the receiver causes a range bias.

*Drift of the Electrical Axis* Sensor imperfections may cause the calibrated vertical axis (electrical axis) to drift with respect to the mechanical vertical axis, causing an azimuth or elevation bias, or both.

### 3.2.3 The Spherical Measurement Model with Cartesian State Vector

In (3.6) both target position and sensor bias are given in the local spherical coordinates of the sensor. In multisensor target tracking it is often desirable to track the target in Cartesian coordinates, since it is straightforward to use linear target motion models in that case. This is especially true if the target position in a universal Cartesian coordinate system is required. Suppose that we want to

track the target in $\{u\}$, hence the target position is given by $\underline{p}_k^u$. To get a proper measurement model we need to express the measurement $\underline{z}_k$ as a function of $\underline{p}_k^u$. An *alignment bias* and *location bias* is also added. The registration errors are modeled as a difference between the assumed nominal coordinate system $\{n\}$ and the true coordinate system $\{s\}$ of the sensor. First we express the target position in the local Cartesian coordinate system $\{s\}$ for the sensor using Cartesian coordinates

$$\underline{p}_k^s = T_n^s T_u^n \left( \underline{p}_k^u - \underline{d}_{un}^u - \underline{b}_{ns}^u \right) \tag{3.10}$$

where the alignment bias is given by the Coordinate Transformation Matrix (CTM) $T_n^s$ from $\{n\}$ to $\{s\}$, and the location bias is given by the displacement vector $\underline{b}_{ns}^u = \begin{bmatrix} b_x & b_y & b_z \end{bmatrix}^T$. The alignment bias is the difference in orientation of the two coordinate systems, while the location bias is the difference in position of the coordinate system origins as seen in Figure 3.1. Note that $\underline{b}_{ns}^u = -\underline{b}_{sn}^u$. The CTM is given by the Euler angles roll $\alpha$, pitch $\beta$ and yaw $\gamma$ which correspond to rotations about the $x$, $y$ and $z$ axes respectively (Zipfel, 2007). The CTM depends on the sequence of Euler angle rotations. In this work we will use the $zyx$ convention where the first rotation is about the $z$ axis, the second about the $y$ axis, and the third about the $x$ axis. The Euler angles of the alignment bias are assumed small, which yields a simplified CTM

$$T_n^s \approx I + S\left(\underline{b}_n^s\right) \tag{3.11}$$

where $I$ is the identity matrix and

$$\underline{b}_n^s = \begin{bmatrix} b_\alpha & b_\beta & b_\gamma \end{bmatrix}^T \tag{3.12}$$

$$S\left(\underline{b}_n^s\right) = \begin{bmatrix} 0 & -b_\gamma & b_\beta \\ b_\gamma & 0 & -b_\alpha \\ -b_\beta & b_\alpha & 0 \end{bmatrix}. \tag{3.13}$$

The simplified CTM in (3.11) is independent of the Euler angle rotation sequence. Using (2.28) on (3.10) and inserting the result into (3.6) yields the measurement model sought

$$\underline{z}_k = \underline{h}_s \left( T_n^s T_u^n \left( \underline{p}_k^u - \underline{d}_{un}^u - \underline{b}_{ns}^u \right) \right) + \underline{b} + \underline{w}_k \tag{3.14a}$$

$$\underline{w}_k \sim \mathcal{N}\left(\underline{0}, R\delta_{kl}\right) \tag{3.14b}$$

$$\underline{b}_{ns}^u \sim \mathcal{N}\left(\underline{0}, B_{ns}^u\right) \tag{3.14c}$$

$$\underline{b}_n^s \sim \mathcal{N}\left(\underline{0}, B_n^s\right) \tag{3.14d}$$

$$\underline{b} \sim \mathcal{N}\left(\underline{0}, B\right) \tag{3.14e}$$

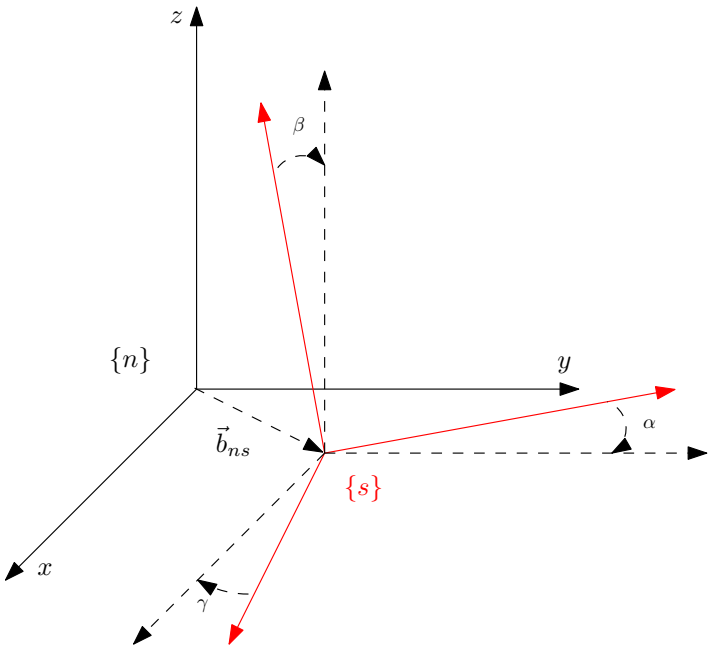



***Figure 3.1:*** *Registration errors separating $\{s\}$ from $\{n\}$ described by an alignment bias using the Euler angles $\alpha$, $\beta$ and $\gamma$, and a location bias $\vec{b}_{ns}$.*

where

$$B_n^s = \begin{bmatrix} \sigma_{b_\alpha}^2 & 0 & 0 \\ 0 & \sigma_{b_\beta}^2 & 0 \\ 0 & 0 & \sigma_{b_\gamma}^2 \end{bmatrix} \tag{3.15a}$$

$$B_{ns}^u = \begin{bmatrix} \sigma_{b_x}^2 & 0 & 0 \\ 0 & \sigma_{b_y}^2 & 0 \\ 0 & 0 & \sigma_{b_z}^2 \end{bmatrix}. \tag{3.15b}$$

### 3.2.4 The Cartesian Measurement Model with Cartesian State Vector

Equation (3.14) is given in the local spherical coordinates of the sensor. It may be useful to model the measurement in Cartesian coordinates, for instance if the sensor performs coordinate conversion to Cartesian coordinates automatically. Transforming (3.6) to Cartesian coordinates in $\{s\}$ yields

$$\underline{z}_k^s = \underline{h}_c\left(\underline{z}_k\right) \tag{3.16}$$

where $\underline{h}_c(\cdot)$ is the Coordinate Transformation Function (CTF) converting spherical coordinates to Cartesian coordinates as given in (2.27). Linearization yields a linear error model with time varying matrices.

$$\begin{aligned}\underline{z}_k^s &= \underline{h}_c(\underline{z}_k) = \underline{h}_c\left(\underline{\varrho}_k + \underline{b} + \underline{w}_k\right)\\ &\approx \underline{h}_c\left(\underline{\varrho}_k\right) + \left.\frac{\partial \underline{h}_c}{\partial \underline{\varrho}_k}\right|_{\underline{\varrho}_k} \underline{b} + \left.\frac{\partial \underline{h}_c}{\partial \underline{\varrho}_k}\right|_{\underline{\varrho}_k} \underline{w}_k\\ &= \underline{p}_k^s + C_k^s \underline{b} + \underline{w}_k^s \qquad (3.17a)\\ \underline{w}_k^s &\sim \mathcal{N}(\underline{0}, R_k \delta_{kl}). \qquad (3.17b)\end{aligned}$$

where $\underline{p}_k^s$ is the local target position in Cartesian coordinates. The matrix $R_k$ is the measurement noise covariance matrix written in Cartesian coordinates

$$R_k = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{12} & R_{22} & R_{23} \\ R_{13} & R_{23} & R_{33} \end{bmatrix} \qquad (3.18)$$

where the matrix diagonal elements are

$$R_{11} = \sigma_\rho^2 \cos^2\tilde{\theta}_k \cos^2\tilde{\phi}_k + \sigma_\theta^2 \tilde{\rho}_k^2 \sin^2\tilde{\theta}_k \cos^2\tilde{\phi}_k + \sigma_\phi^2 \tilde{\rho}_k^2 \cos^2\tilde{\theta}_k \sin^2\tilde{\phi}_k \qquad (3.19a)$$

$$R_{22} = \sigma_\rho^2 \sin^2\tilde{\theta}_k \cos^2\tilde{\phi}_k + \sigma_\theta^2 \tilde{\rho}_k^2 \cos^2\tilde{\theta}_k \cos^2\tilde{\phi}_k + \sigma_\phi^2 \tilde{\rho}_k^2 \sin^2\tilde{\theta}_k \sin^2\tilde{\phi}_k \qquad (3.19b)$$

$$R_{33} = \sigma_\rho^2 \sin^2\tilde{\phi}_k + \sigma_\phi^2 \tilde{\rho}_k^2 \cos^2\tilde{\phi}_k, \qquad (3.19c)$$

and the off-diagonal elements are

$$\begin{aligned}R_{12} &= \sigma_\rho^2 \cos\tilde{\theta}_k \sin\tilde{\theta}_k \cos^2\tilde{\phi}_k - \sigma_\theta^2 \tilde{\rho}_k^2 \sin\tilde{\theta}_k \cos\tilde{\theta}_k \cos^2\tilde{\phi}_k\\ &\quad + \sigma_\phi^2 \tilde{\rho}_k^2 \cos\tilde{\theta}_k \sin\tilde{\theta}_k \sin^2\tilde{\phi}_k \qquad (3.20a)\\ R_{23} &= \sigma_\rho^2 \sin\tilde{\theta}_k \cos\tilde{\phi}_k \sin\tilde{\phi}_k - \sigma_\phi^2 \tilde{\rho}_k \sin\tilde{\theta}_k \cos\tilde{\phi}_k \sin\tilde{\phi}_k \qquad (3.20b)\\ R_{13} &= \sigma_\rho^2 \cos\tilde{\theta}_k \cos\tilde{\phi}_k \sin\tilde{\phi}_k - \sigma_\phi^2 \tilde{\rho}_k^2 \cos\tilde{\theta}_k \cos\tilde{\phi}_k \sin\tilde{\phi}_k. \qquad (3.20c)\end{aligned}$$

The conversion to Cartesian coordinates is done through linearization about the target position $\underline{\tilde{\varrho}}$ in spherical coordinates where the coordinates are obtained using true, predicted or measured values. For more details on this conversion see Appendix B.1. The expression for $R_k$ can be rewritten to

$$R_k = C_k^s R (C_k^s)^T \qquad (3.21)$$

$$C_k^s = \begin{bmatrix} \cos\tilde{\theta}_k \cos\tilde{\phi}_k & -\tilde{\rho}_k \sin\tilde{\theta}_k \cos\tilde{\phi}_k & -\tilde{\rho}_k \cos\tilde{\theta}_k \sin\tilde{\phi}_k \\ \sin\tilde{\theta}_k \cos\tilde{\phi}_k & \tilde{\rho}_k \cos\tilde{\theta}_k \cos\tilde{\phi}_k & -\tilde{\rho}_k \sin\tilde{\theta}_k \sin\tilde{\phi}_k \\ \sin\tilde{\phi}_k & 0 & \tilde{\rho}_k \cos\tilde{\phi}_k \end{bmatrix}. \qquad (3.22)$$

where $R$ is given by (3.7) and $C_k^s$ is called a Sensor Bias Transformation Matrix (SBTM). The measurement in local Cartesian coordinates is obtained by inserting (3.10) into (3.17):

$$\underline{z}_k^s = T_n^s T_u^n \left( \underline{p}_k^u - \underline{d}_{un}^u - \underline{b}_{ns}^u \right) + C_k^s \underline{b} + \underline{w}_k^s \tag{3.23a}$$

$$\underline{w}_k^s \sim \mathcal{N}\left(\underline{0}, R_k \delta_{kl}\right) \tag{3.23b}$$

$$\underline{b}_{ns}^u \sim \mathcal{N}\left(\underline{0}, B_{ns}^u\right) \tag{3.23c}$$

$$\underline{b}_n^s \sim \mathcal{N}\left(\underline{0}, B_n^s\right) \tag{3.23d}$$

$$\underline{b} \sim \mathcal{N}\left(\underline{0}, B\right) \tag{3.23e}$$

### 3.2.5 The Cartesian Measurement Model without Biases

If the measurement biases are known, or they are small enough to be neglected, the Cartesian measurement model without biases is obtained by treating the biases in (3.23) as zero which yields

$$\boxed{\begin{aligned} \underline{z}_k^s &= T_u^s \left( \underline{p}_k^u - \underline{d}_{us}^u \right) + \underline{w}_k^s \qquad (3.24a) \\ \underline{w}_k^s &\sim \mathcal{N}\left(\underline{0}, R_k \delta_{kl}\right). \qquad (3.24b) \end{aligned}}$$

### 3.2.6 The Bias Ignorant Cartesian Measurement Model

It is possible to account for the added uncertainty introduced by the biases without estimating them. This is done by adding a zero-mean white Gaussian noise vector $\underline{a}_k^s$ which accounts for the uncertainty of the unestimated biases. The bias ignorant Cartesian measurement model is obtained by ignoring the biases in (3.23) assuming $\{s\} = \{n\}$ and $\underline{b} = \underline{0}$ which yields

$$\boxed{\begin{aligned} \underline{z}_k^s &\approx T_u^n \left( \underline{p}_k^u - \underline{d}_{un}^u \right) + \underline{a}_k^s + \underline{w}_k^s \qquad (3.25a) \\ \underline{w}_k^s &\sim \mathcal{N}\left(\underline{0}, R_k^s \delta_{kl}\right) \qquad (3.25b) \\ \underline{a}_k^s &\sim \mathcal{N}\left(\underline{0}, A_k^s \delta_{kl}\right). \qquad (3.25c) \end{aligned}}$$

The covariance matrix $A_k^s$ is defined as

$$A_k^s = A_n^s(k) + A_{ns}^u(k) + A_b^s(k) \tag{3.26}$$

where $A_n^s$, $A_{ns}^u$ and $A_b^s$ (time index omitted) are the added measurement noise covariances due to the alignment bias, location bias and sensor bias respectively. The sensor bias covariance matrix is

$$A_b^s(k) = C_k^s B \left(C_k^s\right)^T, \tag{3.27}$$

and the location bias covariance matrix is

$$A_{ns}^u = \begin{bmatrix} \sigma_{b_x}^2 & 0 & 0 \\ 0 & \sigma_{b_y}^2 & 0 \\ 0 & 0 & \sigma_{b_z}^2 \end{bmatrix} \tag{3.28}$$

where the location bias variances in $\{u\}$ are found as elements of the diagonal. We will now calculate the alignment bias covariance matrix

$$A_n^s = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{12} & A_{22} & A_{23} \\ A_{13} & A_{23} & A_{33} \end{bmatrix}. \tag{3.29}$$

The calculation is based on linearization about a known target position given by $\tilde{x}$, $\tilde{y}$ and $\tilde{z}$, which could be estimates or measurements. Uncertainty in target position is neglected. The true target position is

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & -b_\gamma & b_\beta \\ b_\gamma & 1 & -b_\alpha \\ -b_\beta & b_\alpha & 1 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} = \begin{bmatrix} \tilde{x} - b_\gamma \tilde{y} + b_\beta \tilde{z} \\ \tilde{y} + b_\gamma \tilde{x} - b_\alpha \tilde{z} \\ \tilde{z} - b_\beta \tilde{x} + b_\alpha \tilde{y} \end{bmatrix}. \tag{3.30}$$

Assuming

$$E\{b_\alpha\} = E\{b_\beta\} = E\{b_\gamma\} = E\{b_\alpha b_\beta\} = E\{b_\alpha b_\gamma\} = E\{b_\beta b_\gamma\} = 0 \tag{3.31}$$

the variances are

$$\begin{aligned} A_{11} &= E\left\{x^2\right\} - E\{x\}^2 \\ &= \tilde{y}^2 \sigma_{b_\gamma}^2 + \tilde{z}^2 \sigma_{b_\beta}^2 && \text{(3.32a)} \\ A_{22} &= \tilde{x}^2 \sigma_{b_\gamma}^2 + \tilde{z}^2 \sigma_{b_\alpha}^2 && \text{(3.32b)} \\ A_{33} &= \tilde{x}^2 \sigma_{b_\beta}^2 + \tilde{y}^2 \sigma_{b_\alpha}^2 && \text{(3.32c)} \end{aligned}$$

and the cross covariances are:

$$\begin{aligned} A_{12} &= E\{xy\} - E\{x\} E\{y\} \\ &= -\tilde{x}\tilde{y}\sigma_{b_\gamma}^2 && \text{(3.33a)} \\ A_{13} &= -\tilde{x}\tilde{z}\sigma_{b_\beta}^2 && \text{(3.33b)} \\ A_{23} &= -\tilde{y}\tilde{z}\sigma_{b_\alpha}^2 && \text{(3.33c)} \end{aligned}$$

### 3.2.7 The Cartesian Measurement Model with Spherical State Vector

In this section we will derive a nonlinear Cartesian measurement model which will be used to prove Proposition 3.1 in Section 3.2.8. The measurement of target position with sensor bias in spherical coordinates is given by

$$\underline{z}_k^n = T_s^n \underline{h}_c \left( \underline{\varrho}_k + \underline{b} \right) - \underline{b}_{ns}^n + \underline{w}_k^n. \tag{3.34}$$

We choose the measurement in $\{n\}$ to include alignment, location and sensor biases in the measurement model. In the following we omit the time index $k$ and define

$$\underline{p}_k^n = T_s^n \underline{h}_c \left( \underline{\varrho}_k + \underline{b} \right) - \underline{b}_{ns}^n = \begin{bmatrix} x_b & y_b & z_b \end{bmatrix}^T. \tag{3.35}$$

We start with the measurement in spherical coordinates and its biases

$$\rho_b = \rho + b_\rho \tag{3.36a}$$
$$\theta_b = \theta + b_\theta \tag{3.36b}$$
$$\phi_b = \phi + b_\phi. \tag{3.36c}$$

This measurement is transformed to Cartesian coordinates using (2.27), obtaining

$$x_z = \rho_b \cos\theta_b \cos\phi_b \tag{3.37a}$$
$$y_z = \rho_b \sin\theta_b \cos\phi_b \tag{3.37b}$$
$$z_z = \rho_b \sin\phi_b. \tag{3.37c}$$

Adding the alignment and location biases we get the deterministic biased position measurement in Cartesian coordinates

$$x_b = x_z - b_\gamma y_z + b_\beta z_z - b_x \tag{3.38a}$$
$$y_b = y_z + b_\gamma x_z - b_\alpha z_z - b_y \tag{3.38b}$$
$$z_b = z_z - b_\beta x_z + b_\alpha y_z - b_z \tag{3.38c}$$

Using (3.36), (3.37), (3.38) we get

$$\begin{aligned} x_b &= (\rho + b_\rho)\cos(\theta + b_\theta)\cos(\phi + b_\phi) \\ &\quad - (\rho + b_\rho)\, b_\gamma \sin(\theta + b_\theta)\cos(\phi - b_\phi) \\ &\quad + (\rho + b_\rho)\, b_\beta \sin(\phi + b_\phi) - b_x \end{aligned} \tag{3.39a}$$

$$\begin{aligned} y_b &= (\rho + b_\rho)\sin(\theta + b_\theta)\cos(\phi + b_\phi) \\ &\quad + (\rho + b_\rho)\, b_\gamma \cos(\theta + b_\theta)\cos(\phi - b_\phi) \\ &\quad - (\rho + b_\rho)\, b_\alpha \sin(\phi + b_\phi) - b_y \end{aligned} \tag{3.39b}$$

$$\begin{aligned} z_b &= (\rho + b_\rho)\sin(\phi + b_\phi) \\ &\quad - (\rho + b_\rho)\, b_\beta \cos(\theta + b_\theta)\cos(\phi + b_\phi) \\ &\quad + (\rho + b_\rho)\, b_\alpha \sin(\theta + b_\theta)\cos(\phi + b_\phi) - b_z. \end{aligned} \tag{3.39c}$$

Using the following trigonometric relations

$$\begin{aligned} \sin(\theta + b_\theta) &= \sin\theta\cos b_\theta + \cos\theta \sin b_\theta \\ &\approx \sin\theta + b_\theta \cos\theta \end{aligned} \tag{3.40a}$$

$$\begin{aligned} \cos(\theta + b_\theta) &= \cos\theta\cos b_\theta - \sin\theta \sin b_\theta \\ &\approx \cos\theta - b_\theta \sin\theta \end{aligned} \tag{3.40b}$$

where the approximation is obtained through the small angle bias assumption, and only including first order terms, we obtain

$$\begin{aligned} x_b &\approx (\rho + b_\rho)\cos\theta\cos\phi - (b_\theta + b_\gamma)\,\rho\sin\theta\cos\phi \\ &\quad + (b_\beta - b_\phi\cos\theta)\,\rho\sin\phi - b_x \end{aligned} \tag{3.41a}$$

$$\begin{aligned} y_b &\approx (\rho + b_\rho)\sin\theta\cos\phi + (b_\theta + b_\gamma)\,\rho\cos\theta\cos\phi \\ &\quad - (b_\phi\sin\theta + b_\alpha)\,\rho\sin\phi - b_y \end{aligned} \tag{3.41b}$$

$$z_b \approx (\rho + b_\rho)\sin\phi + (b_\phi - b_\beta\cos\theta + b_\alpha\sin\theta)\,\rho\cos\phi - b_z. \tag{3.41c}$$

### 3.2.8 The Alignment Bias and the Sensor Angle Biases

Both the alignment bias $\underline{b}_n^s$ and the sensor bias $\underline{b}$ include angle biases, five in total. Assuming that all angle biases are small, the CTM $T_n^s$ represents all possible rotations of $\{s\}$ with respect to $\{n\}$. This means that it may be possible to reduce the number of angles to estimate, without sacrificing performance.

**Proposition 3.1**
*The effect of the sensor angle bias in azimuth can without loss of generality be included into the alignment bias Euler angle yaw.*

**Proof:** Looking at (3.41) we see that $b_\theta$ and $b_\gamma$ always appear as a sum. Since both are constant they cannot be separated in a bias estimator. Only the sum is observable, which is expected since $b_\theta$ and $b_\gamma$ both are defined as a rotation about the $z$ axis. Thus we can choose to represent the sum of $b_\theta$ and $b_\gamma$ either as $b_\gamma$, which leads to $b_\theta = 0$, or vice versa. To prove the proposition we state that $b_\gamma$ represents the sum of these biases, and set $b_\theta = 0$. □

### 3.2.9 Universal, Absolute and Relative Bias Estimation

One way of estimating sensor measurement biases is to assume that all sensors are biased with respect to a universal coordinate system, and try to estimate all biases for all sensors. This is called *universal* bias estimation. Universal estimation is not necessarily possible for all biases, and this will be discussed later. The advantage of this method is that if successful, fused target estimates are unbiased in a sensor independent coordinate system. An estimator using this method is called a Universal Bias Estimator (UBE).

Another way of modeling biases is to choose one sensor as the *master* sensor. The biases of the other sensors are estimated relative to this sensor. This makes it possible to simplify the mathematical model for the measurement biases. Some biases may for instance be additive under this assumption, and it is thus sufficient to estimate a sum of biases instead of individual biases. The latter is usually the case in universal bias estimation. If the resulting mathematical model in theory leads to unbiased estimates in the master sensor coordinates, the estimator is called an *absolute* bias estimator. The advantage of absolute bias estimation with respect to universal bias estimation is that it is simpler to do successfully. The disadvantage is that target tracks will be biased when they are transformed to a universal coordinate system. If it is desired to track the targets in a coordinate system associated to the master sensor, this method is well suited. An estimator using this method is called an Absolute Bias Estimator (ABE).

A common method in the literature is to choose a master sensor and assume that this sensor has no biases. The biases of the other sensors are estimated *relative* to this sensor. Successful use of this method leads to fused target estimates that are biased with the neglected master sensor biases only. An estimator using this method is called a Relative Bias Estimator (RBE).

### 3.2.10 General Measurement Model for Universal Bias Estimation

We seek a general measurement model for universal bias estimation which takes into account all the sensor measurement biases discussed in Section 3.2. First it is necessary to define in which coordinate system the targets will be tracked, and we choose the coordinate system $\{u\}$ which is independent of the sensors. The transformation between $\{u\}$ and the nominal sensor coordinate system $\{n\}$ is assumed known. Thus we use the results from Section 3.2.3. The target measurement from Sensor $i \in \{0, 1, \ldots, M\}$ is according to (3.14)

$$\underline{z}_{i,k}^{i} = \underline{h}_s \left( T_{n_i}^{s_i} T_u^{n_i} \left( \underline{p}_k^u - \underline{d}_{un_i}^u - \underline{b}_{n_i s_i}^u \right) \right) + \underline{b}_i + \underline{w}_{i,k} \tag{3.42a}$$

$$\underline{w}_{i,k} \sim \mathcal{N}\left(\underline{0}, R\delta_{kl}\right) \tag{3.42b}$$

where $\underline{z}_{i,k}^{i}$ is the measurement in local spherical sensor coordinates, $\underline{h}_s(\cdot)$ is the CTF from Cartesian to spherical coordinates, $T_{n_i}^{s_i}$ is the CTM representing the alignment bias, $\underline{p}_k^u$ is the target position, $\underline{b}_{n_i s_i}^u$ is the location bias, $\underline{b}_i$ is the sensor bias, $\underline{w}_{i,k}$ is the white measurement noise and $R$ is its covariance matrix given by (3.7). The sensor bias includes biases in range and elevation since according to Section 3.2.8 the sensor angle bias in azimuth is included in the alignment bias Euler angle yaw. As stated in Section 3.2.4 it is possible to convert spherical measurements to Cartesian coordinates. If this is desirable the target measurement from Sensor $i$ is according to (3.23)

$$\underline{z}_{i,k}^{s_i} = T_{n_i}^{s_i} T_u^{n_i} \left( \underline{p}_k^u - \underline{d}_{un_i}^u - \underline{b}_{n_i s_i}^u \right) + C_k^{s_i} \underline{b}_i + \underline{w}_{i,k}^{s_i} \tag{3.43a}$$

$$\underline{w}_{i,k}^{s_i} \sim \mathcal{N}\left(\underline{0}, R_{i,k}^{s_i} \delta_{kl}\right) \tag{3.43b}$$

where $C_k^{s_i}$ is the SBTM from spherical to Cartesian coordinates given by (3.22), and $R_{i,k}^{s_i}$ is the measurement covariance matrix given by (3.21). These matrices are calculated based on linearization about the true values of $\rho$, $\theta$ and $\phi$ for the target. Measured or estimated values can be used instead of the true values when they are not available.

### 3.2.11 General Measurement Model for Absolute Bias Estimation

In this section the models are derived assuming converted Cartesian target measurements in order to simplify expressions and clarify the coordinate transformations involved. In absolute bias estimation a master sensor is chosen, whose true

local Cartesian coordinates are denoted $\{s_0\}$. The targets are tracked in these coordinates.

#### 3.2.11.1 Biased Sensor

In order to simplify the number of biases to estimate, some of the master sensor biases are not estimated. To retain absolute accuracy these biases are modeled as part of the biases of the biased sensors. Thus the biased sensor measurement is affected by its own biases as well as the master sensor biases. Since the sensor bias effect in Cartesian coordinates depends on target position, the sensor bias needs to be estimated for all sensors. The alignment and location biases however, are constant in Cartesian coordinates. Thus we will model the alignment and location biases of the master sensor as part of the biases of the biased sensors.

The target is tracked in $\{s_0\}$, thus the target position in the measurement model must be $\underline{p}_k^{s_0}$. The target position in $\{n_0\}$ is given by

$$\underline{p}_k^{n_0} = T_{s_0}^{n_0} \left( \underline{p}_k^{s_0} - \underline{b}_{s_0 n_0}^{s_0} \right) \tag{3.44}$$

where the alignment bias $T_{s_0}^{n_0}$ and the location bias $\underline{b}_{s_0 n_0}^{s_0}$ of the master sensor are included. Since tracking is performed in the master sensor coordinate system $\{s_0\}$, the target measurement from a biased sensor should be expressed as a function of the target position in $\{s_0\}$. The target position in $\{s_i\}$ is

$$\underline{p}_k^{s_i} = T_{n_i}^{s_i} T_{n_0}^{n_i} \underline{p}_k^{n_0} - \underline{d}_{n_0 n_i}^{s_i} - \underline{b}_{n_i s_i}^{s_i} + C_k^{s_i} \underline{b}_i \tag{3.45}$$

where the alignment bias $T_{n_i}^{s_i}$, the location bias $\underline{b}_{n_i s_i}^{s_i}$ and the sensor bias $\underline{b}_i$ of the biased sensor have been added. To obtain the measurement equation of the biased sensor we insert (3.44) into (3.45) and add measurement noise. The target measurement from a biased sensor (Sensor $i$) at time $k$ is

$$\underline{z}_{i,k}^{s_i} = T_{n_i}^{s_i} T_{n_0}^{n_i} T_{s_0}^{n_0} \left( \underline{p}_k^{s_0} - \underline{b}_{s_0 n_0}^{s_0} - \underline{d}_{n_0 n_i}^{s_0} - \underline{b}_{n_i s_i}^{s_0} \right) + C_k^{s_i} \underline{b}_i + \underline{w}_{i,k}^{s_i} \tag{3.46a}$$

$$\underline{w}_{i,k}^{s_i} \sim \mathcal{N} \left( \underline{0}, R_{i,k}^{s_i} \delta_{kl} \right) \tag{3.46b}$$

where the alignment bias is $T_{n_i}^{s_i}$, $\underline{b}_{n_i s_i}^{s_0}$ is the location bias, and $\underline{b}_i$ is the sensor bias of Sensor $i$. Note that $T_{s_0}^{n_0}$ and $\underline{b}_{s_0 n_0}^{s_0}$ are the alignment and location biases respectively of the master sensor. In absolute bias estimation we seek to reduce the number of biases to be estimated, but retain the accuracy of the measurement model. We manipulate (3.46a) to achieve this by including the master sensor

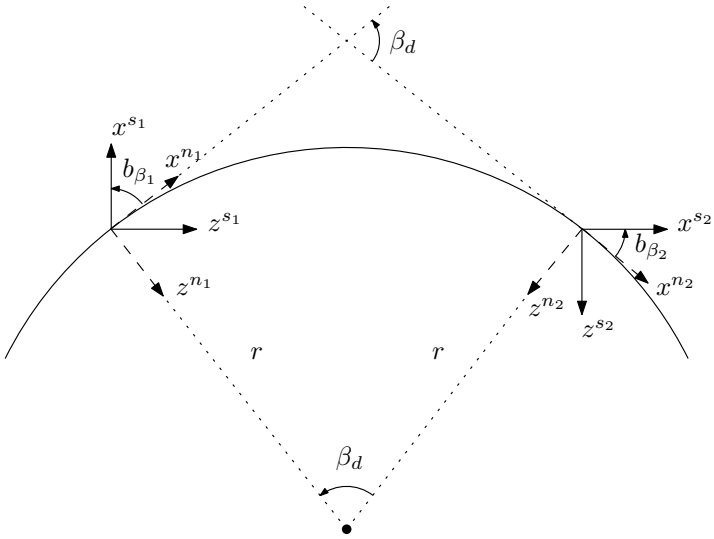



***Figure 3.2:*** *Elevation alignment biases $b_{\beta_i}$ and elevation angle $\beta_d$ due to the curvature of the Earth.*

biases in the biases of the other sensors. The location bias is straightforward, by defining the relative location bias as

$$\underline{b}_{s_0 s_i}^{s_0} = \underline{b}_{s_0 n_0}^{s_0} + \underline{b}_{n_i s_i}^{s_0}. \tag{3.47}$$

The location bias of Sensor 0 is thus included in all the relative location biases of Sensor $i$. Hence these biases are correlated. The cross-covariances for Sensor $i$ and $j$, which follow from Definition 3.1, are

$$\sigma_{b_{i,x} b_{j,x}} = \sigma_{b_{0,x}}^2 \tag{3.48a}$$
$$\sigma_{b_{i,y} b_{j,y}} = \sigma_{b_{0,y}}^2 \tag{3.48b}$$
$$\sigma_{b_{i,z} b_{j,z}} = \sigma_{b_{0,z}}^2 \tag{3.48c}$$

where $\sigma_{b_{0,x}}^2$, $\sigma_{b_{0,x}}^2$ and $\sigma_{b_{0,x}}^2$ are the assumed location bias variances of Sensor $0$.

To obtain one alignment bias in (3.46a) we define $T_{s_0}^{s_i} = T_{n_i}^{s_i} T_{n_0}^{n_i} T_{s_0}^{n_0}$ as the alignment bias to estimate. In general the Euler angles of $T_{n_0}^{n_i}$ are arbitrary, thus the Euler angles of $T_{s_0}^{s_i}$ are not assumed small. Note that these Euler angles will be correlated across the biased sensors.

**Example 3.1**

If the Euler angles of $T_{n_0}^{n_i}$ are small enough, then $T_{s_0}^{s_i}$ can be assumed small. Given the maximum range of sensor coverage and the curvature of the Earth, it

is possible to investigate whether the Euler angles of $T_{s_0}^{s_i}$ can be assumed small. Approximating the Earth as a sphere, its radius $r$ can be approximated to 6371 km when conserving the area or volume of the Earth ellipsoid. An angle $\beta$ is assumed small when $\sin\beta \approx \beta$. Defining the angle $\beta$ to be small when this is true by a 1 $\%$ margin, it must obey

$$\beta \leq \frac{\pi}{13}. \tag{3.49}$$

For two sensors this angle is the sum of an angle $\beta_d$ due to the distance between them and the curvature of the Earth, and their respective alignment biases. As seen from Figure 3.2

$$\beta_d = \frac{d}{r} \tag{3.50}$$

where $d$ is the distance between the sensors. Denoting $b_{\beta_i}$ as the worst case pitch bias of sensor $i$, and using (3.49) and (3.50) we get

$$\begin{aligned} \beta_d + b_{\beta_1} + b_{\beta_2} &= \beta \\ \frac{d}{r} + b_{\beta_1} + b_{\beta_2} &\leq \frac{\pi}{13} \end{aligned} \tag{3.51}$$

which must hold for the small angle assumption to be true by a 1 $\%$ margin. Suppose that both sensors have a maximum range of 500 km, and that they have the same worst case pitch bias. Setting $d = 500$ km, we get a worst case pitch bias for each sensor of $82$ mrad $= 4.7°$. So if the biases are larger than this value, the Euler angles of $T_{s_0}^{s_i}$ cannot be assumed small. This discussion applies equally to the roll bias $b_{\alpha_i}$, but not the yaw bias $b_{\gamma_i}$ which is not affected by the curvature of the Earth in this manner.

In the general case it is possible to define one alignment bias with small angles for each sensor as described in the following proposition.

**Proposition 3.2**
*The alignment bias for Sensor $i$ can be approximated to*

$$T_{s_0}^{s_i} = T_{n_i}^{s_i} T_{n_0}^{n_i} T_{s_0}^{n_0} \approx T^{s_i} T_{n_0}^{n_i} \tag{3.52a}$$

$$T^{s_i} = I + S\left(\underline{b}^{s_i}\right) \tag{3.52b}$$

$$S\left(\underline{b}^{s_i}\right) = S\left(\underline{b}_{n_i}^{s_i}\right) - T_{n_0}^{n_i} S\left(\underline{b}_{s_0}^{n_0}\right) T_{n_i}^{n_0} \tag{3.52c}$$

*ignoring higher order terms where $T^{s_i}$ is an alignment bias with small angles and the angles of $T_{n_0}^{n_i}$ are arbitrary and known.*

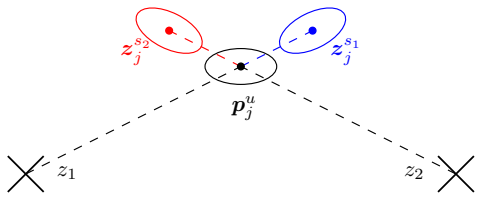



***Figure 3.3:*** *Two sensors with locations denoted by their vertical axes (cross), $z_1$ and $z_2$, are each subject to a range bias. Each sensor measures the position of Target $j$, and the measurements are represented by their value (blue and red dots) and their covariance (blue and red ellipses). Requiring the measurements to overlap yields a unique solution. The range biases are exaggerated for illustrative purposes.*

**Proof:** We will solve for the alignment bias $T^{s_i}$ using the following equation

$$T_{n_i}^{s_i} T_{n_0}^{n_i} T_{s_0}^{n_0} = T^{s_i} T_{n_0}^{n_i} \tag{3.53a}$$

$$\left(I + S\left(\underline{b}_{n_i}^{s_i}\right)\right) T_{n_0}^{n_i} \left(I - S\left(\underline{b}_{s_0}^{n_0}\right)\right) = \left(I + S\left(\underline{b}^{s_i}\right)\right) T_{n_0}^{n_i} \tag{3.53b}$$

where we have used (3.11). Performing matrix multiplications on the left hand side and using the fact that $I = T_{n_i}^{n_0} T_{n_0}^{n_i}$ we get

$$T^{s_i} T_{n_0}^{n_i} = \left(I + S\left(\underline{b}_{n_i}^{s_i}\right) - T_{n_0}^{n_i} S\left(\underline{b}_{s_0}^{n_0}\right) T_{n_i}^{n_0} - S\left(\underline{b}_{n_i}^{s_i}\right) T_{n_0}^{n_i} S\left(\underline{b}_{s_0}^{n_0}\right) T_{n_i}^{n_0}\right) T_{n_0}^{n_i}. \tag{3.54}$$

Neglecting the last higher order term we get

$$S\left(\underline{b}^{s_i}\right) \approx S\left(\underline{b}_{n_i}^{s_i}\right) - T_{n_0}^{n_i} S\left(\underline{b}_{s_0}^{n_0}\right) T_{n_i}^{n_0} \tag{3.55}$$

which has a solution if the right hand side is a skew symmetric matrix according to (3.13). This is the case since both terms are skew symmetric matrices. The second term is skew symmetric since it is a similarity transformation of a skew symmetric matrix using orthogonal transformation matrices (Weisstein, 2013). □

In general the master sensor bias cannot be included in the sensor bias of the biased sensors. This can be seen from $C_k^{s_0}$ which is not constant and from Figure 3.3 which shows that the range directions of two sensors are not the same. In fact this is only the case if the targets are located along the line connecting the two sensors, or if the targets are so far away that the distance between the sensors can be neglected. None of these assumptions are made herein. Finally we arrive at

the target measurement from a biased sensor (Sensor $i$) at time $k$ which is

$$\boxed{\begin{aligned} \underline{z}_{i,k}^{s_i} &= T^{s_i}\left(\underline{p}_k^{s_0} - \underline{d}_{n_0 n_i}^{s_0} - \underline{b}_{s_0 s_i}^{s_0}\right) + C_k^{s_i}\underline{b}_i + \underline{w}_{i,k}^{s_i} && \text{(3.56a)} \\ \underline{w}_{i,k}^{s_i} &\sim \mathcal{N}\left(\underline{0}, R_{i,k}^{s_i}\delta_{kl}\right). && \text{(3.56b)} \end{aligned}}$$

#### 3.2.11.2 Master Sensor

The master sensor alignment bias is estimated through the estimation of the alignment biases of the other sensors. The same applies to the location bias, hence the only master sensor bias remaining is the sensor bias. The target measurement in Cartesian coordinates from the master sensor (Sensor 0) at time $k$ is given by (3.17)

$$\boxed{\begin{aligned} \underline{z}_{0,k}^{s_0} &= \underline{p}_k^{s_0} + C_k^{s_0}\underline{b}_0 + \underline{w}_{0,k}^{s_0} && \text{(3.57a)} \\ \underline{w}_{0,k}^{s_0} &\sim \mathcal{N}\left(\underline{0}, R_{0,k}^{s_0}\delta_{kl}\right). && \text{(3.57b)} \end{aligned}}$$

### 3.2.12 General Measurement Model for Relative Bias Estimation

In relative bias estimation a master sensor is chosen, whose true local Cartesian coordinates are denoted $\{s_0'\}$. In this case the master sensor biases will not be estimated, which leads to tracking being performed in $\{s_0'\}$ rather than $\{s_0\}$. The alignment and location biases of the master sensor are included in the biases of the biased sensors as in the previous section. However the sensor bias of the master sensor is neglected and compensated for by increasing the measurement noise.

#### 3.2.12.1 Biased Sensor

We start by defining a zero-mean white Gaussian noise vector $\underline{a}_{i,k}^{s_0}$ which will replace the sensor bias of the master sensor

$$\begin{aligned} \underline{a}_{0,k}^{s_0} &\approx C_k^{s_0}\underline{b}_0 && \text{(3.58a)} \\ \underline{a}_{0,k}^{s_0} &\sim \mathcal{N}\left(\underline{0}, A_{0,k}^{s_0}\delta_{kl}\right) && \text{(3.58b)} \\ A_{0,k}^{s_0} &= C_k^{s_0} B_0 \left(C_k^{s_0}\right)^T && \text{(3.58c)} \end{aligned}$$

where $B_0$ is given by (3.9). The target measurement in Cartesian coordinates for a biased sensor (Sensor i) at time $k$ is approximated to

$$\underline{z}_{i,k}^{s_i} \approx T^{s_i}\left(\underline{p}_k^{s_0'} + \underline{a}_{0,k}^{s_0} - \underline{d}_{n_0 n_i}^{s_0} - \underline{b}_{s_0 s_i}^{s_0}\right) + C_k^{s_i}\underline{b}_i + \underline{w}_{i,k}^{s_i}. \tag{3.59}$$

Defining

$$\underline{a}_{i,k}^{s_i} = T^{s_i}\underline{a}_{0,k}^{s_0} \tag{3.60a}$$
$$\underline{a}_{i,k}^{s_i} \sim \mathcal{N}\left(\underline{0}, A_{i,k}^{s_i}\delta_{kl}\right) \tag{3.60b}$$
$$A_{i,k}^{s_i} = T^{s_i}C_k^{s_0}B_0\left(T^{s_i}C_k^{s_0}\right)^T \tag{3.60c}$$

we get

$$\boxed{\begin{aligned}\underline{z}_{i,k}^{s_i} &\approx T^{s_i}\left(\underline{p}_k^{s_0'} - \underline{d}_{n_0 n_i}^{s_0} - \underline{b}_{s_0 s_i}^{s_0}\right) + C_k^{s_i}\underline{b}_i + \underline{a}_{i,k}^{s_i} + \underline{w}_{i,k}^{s_i} && (3.61a)\\ \underline{w}_{i,k}^{s_i} &\sim \mathcal{N}\left(\underline{0}, R_{i,k}^{s_i}\delta_{kl}\right). && (3.61b)\end{aligned}}$$

Note that because of this approximation we get a residual bias which may or may not be small enough to be neglected.

#### 3.2.12.2 Master Sensor

The target measurement in Cartesian coordinates from the master sensor (Sensor 0) at time $k$ is given by

$$\boxed{\begin{aligned}\underline{z}_{0,k}^{s_0'} &= \underline{p}_k^{s_0'} + \underline{w}_{0,k}^{s_0'} && (3.62a)\\ \underline{w}_{0,k}^{s_0'} &\sim \mathcal{N}\left(\underline{0}, R_{0,k}^{s_0'}\delta_{kl}\right). && (3.62b)\end{aligned}}$$

## 3.3 Measurement Model Discussion

### 3.3.1 Summary

The system model uses the General Measurement Model for Universal Bias Estimation, i.e. (3.42) for all sensors, to produce measurements. The presented measurement models will be used in the following estimators:

*UBE* The Universal Bias Estimator uses the General Measurement Model for Universal Bias Estimation, i.e. (3.42) for all sensors.

*ABE* The Absolute Bias Estimator uses the General Measurement Model for Absolute Bias Estimation, i.e. (3.56) for all sensors except the master sensor which is modeled by (3.57).

*RBE* The Relative Bias Estimator uses the General Measurement Model for Relative Bias Estimation, i.e. (3.61) for all sensors except the master sensor which is modeled by (3.62).

### 3.3.2 General Measurement Model for Universal Bias Estimation

The measurement model for the UBE is designed for tracking in a sensor independent coordinate system. The measurement (3.43) is given in spherical coordinates which coincides with the basic model for the sensor measurement (3.6). The primary benefit of this model is that it enables estimation of all biases for all sensors, which makes it possible to transform a target position to any known coordinate system as long as the biases are successfully estimated. The drawbacks of this model are a large number of unknowns (target state and biases) to be estimated, and the fact that the location bias is unestimable if all sensor locations are biased and all targets are moving.

### 3.3.3 General Measurement Model for Absolute Bias Estimation

The measurement model for the ABE is designed for tracking in the coordinate system of a master sensor. The measurement (3.56) is given in Cartesian coordinates because that allows us to have a linear measurement model for the master sensor. The measurement model is mathematically equivalent to the measurement model of the UBE, but tracking in master sensor coordinates leads to a reduction of unknowns with respect to the UBE. The ABE is able to estimate all of the biases treated herein. In the development of the ABE it is shown that the model holds for small alignment biases even if the coordinate transformation from a biased sensor to the master sensor involves large angles.

### 3.3.4 General Measurement Model for Relative Bias Estimation

The measurement model for the RBE is designed for tracking in the coordinate system of a master sensor where its biases are not estimated. The measurement

(3.61) is given in Cartesian coordinates because that allows us to have a linear measurement model for the master sensor. The number of unknowns is reduced with respect to the ABE since the master sensor biases are not estimated. They are however compensated for by adding measurement noise. The primary advantage of the RBE is a further reduction of unknowns yielding a simpler model than the ABE. It is in fact an approximation of the ABE, and in many cases the RBE may perform just as well as the ABE. Note that this approximation leads to tracking in different coordinate systems for the master sensor and the biased sensors respectively. Since the master sensor biases are not estimated, the measurements from the biased sensors cannot be transformed to the coordinate system where the master sensor measurement is represented and vice versa. This leads to a residual bias, and its size determines the difference in performance of the RBE with respect to the ABE.

## 3.4 The Static Cramer-Rao Lower Bound

Now that the measurement models include all the biases of interest, the fundamental question becomes: *Is it possible to find all the biases present, and the positions of all the targets?* In this section we will look at stochastic *estimability* conditions, that is what conditions need to be fulfilled in order to estimate successfully all the biases present and all the target positions. Note that a bias can be strongly or weakly estimable, which is different from deterministic observability where the bias is either observable or not. Observability is a necessary condition in order to have estimability. We limit the discussion to cases where the number of targets and sensors is limited.

### 3.4.1 The Estimability Index

The static Cramer-Rao Lower Bound (CRLB) is a useful tool to quantify estimability, since it yields the lower bound for estimator performance.

**Definition 3.2.** A variable is estimable if the measurements available lead to a decrease in its variance with respect to its prior variance using the static CRLB.

Prior variance is included as additional measurements of the state vector. Using the CRLB to define estimability has previously been suggested in Bar-Shalom et al. (2001). The bias is weakly estimable if the decrease is small, and strongly estimable if the decrease is large. If there is no decrease, the bias is *unestimable*. For details on the static CRLB see Section 2.1.1. We will study estimability by plotting an estimability index defined as follows.

**Definition 3.3.** The estimability index $\eta$ for the state vector $\underline{x}$ with normalized CRLB positive definite covariance matrix $P_n$ is

$$\eta = 1 - \sqrt{\lambda_{max}} \tag{3.63}$$

where $\lambda_{max} > 0$ is the largest eigenvalue of $P_n$ giving $\eta \leq 1$. The covariance matrix $P_n$ is normalized using the current and prior CRLB covariance matrices $P$ and $P_0$ respectively. The normalization of the matrix elements is done according to

$$P_m = M_0^T P M_0 \tag{3.64a}$$

$$P_n(i,j) = P_m(i,j) / \sqrt{\Lambda_0(i,i)\,\Lambda_0(j,j)} \tag{3.64b}$$

where $i$ and $j$ are matrix indices and $P$ is given by (2.5). The diagonal Eigenvalue matrix $\Lambda_0$ is obtained from

$$\Lambda_0 = M_0^T P_0 M_0 \tag{3.65}$$

where $\Lambda_0$ contains the Eigenvalues of $P_0$ along its diagonal, and $M_0$ is the orthonormal Eigenvector matrix of $P_0$.

It is possible to simplify (3.64) in the case where $P_0$ is diagonal. In that case it is possible to choose $\Lambda_0 = P_0$ and $M_0 = I$ according to (3.65). Thus (3.64) simplifies to

$$P_n(i,j) = P(i,j) / \sqrt{P_0(i,i)\,P_0(j,j)}. \tag{3.66}$$

Note that using $\lambda_{max}$ in (3.63) is equivalent to using the matrix 2-norm for positive definite symmetric matrices. If $P(i,j) \leq \sqrt{\Lambda_0(i,i)\,\Lambda_0(j,j)}$, all elements of $P_n$ are between 0 and 1 and $\eta \in [0,1]$. If the matrix 2-norm of $P$ has increased with respect to the matrix 2-norm of $P_0$, $\lambda_{max} > 1$ and $\eta < 0$. From (3.63) it is seen that $\eta \leq 1$ which should be interpreted as follows.

$\eta = 1$ The state vector elements are estimable and known exactly using the available measurements.

$0 < \eta < 1$ The state vector is estimable, and the variance of all elements has decreased.

$\eta = 0$ The state vector is unestimable since at least one linear combination of the state vector elements has unchanged variance. Variance may have decreased in other directions in the state space.

$\eta < 0$ The state vector is unestimable since at least one linear combination of the state vector elements has increased variance. Variance may have decreased in other directions in the state space.

We are now ready to define estimability for a set of biases.

**Definition 3.4.** A set of biases is estimable if $0 < \eta \leq 1$ for the set. The estimability index is calculated for the submatrix of $P_n$ which corresponds to the elements of the set.

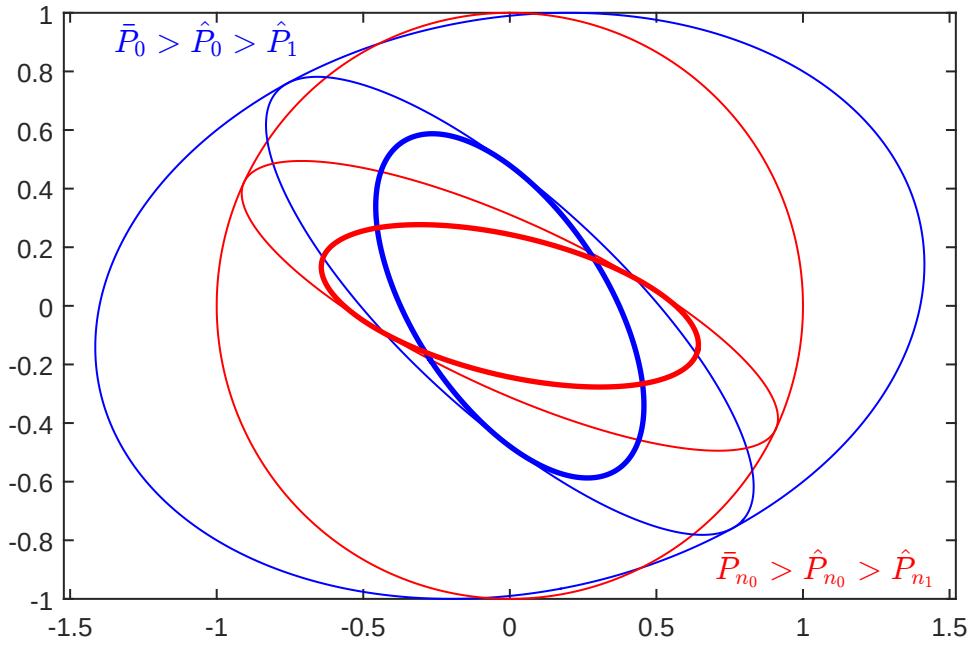
In this work examples of sets of biases are the alignment bias, location bias and sensor bias. Note that $\sqrt{\lambda_{max}}$ is the semimajor axis of the covariance ellipsoid of $P_n$. The square root in (3.63) is used to quantify improvement in standard deviation, instead of variance. This choice is made because it is common to refer to uncertainty in standard deviation, for instance when plotting one sigma curves. The following example provides more explanations regarding the estimability index.

**Example 3.2**

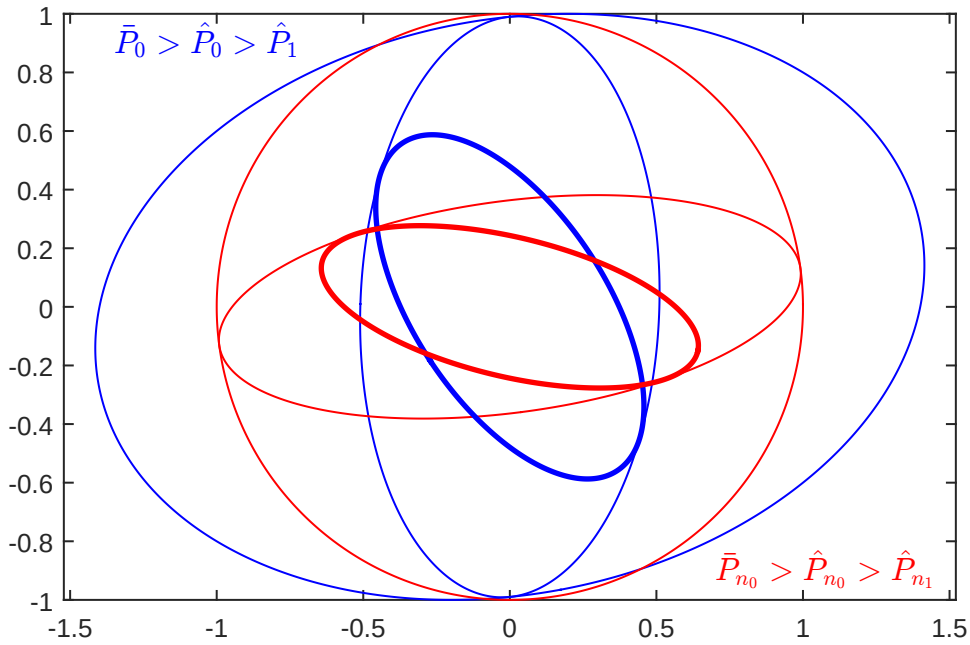
In this example we will look at how the normalized covariance matrix $P_n$ changes after consecutive measurement updates, and the associated values of the estimability index $\eta$. Suppose that we have the prior 2D covariance matrix

$$\bar{P}_0 = \begin{bmatrix} 2 & 0.2 \\ 0.2 & 1 \end{bmatrix}. \tag{3.67}$$

Furthermore we assume that we receive two measurement updates, and we will verify that the resulting normalized covariance matrix is the same regardless of the processing order of these measurements. One measurement measures the sum of the state vector elements with the measurement matrix $H_0 = \begin{bmatrix} 1 & 1 \end{bmatrix}$, and the other measurement is of the first element where $H_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}$. The measurement noise variance is $R = 0.3$. In the first case $H_0$ is processed first, then $H_1$. Figure 3.4a shows the covariance ellipses of $\bar{P}_0$, $\hat{P}_0$ and $\hat{P}_1$ where $\hat{P}_0$ and $\hat{P}_1$ are the resulting covariance matrices after the first and second measurement updates respectively. The corresponding normalized covariance matrices $\bar{P}_{n_0}$, $\hat{P}_{n_0}$ and $\hat{P}_{n_1}$ are also shown. These are normalized with respect to $\bar{P}_0$. Figure 3.4b shows the same kind of plot, but in this case the measurement updates are reversed. We see that $\hat{P}_1$ and $\hat{P}_{n_1}$ are the same in both cases with $\eta = 0.3412$ showing that the problem is estimable. The covariance matrices after one measurement update $\hat{P}_0$ and $\hat{P}_{n_0}$ differ in Figure 3.4a and Figure 3.4b, but in both figures the associated $\eta = 0$. We understand that both measurement updates are needed to get estimability with respect to $\bar{P}_0$.




***(a)***




***(b)***

***Figure 3.4:*** *The plots show unnormalized (blue) and normalized (red) covariance ellipsoids. In (a) the measured sum of the state vector elements is processed first, and in (b) the first element is measured first.*

In general when using the estimability index, care must be taken in the choice of the covariance matrix used to normalize the posterior covariance matrix. In this work the measurements are processed simultaneously in an augmented state vector, and the normalization is done with respect to the prior covariance matrix. If measurements are processed sequentially, one should in general not normalize with respect to the previously computed covariance matrix, i.e. the covariance matrix obtained after processing the previous measurement. The reason for this is that it is possible to get an estimability index of 0 after processing each measurement, even if after processing several measurements the covariance in all state space directions has decreased. This is possible because each measurement may decrease covariance in some directions, and not in others. In fact if covariance is not decreased in at least one direction after each measurement, and this direction changes with each measurement, covariance will decrease in all directions after processing several measurements, but the estimability index may still be zero for each covariance update.

### 3.4.2 Linearization of Measurement Error Model

To compute the CRLB the measurement error is linearized about the augmented true state vector $\underline{x}$. The nominal state vector is denoted $\underline{\tilde{x}}$. For $M+1$ sensors and $N$ targets

$$\underline{z} = \underline{h}\left(\underline{x}\right) + \underline{w} \tag{3.68}$$

where $\underline{h}$ is the nonlinear measurement model which we linearize about $\underline{x}$ to

$$\delta\underline{z} = H\delta\underline{x} + \underline{w} \tag{3.69}$$

where

$$\delta\underline{x} = \underline{x} - \underline{\tilde{x}} \tag{3.70a}$$

$$\delta\underline{z} = \underline{z} - \underline{\tilde{z}} \tag{3.70b}$$

$$H = \left.\frac{\partial \underline{h}\left(\underline{x}\right)}{\partial \underline{x}^T}\right|_{\underline{x}}. \tag{3.70c}$$

To simplify notation below we define the target position $\underline{p}_j$ to be equal to $\underline{p}_j^u$ or $\underline{p}_j^{s_0}$ in the universal and absolute/relative case respectively. Analogously the bias vector $\underline{b}_i$ below contains all appropriate alignment, location and sensor biases. The state vector $\underline{x}$ and measurement vector $\underline{z}$ for Sensor $i \in 0, 1, \ldots, M$ and

Target $j \in 1, 2, \ldots, N$ are

$$\underline{x} = \begin{bmatrix} (\underline{p}_1)^T & (\underline{p}_2)^T & \ldots & (\underline{p}_N)^T & (\underline{b}_0)^T & (\underline{b}_1)^T & \ldots & (\underline{b}_M)^T \end{bmatrix}^T \tag{3.71}$$

$$\underline{z} = \begin{bmatrix} (\bar{\underline{b}}_0)^T & (\bar{\underline{b}}_1)^T & \ldots & (\bar{\underline{b}}_M)^T & (\underline{z}_1^{s_0})^T & (\underline{z}_2^{s_0})^T & \ldots & (\underline{z}_N^{s_0})^T & (\underline{z}_1^{s_1})^T \\ (\underline{z}_2^{s_1})^T & \ldots & (\underline{z}_N^{s_1})^T & \ldots & (\underline{z}_1^{s_M})^T & (\underline{z}_2^{s_M})^T & \ldots & (\underline{z}_N^{s_M})^T \end{bmatrix}^T \tag{3.72}$$

where $\tilde{\underline{x}} = \underline{0}$, $\tilde{\underline{z}} = \underline{0}$ and the prior bias mean $\bar{\underline{b}}_i = \underline{0}$. The augmented measurement error covariance matrices for a Cartesian measurement are given by

$$R_a = \begin{bmatrix} B_a & 0 & 0 & \ldots & 0 \\ 0 & R^{s_0} & 0 & \ldots & 0 \\ 0 & 0 & R^{s_1} & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & R^{s_M} \end{bmatrix} \tag{3.73a}$$

$$B_a = \begin{bmatrix} B^{s_0} & 0 & \ldots & 0 \\ 0 & B^{s_1} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & B^{s_M} \end{bmatrix} \tag{3.73b}$$

$$B^{s_i} = \begin{bmatrix} B_{n_i}^{s_i} & 0 & 0 \\ 0 & B_{n_i s_i}^{s_i} & 0 \\ 0 & 0 & B_i \end{bmatrix} \tag{3.73c}$$

$$R^{s_i} = \begin{bmatrix} R_1^{s_i} & 0 & \ldots & 0 \\ 0 & R_2^{s_i} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & R_N^{s_i} \end{bmatrix} \tag{3.73d}$$

$$R_j^{s_i} = C_{i,j}^T R C_{i,j} \tag{3.73e}$$

where the submatrices of $B^{s_i}$ are given by (3.9) and (3.15), $R$ is the measurement noise covariance matrix in spherical coordinates given by (3.7), and $C_{i,j} = C_k^{s_i}$ for Target $j$ given by (3.22). Note that we linearize about the true state vector because we want to calculate the CRLB without errors caused by incorrect linearization. When the truth is unknown the estimated or the predicted state vector can be used. Regarding computation of the estimability index, we see that $B_a$ is diagonal, hence in the following we can use (3.66) to compute $\eta$ with (3.63).

### 3.4.3 The Effect of Geometry and Measurement Noise

Measurement noise and the geometry of the problem decides if there is enough information to estimate the biases. This includes:

1. Number of targets and their positions.
2. Number of sensors and their locations.
3. Target motion models.

White measurement noise has an important impact on estimability, since an estimable measurement bias can become unestimable as the noise increases. This means that the effect of the bias on the measurement is hidden because of the noise. Assuming that the target is positioned at long range, the angular measurement noise variances $\sigma_\theta^2$ and $\sigma_\phi^2$ are the dominant noise sources. This is because their contribution to the measurement noise in Cartesian coordinates increases with range. Thus $\sigma_\rho^2$ can be neglected. At long range $\phi$ is assumed small. Using these assumptions on the covariance conversion formulas in (3.19) we get

$$R_{11} \approx \sigma_\theta^2 \tilde{\rho}^2 \sin^2 \tilde{\theta} \tag{3.74a}$$
$$R_{22} \approx \sigma_\theta^2 \tilde{\rho}^2 \cos^2 \tilde{\theta} \tag{3.74b}$$
$$R_{33} \approx \sigma_\phi^2 \tilde{\rho}^2 \tag{3.74c}$$

where $\tilde{(\cdot)}$ denotes the value used for linearization. Equation (3.74) shows that $\sigma_\theta^2$ is dominant in the $xy$ plane, and that $\sigma_\phi^2$ only affects the $z$ axis. The corresponding cross covariance is given by using (3.20)

$$R_{12} \approx -\sigma_\theta^2 \tilde{\rho}^2 \sin \tilde{\theta} \cos \tilde{\theta} \tag{3.75a}$$
$$R_{23} \approx -\tilde{\phi} \sigma_\phi^2 \tilde{\rho}^2 \sin \tilde{\theta} \tag{3.75b}$$
$$R_{13} \approx -\tilde{\phi} \sigma_\phi^2 \tilde{\rho}^2 \cos \tilde{\theta}. \tag{3.75c}$$

### 3.4.4 Estimability Contours and Sensitivity Analysis

We assume $T_u^{n_i} = T_{n_0}^{n_i} = I$ and that the Euler angles in $T_u^{s_i}$ and $T_{s_0}^{s_i}$ are small. Hence the alignment bias is represented by $T_u^{s_i}$ and $T_{s_0}^{s_i}$ in the universal and absolute/relative cases respectively, and (3.11) applies. To study the estimability given by the CRLB matrix we make a contour plot of the smallest estimability index of the biases sought using a grid in the state space. These plots are geometric plots where the position of Target 1 changes. The measurements are originally in spherical coordinates, and then converted to Cartesian coordinates. Table 3.1 shows the simulation parameters used. The plots are presented with a

value at a given position together with contour resolution in order to derive values elsewhere in the state space. In addition to the geometric plots we present a plot of estimability in function of the angular measurement noise standard deviations $\sigma_\theta$ and $\sigma_\phi$. These plots are used for sensitivity analysis where Target 1 has been placed in a given horizontal position in the geometric plot. The plots are given for the simplest bias estimation scenarios, and then expanded to more complicated scenarios which can be used to compare universal, absolute and relative bias estimation.

***Table 3.1:*** *Altitude of sensors ($z$) and targets ($p_z$), prior bias standard deviations ($\sigma_{b_\alpha}, \sigma_{b_\beta}, \sigma_{b_\gamma}, \sigma_{b_x}, \sigma_{b_y}, \sigma_{b_z}, \sigma_{b_\rho}, \sigma_{b_\theta}, \sigma_{b_\phi}$), and measurement standard deviations ($\sigma_\rho, \sigma_\theta, \sigma_\phi$).*

| $z$ | $p_z$ | $\sigma_{b_\alpha}, \sigma_{b_\beta}, \sigma_{b_\gamma}, \sigma_{b_\theta}, \sigma_{b_\phi}$ | $\sigma_{b_x}, \sigma_{b_y}, \sigma_{b_z}, \sigma_{b_\rho}$ | $\sigma_\rho$ | $\sigma_\theta, \sigma_\phi$ |
|---|---|---|---|---|---|
| 0 m | 1000 m | 5 mrad | 20 m | 20 m | 0.5 mrad |

### 3.4.5 The Alignment Bias

Since in this case the only measurement bias present is the alignment bias, (3.43a) and (3.46a) are used with $\underline{b}^u_{n_i s_i} = \underline{b}^{s_0}_{s_0 n_0} = \underline{b}^{s_0}_{n_i s_i} = \underline{b}_i = \underline{b}_0 = \underline{0}$ which leads to

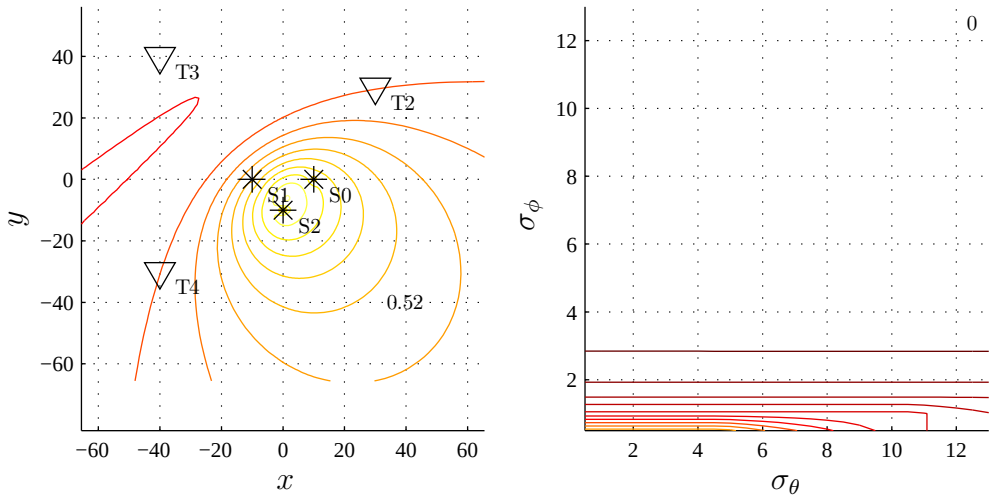
$$\underline{z}^{s_i}_j = T^{s_i}_u \left( \underline{p}^u_j - \underline{d}^u_{un_i} \right) + \underline{w}^{s_i}_j \tag{3.76}$$

for the universal case and

$$\underline{z}^{s_i}_j = T^{s_i}_{s_0} \left( \underline{p}^{s_0}_j - \underline{d}^{s_0}_{n_0 n_i} \right) + \underline{w}^{s_i}_j \tag{3.77}$$

for a biased sensor in the absolute and relative case. The displacement vectors $\underline{d}^u_{un_i}$ and $\underline{d}^{s_0}_{n_0 n_i}$ are known, and $j$ is a target index. The time index $k$ is omitted since all states are assumed static in the following analysis. The measurement vector and noise vector notation is simplified to $\underline{z}^{s_i}_{i,j} = \underline{z}^{s_i}_j$ and $\underline{w}^{s_i}_j = \underline{w}^{s_i}_{i,j}$ respectively. The subscript $i$ indices are removed because the measurement coordinates do not change. Thus the $i$ index of $\{s_i\}$ denotes the measurement origin as well as coordinate system.

Figure 3.5a and Figure 3.6a show the smallest estimability index for simple scenarios where all the alignment biases become estimable. In these scenarios there is no such thing as a least estimable bias. Rather there is a least estimable

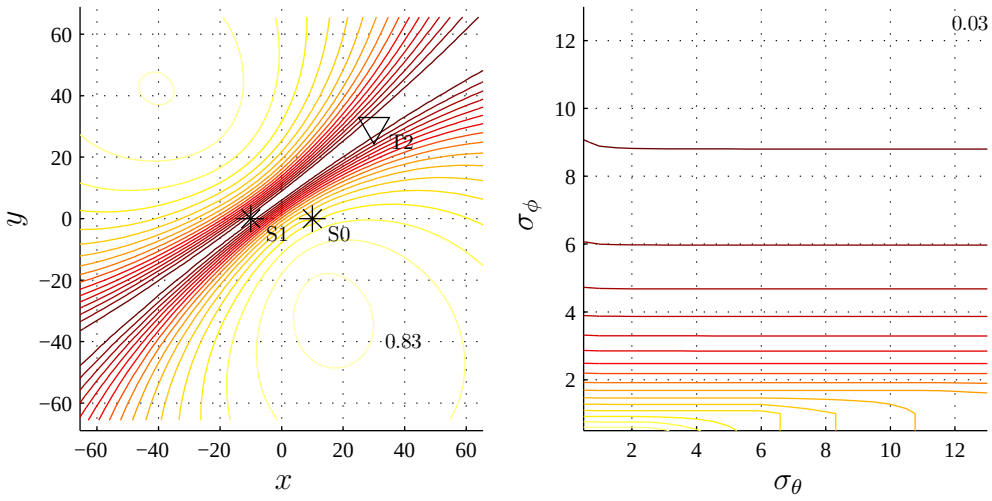



***(a)*** ***(b)***

***Figure 3.5:*** *Universal estimability index of all $\underline{b}_{n_i}^{s_i}$ in function of geometry and measurement noise where the contour resolution is 0.05, and lighter contours indicate a larger estimability index. Figure (a) shows a geometrical estimability plot with axis unit km, and the contour value at $(41, -41)$. Triangles and stars show targets and sensors respectively. Figure (b) shows a measurement sensitivity plot when Target 1 is in $(41, -41)$ in (a), and the contour value at $(12.5, 12.5)$ is displayed. The axis unit is mrad.*

direction in the bias space, since the Euler angles are all correlated. This direction can be found through the eigenvectors of the submatrix of the CRLB containing the biases. Note that the estimability contours for two sensors show that all three Euler angle biases are unestimable in the universal sense, which is because any rotation of the targets about the line connecting the two sensors will yield the same measurements, as explained in Example 3.3. Figure 3.5a and Figure 3.6a show the importance of having well spread targets. Placing target 1 near another target has a low impact on estimability, which is expected as there is little new geometric information in that case. In fact Figure 3.6a shows that the absolute and relative alignment bias is unestimable if the targets and the biased sensor are located along the same line. This is expected according to Example 3.5.

Figure 3.5b and Figure 3.6b show sensitivity analysis plots. Note the substantial difference of universal versus absolute/relative estimability. In Figure




***(a)*** ***(b)***

***Figure 3.6:*** *Absolute and relative estimability index of $\underline{b}_{n_1}^{s_1}$ in function of geometry and measurement noise where the contour resolution is 0.05, and lighter contours indicate a larger estimability index. Figure (a) shows a geometrical estimability plot with axis unit km, and the contour value at $(41, -41)$. Triangles and stars show targets and sensors respectively. Figure (b) shows a measurement sensitivity plot when Target 1 is in $(41, -41)$ in (a), and the contour value at $(12.5, 12.5)$ is displayed. The axis unit is mrad.*

3.5b the contour for $\eta = 0.05$, denoted $\eta_{0.05}$, intersects with $\sigma_\phi \approx 3$ mrad, while $\eta_{0.05}$ intersects with $\sigma_\phi \approx 9$ mrad in Figure 3.6b. The sensitivity to measurement noise is greater in the universal case since more information is extracted from the measurements as there are more biases to estimate. Note that Figure 3.5b and Figure 3.6b show stronger sensitivity with respect to $\sigma_\phi$ compared to $\sigma_\theta$. This can be explained from (3.74) where it is seen that when $\sigma_\theta = \sigma_\phi$, the covariance in the $xz$ and $yz$ planes is larger than in the $xy$ plane. Increasing $\sigma_\phi$ yields a strong increase in covariance for the $xz$ and $yz$ planes. An increase in $\sigma_\theta$ has less effect since $\sigma_\theta^2$ is multiplied by $\sin^2 \hat{\theta}$ or $\cos^2 \hat{\theta}$.

**Example 3.3**

Suppose that we have an arbitrary number of sensors that are located along a line. Without loss of generality, assume that they have the same $x$ and $y$ coordinates in $\{u\}$. The sensors are thus located along the $z$ axis. Assume that Sensor $i$ is only subject to a small alignment bias angle $\gamma_i$ about the $x$ axis. The deterministic

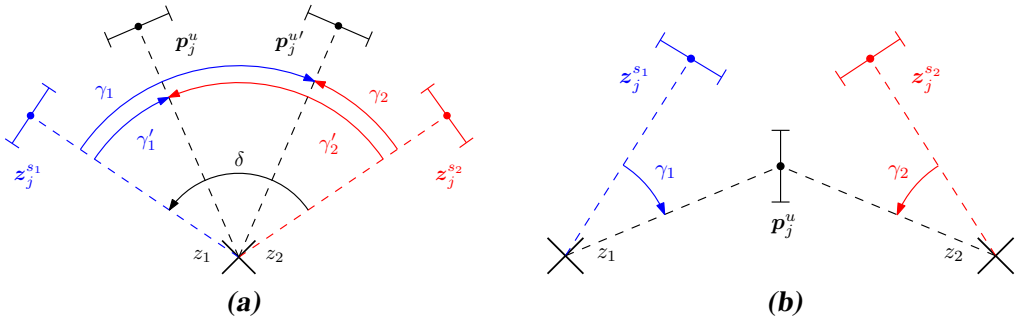



***Figure 3.7:*** *Figure (a) shows a scenario where two sensors are located along the same axis $z_1 = z_2$, denoted by a cross. Each sensor measures the position of Target $j$, and the measurements are represented by their value (blue and red dots) and their covariance (blue and red uncertainty intervals). Requiring the measurements to overlap we obtain multiple solutions. One solution is given by $\gamma_i$, and another by $\gamma_i'$. In Figure (b) the two sensors are located along different axes, $z_1$ and $z_2$, and a unique solution is found. The angles are exaggerated for illustrative purposes.*

target measurement is

$$x_{z_i} = x - \gamma_i y \tag{3.78a}$$
$$y_{z_i} = y + \gamma_i x \tag{3.78b}$$
$$z_{z_i} = z \tag{3.78c}$$

where $x_{z_i}$, $y_{z_i}$, $z_{z_i}$ are the measured target coordinates, and $x$, $y$, $z$ are the true target coordinates. Requiring the true target coordinates to be equal for all sensors $i \in \{1, 2, \ldots, M\}$, and recalling that all sensors have the same $x$ and $y$ coordinates, yields $\gamma_i$ equal for all sensors. Thus $\gamma_i$ can have any value as long as it is small. Figure 3.7a shows a scenario where there are two sensors receiving one measurement of the position of Target $j$ each. The measurements are given by their measured position values and covariances. The alignment bias $\gamma_i$ of Sensor $i$ is estimated by requiring the measurements and their covariance to overlap under the constraint that $\gamma_i$ is small. In Figure 3.7a this problem has multiple solutions since any solution satisfying $\gamma_1 + \gamma_2 = \delta$ can be used. For $M$ sensors we get

$$\gamma_1 + \gamma_2 + \ldots + \gamma_M = \delta_{max} \tag{3.79}$$

where $\delta_{max}$ is the largest possible angle difference $\delta$ about the $x$ axis of any two measurements. Because there are multiple solutions to the problem, all $\gamma_i$ are

unobservable when performing universal bias estimation. This problem only occurs for a rotation angle about the line connecting the sensors. The unobservable angle always exists when the sensors are located along a line, and will in general be a linear combination of the Euler angle biases of each sensor. Thus it is impossible to estimate the alignment bias of individual sensors located along the same line.

**Example 3.4**

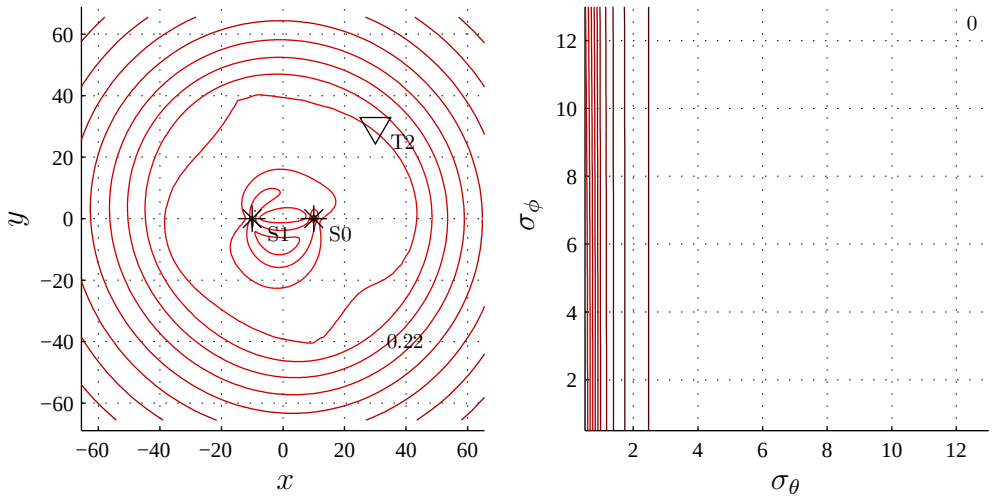
Universal observability can be achieved if at least one sensor is not located along a line connecting the other sensors, as shown in Figure 3.7b for two sensors where Sensor $i$ has an $\gamma_i$ bias. Requiring the measurements to overlap yields a unique solution, with associated covariance, under the assumption that $\gamma_i$ is small. Note that if $\gamma_i$ is arbitrary, the problem has two solutions, and $\gamma_i$ is thus unobservable. Hence three sensors are required to get observability for the alignment biases of all sensors, since two sensors are always located along a line.

**Example 3.5**

Suppose that there are two targets (Target 1 and Target 2), one master sensor (Sensor 0) and one biased sensor (Sensor 1). Assume that Target 1 is located on the line connecting Sensor 1 and Target 2. In that case the position of Target 1 is a linear function of the position of Target 2, thus making Target 1 linearly dependent on Target 2. Thus Target 2 does not provide more information on the alignment bias of Sensor 1, hence the bias is unestimable.

### 3.4.6 The Location Bias

The location bias can only be found by adding a sensor without a location bias, otherwise only relative locations are observable. The reason is that without a known location of a sensor, a displacement applied to all sensors and targets conserving the sensor's latitude will yield the same measurements. This is because the Earth ellipsoid is rotationally symmetric about the polar axis, i.e. a spheroid. If a flat Earth is assumed, a location bias applied to all sensors will result in a case where no points in space are known. Thus the location bias is unobservable and unestimable universally. In the absolute and relative case where




***(a)*** ***(b)***

***Figure 3.8:*** *Absolute and relative estimability index of $\underline{b}_{s_0 s_1}^{s_0}$ in function of geometry and measurement noise where the contour resolution is 0.02, and lighter contours indicate a larger estimability index. Figure (a) shows a geometrical estimability plot with axis unit km, and the contour value at $(41, -41)$. Triangles and stars show targets and sensors respectively. Figure (b) shows a measurement sensitivity plot when Target 1 is in $(41, -41)$ in (a), and the contour value at $(12.5, 12.5)$ is displayed. The axis unit is mrad.*

Sensor 0 is a master sensor, and the location bias is the only measurement bias, the measurement of a biased Sensor $i$ is (3.46a)

$$\underline{z}_j^{s_i} = \underline{p}_j^{s_0} - \underline{d}_{n_0 n_i}^{s_0} - \underline{b}_{s_0 s_i}^{s_0} + \underline{w}_j^{s_i} \tag{3.80}$$

where we have used that $T_{n_i}^{s_i} = T_{n_0}^{s_0} = T_{n_0}^{n_i} = I$ and $\underline{b}_i = \underline{b}_0 = \underline{0}$. Note that (3.80) is linear, and that the effect of the location bias does not change with target position, as opposed to the alignment bias. Hence no new geometrical information is gained when adding targets. Figure 3.8a shows an estimability plot, where it is seen that the bias is estimable everywhere in the plot region for the two target case. Estimability decreases with target range since the measurement noise increases. The measurement noise sensitivity analysis is presented in Figure 3.8b where $\eta_{0.05}$ is situated at $\sigma_\theta \approx 2.5$ mrad. We see that there is strong sensitivity to $\sigma_\theta$, and weak sensitivity to $\sigma_\phi$. This can be explained by examining (3.74) and (3.75). An increase in $\sigma_\theta$ leads to an increase in measurement noise in both $x$ and $y$ directions, affecting the estimability of both $b_x$ and $b_y$. Furthermore the

measurement cross covariance in the $xy$ plane is also increased. This increase is larger than the corresponding effect of increasing $\sigma_\phi$.

### 3.4.7 The Sensor Bias

In this section we analyse the sensor bias $\underline{b}_i$ which is assumed to be the only measurement bias present. Using $T_u^{s_i} = I$ and $\underline{b}_{n_i s_i}^u = \underline{0}$, (3.43a) yields

$$\underline{z}_j^{s_i} = \underline{p}_j^u - \underline{d}_{un_i}^u + C_j^{s_i} \underline{b}_i + \underline{w}_j^{s_i} \tag{3.81}$$

for the universal/absolute case. For the relative case using $T_{n_i}^{s_i} = T_{n_0}^{n_i} = I$ and $\underline{b}_{s_0 n_0}^{s_0} = \underline{b}_{n_i s_i}^{s_0} = \underline{0}$ (3.56a) becomes

$$\underline{z}_j^{s_i} = \underline{p}_j^{s_0} - \underline{d}_{n_0 n_i}^{s_0} + C_j^{s_i} \underline{b}_i + \underline{w}_j^{s_i}. \tag{3.82}$$

Neither (3.81) nor (3.82) are well suited to interpret how the sensor bias evolves in Cartesian coordinates, since the matrix $C_j^{s_i}$ is a function of spherical coordinates (3.22). To get a better understanding of the sensor bias evolution, it is desirable to have a Cartesian sensor measurement which is a function of Cartesian coordinates. This can be achieved by using

$$\begin{aligned} \underline{z}_j^{s_i} &= \underline{h}_c \left( \underline{h}_s \left( \underline{p}_j^u - \underline{d}_{un_i}^u \right) + \underline{b}_i \right) + \underline{w}_j^{s_i} \\ &= \underline{h} \left( \underline{p}_j^u, \underline{b}_i \right) + \underline{w}_j^{s_i}. \end{aligned} \tag{3.83}$$

Denoting

$$\underline{p}_j^{n_i} = \underline{p}_j^u - \underline{d}_{un_i}^u \tag{3.84}$$

$$\underline{p}_j = \underline{h}_s \left( \underline{p}_j^{n_i} \right) + \underline{b}_i, \tag{3.85}$$

using (2.28) and $\underline{p}_j^{n_i} = \begin{bmatrix} x_n & y_n & z_n \end{bmatrix}^T$, $\underline{p}_j = \begin{bmatrix} \rho & \theta & \phi \end{bmatrix}^T$ in spherical coordinates is given by

$$\rho = \sqrt{x_n^2 + y_n^2 + z_n^2} + b_\rho \tag{3.86a}$$

$$\theta = \arctan \frac{y_n}{x_n} + b_\theta \tag{3.86b}$$

$$\phi = \arcsin \frac{z_n}{\sqrt{x_n^2 + y_n^2 + z_n^2}} + b_\phi. \tag{3.86c}$$

Using (2.27) and

$$\underline{p}_j^{s_i} = \underline{h}_c \left( \underline{p}_j \right) \tag{3.87}$$

we calculate $\underline{p}_j^{s_i} = \begin{bmatrix} x_b & y_b & z_b \end{bmatrix}^T$:

$$x_b = \rho \cos\theta \cos\phi = \rho \frac{\sin\theta}{\tan\theta} \frac{\sin\phi}{\tan\phi} \tag{3.88a}$$

$$y_b = \rho \sin\theta \cos\phi = \rho \sin\theta \frac{\sin\phi}{\tan\phi} \tag{3.88b}$$

$$z_b = \rho \sin\phi \tag{3.88c}$$

Using (3.40), the trigonometric equalities

$$\arctan a = \arcsin \frac{a}{\sqrt{a^2+1}} \tag{3.89a}$$

$$\arcsin a = \arctan \frac{a}{\sqrt{1-a^2}} \tag{3.89b}$$

and neglecting products of the sensor bias elements we get the derivatives of $\underline{h} = \begin{bmatrix} h_x & h_y & h_z \end{bmatrix}^T$ with respect to $b_\rho$:

$$\frac{\partial h_x}{\partial b_\rho} = \frac{x_n \sqrt{1 - \frac{z_n^2}{x_n^2+y_n^2+z_n^2}}}{\sqrt{x_n^2+y_n^2}} \tag{3.90a}$$

$$\frac{\partial h_y}{\partial b_\rho} = \frac{y_n \sqrt{1 - \frac{z_n^2}{x_n^2+y_n^2+z_n^2}}}{\sqrt{x_n^2+y_n^2}} \tag{3.90b}$$

$$\frac{\partial h_z}{\partial b_\rho} = \frac{z_n}{\sqrt{x_n^2+y_n^2+z_n^2}} \tag{3.90c}$$

It is seen that there is much information about $b_\rho$ in the Cartesian coordinates $x$ and $y$, and less information in $z$. This is expected since almost all information about $b_\rho$ is in the $xy$ plane for small elevation angles. There is no information in $x$ and $y$ when the target and sensor share the same $x$ and $y$ coordinates. Since information about the range bias $b_\rho$ is strong in $x$ and $y$, it is universally estimable. Given one target and two sensors with range biases, and neglecting the information in $z$, there are four measurements and four unknowns. The measurement equations are linearly independent since the effect of the range bias depends on the sensor locations, and these are assumed different. Hence universal estimability is achieved.

The derivatives of $\underline{h}$ with respect to $b_\theta$ are

$$\frac{\partial h_x}{\partial b_\theta} = -y_n \tag{3.91a}$$

$$\frac{\partial h_y}{\partial b_\theta} = x_n \tag{3.91b}$$

$$\frac{\partial h_z}{\partial b_\theta} = 0 \tag{3.91c}$$

and the derivatives of $\underline{h}$ with respect to $b_\phi$ are

$$\frac{\partial h_x}{\partial b_\phi} = -\frac{x_n z_n}{\sqrt{x_n^2 + y_n^2}} \tag{3.92a}$$

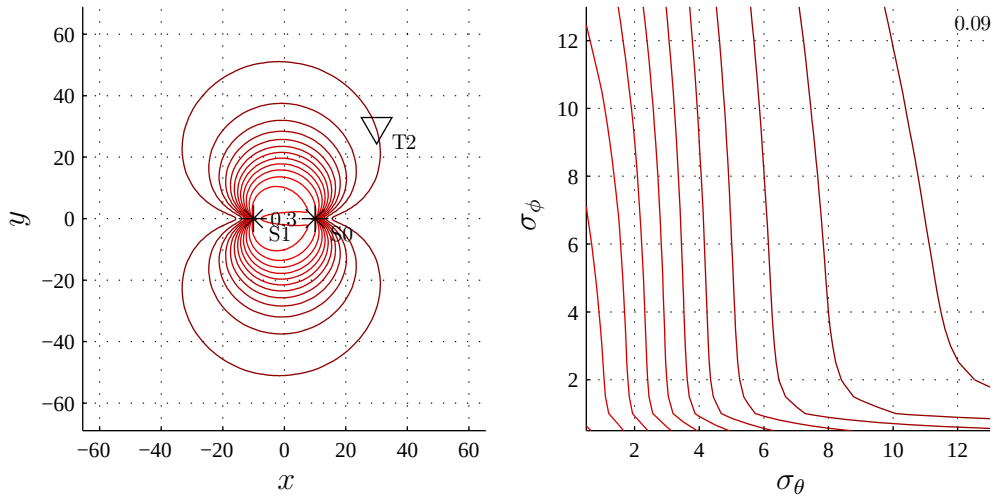
$$\frac{\partial h_y}{\partial b_\phi} = -\frac{y_n z_n}{\sqrt{x_n^2 + y_n^2}} \tag{3.92b}$$

$$\frac{\partial h_z}{\partial b_\phi} = \sqrt{1 - \frac{z_n^2}{x_n^2 + y_n^2 + z_n^2}}. \tag{3.92c}$$

There is much information about the azimuth bias $b_\theta$ in the $xy$ plane, except for the case where the target is very close to the sensor, and no information in the $z$ direction. Information about the elevation bias $b_\phi$ is primarily found in the $z$ direction, but as long as $z_n$ is not close to zero, some information is found in the $xy$ plane. If $z_n$ is fixed and target range increases, information in the $xy$ plane is reduced.

Figure 3.9 and Figure 3.10 show estimability plots for the range bias in the universal/absolute and relative case respectively. Note the large differences in estimability for Figure 3.9a and Figure 3.10a. In Figure 3.9a strong estimability only occurs at close range, while Figure 3.10a shows strong estimability everywhere in the plotted region. The measurement noise sensitivity can be found in Figure 3.9b and Figure 3.10b. Both plots show strong sensitivity to $\sigma_\theta$, and almost no sensitivity to $\sigma_\phi$.

The weak estimability in Figure 3.9a is caused by two factors. At $y = 0$ (3.90b) becomes zero for both sensors. Given small elevation angles (3.90c) is approximately zero, hence information about $b_\rho$ is only available from the $x$ position of the target. Thus there are two $x$ measurements that can be used to determine the two biases $b_\rho$, yielding 2 equations and 3 unknowns (target $x$ position and two biases). This problem occurs when $y = 0$ for both sensors and



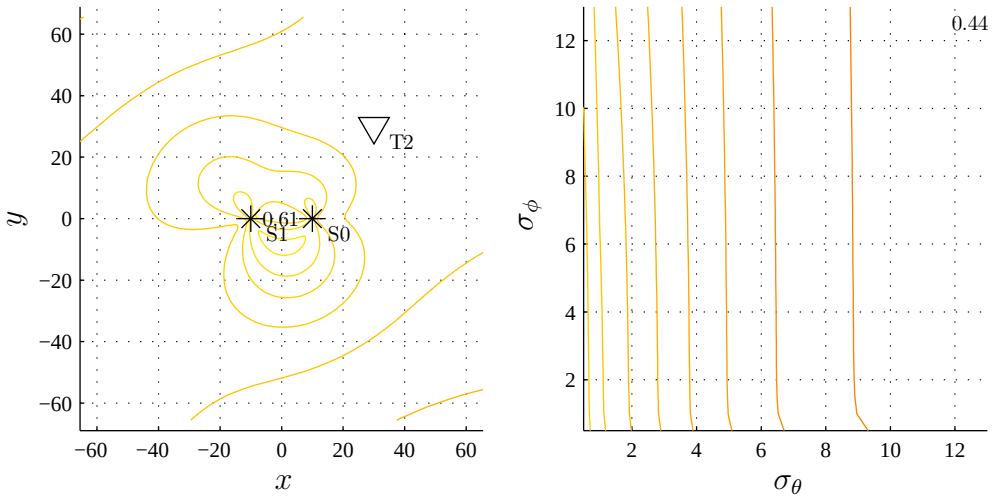

***(a)*** ***(b)***

***Figure 3.9:*** *Universal and absolute estimability index of $\underline{b}_{\rho_1}$ (range only) in function of geometry and measurement noise where the contour resolution is 0.02, and lighter contours indicate a larger estimability index. Figure (a) shows a geometrical estimability plot with axis unit km, and the contour value at $(0, 0)$. Triangles and stars show targets and sensors respectively. Figure (b) shows a measurement sensitivity plot when Target 1 is in $(0, 0)$ in (a), and the contour value at $(12.5, 12.5)$ is displayed. The axis unit is mrad.*

$z \ll \sqrt{x^2 + y^2 + z^2}$ for at least one of the sensors. The latter is not true between the sensors, which explains the stronger estimability in that region. Note that $y = 0$ corresponds to the general case where the target is on the line connecting the two sensors.

The other factor that creates weak estimability in Figure 3.9a is the increasing noise in Cartesian coordinates where the noise depends on target range. This is due to the angular measurement noise in azimuth and elevation. At long range the noise leads to weaker estimability for universal and absolute range biases.

Figure 3.11 and Figure 3.12 show estimability plots for the azimuth and elevation biases in the universal/absolute and relative case respectively. Figure 3.11a shows that in the universal and absolute case, well spread targets is important to achieve strong estimability. Figure 3.11b shows a strong sensitivity to




***(a)*** ***(b)***

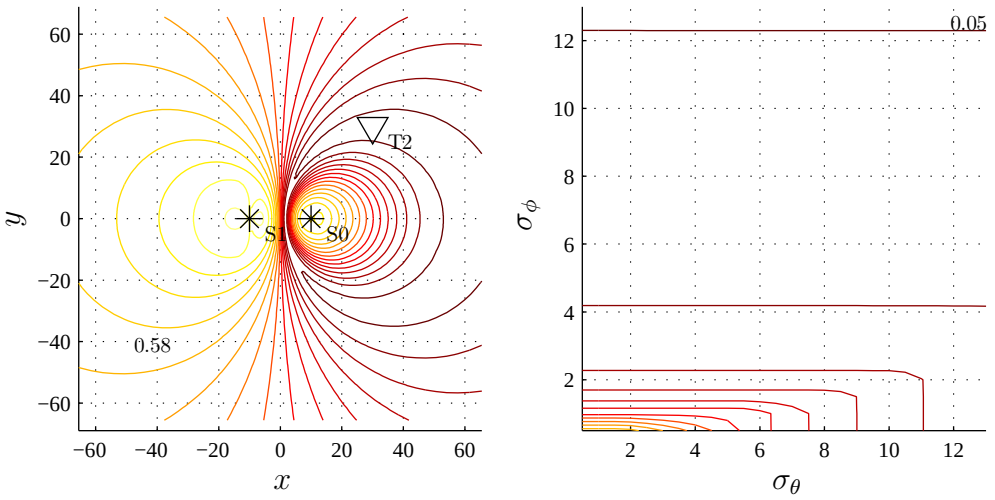
***Figure 3.10:*** *Relative estimability index of $\underline{b}_{\rho_1}$ (range only) in function of geometry and measurement noise where the contour resolution is 0.02, and lighter contours indicate a larger estimability index. Figure (a) shows a geometrical estimability plot with axis unit km, and the contour value at $(0, 0)$. Triangles and stars show targets and sensors respectively. Figure (b) shows a measurement sensitivity plot when Target 1 is in $(0, 0)$ in (a), and the contour value at $(12.5, 12.5)$ is displayed. The axis unit is mrad.*

measurement noise in elevation. Figure 3.12a shows strong estimability everywhere in the relative case, except when the target is close to the biased sensor. Figure 3.12b shows strong sensitivity to measurement noise in azimuth.

In the universal and absolute case, $b_\phi$ is less estimable than $b_\theta$ since $z \ll \sqrt{x^2 + y^2}$ everywhere except close to the sensors. This leads to larger measurement noise in the $z$ direction. Therefore, an increase in $\sigma_\phi$ has a strong effect on estimability since it affects estimability of $b_\phi$ directly. In the relative case the difference in estimability is smaller because the master sensor biases are not estimated.

### 3.4.8 Universal Alignment and Sensor Bias

The following combinations of universal alignment and sensor bias are unestimable:




***(a)*** ***(b)***

***Figure 3.11:*** *Universal and absolute estimability index of $\underline{b}_1$ (azimuth and elevation) in function of geometry and measurement noise where the contour resolution is 0.02, and lighter contours indicate a larger estimability index. Figure (a) shows a geometrical estimability plot with axis unit km, and the contour value at $(-41, -41)$. Triangles and stars show targets and sensors respectively. Figure (b) shows a measurement sensitivity plot when Target 1 is in $(-41, -41)$ in (a), and the contour value at $(12.5, 12.5)$ is displayed. The axis unit is mrad.*

- Alignment and sensor bias.
- Alignment and range bias.
- Alignment and elevation bias.

Recall that the alignment yaw bias and sensor azimuth bias are the same angle. When the estimator is allowed to adjust sensor alignment and target range or elevation for all sensors, the bias estimation problem has several solutions. This means that several values of each individual bias yield the same target measurements. Since the alignment and sensor bias are estimable individually, a possible solution is to use some targets to estimate the alignment bias, ignoring the sensor bias, and vice versa.

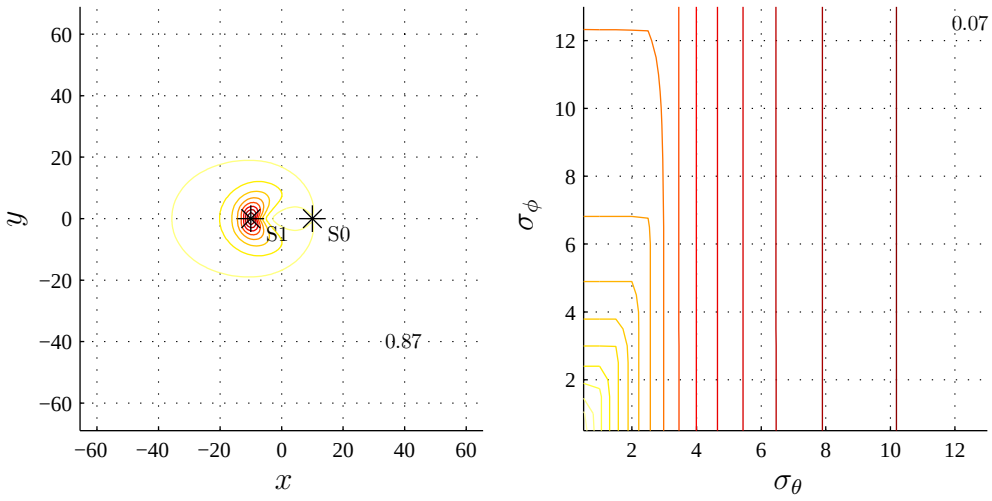



***Figure 3.12:*** *Relative estimability index of $\underline{b}_1$ (azimuth and elevation) in function of geometry and measurement noise where the contour resolution is 0.02, and lighter contours indicate a larger estimability index. Figure (a) shows a geometrical estimability plot with axis unit km, and the contour value at $(-41, -41)$. Triangles and stars show targets and sensors respectively. Figure (b) shows a measurement sensitivity plot when Target 1 is in $(-41, -41)$ in (a), and the contour value at $(12.5, 12.5)$ is displayed. The axis unit is mrad.*

### 3.4.9 Absolute and Relative Alignment, Location and Sensor Bias

The alignment, location and sensor biases are all estimable in the absolute and relative cases. The biggest difference lies in estimation of the sensor range bias which in the absolute case requires targets at shorter range to become estimable.

# 4

# Bias Estimation and Data Fusion for Distributed Radars in 3D

An important property of a multisensor data fusion scheme in a target tracking scenario is to be able to improve tracking by adding more sensors. This property is challenging to implement because of unknown sensor measurement biases. Not accounting for these may lead to more false tracks and poor performance. It is desirable to find all of these biases from measurements from common targets. We consider a network of 3D sensors observing targets of opportunity which return measurements in spherical coordinates. The measurements are subject to biases due to uncertain sensor alignment and location, and biases from sensor imperfections.

Section 4.1 presents simulation results where the estimability conclusions from Section 3.4 are used to create interesting simulation scenarios for targets moving at constant velocity. In Section 4.2 the targets are moving at constant altitude, and the curvature of the Earth is taken into account.

## 4.1 Constant Velocity Flight with Monte Carlo Simulations

The estimability results in the previous section are valid for static targets. In this section the targets are moving at constant velocity, however the estimability plots for static targets can be used to predict estimability for moving targets. A target moving at constant velocity yields information about a line, whereas a

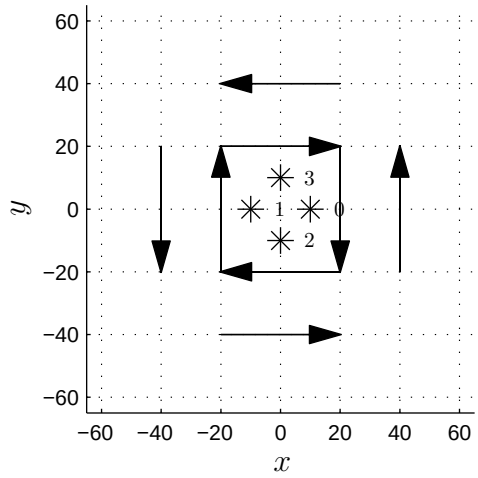



***Figure 4.1:*** *Simulation scenario for bias estimation. The stars represent sensors, and the arrows represent targets which start at the tail of the arrow, and end at the arrow tip. The axis unit is km.*

static target yields a point. A line includes more geometric information than a point, thus bias estimability should be improved. In the following we investigate bias estimator performance in Monte Carlo (MC) simulations. Since there is an infinity of simulation scenarios possible, we will use the estimability plots from Section 3.4 to choose interesting ones.

***Table 4.1:*** *Target altitude, prior bias uncertainty, measurement noise and sampling interval.*

| $p_z$ | $\sigma_{b_{\alpha_0}}, \sigma_{b_{\beta_0}}, \sigma_{b_{\gamma_0}}, \sigma_{b_{\theta_0}}, \sigma_{b_{\phi_0}}$ | $\sigma_{b_{x_0}}, \sigma_{b_{y_0}}, \sigma_{b_{z_0}}, \sigma_{b_{\rho_0}}$ | $\sigma_\rho$ | $\sigma_\theta, \sigma_\phi$ | $\Delta t$ |
|---|---|---|---|---|---|
| 1 km | 5 mrad | 20 m | 20 m | 5 mrad | 4 s |

## 4.1.1 Simulation Scenario and Parameters

We will now run 1000 MC runs to compare estimator performance. Two simulation scenarios are considered where the targets move as in Figure 4.1, and the simulation parameters are given in Table 4.1. The movement in Figure 4.1 is chosen to ensure universal estimability of the alignment and sensor bias, which was studied in Figure 3.5 and Figure 3.9. This is achieved by using the long range targets to estimate the alignment bias, and the close range targets to estimate the sensor bias. Furthermore this geometry ensures well spread targets.

Eight targets are required to estimate all of the unknowns in all simulation cases. The unknowns are position and speed for all targets, and the biases to be estimated for all sensors.

The first scenario evaluates estimator performance when alignment, location and sensor biases are present for all sensors, while the second scenario removes elevation from the sensor bias. The third scenario includes alignment and sensor bias only for all sensors, and the fourth scenario simulates location and sensor bias. The measurement biases for each sensor are drawn individually in each MC run from a normal distribution with mean zero and standard deviations as stated in Table 4.1. The initial bias values for the estimators are zero with standard deviations as in Table 4.1. The target tracks are initialized using two point differencing (Bar-Shalom et al., 2001).

Bias estimation performance in universal coordinates is compared to the performance in master sensor coordinates. It is important that the estimators are compared in equal scenarios in order to present a fair comparison. All sensors have biases, thus the sensor chosen as master sensor is also biased. This gives an advantage to the estimators performing universal and absolute bias estimation, the Universal Bias Estimator (UBE) and the Absolute Bias Estimator (ABE) respectively. The ABE and Relative Bias Estimator (RBE) has the advantage of having fewer biases to estimate, and they are able to estimate location biases with respect to the master sensor. The final estimators to be evaluated are the Standard Universal Bias Estimator (SUBE) and the Standard Absolute Bias Estimator (SABE), which are classic bias estimators from the tracking literature. The SUBE and SABE perform bias estimation in universal coordinates and sensor coordinates respectively. They estimate the sensor bias for all sensors, which consists of range, azimuth and elevation. The alignment bias is not estimated, and the location bias is estimated relative to a master sensor in the SABE. Some of the alignment bias is accounted for by estimating the biases in azimuth and elevation.

### 4.1.2 Process Model and Estimator

The targets are assumed to move at constant velocity for the system and filter models, thus the linear target process model in continuous time is

$$\dot{\underline{p}}_j^u = \underline{v}_j^u \tag{4.1a}$$

$$\dot{\underline{v}}_j^u = \underline{n}_j^u \sim \mathcal{N}\left(\underline{0}, \tilde{Q}\delta\left(t-\tau\right)\right) \tag{4.1b}$$

where $\underline{v}_j^u$ is the target velocity and $\underline{n}_j^u$ is the white process noise. This model is discretized, (Gelb, 1974). An Extended Kalman Filter (EKF) is used to estimate the target positions and velocities, and the measurement biases of the sensors. The measurement biases are modeled as in (3.2). An EKF is used to estimate the biases, where all sensor measurements for one time step is included in an augmented measurement vector. The state vector $\underline{x}_k$ and measurement vector $\underline{z}_k$ for Sensor $0, 1, \ldots, M$ and Target $1, 2, \ldots, N$ are

$$\underline{x}\,(t) = \left[(\underline{x}^u)^T \quad (\underline{x}^b)^T\right]^T \tag{4.2}$$

$$\underline{z}_k = \left[(\underline{z}_1^{s_1})^T \quad \ldots \quad (\underline{z}_N^{s_1})^T \quad \ldots \quad (\underline{z}_1^{s_M})^T \quad \ldots \quad (\underline{z}_N^{s_M})^T\right]^T \tag{4.3}$$

where

$$\underline{x}^u = \left[(\underline{p}_1^u)^T \quad (\underline{v}_1^u)^T \quad \ldots \quad (\underline{p}_N^u)^T \quad (\underline{v}_N^u)^T\right]^T \tag{4.4}$$

$$\underline{x}^b = \left[(\underline{b}_{n_0}^{s_0})^T \quad (\underline{b}_{n_0 s_0}^u)^T \quad (b_0)^T \quad \ldots \quad (\underline{b}_{n_M}^{s_M})^T \quad (\underline{b}_{n_M s_M}^u)^T \quad (b_M)^T\right]^T \tag{4.5}$$

for universal bias estimation. When performing relative bias estimation $\underline{x}^b$ contains biases from Sensor $1, 2, \ldots, M$, and $\underline{b}_{n_i s_i}^u$ is replaced by $\underline{b}_{s_0 s_i}^{s_0}$. For absolute bias estimation $\underline{x}^b$ is unchanged from the relative case, with the exception that the sensor bias of Sensor 0 $\underline{b}_0$ is added as a vector element.

## 4.1.3 Measurement Models

To simplify notation, the target index $j$ is omitted from the measurement models.

### 4.1.3.1 Measurement Noise Covariance

The measurement models presented in this section use the following matrices to determine the white measurement noise covariance. The covariance matrix of a spherical target measurement is given by (3.7)

$$R = \begin{bmatrix} \sigma_\rho^2 & 0 & 0 \\ 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & \sigma_\phi^2 \end{bmatrix}, \tag{4.6}$$

while the Sensor Bias Transformation Matrix (SBTM) is given by (3.22)

$$C_k^{s_i} = \begin{bmatrix} \cos\tilde{\theta}_k \cos\tilde{\phi}_k & -\tilde{\rho}_k \sin\tilde{\theta}_k \cos\tilde{\phi}_k & -\tilde{\rho}_k \cos\tilde{\theta}_k \sin\tilde{\phi}_k \\ \sin\tilde{\theta}_k \cos\tilde{\phi}_k & \tilde{\rho}_k \cos\tilde{\theta}_k \cos\tilde{\phi}_k & -\tilde{\rho}_k \sin\tilde{\theta}_k \sin\tilde{\phi}_k \\ \sin\tilde{\phi}_k & 0 & \tilde{\rho}_k \cos\tilde{\phi}_k \end{bmatrix} \tag{4.7}$$

where the measured values are used for $\tilde{\rho}_k$, $\tilde{\theta}_k$ and $\tilde{\phi}_k$.

#### 4.1.3.2 The System Sensor Measurement Model

The measurement system model is given by (3.42)

$$\underline{z}_{i,k} = \underline{h}_s \left( T_{n_i}^{s_i} T_u^{n_i} \left( \underline{p}_k^u - \underline{d}_{un_i}^u - \underline{b}_{n_i s_i}^u \right) \right) + \underline{b}_i + \underline{w}_{i,k} \tag{4.8a}$$

$$\underline{w}_{i,k} \sim \mathcal{N}\left(\underline{0}, R\delta_{kl}\right) \tag{4.8b}$$

where $i \in \{0, 1, \ldots, M\}$. The spherical measurement $\underline{z}_{i,k}$ is produced by first calculating its true value

$$\underline{\varrho}_{i,k} = \underline{h}_s \left( T_{n_i}^{s_i} T_u^{n_i} \left( \underline{p}_k^u - \underline{d}_{un_i}^u - \underline{b}_{n_i s_i}^u \right) \right) + \underline{b}_i \tag{4.9}$$

and draw the corresponding measured value using

$$\underline{z}_{i,k} \sim \mathcal{N}\left( \underline{\varrho}_{i,k}, R\delta_{kl} \right). \tag{4.10}$$

The Cartesian measurement $\underline{z}_{i,k}^{s_i}$ is given by

$$\underline{z}_{i,k}^{s_i} = \underline{h}_c \left( \underline{z}_{i,k} \right) \tag{4.11}$$

where $\underline{h}_c(\cdot)$ is the Coordinate Transformation Function (CTF) from spherical to Cartesian coordinates given by (2.27). The bias values are drawn once in each MC run at time $k = 0$ according to

$$\underline{b}_{n_i}^{s_i} \sim \mathcal{N}\left(\underline{0}, B_{n_i}^{s_i}\right) \tag{4.12a}$$

$$\underline{b}_{n_i s_i}^u \sim \mathcal{N}\left(\underline{0}, B_{n_i s_i}^u\right) \tag{4.12b}$$

$$\underline{b}_i \sim \mathcal{N}\left(\underline{0}, B_i\right) \tag{4.12c}$$

where

$$B_{n_i}^{s_i} = \begin{bmatrix} \sigma_{b_{\alpha_0}}^2 & 0 & 0 \\ 0 & \sigma_{b_{\beta_0}}^2 & 0 \\ 0 & 0 & \sigma_{b_{\gamma_0}}^2 \end{bmatrix} \tag{4.13a}$$

$$B_{n_i s_i}^u = \begin{bmatrix} \sigma_{b_{x_0}}^2 & 0 & 0 \\ 0 & \sigma_{b_{y_0}}^2 & 0 \\ 0 & 0 & \sigma_{b_{z_0}}^2 \end{bmatrix} \tag{4.13b}$$

$$B_i = \begin{bmatrix} \sigma_{b_{\rho_0}}^2 & 0 \\ 0 & \sigma_{b_\phi}^2 \end{bmatrix}, \tag{4.13c}$$

and the associated values can be found in Table 4.1.

#### 4.1.3.3 The Unbiased Kalman Filter

The measurement filter model for the unbiased Kalman Filter (KFU) is given by (3.24)

$$\underline{z}_{i,k}^{s_i} = T_u^{s_i}\left(\underline{p}_k^u - \underline{d}_{us_i}^u\right) + \underline{w}_{i,k}^{s_i} \tag{4.14a}$$

$$\underline{w}_{i,k}^{s_i} \sim \mathcal{N}\left(\underline{0}, R_{i,k}^{s_i}\delta_{kl}\right) \tag{4.14b}$$

$$R_{i,k}^{s_i} = C_k^{s_i} R \left(C_k^{s_i}\right)^T \tag{4.14c}$$

where $i \in \{0, 1, \ldots, M\}$. The measurement biases are assumed known values for this filter, hereby its name, so only the target states are estimated. This filter serves as a reference with respect to the other filters since a filter which succeeds in compensating for all bias effects will have performance close to the KFU. Note that this filter will always have superior performance compared to the other filters.

#### 4.1.3.4 The Biased Kalman Filter

The measurement filter model for the bias ignorant Kalman Filter which tracks in $\{u\}$ (KFBU) is given by (3.25)

$$\underline{z}_{i,k}^{s_i} = T_u^{n_i}\left(\underline{p}_k^u - \underline{d}_{un_i}^u\right) + \underline{a}_{i,k}^{s_i} + \underline{w}_{i,k}^{s_i} \tag{4.15a}$$

$$\underline{w}_{i,k}^{s_i} \sim \mathcal{N}\left(\underline{0}, R_{i,k}^{s_i}\delta_{kl}\right) \tag{4.15b}$$

$$R_{i,k}^{s_i} = C_k^{s_i} R \left(C_k^{s_i}\right)^T \tag{4.15c}$$

$$\underline{a}_{i,k}^{s_i} \sim \mathcal{N}\left(\underline{0}, A_{i,k}^{s_i}\delta_{kl}\right) \tag{4.15d}$$

where $i \in \{0, 1, \ldots, M\}$ and the added noise covariance matrix $A_{i,k}^{s_i}$ is given by (3.26) and derived in Section 3.2.6. The measurement filter model for the bias ignorant Kalman Filter which tracks in $\{s_0\}$ (KFBS) is given by

$$\underline{z}_{i,k}^{s_i} = T_{n_0}^{n_i}\left(\underline{p}_k^{s_0} - \underline{d}_{n_0 n_i}^{n_0}\right) + \underline{a}_{i,k}^{s_i} + \underline{w}_{i,k}^{s_i} \tag{4.16a}$$

$$\underline{w}_{i,k}^{s_i} \sim \mathcal{N}\left(\underline{0}, R_{i,k}^{s_i}\delta_{kl}\right) \tag{4.16b}$$

$$R_{i,k}^{s_i} = C_k^{s_i} R \left(C_k^{s_i}\right)^T \tag{4.16c}$$

$$\underline{a}_{i,k}^{s_i} \sim \mathcal{N}\left(\underline{0}, A_{i,k}^{s_i}\delta_{kl}\right) \tag{4.16d}$$

for $i \in \{1, 2, \ldots, M\}$ where $A_{i,k}^{s_i}$ is given by (3.26). The master sensor is modeled by (3.62)

$$\underline{z}_{0,k}^{s_0'} = \underline{p}_k^{s_0'} + \underline{w}_{0,k}^{s_0'} \tag{4.17a}$$

$$\underline{w}_{0,k}^{s_0'} \sim \mathcal{N}\left(\underline{0}, R_{0,k}^{s_0'}\delta_{kl}\right) \tag{4.17b}$$

$$R_{0,k}^{s_0'} = C_k^{s_0'} R \left(C_k^{s_0'}\right)^T. \tag{4.17c}$$

This filter accounts for measurement bias uncertainty by increasing the white noise in the measurement model, but the biases are not estimated. Only the target states are estimated. This filter serves as a worst case reference. If a filter performs worse than this filter, it means that it is better to increase measurement noise instead of estimating the measurement biases.

#### 4.1.3.5 The Universal Bias Estimator

The measurement filter model for the UBE is given by (3.42) with $\underline{b}_{n_i s_i}^u = \underline{0}$

$$\underline{z}_{i,k} = \underline{h}_s\left(T_{n_i}^{s_i} T_u^{n_i}\left(\underline{p}_k^u - \underline{d}_{un_i}^u\right)\right) + \underline{b}_i + \underline{w}_{i,k} \tag{4.18a}$$

$$\underline{w}_{i,k} \sim \mathcal{N}\left(\underline{0}, R\delta_{kl}\right) \tag{4.18b}$$

where $i \in \{0, 1, \ldots, M\}$. This filter estimates the alignment bias $\underline{b}_{n_i}^{s_i}$ and sensor bias $\underline{b}_i$ for all sensors, and the target states. The location bias $\underline{b}_{n_i s_i}^u$ is neglected.

#### 4.1.3.6 The Absolute Bias Estimator

The measurement filter model for the ABE is given by (3.56)

$$\underline{z}_{i,k}^{s_i} = T_{s_0}^{s_i}\left(\underline{p}_k^{s_0} - \underline{d}_{n_0 n_i}^{s_0} - \underline{b}_{s_0 s_i}^{s_0}\right) + C_k^{s_i}\underline{b}_i + \underline{w}_{i,k}^{s_i} \tag{4.19a}$$

$$\underline{w}_{i,k}^{s_i} \sim \mathcal{N}\left(\underline{0}, R_{i,k}^{s_i}\delta_{kl}\right) \tag{4.19b}$$

$$R_{i,k}^{s_i} = C_k^{s_i} R \left(C_k^{s_i}\right)^T \tag{4.19c}$$

where $i \in \{1, 2, \ldots, M\}$, and (3.57)

$$\underline{z}_{0,k}^{s_0} = \underline{p}_k^{s_0} + C_k^{s_0}\underline{b}_0 + \underline{w}_{0,k}^{s_0} \tag{4.20a}$$

$$\underline{w}_{0,k}^{s_0} \sim \mathcal{N}\left(\underline{0}, R_{0,k}^{s_0}\delta_{kl}\right) \tag{4.20b}$$

$$R_{0,k}^{s_0} = C_k^{s_0} R \left(C_k^{s_0}\right)^T. \tag{4.20c}$$

This filter estimates the alignment bias $\underline{b}_{s_0}^{s_i}$ and location bias $\underline{b}_{s_0 s_i}^{s_0}$ for Sensor $i \in \{1, 2, \ldots, M\}$, the sensor bias $\underline{b}_i$ for Sensor $i \in \{0, 1, \ldots, M\}$, and the target states.

#### 4.1.3.7 The Relative Bias Estimator

The measurement filter model for the RBE is given by (3.61)

$$\underline{z}_{i,k}^{s_i} = T_{s_0}^{s_i}\left(\underline{p}_k^{s_0'} - \underline{d}_{n_0 n_i}^{s_0} - \underline{b}_{s_0 s_i}^{s_0}\right) + C_k^{s_i}\underline{b}_i + \underline{a}_{i,k}^{s_i} + \underline{w}_{i,k}^{s_i} \tag{4.21a}$$

$$\underline{w}_{i,k}^{s_i} \sim \mathcal{N}\left(\underline{0}, R_{i,k}^{s_i}\delta_{kl}\right) \tag{4.21b}$$

$$R_{i,k}^{s_i} = C_k^{s_i} R \left(C_k^{s_i}\right)^T \tag{4.21c}$$

$$\underline{a}_{i,k}^{s_i} \sim \mathcal{N}\left(\underline{0}, A_{i,k}^{s_i}\delta_{kl}\right) \tag{4.21d}$$

$$A_{i,k}^{s_i} = T_{s_0}^{s_i} C_k^{s_0} B_0 \left(T_{s_0}^{s_i} C_k^{s_0}\right)^T \tag{4.21e}$$

where the sensor bias covariance matrix $B_0$ is defined by (3.9), and $i \in \{1, 2, \ldots, M\}$. The master sensor is modeled by (3.62)

$$\underline{z}_{0,k}^{s_0'} = \underline{p}_k^{s_0'} + \underline{w}_{0,k}^{s_0'} \tag{4.22a}$$

$$\underline{w}_{0,k}^{s_0'} \sim \mathcal{N}\left(\underline{0}, R_{0,k}^{s_0'}\delta_{kl}\right) \tag{4.22b}$$

$$R_{0,k}^{s_0'} = C_k^{s_0'} R \left(C_k^{s_0'}\right)^T. \tag{4.22c}$$

This filter estimates the alignment bias $\underline{b}_{s_0}^{s_i}$, location bias $\underline{b}_{s_0 s_i}^{s_0}$ and the sensor bias $\underline{b}_i$ for Sensor $i \in \{1, 2, \ldots, M\}$, and the target states.

#### 4.1.3.8 The Standard Universal Bias Estimator

The measurement filter model for the SUBE is given by (3.42) with $T_{n_i}^{s_i} = I$ and $\underline{b}_{n_i s_i}^{u} = \underline{0}$

$$\underline{z}_{i,k} = \underline{h}_s\left(T_u^{n_i}\left(\underline{p}_k^u - \underline{d}_{u n_i}^u\right)\right) + \underline{b}_i + \underline{w}_{i,k} \tag{4.23a}$$

$$\underline{w}_{i,k} \sim \mathcal{N}\left(\underline{0}, R\delta_{kl}\right) \tag{4.23b}$$

where $i \in \{0, 1, \ldots, M\}$. This filter estimates the sensor bias $\underline{b}_i$ only, which includes biases in range $b_\rho$ and azimuth $b_\theta$. The sensor elevation bias $b_\phi$, the alignment biases roll $b_\alpha$ and pitch $b_\beta$, and the location bias $\underline{b}_{n_i s_i}^u$ are neglected. The alignment bias $b_\gamma$ is estimated through the estimation of $b_\theta$ since these are different representations of the same angle. Note that although $b_\phi$ can be estimated by the SUBE, it is neglected because of poor estimator performance when estimating $b_\phi$ together with $b_\rho$ and $b_\theta$. This is due to disturbances from the unestimated alignment biases $b_\alpha$ and $b_\beta$ and the location bias.

#### 4.1.3.9 The Standard Absolute Bias Estimator

The measurement filter model for the SABE is given by (3.56) with $T_{n_i}^{s_i} = I$

$$\underline{z}_{i,k}^{s_i} = T_{n_0}^{n_i}\left(\underline{p}_k^{s_0} - \underline{d}_{n_0 n_i}^{s_0} - \underline{b}_{s_0 s_i}^{s_0}\right) + C_k^{s_i}\underline{b}_i + \underline{w}_{i,k}^{s_i} \tag{4.24a}$$

$$\underline{w}_{i,k}^{s_i} \sim \mathcal{N}\left(\underline{0}, R_{i,k}^{s_i}\delta_{kl}\right) \tag{4.24b}$$

$$R_{i,k}^{s_i} = C_k^{s_i} R \left(C_k^{s_i}\right)^T \tag{4.24c}$$

where $i \in \{1, 2, \ldots, M\}$, and (3.57)

$$\underline{z}_{0,k}^{s_0} = \underline{p}_k^{s_0} + C_k^{s_0}\underline{b}_0 + \underline{w}_{0,k}^{s_0} \tag{4.25a}$$

$$\underline{w}_{0,k}^{s_0} \sim \mathcal{N}\left(\underline{0}, R_{0,k}^{s_0}\delta_{kl}\right) \tag{4.25b}$$

$$R_{0,k}^{s_0} = C_k^{s_0} R \left(C_k^{s_0}\right)^T. \tag{4.25c}$$

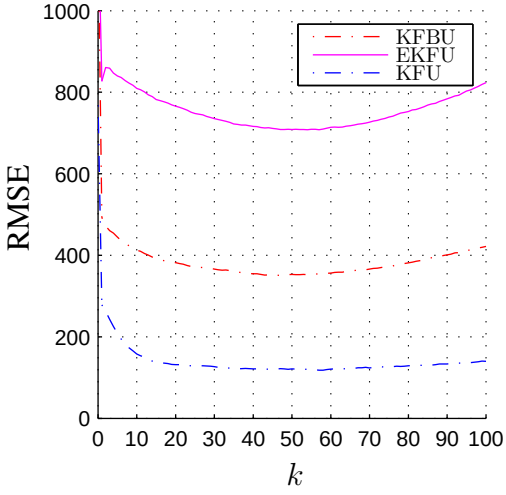
This filter estimates the sensor bias $\underline{b}_i$, including range $b_\rho$, azimuth $b_\theta$ and elevation $b_\phi$, for Sensor $i \in \{0, 1, \ldots, M\}$, and the location bias $\underline{b}_{s_0 s_i}^{s_0}$ for Sensor $i \in \{1, 2, \ldots, M\}$. The alignment biases roll $b_\alpha$ and pitch $b_\beta$ are neglected. The alignment bias $b_\gamma$ is estimated through the estimation of $b_\theta$ since these are different representations of the same angle.

### 4.1.4 Alignment, Location and Sensor Bias

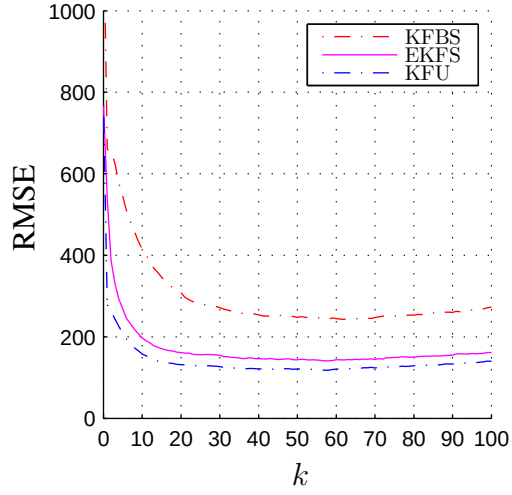
Table 4.2 shows simulation and filter models for the first scenario with alignment, location and sensor biases. The system model is used to produce measurements for the estimators, while the estimators are based on filter models. The UBE is omitted from the results since it performs worse than both the KFBU and the KFBS.

#### 4.1.4.1 The Absolute and Relative Bias Estimator

Figure 4.2 shows simulation results for the ABE, implemented as an EKF. In Figure 4.2 the EKF is compared to two Kalman Filters (KFs). The KFU receives unbiased measurements, which is equivalent to saying that the biases are known. If the EKF converges close to the KFU, the biases have been successfully accounted for. The KFBU tracks targets in $\{u\}$, while the KFBS tracks targets in $\{s_0\}$. The EKFU and EKFS show the performance of the EKF in $\{u\}$ and $\{s_0\}$ respectively. If the EKFU and EKFS performances are similar to or worse than the KFBU and KFBS respectively, the bias estimation does not improve target tracking performance.

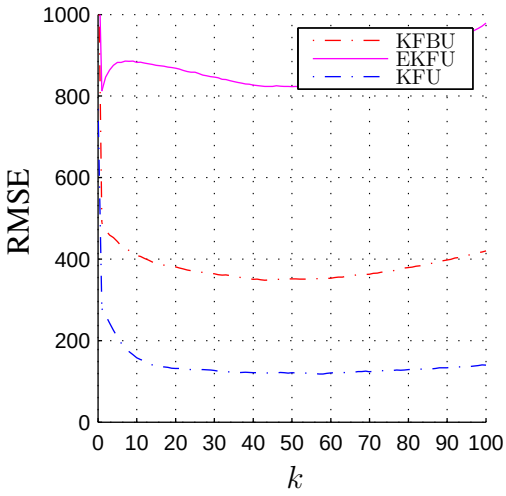



***(a)*** *Position RMSE in $\{u\}$ (EKFU).*

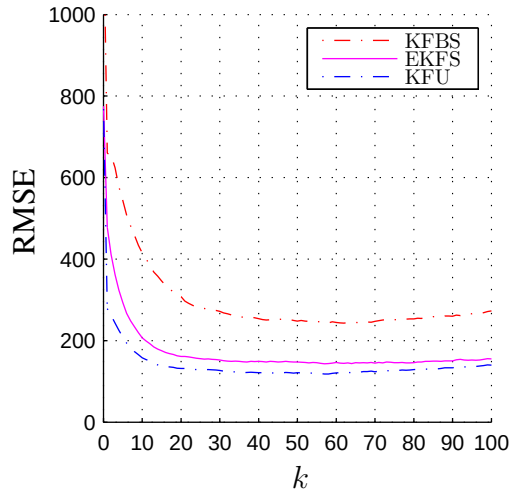



***(b)*** *Position RMSE in $\{s_0\}$ (EKFS).*

***Figure 4.2:*** *Position RMSE in meters versus time steps $k$ for the ABE.*




***(a)*** *Position RMSE in $\{u\}$ (EKFU).*



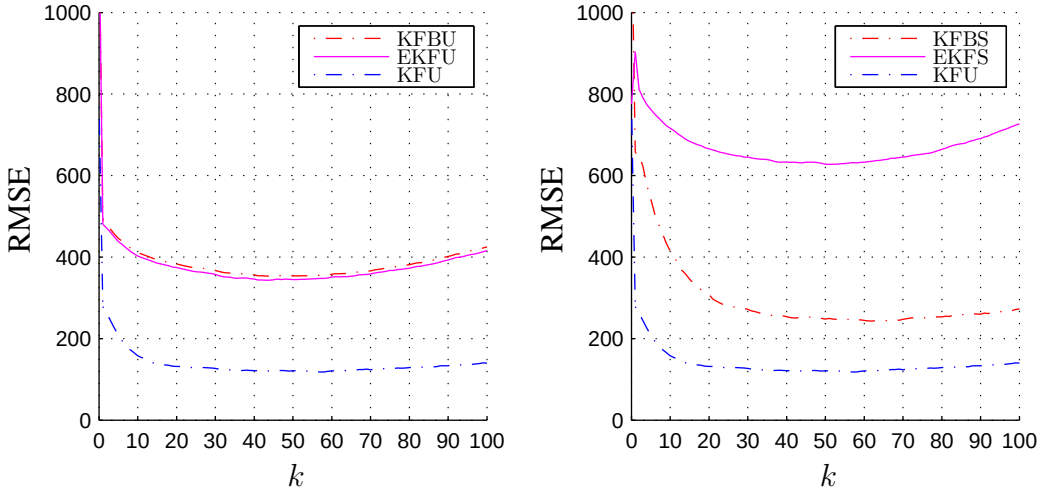

***(b)*** *Position RMSE in $\{s_0\}$ (EKFS).*

***Figure 4.3:*** *Position RMSE in meters versus time steps $k$ for the RBE.*

***Table 4.2:*** *Simulation and filter (estimator) measurement models for alignment, location and sensor bias estimation. The SUBE should ideally estimate all elements of the sensor bias $\underline{b}_i$, but due to poor performance in estimating the elevation bias $b_{\phi_i}$, the estimated biases are reduced to range $b_{\rho_i}$ and azimuth $b_{\theta_i}$.*

| Measurement model | Bias model |
|---|---|
| System model (4.8) | Present: $\underline{b}_{n_i}^{s_i}$, $\underline{b}_{n_i s_i}^{u}$, $b_{\rho_i}$, $b_{\phi_i}$, $i \in \{0, 1, \ldots, M\}$ |
| KFU model (4.14) | Known: $\underline{b}_{n_i}^{s_i}$, $\underline{b}_{n_i s_i}^{u}$, $b_{\rho_i}$, $b_{\phi_i}$, $i \in \{0, 1, \ldots, M\}$ |
| KFBU model (4.15) | White noise vector $\underline{a}_{i,k}^{s_i}$ accounts for: $\underline{b}_{n_i}^{s_i}$, $\underline{b}_{n_i s_i}^{u}$, $b_{\rho_i}$, $b_{\phi_i}$, $i \in \{0, 1, \ldots, M\}$ |
| KFBS model (4.16) & (4.17) | White noise vector $\underline{a}_{i,k}^{s_i}$ accounts for: $\underline{b}_{n_i}^{s_i}$, $\underline{b}_{n_i s_i}^{u}$, $b_{\rho_i}$, $b_{\phi_i}$, $i \in \{1, 2, \ldots, M\}$ |
| UBE model (4.18) | Estimates: $\underline{b}_{n_i}^{s_i}$, $b_{\rho_i}$, $b_{\phi_i}$, $i \in \{0, 1, \ldots, M\}$ |
| ABE model (4.19) & (4.20) | Estimates: $\underline{b}_{s_0}^{s_i}$, $\underline{b}_{s_0 s_i}^{s_0}$, $i \in \{1, 2, \ldots, M\}$, $b_{\rho_i}$, $b_{\phi_i}$, $i \in \{0, 1, \ldots, M\}$ |
| RBE model (4.21) & (4.22) | Estimates: $\underline{b}_{s_0}^{s_i}$, $\underline{b}_{s_0 s_i}^{s_0}$, $i \in \{1, 2, \ldots, M\}$, $b_{\rho_i}$, $b_{\phi_i}$, $i \in \{0, 1, \ldots, M\}$ |
| SUBE model (4.23) | Estimates: $b_{\rho_i}$, $b_{\theta_i}$, $i \in \{0, 1, \ldots, M\}$ |
| SABE model (4.24) & (4.25) | Estimates: $\underline{b}_{s_0 s_i}^{s_0}$, $i \in \{1, 2, \ldots, M\}$, $\underline{b}_i$, $i \in \{0, 1, \ldots, M\}$ |

In this simulation the estimation performance of the ABE and the RBE is approximately identical in $\{s_0\}$, as shown in Figure 4.2b and Figure 4.3b respectively. The EKFS converges close to the KFU in both cases. In $\{u\}$ both the ABE and the RBE perform worse than the KFBU, which is expected since the master sensor (Sensor 0) has an alignment bias which is not estimated. Thus the transformation from $\{s_0\}$ to $\{u\}$ is not known accurately. Figure 4.2a shows better performance of the ABE in $\{u\}$ compared to the RBE in Figure 4.3a. This is explained by the unestimated master sensor bias in the RBE.



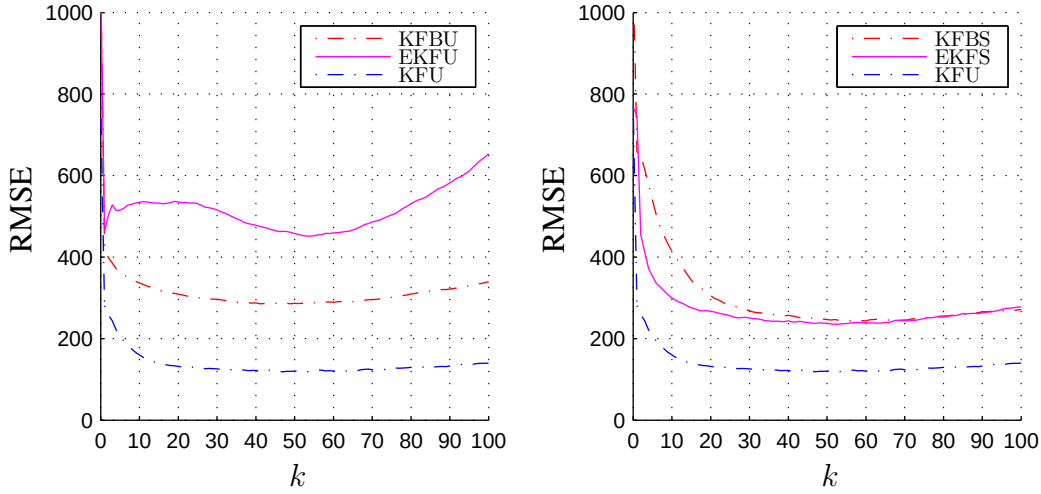

***(a)*** *Position RMSE in $\{u\}$ (EKFU).* ***(b)*** *Position RMSE in $\{s_0\}$ (EKFS).*

***Figure 4.4:*** *Position RMSE in meters versus time steps $k$ for the SUBE.*

#### 4.1.4.2 The Standard Universal Bias Estimator

As noted in Table 4.2, the sensor bias of the SUBE only includes biases in range and azimuth since elevation bias estimation performs poorly in this scenario. This poor performance is explained by two factors. From (3.74) it is clear that at long range there is more measurement noise along the $z$ axis than in the $xy$ plane. This makes it more challenging to estimate the elevation bias. The other factor is the bias model mismatch. The system model produces sensor measurements where all sensors have an alignment bias, which consists of the three Euler angles. The SUBE tries to compensate for the alignment bias using azimuth and elevation biases. It succeeds in compensating for the yaw alignment bias with its azimuth bias model because these angle biases are geometrically the same, and they are situated in the $xy$ plane which has less measurement noise than planes including the $z$ axis. Estimation of the elevation bias is subject to disturbance from both larger measurement noise and the unmodeled roll and pitch biases.

Figure 4.4 shows the estimation performance of the SUBE. The EKFU has similar performance as the KFBU, while the EKFS performs worse than the KFBS. The measurements are processed in $\{u\}$ which leads to the superior performance of the EKFU compared to the EKFS. The EKFS gets larger errors because the coordinate transformation from $\{u\}$ to $\{s_0\}$ is not accurately estimated.




***(a)*** *Position RMSE in $\{u\}$ (EKFU).* ***(b)*** *Position RMSE in $\{s_0\}$ (EKFS).*

***Figure 4.5:*** *Position RMSE in meters versus time steps $k$ for the SABE.*

#### 4.1.4.3 The Standard Absolute Bias Estimator

Figure 4.5 shows the estimation performance for the SABE. The EKFU performs poorly, while the EKFS has about the same performance as the KFBS. The superior performance of the EKFS compared to the EKFU is explained by the fact that measurements are processed in $\{s_0\}$.

#### 4.1.4.4 Conclusions

Figure 4.2 and Figure 4.3 show the estimator performance on a simulation scenario where the alignment, location and sensor biases are present for the ABE and RBE respectively. In this scenario the angles involved in all known and unknown coordinate system transformations are small. Hence the measurement model for a biased sensor in absolute bias estimation (4.19) is correct, since the angles of the Coordinate Transformation Matrix (CTM) from master Sensor 0 to Sensor $i$ $T_{s_0}^{s_i}$ are assumed small. The measurement model for a biased sensor in relative bias estimation (4.21) is almost correct. The only bias missing is the master sensor range and elevation biases. In fact the performance of the ABE and the RBE is almost identical, which is explained by this small difference. The additional error introduced by the RBE is small enough not to influence the tracking performance.

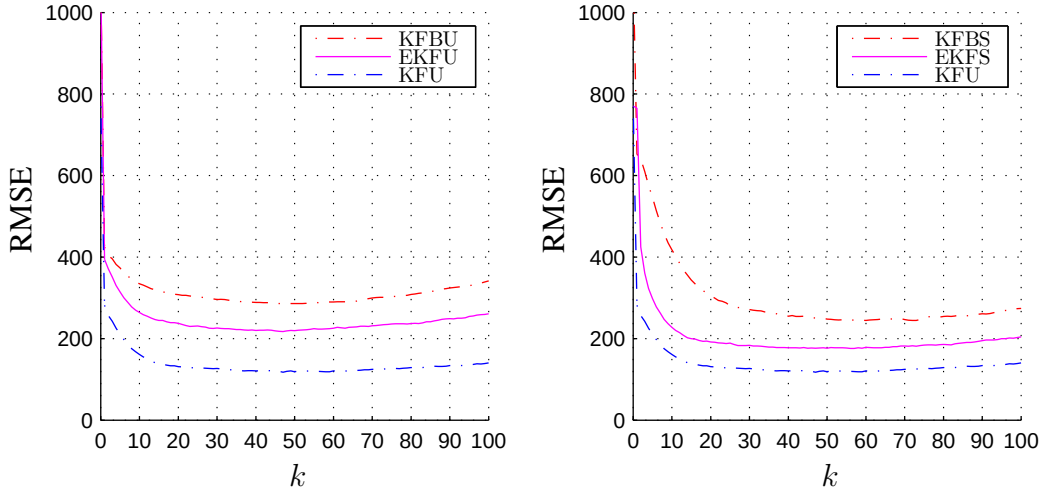
***Table 4.3:*** *Simulation and filter (estimator) measurement models for alignment, location and sensor range bias estimation.*

| Measurement model | Bias model |
|---|---|
| System model (4.8) | Present: $\underline{b}_{n_i}^{s_i}$, $\underline{b}_{n_i s_i}^{u}$, $b_{\rho_i}$, $i \in \{0, 1, \ldots, M\}$ |
| KFU model (4.14) | Known: $\underline{b}_{n_i}^{s_i}$, $\underline{b}_{n_i s_i}^{u}$, $b_{\rho_i}$, $i \in \{0, 1, \ldots, M\}$ |
| KFBU model (4.15) | White noise vector $\underline{a}_{i,k}^{s_i}$ accounts for: $\underline{b}_{n_i}^{s_i}$, $\underline{b}_{n_i s_i}^{u}$, $b_{\rho_i}$, $i \in \{0, 1, \ldots, M\}$ |
| KFBS model (4.16) & (4.17) | White noise vector $\underline{a}_{i,k}^{s_i}$ accounts for: $\underline{b}_{n_i}^{s_i}$, $\underline{b}_{n_i s_i}^{u}$, $b_{\rho_i}$, $i \in \{1, 2, \ldots, M\}$ |
| UBE model (4.18) | Estimates: $\underline{b}_{n_i}^{s_i}$, $b_{\rho_i}$, $i \in \{0, 1, \ldots, M\}$ |
| ABE model (4.19) & (4.20) | Estimates: $\underline{b}_{s_0}^{s_i}$, $\underline{b}_{s_0 s_i}^{s_0}$, $i \in \{1, 2, \ldots, M\}$, $b_{\rho_i}$, $i \in \{0, 1, \ldots, M\}$ |
| RBE model (4.21) & (4.22) | Estimates: $\underline{b}_{s_0}^{s_i}$, $\underline{b}_{s_0 s_i}^{s_0}$, $b_{\rho_i}$, $i \in \{1, 2, \ldots, M\}$ |
| SUBE model (4.23) | Estimates: $b_{\rho_i}$, $b_{\theta_i}$, $i \in \{0, 1, \ldots, M\}$ |
| SABE model (4.24) & (4.25) | Estimates: $\underline{b}_{s_0 s_i}^{s_0}$, $i \in \{1, 2, \ldots, M\}$, $b_{\rho_i}$, $b_{\theta_i}$, $i \in \{0, 1, \ldots, M\}$ |

Figure 4.4 shows the estimator performance of the SUBE. In $\{u\}$ it has similar performance as the bias ignorant KFBU. In $\{s_0\}$ it performs worse than the KFBS. Since the KFBU and the KFBS are simpler mathematically, they seem to be better choices.

Figure 4.5 shows the estimator performance of the SABE. In $\{u\}$ it performs poorly, and in $\{s_0\}$ it has similar performance as the KFBS. This means that its performance is never better than the KFBU nor the KFBS which ignore all measurement biases. This poor performance is explained by the bias model mismatch since the angular biases in roll and pitch are not estimated.

The UBE performs poorly both compared to the KFBU and the KFBS. The location bias degrades the UBE performance as it is not estimated.




***(a)*** *Position RMSE in* $\{u\}$ *(EKFU).* ***(b)*** *Position RMSE in* $\{s_0\}$ *(EKFS).*

***Figure 4.6:*** *Position RMSE in meters versus time steps* $k$ *for the UBE.*

### 4.1.5 Alignment, Location and Sensor Range Bias

Simulation and filter models are shown in Table 4.3 where alignment, location and sensor range biases are present.

#### 4.1.5.1 The Universal Bias Estimator

In Section 3.4 it is concluded that joint estimation of target position, alignment bias and sensor bias in universal coordinates is impossible. However these biases are estimable separately. The effect of the alignment bias and sensor elevation bias on target position is small at close range and large at long range, while the sensor range bias is constant. According to Figure 3.9a the range bias has stronger estimability close to the sensors. Thus it seems to be a good assumption to neglect the range bias at long range, and neglect the alignment bias at short range. This means that targets at long range are assumed to only be affected by the alignment bias, while short range targets are only affected by the range bias. The problem is now estimable. In the scenario shown in Figure 4.1 a target is defined to be at short range when the predicted position of each Cartesian coordinate $x$, $y$ and $z$ is in the interval $[-25\ \mathrm{km}, 25\ \mathrm{km}]$. If the target exceeds this interval in any coordinate it is at long range. This threshold is chosen based on Figure 3.9a, but recall that it is only approximate since the constant velocity assumption is not included in the static estimability plots.

Figure 4.6 shows the estimation performance for the UBE. The EKFU performs better than the KFBU in Figure 4.6a, while the EKFS has superior performance compared to the KFBS in Figure 4.6b. However neither the EKFU nor the EKFS come close to the KFU.

#### 4.1.5.2 The Absolute and Relative Bias Estimator

In this simulation the estimation performance of the ABE and the RBE are similar to the performance in Figure 4.2 and Figure 4.3 respectively. The EKFS converges close to the KFU in both cases. The performance of the EKFU is poor in both cases, which is expected since the master sensor (Sensor 0) has an alignment bias which is not estimated.

#### 4.1.5.3 The Standard Universal Bias Estimator

In this simulation the estimator performance of the SUBE is similar to the performance in Figure 4.4. The EKFU performs slightly better than the KFBU, while the EKFS performs worse than the KFBS.

#### 4.1.5.4 The Standard Absolute Bias Estimator

In this simulation the estimator performance of the SABE is similar to the perfomance in Figure 4.5. The EKFS and KFBS have similar performance while the EKFU performs worse than the KFBU.

#### 4.1.5.5 Conclusions

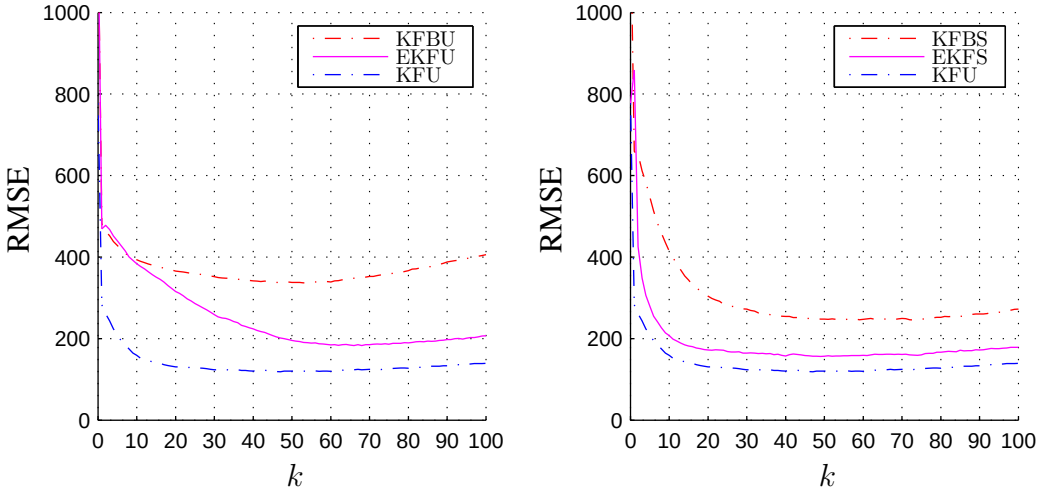
Comparing Figure 4.4 with the performance of the UBE in Figure 4.6 it is clear that the UBE has superior performance in both $\{u\}$ and $\{s_0\}$. The inferior performance of the SUBE is explained by the bias model mismatch. The azimuth bias estimation of the SUBE is only able to account for the yaw alignment bias, but not the roll and pitch biases. Since location biases are present the UBE performance is degraded since it is not able to estimate the location bias. The objective of this simulation is to find a scenario where location biases are present and the UBE estimates as many biases as possible. This is achieved by removing the sensor elevation bias. Other alternatives are to remove the alignment biases roll or pitch.

The performance of the ABE and RBE is approximately equal to the performance presented in Section 4.1.4.1, Figure 4.2 and Figure 4.3 respectively. The

***Table 4.4:*** *Simulation and filter (estimator) measurement models for alignment and sensor bias estimation. The SUBE should ideally estimate all elements of the sensor bias $\underline{b}_i$, but due to poor performance in estimating the elevation bias $b_{\phi_i}$, the estimated biases are reduced to range $b_{\rho_i}$ and azimuth $b_{\theta_i}$.*

| Measurement model | Bias model |
|---|---|
| System model (4.8) | Present: $\underline{b}_{n_i}^{s_i},\ b_{\rho_i},\ b_{\phi_i},\ i \in \{0, 1, \ldots, M\}$ |
| KFU model (4.14) | Known: $\underline{b}_{n_i}^{s_i},\ b_{\rho_i},\ b_{\phi_i},\ i \in \{0, 1, \ldots, M\}$ |
| KFBU model (4.15) | White noise vector $\underline{a}_{i,k}^{s_i}$ accounts for: $\underline{b}_{n_i}^{s_i},\ b_{\rho_i},\ b_{\phi_i},\ i \in \{0, 1, \ldots, M\}$ |
| KFBS model (4.16) & (4.17) | White noise vector $\underline{a}_{i,k}^{s_i}$ accounts for: $\underline{b}_{n_i}^{s_i},\ b_{\rho_i},\ b_{\phi_i},\ i \in \{1, 2, \ldots, M\}$ |
| UBE model (4.18) | Estimates: $\underline{b}_{n_i}^{s_i},\ b_{\rho_i},\ b_{\phi_i},\ i \in \{0, 1, \ldots, M\}$ |
| ABE model (4.19) & (4.20) | Estimates: $\underline{b}_{s_0}^{s_i},\ i \in \{1, 2, \ldots, M\},\ b_{\rho_i},\ b_{\phi_i},\ i \in \{0, 1, \ldots, M\}$ |
| RBE model (4.21) & (4.22) | Estimates: $\underline{b}_{s_0}^{s_i},\ b_{\rho_i},\ b_{\phi_i},\ i \in \{1, 2, \ldots, M\}$ |
| SUBE model (4.23) | Estimates: $b_{\rho_i},\ b_{\theta_i},\ i \in \{0, 1, \ldots, M\}$ |
| SABE model (4.24) & (4.25) | Estimates: $\underline{b}_i,\ i \in \{0, 1, \ldots, M\}$ |

fact that the ABE model is correct, and the RBE is almost correct, explains the superior performance in $\{s_0\}$ of the ABE and RBE in Figure 4.2b and Figure 4.3b with respect to the UBE in Figure 4.6b. In the absolute and relative case there are fewer biases to estimate, and the mathematical model is correct. The superior performance is obtained in $\{s_0\}$ which is where tracking is done in the absolute and relative case. In $\{u\}$ the ABE and RBE perform poorly, as seen in Figure 4.2a and Figure 4.3a with respect to Figure 4.6a. This is because it assumes that the CTM $T_{s_0}^u = I$, which is not the case. The performance of the bias ignorant KFBU is superior, which means that there is no point in using the ABE or the RBE when tracking targets in $\{u\}$. This highlights the importance of universal bias estimation if tracking is done in sensor independent coordinates.




*(a) Position RMSE in $\{u\}$ (EKFU).* *(b) Position RMSE in $\{s_0\}$ (EKFS).*

***Figure 4.7:*** *Position RMSE in meters versus time steps $k$ for the UBE.*

### 4.1.6 Alignment and Sensor Bias

Simulation and filter models are shown in Table 4.4 where alignment and sensor biases are present.

#### 4.1.6.1 The Universal Bias Estimator

Since joint estimation of the alignment and sensor elevation bias is unestimable, we neglect the sensor elevation bias at long range. This means that targets at long range are assumed to only be affected by the alignment bias, while short range targets are only affected by the sensor bias. The problem is now estimable.

Figure 4.7 shows the RMSE in position for the UBE. In Figure 4.7 we see that both the EKFU and EKFS converge close to the KFU, but the KFU does perform noticeably better.

#### 4.1.6.2 The Absolute and Relative Bias Estimator

In this simulation the estimation performance of the ABE and the RBE are similar to the performance in Figure 4.2 and Figure 4.3 respectively. The EKFS converges close to the KFU in both cases. The performance of the EKFU is poor in both cases, which is expected since the master sensor (Sensor 0) has an

alignment bias which is not estimated.

#### 4.1.6.3 The Standard Universal Bias Estimator

In this simulation the estimator performance of the SUBE is similar to the performance in Figure 4.4. The EKFU performs slightly better than the KFBU, while the EKFS performs worse than the KFBS.

#### 4.1.6.4 The Standard Absolute Bias Estimator

In this simulation the estimator performance of the SABE is similar to the perfomance in Figure 4.5. The EKFS and KFBS have similar performance while the EKFU performs worse than the KFBU.

#### 4.1.6.5 Conclusions

Figure 4.7 shows the estimator performance on a simulation scenario where the alignment and sensor biases are the only measurement biases present for the UBE. The UBE converges close to the unbiased KFU in both $\{u\}$ and $\{s_0\}$. Note that the UBE has slightly better performance in $\{s_0\}$ than in $\{u\}$ as seen in Figure 4.7. The reason is that target position measurements in any sensor coordinate system $\{s_i\}$ are unbiased in that system, so performance is slightly improved. When fusing the measurements in $\{s_i\}$ the unbiased measurements are given more weight in the EKF than the biased measurements, since the bias estimates have not converged. The unbiased measurements ensure that the estimator gets nearly unbiased tracks on the targets quickly. In a sensor independent coordinate system such as $\{u\}$ all measurements are biased, thus performance suffers until the biases of at least one sensor has converged. Since one cannot expect the bias estimates to converge exactly to the true values, performance is slightly better in a sensor coordinate system. This suggests that tracking should always be done in a sensor coordinate system if the only measurement biases present are coordinate system transformations.

It is interesting to compare the UBE performance in Figure 4.6 with its performance in Figure 4.7 where the alignment and sensor biases are the only biases present. Since the location bias is ignored by the UBE it is expected that performance degrades when location biases are present. This is clearly seen by comparing Figure 4.6 with Figure 4.7. However the performance degradation is clearly worse in $\{u\}$ than in $\{s_0\}$.

***Table 4.5:*** *Simulation and filter (estimator) measurement models for location and sensor bias estimation.*

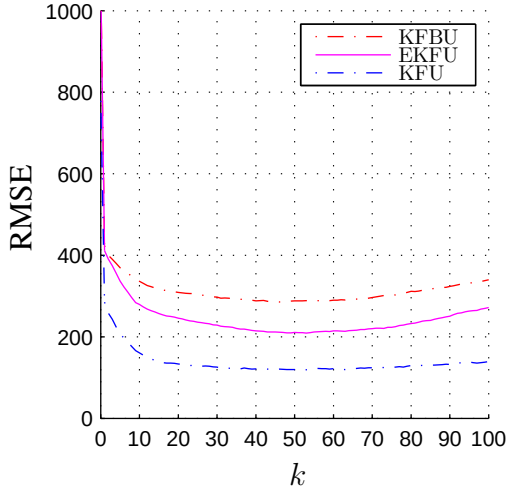
| Measurement model | Bias model |
|---|---|
| System model (4.8) | Present: $\underline{b}^{u}_{n_i s_i},\ \underline{b}_i,\ i \in \{0, 1, \ldots, M\}$ |
| KFU model (4.14) | Known: $\underline{b}^{u}_{n_i s_i},\ \underline{b}_i,\ i \in \{0, 1, \ldots, M\}$ |
| KFBU model (4.15) | White noise vector $\underline{a}^{s_i}_{i,k}$ accounts for: $\underline{b}^{u}_{n_i s_i},\ \underline{b}_i,\ i \in \{0, 1, \ldots, M\}$ |
| KFBS model (4.16) & (4.17) | White noise vector $\underline{a}^{s_i}_{i,k}$ accounts for: $\underline{b}^{u}_{n_i s_i},\ \underline{b}_i,\ i \in \{1, 2, \ldots, M\}$ |
| UBE model (4.18) | Estimates: $b_{\gamma_i},\ b_{\rho_i},\ b_{\phi_i},\ i \in \{0, 1, \ldots, M\}$ |
| ABE model (4.19) & (4.20) | Estimates: $\underline{b}^{s_0}_{s_0 s_i},\ i \in \{1, 2, \ldots, M\},\ b_{\gamma_i},\ b_{\rho_i},\ b_{\phi_i},\ i \in \{0, 1, \ldots, M\}$ |
| RBE model (4.21) & (4.22) | Estimates: $\underline{b}^{s_0}_{s_0 s_i},\ b_{\gamma},\ b_{\rho_i},\ b_{\phi_i},\ i \in \{1, 2, \ldots, M\}$ |
| SUBE model (4.23) | Estimates: $\underline{b}_i,\ i \in \{0, 1, \ldots, M\}$ |
| SABE model (4.24) & (4.25) | Estimates: $\underline{b}^{s_0}_{s_0 s_i},\ i \in \{1, 2, \ldots, M\},\ \underline{b}_i,\ i \in \{0, 1, \ldots, M\}$ |

## 4.1.7 Location and Sensor Bias

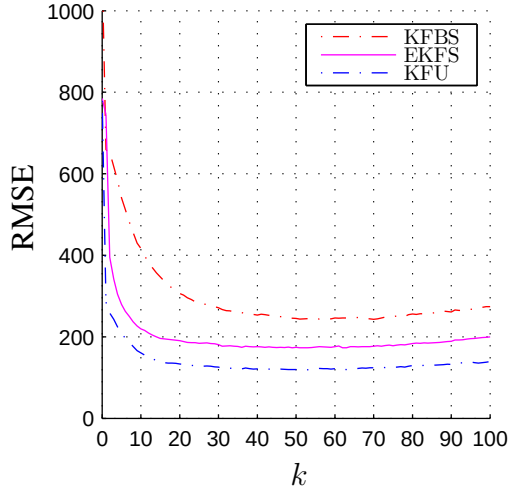
Simulation and filter models are shown in Table 4.5 where the location and sensor biases are present. This simulation case serves as a verification of the SUBE and the SABE, since they do not model the alignment bias. In fact, in this simulation the mathematical models of the UBE and the ABE become equal to the SUBE and SABE respectively. Since the RBE performance is approximately equal to the performance of the ABE and SABE in this case, we only present the plots for the SUBE and SABE.

### 4.1.7.1 The Standard Universal Bias Estimator

Figure 4.8 shows the estimation performance of the SUBE. The EKFU performance is about half way between the KFU and the KFBU, while the EKFS performance is about half way between the KFU and the KFBS.

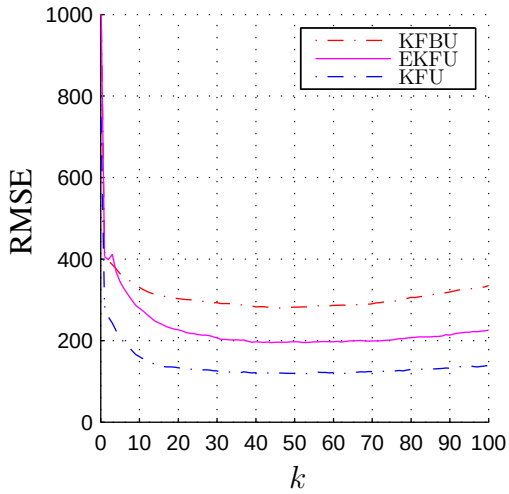



***(a)*** *Position RMSE in* $\{u\}$ *(EKFU).*

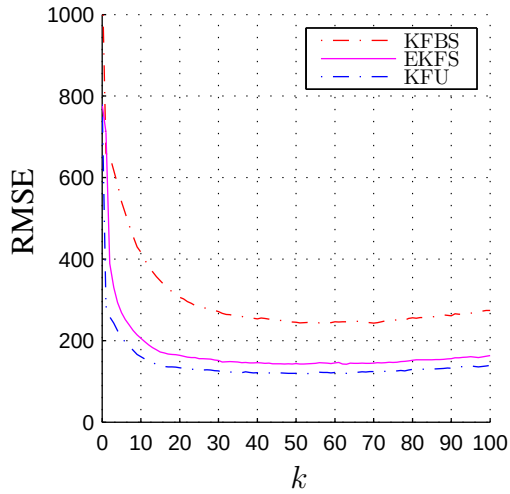



***(b)*** *Position RMSE in* $\{s_0\}$ *(EKFS).*

***Figure 4.8:*** *Position RMSE in meters versus time steps* $k$ *for the SUBE.*




***(a)*** *Position RMSE in* $\{u\}$ *(EKFU).*




***(b)*** *Position RMSE in* $\{s_0\}$ *(EKFS).*

***Figure 4.9:*** *Position RMSE in meters versus time steps* $k$ *for the SABE.*

#### 4.1.7.2 The Standard Absolute Bias Estimator

Figure 4.9 shows the estimation performance for the SABE. The EKFU performance is about half way between the KFU and the KFBU, while the EKFS has about the same performance as the KFBS. The superior performance of the EKFS compared to the EKFU is explained by the fact that measurements are processed in $\{s_0\}$.

#### 4.1.7.3 Conclusions

Figure 4.8 shows the estimator performance on a simulation scenario where the location and sensor biases are the only measurement biases present for the SUBE. Since the UBE in this case yields the same mathematical model as for the SUBE, Figure 4.8 represents UBE performance as well. Neither of these estimators estimate the location bias, thus their performance is worse than the KFU. However their performance is better than both the KFBU and the KFBS.

Figure 4.9 shows the estimator performance of the SABE. Since the ABE in this case yields the same mathematical model as for the SABE, and the RBE performance is approximately equal to the ABE, Figure 4.9 shows the performance of both the ABE and RBE in addition to the SABE. Note the substantial performance difference in $\{u\}$ shown in Figure 4.9a compared to Figure 4.5a and Figure 4.5a. Since there is no alignment bias in this case, there are fewer angular biases causing errors when transforming position estimates from $\{s_0\}$ to $\{u\}$. This leads to acceptable performance in $\{u\}$.

## 4.2 Constant Altitude Flight Using the Curvature of the Earth

In Section 4.1 the targets were considered to move in straight lines. In Air Traffic Control (ATC) aircraft may be told to fly at a certain altitude. The Earth can be approximated to an ellipsoid, hence an aircraft flying at constant altitude will move in a slight curve, and not along a straight line. This type of motion is used in this section to perform universal estimation of the alignment bias.

### 4.2.1 Measurement model

To model motion on the Earth, it is convenient to use longitude, latitude and altitude, i.e. the geodetic coordinate system as described in Section 2.2.1. The transformation $\underline{h}_e(\cdot)$ transforms a vector in Geodetic coordinates in $\{g\}$ to Cartesian coordinates in $\{e\}$. In Sudano (1997) more details on this transformation and its inverse can be found. The measurement vector $\underline{z}_{i,k}^{s_i}$ for Sensor $i$ is given by

$$\underline{z}_{i,k}^{s_i} = \underline{h}_s\left(T_{n_i}^{s_i}\left(T_e^{n_i}\left(\underline{h}_e\left(\underline{p}_k^g\right) - \underline{d}_{en_i}^e\right) + \underline{b}_{n_i s_i}^{n_i}\right)\right) + C_k^{s_i}\underline{b}_i + \underline{w}_k^{s_i}. \tag{4.26}$$

The transformation matrix $T_{n_i}^{s_i}$ is the alignment bias, $\underline{b}_{n_i s_i}^{n_i}$ is the location bias, and $\underline{b}_i$ is the sensor bias. The SBTM $C_k^{s_i}$, the transformation matrix $T_e^{n_i}$ and the displacement vector $\underline{d}_{en_i}^e$ are assumed known. Every measurement from Sensor $i$ is now given as a function of the target state in $\{g\}$ and the biases we wish to estimate. Note that both CTFs $\underline{h}_c(\cdot)$ and $\underline{h}_e(\cdot)$, and the unknown CTM $T_{n_i}^{s_i}$ make (4.26) nonlinear. The geometry of the transformations is shown in Figure 4.10.

### 4.2.2 Target Motion Model

The targets move at constant altitude, speed and heading, thus

- the target is moving on the surface of an ellipsoid given by the WGS-84 Earth ellipsoid and the constant altitude,
- and the target is moving at constant speed and heading on this surface.

In the following the target motion model is derived. The aircraft moves on an ellipsoid at constant speed and heading, i.e. its trajectory is a part of an ellipse in 3D space. This corresponds to a 3D planar turn model, which in geodetic

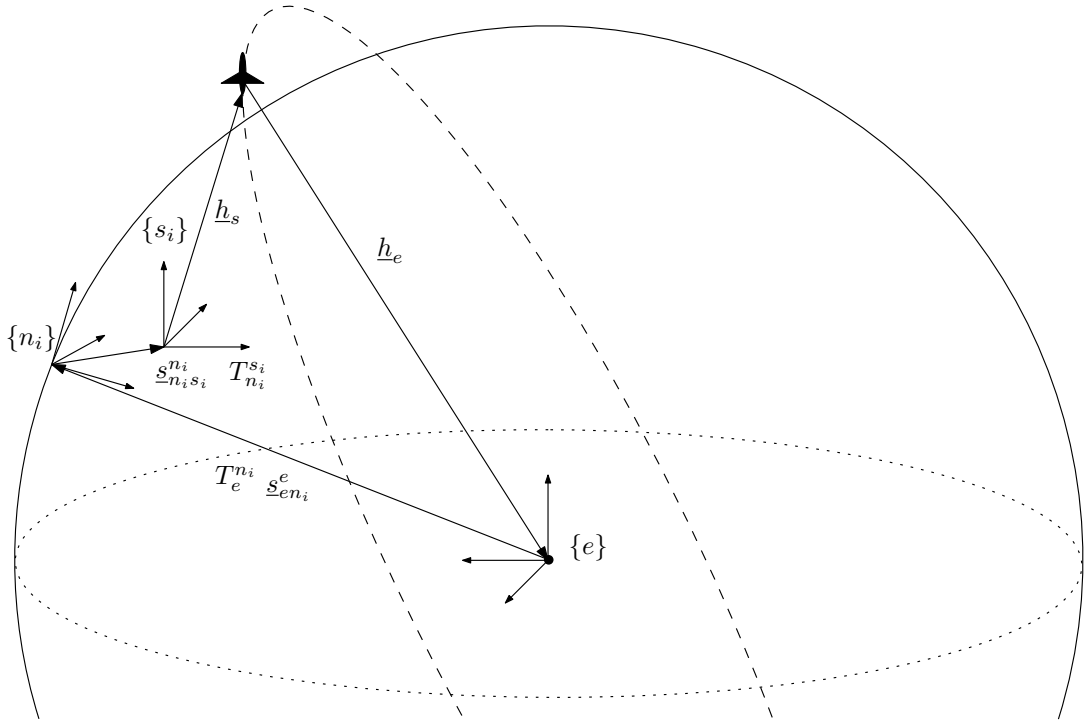



***Figure 4.10:*** *The geometrical transformations in the measurement equation. From Topland et al. (2007); reprinted with permission of the Institute of Electrical and Electronics Engineers.*

coordinates is given by

$$\dot{\mu} = \omega_\mu \tag{4.27a}$$
$$\dot{\lambda} = \omega_\lambda \tag{4.27b}$$
$$\dot{h} = 0 \tag{4.27c}$$
$$\dot{\omega}_\mu = \nu_\mu \sim \mathcal{N}\left(0, q_\mu \delta\left(t-\tau\right)\right) \tag{4.27d}$$
$$\dot{\omega}_\lambda = \nu_\lambda \sim \mathcal{N}\left(0, q_\lambda \delta\left(t-\tau\right)\right) \tag{4.27e}$$

where $\mu$, $\lambda$ and $h$ are latitude, longitude and altitude respectively. The velocity is modeled as constant in geodetic coordinates where $\nu_\mu$ and $\nu_\lambda$ are white noise scalars. The target state vector is

$$\underline{x}^u = \left[\begin{array}{ccccccccccc} \mu_1 & \lambda_1 & h_1 & \omega_{\mu,1} & \omega_{\lambda,1} & \ldots & \mu_N & \lambda_N & h_N & \omega_{\mu,N} & \omega_{\lambda,N} \end{array}\right]^T \tag{4.28}$$

for Target $1, 2, \ldots, N$, and the bias state vector is

$$\underline{x}^b = \begin{bmatrix} \alpha_1 & \beta_1 & \gamma_1 & \ldots & \alpha_M & \beta_M & \gamma_M \end{bmatrix}^T, \tag{4.29}$$

for Sensor $1, 2, \ldots, M$. The augmented state and measurement vectors are

$$\underline{x} = \begin{bmatrix} (\underline{x}^u)^T & (\underline{x}^b)^T \end{bmatrix}^T \tag{4.30}$$

$$\underline{z}_k = \begin{bmatrix} (\underline{z}_1^{s_1})^T & \ldots & (\underline{z}_N^{s_1})^T & \ldots & (\underline{z}_1^{s_M})^T & \ldots & (\underline{z}_N^{s_M})^T \end{bmatrix}^T \tag{4.31}$$

where the discretization of the process model is done according to Gelb (1974). The process model consists of the target motion model (4.27) and the bias model (3.1) augmented appropriately.

### 4.2.3 Simulation

The above model is used to estimate the target states for one sensor and one target using the Unscented Kalman Filter (UKF) (Julier and Uhlmann, 2004), mainly to avoid differentiation of (4.26). This filter is known performance-wise to be equivalent or better than the EKF. To study the observability of the biases, we use measurements without white noise. The filter however assumes both process and measurement white noise. The alignment bias is of special interest, since the observability of $\underline{b}^{s_i}$ and $\underline{b}_{n_i s_i}^{n_i}$ has been reported elsewhere. In the presented simulation the target is moving at constant altitude and heading, traveling 160 km on the Earth ellipsoid with a constant speed of 200 m/s. The sampling interval is 4 seconds, and the alignment bias is the only bias investigated. Figure 4.11 shows the successful determination of $\alpha_i$, and the target position error in $\{e\}$. Once the Euler angle bias has been determined, the error becomes small as expected. Other simulations show that $\beta_i$ is observable in the same manner, but $\alpha_i$ and $\beta_i$ are not observable when estimated at the same time. The third rotation $\gamma_i$ is never observable. It is noted that it takes some time for $\alpha_i$ to converge. This is because it takes some time for the Earth's curvature to appear for the estimator. It is essential that the curved motion is observed to find the alignment biases.

### 4.2.4 Observability and Estimability

In order to estimate all the biases, they need to be observable in the deterministic case. We will investigate the observability of the continuous system given by (4.27) and (3.1), and a continuous deterministic version of (4.26),

$$\begin{aligned} \underline{z}^{s_i}(t) &= \underline{h}_s \left( T_{n_i}^{s_i} \left[ T_e^{n_i} \left[ \underline{h}_e \left( \underline{p}^g(t) \right) - \underline{d}_{en_i}^e \right] + \underline{b}_{n_i s_i}^{n_i} \right] \right) + \underline{b}^{s_i} \\ &= \underline{h} \left( \underline{p}^g(t), \underline{b}_{n_i}^{s_i}, \underline{b}_{n_i s_i}^{n_i}, \underline{b}^{s_i} \right). \end{aligned} \tag{4.32}$$

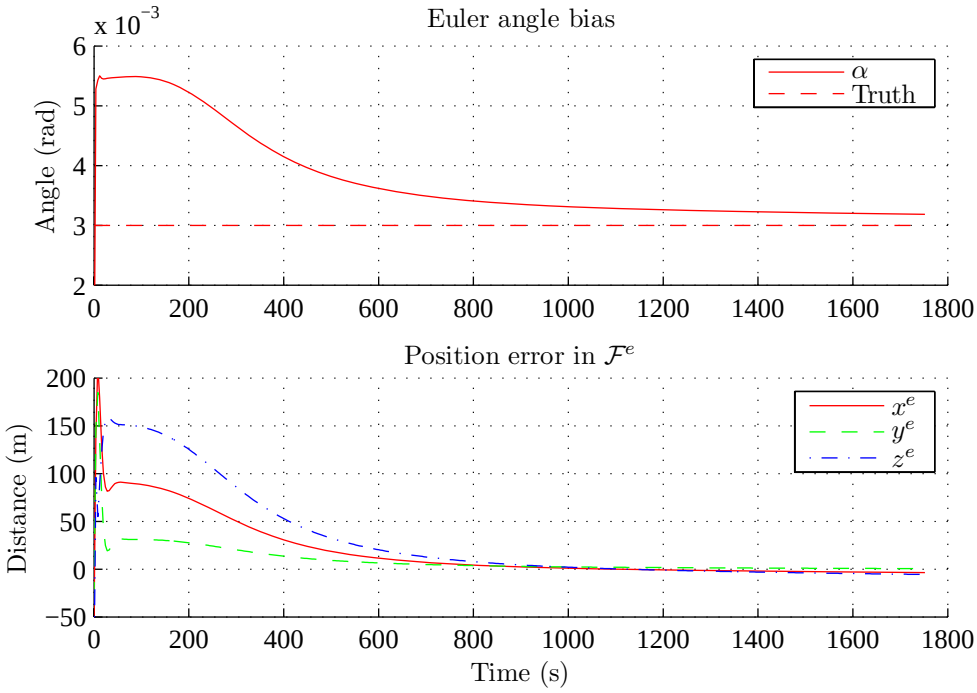



***Figure 4.11:*** *Alignment bias and target position error, where $\mathcal{F}^e = \{e\}$. From Topland et al. (2007); reprinted with permission of the Institute of Electrical and Electronics Engineers.*

Note that only the target position is time dependent. This new model adds more information to the estimation of the alignment bias, compared to the discussion in Section 3.4.5. The discussion on the location bias in Section 3.4.6 and the sensor bias in Section 3.4.7 is unchanged. In the following we discuss Earth approximations and their consequences for the observability of the alignment bias. The flat Earth assumption is treated in Section 3.4.5.

#### 4.2.4.1 Spherical Earth

If a target is moving on a spherical Earth with constant altitude and speed, it will have an acceleration which is a function of velocity and altitude. Thus it has a known acceleration, and we can expect to observe some of the alignment biases without the aid of another sensor. The expected trajectory is the equivalent of a straight line on a sphere, i.e. a great circle. The shortest path between two locations on a sphere is always a great circle. This is a circle whose center coincides with the center of the Earth. The orientation of the circle in 3D space is

unknown. In order to observe any alignment bias, its presence must yield an observed trajectory which is different from the expected trajectory. The rotational effect of these biases will keep the circular shape of the trajectory, but the center of the circle may deviate from the center of the Earth. Biases which generate this effect will be observable. Since we know the true Earth center location we can determine the biases that change the observed Earth center.

**Proposition 4.1**
*Assuming the Earth is spherical the alignment biases of Sensor $i$, $\alpha_i$ and $\beta_i$, will move the observed center of the circular trajectory away from the center of the Earth since the $x^{n_i}$- and $y^{n_i}$-axes of $\{n_i\}$ do not pass through it. The alignment bias $\gamma_i$ is not observable in this manner, since a rotation about an axis passing through the center of a sphere will not change the location of the sphere.*

**Proof:** Consider a sensor located at the equator, along the $x^e$-axis. Because of a sphere's symmetry this location is arbitrary. This sensor is tracking a target, whose position in $\{e\}$ is

$$\underline{p}_1^e = \begin{bmatrix} x^e & y^e & z^e \end{bmatrix}^T . \tag{4.33}$$

The subscript indicates position 1. The target is moving on a sphere, thus

$$(x^e)^2 + (y^e)^2 + (z^e)^2 = (r+h)^2 = r_h^2 \tag{4.34}$$

where $r$ is the Earth's radius and $h$ is the target altitude. The target location in $\{n\}$ is

$$\begin{aligned} \underline{p}^n &= T_e^n \underline{p}_1^e + \underline{s}_{en}^n = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x^e \\ y^e \\ z^e \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ r_h \end{bmatrix} \\ &= \begin{bmatrix} z^e \\ y^e \\ -x^e + r_h \end{bmatrix} . \end{aligned} \tag{4.35}$$

We will use a linearized rotation matrix $T_n^s$, assuming small biases,

$$T_n^s = \begin{bmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{bmatrix} . \tag{4.36}$$

In $\{s\}$ we obtain

$$\begin{aligned}\underline{p}^s &= \begin{bmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{bmatrix} \begin{bmatrix} z^e \\ y^e \\ -x^e + r_h \end{bmatrix} \\ &= \begin{bmatrix} z^e + \beta\,(r_h - x^e) - y^e\gamma \\ y^e - \alpha\,(r_h - x^e) + z^e\gamma \\ r_h - x^e + y^e\alpha - z^e\beta \end{bmatrix}. \end{aligned} \tag{4.37}$$

We now transform back to $\{e\}$, expecting that $\{n\} = \{s\}$,

$$\begin{aligned}\underline{p}_2^e &= \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} z^e + \beta\,(r_h - x^e) - y^e\gamma \\ y^e - \alpha\,(r_h - x^e) + z^e\gamma \\ r_h - x^e + y^e\alpha - z^e\beta \end{bmatrix} \\ &+ \begin{bmatrix} r_h \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} x^e - y^e\alpha + z^e\beta \\ y^e - r_h\alpha + x^e\alpha + z^e\gamma \\ z^e + r_h\beta - x^e\beta - y^e\gamma \end{bmatrix} \\ &= \begin{bmatrix} x^e - y^e\alpha + z^e\beta \\ y^e + x^e\alpha + z^e\gamma \\ z^e - x^e\beta - y^e\gamma \end{bmatrix} + \begin{bmatrix} 0 \\ -r_h\alpha \\ r_h\beta \end{bmatrix}. \end{aligned} \tag{4.38}$$

If we complete the same transformations, but exclude the displacements we get

$$\underline{p}_3^e = T_n^e T_n^s T_e^n \underline{p}^e = \begin{bmatrix} x^e - y^e\alpha + z^e\beta \\ y^e + x^e\alpha + z^e\gamma \\ z^e - x^e\beta - y^e\gamma \end{bmatrix}. \tag{4.39}$$

Thus we can write

$$\underline{p}_2^e = \underline{p}_3^e + \underline{r}_b \tag{4.40}$$

where $\underline{r}_b$ is a displacement bias caused by the alignment biases. This displacement bias is a displacement of the observed Earth center because of $\alpha$ and $\beta$. Note that $\gamma$ does not contribute in this manner. Since we know that $\underline{p}_3^e$ is only rotated, and thus the Earth sphere has not changed, the new sphere equation for the target

$$\begin{aligned}(x^e - y^e\alpha + z^e\beta)^2 + (y^e - r_h\alpha + x^e\alpha + z^e\gamma)^2 + (z^e + r_h\beta - x^e\beta - y^e\gamma)^2 \\ = r_h^2 \end{aligned} \tag{4.41}$$

is equivalent to

$$(x^e)^2 + (y^e - r_h\alpha)^2 + (z^e + r_h\beta)^2 = r_h^2. \tag{4.42}$$

Thus it is proven that a sensor misalignment in $\alpha$ or $\beta$ causes a circular trajectory with a displaced center. □

From the measurements we get the observed Earth sphere, while we know the true Earth sphere. However (4.42) is one equation with two unknowns, suggesting that we may only be able to observe $\alpha_i$ or $\beta_i$ or a combination. This is indeed what was observed using the model previously considered. To find the other bias, another constraint is needed. Such a constraint could be that the target is moving on a great circle. This constraint is not included in the simulated model which allows motion on a circle whose center is different from the center of the sphere. The shortest path between two points on the Earth sphere is always a section of a great circle, so a possible assumption is that an aircraft will move on a great circle. The plane of a great circle is given by (Weisstein, 2002)

$$x^e \sin c_2 + (y^e - r\alpha)\cos c_2 - \frac{(z^e + r\beta)}{\sqrt{(r+h)^2/c_1^2 - 1}} = 0 \tag{4.43}$$

where $\alpha_i$ and $\beta_i$ is included, and $c_1$ and $c_2$ are constants which can be determined from two positions on the observed great circle. So we now have two equations and two unknowns, which make $\alpha_i$ and $\beta_i$ observable. Note that this is equivalent to saying that the tilt of the horizontal plane tangent to the Earth's surface at a sensor's location is observable.

#### 4.2.4.2 Elliptical Earth

A target moving on the Earth ellipsoid with constant altitude and speed will have an acceleration which is a function of position and velocity. The expected trajectory of a target is an ellipse, whose center coincides with the center of the Earth. In this case all the alignment biases in $\underline{b}_{n_i}^{s_i}$ move the center of the observed Earth in different directions through the three Euler rotations. The reason why $\gamma_i$ also causes the center to move is that the $z^{n_i}$-axis in $\{n_i\}$ no longer passes through the Earth center, except at the poles and at the equator. The alignment biases cause the observed ellipsoid to be rotated in reference to the true ellipsoid. Note that when using an estimator to determine $\gamma_i$, the performance will be better the more elliptical the surface. For movement on an ellipsoid close to a sphere, $\gamma_i$ will be more difficult to estimate since the effect will be small, and thus sensitive to measurement noise. This applies to the Earth, as it is close to spherical.

# 5

# Aircraft Altitude Prediction Using Mode C Measurements

The altitude of an aircraft in civilian airspace is determined by Mode C transponder transmissions using Secondary Surveillance Radars (SSRs). This measurement has a resolution of 100 ft due to bandwidth constraints. The coarse resolution creates difficulties in estimating the vertical velocity of the aircraft which influences altitude prediction. Altitude prediction is important in conflict alarm systems. This problem is investigated and several Bayesian estimators, including the Interacting Multiple Model (IMM) algorithm, are tuned and tested on real measurement data for a performance comparison. It is shown that a single model filter with maneuver handling using the Unscented Kalman Filter (UKF) is superior to IMM estimators and Kalman Filters (KFs) for long term predictions.

## 5.1 Modeling

### 5.1.1 Altimeters and the Mode C Measurement

Aircraft operating in civilian airspace use altimeters to measure their own barometric altitude. This altitude is then transmitted using transponders to Air Traffic Control (ATC). Due to bandwidth constraints the altitude is transmitted with a resolution of 100 ft. The Mode C measurement $z_k$ is discrete in time and mod-

eled by

$$z_k = h\left(x_k, w_k, \Delta\right) = \Delta \text{round}\left(\frac{x_k + w_k}{\Delta}\right) \tag{5.1a}$$

$$w_k \sim \mathcal{N}\left(0, \sigma_w^2 \delta_{kl}\right) \tag{5.1b}$$

where $x_k$ is aircraft altitude, $w_k$ is white noise from the onboard altimeter and $\Delta$ is the Mode C resolution of 100 ft. The quantization function $\text{round}\left(\cdot\right)$ returns the integer closest to the input. Note that this function is nonlinear, discontinuous and many-to-one, and that the effect of the measurement noise in (5.1) depends on $x_k$ because of this nonlinearity. If the quantization is the dominant error source $w_k = 0$ is a good assumption (Sviestins and Wigren, 2001). Filter models are discussed next.

### 5.1.2 Aircraft Vertical Motion Filter Process Models

The vertical motion of an aircraft typically consists of two modes; level flight and altitude change. In level flight the aircraft will oscillate around a set altitude with low vertical velocity. When the aircraft changes altitude, the vertical velocity oscillates around a set velocity. Both of these modes will be described using stochastic linear models. The continuous time models are discretized (Gelb, 1974).

#### 5.1.2.1 Level Flight

We propose two different models for this mode. Model (5.2), which describes constant altitude, is a position random walk which is given in discrete time by

$$x_1\left(k+1\right) = x_1\left(k\right) + n_1 \tag{5.2a}$$

$$n_1 \sim \mathcal{N}\left(0, \Delta t \sigma_{n_1}^2 \delta_{kl}\right), \tag{5.2b}$$

where $x_1$ is the aircraft air pressure altitude (time index omitted), $n_1$ is the white process noise and $\Delta t$ is the sampling interval. Choosing this model to represent the level flight mode means that the predicted altitude is the same as the current altitude. It follows that the predicted vertical velocity is zero and its discretized variance is $2\sigma_{n_1}^2/\Delta t$. Model (5.3) describes motion with slowly oscillating vertical velocity $x_2$ about zero. This can be modeled appropriately

by using a continuous first order Markov process for the vertical velocity of the aircraft,

$$\dot{x}_1 = x_2 \tag{5.3a}$$

$$\dot{x}_2 = -\frac{1}{\kappa} x_2 + n_2 \tag{5.3b}$$

$$n_2 \sim \mathcal{N}\left(0, \sigma_{n_2}^2 \delta\left(t - \tau\right)\right), \tag{5.3c}$$

where $n_2$ is the white process noise and $\kappa$ is the correlation time of the Markov process (Gelb, 1974). We see that these two models differ with respect to state and covariance predictions.

#### 5.1.2.2 Altitude Change

In this mode we choose a velocity random walk model. The continuous time Model (5.4) is given by

$$\dot{x}_1 = x_2 \tag{5.4a}$$

$$\dot{x}_2 = n_3 \tag{5.4b}$$

$$n_3 \sim \mathcal{N}\left(0, \sigma_{n_3}^2 \delta\left(t - \tau\right)\right) \tag{5.4c}$$

where $n_3$ is the white process noise. Note that the vertical velocity prediction of this model is not constrained about zero as in model (5.3). The predicted vertical velocity at discrete time $k + 1$ equals the vertical velocity estimate at time $k$.

## 5.2 Estimation

### 5.2.1 Nonlinear Estimators

Since the Mode C measurement (5.1) is nonlinear, nonlinear filtering algorithms should be used. We will investigate Bayesian methods which can be divided into methods that estimate entire probability density functions (pdfs) and methods that estimate moments of pdfs.

#### 5.2.1.1 Estimation of Probability Density Functions

The general Bayesian filtering equations

$$p\left(\underline{x}_k \mid Z_{k-1}\right) = \int p\left(\underline{x}_k \mid \underline{x}_{k-1}\right) p\left(\underline{x}_{k-1} \mid Z_{k-1}\right) d\underline{x}_{k-1} \tag{5.5a}$$

$$p\left(\underline{x}_k \mid Z_k\right) = \frac{p\left(\underline{z}_k \mid \underline{x}_k\right) p\left(\underline{x}_k \mid Z_{k-1}\right)}{\int p\left(\underline{z}_k \mid \underline{x}_k\right) p\left(\underline{x}_k \mid Z_{k-1}\right) d\underline{x}_k} \tag{5.5b}$$

return the posterior conditional probability density function (cpdf) $p\left(\underline{x}_k \mid Z_k\right)$ given the prior cpdf $p\left(\underline{x}_{k-1} \mid Z_{k-1}\right)$, the process cpdf $p\left(\underline{x}_k \mid \underline{x}_{k-1}\right)$, the likelihood function $p\left(\underline{z}_k \mid \underline{x}_k\right)$, the initial pdf $p\left(\underline{x}_0\right)$ and the measurement sequence $Z_k = \{\underline{z}_0, \underline{z}_1, \ldots, \underline{z}_k\}$. In this case $p\left(\underline{x}_0\right)$ is a Gaussian distribution. However it is passed through (5.1), so the resulting pdf is not Gaussian. The likelihood function $p\left(\underline{z}_k \mid \underline{x}_k\right)$ is

$$p\left(z_k \mid \underline{x}_k\right) = \begin{cases} 1 \text{ for } x_1\left(k\right) \in \left[z_k - \frac{\Delta}{2}, z_k + \frac{\Delta}{2}\right\rangle \\ 0 \text{ otherwise} \end{cases} \tag{5.6}$$

where we have assumed a deterministic measurement, i.e. $w_k = 0$ in (5.1). One of the most common ways of computing pdfs is to use the Particle Filter (PF) (Ristic et al., 2004). Note that the PF may diverge if no particles are validated by the measurement. This problem can be treated by increasing process noise to increase the probability that there are measurement validated particles. Details on the PF can be found in Appendix C.1.

#### 5.2.1.2 Estimation of Moments

Calculating (5.5a) and (5.5b) is computationally expensive. An alternative is to calculate the first two moments of the pdf, which can be done considerably faster. Typical moment matching filters include the Extended Kalman Filter (EKF) and the more recent UKF (Julier and Uhlmann, 2004). The latter is presented in Appendix C.2, and uses the unscented transform to match moments instead of performing linearization like the EKF. It has been shown that the EKF produces poor results applied to our problem of measurement quantization (Sviestins and Wigren, 2001). This is because (5.1) is not smooth. Thus it needs to be approximated by a smooth function, yielding an incorrect measurement model. The UKF does not have this weakness, and it has the potential to match the fourth central moment of the pdf in addition to its mean and covariance. The measurement model (5.1) needs to be modified when using the UKF. The UKF uses sigma points to match the moments of the pdf. It does not include the measurement noise of the quantization function in (5.1) when all sigma points are within

the quantization interval. Therefore the measurement noise is added. We will use the filter measurement model

$$z_k = \Delta \text{round}\left(\frac{x_1}{\Delta}\right) + w_k \tag{5.7a}$$

$$w_k \sim \mathcal{N}\left(0, \sigma_w^2 \delta_{kl}\right) \tag{5.7b}$$

where the white measurement noise variance $\sigma_w^2$ is chosen to match the variance of a uniform distribution (Soderstrom, 2002) with $\sigma_w^2 = \Delta^2/12 \approx 833$ ft$^2$.

### 5.2.2 Linear Estimator

Because of its simplicity and speed, the KF is a popular choice for filtering problems. It is optimal for linear systems with Gaussian pdfs. For nonlinear problems it is suboptimal, but it may still provide satisfactory results. We will use the filter measurement model

$$z_k = x_1(k) + w_k \tag{5.8a}$$

$$w_k \sim \mathcal{N}\left(0, \sigma_w^2\right) \tag{5.8b}$$

when we use the KF. Equation (5.8) is clearly different from (5.1) and (5.7). Finding a value for the filter model measurement noise variance $\sigma_w^2$ is challenging, since the measurement model should be as close to (5.1) as possible. It is tempting to choose a constant approximation of $\sigma_w^2$, for example matching it with the variance of a uniform measurement as in (5.7). In this way the normal measurement noise distribution will have a variance matched to a uniform distribution, but higher order moments are not matched. This is done for all KF based estimators in the following.

### 5.2.3 Interacting Multiple Model Estimator

Since we have two modes for the motion of the aircraft, we would like to include these two modes in our altitude and vertical velocity estimator. One of the best ways to do multiple model estimation for target tracking is to use the IMM framework (Bar-Shalom et al., 2001). Using the IMM estimator on the vertical motion of an aircraft has previously been suggested in Bar-Shalom et al. (2001), but only the horizontal motion is treated in that work. The IMM estimator works by processing several filters in parallel where each filter uses a different

mathematical model for the target motion. Transition probabilities between the possible models are given as prior information.

### 5.2.4 Maneuver Handling

When an estimator reaches steady state it is vulnerable to maneuvers. Steady state means that the covariance estimates have converged and the measurements are arriving close to predictions. If the aircraft changes velocity, the estimator will lag behind for a period of time before adjusting to the new velocity. We want this period of time to be short, in order to avoid too many inaccurate predictions.

#### 5.2.4.1 Adjustable Level Process Noise

A simple way to handle maneuvers is to use process noise level switching (Bar-Shalom et al., 2001). The Normalized Innovation Squared (NIS)

$$\underline{\epsilon}_k = \underline{\nu}_k^T S_k^{-1} \underline{\nu}_k \tag{5.9}$$

is monitored, and if it exceeds a certain threshold the process noise is increased. The vector $\underline{\nu}_k$ is the innovation (prediction error) and $S_k$ is its covariance. When the NIS is no longer exceeded, the process noise switches back. If we assume that the innovation and its covariance is linear and Gaussian, the NIS is chi-square distributed. The threshold is chosen so that there is a certain chance of exceeding it (tail probability). An alternative to process noise switching is to reinitialize the estimator when the threshold is exceeded.

#### 5.2.4.2 The IMM Estimator

A more advanced method is to use the IMM framework for maneuver handling (Bar-Shalom et al., 2001). In fact this is probably the most common way of employing the IMM structure. One of the parallel filters has low process noise while the other has increased process noise to handle maneuvers. The IMM estimator will automatically switch between the models as necessary. In fact the estimate is always a weighted average of the two models. Hence it does not switch completely as with the previous method, but increases or decreases the model weights.

## 5.3 Results

### 5.3.1 Estimator Tuning on Real Trajectories

The estimators are tuned on 130 measurement sequences generated from previously filtered trajectories based on real Mode C measurements. First long sequences of level flight are removed from the trajectories, and then they are quantized to recreate Mode C measurements. The long sequences of level flight are long sequences of the same Mode C measurement. All estimators can be tuned to perform well on such flight behavior, and since this flight mode is likely to be dominant it may hide differences in estimator performance. When these long sequences are removed, staircase measurements from constant velocity motion are dominant. This is the measurement behavior which is problematic for the estimators. We will refer to the remaining sequences as the tuning data. The tuning variables are process noise levels, the sigma point weights of the UKF and the tail probability of the NIS threshold. The tuning is based on minimizing the Least Squares (LS) cost function (Bar-Shalom et al., 2001)

$$c_{LS}\left(\underline{\xi}\right) = \arg\min_{\underline{\xi}} \sum \underline{\nu}_k^T \underline{\nu}_k \tag{5.10}$$

using numerical search when processing all the measurements. The vector $\underline{\xi}$ contains the tuning variables and $\underline{\nu}_k$ is the innovation for five step predictions. We use five step predictions to emphasize the velocity estimates, since the position estimates are too dominant in one step predictions. This tuning scheme applies to all estimators except the PF which is tuned using the Maximum Likelihood (ML) criterion (Bar-Shalom et al., 2001) on five step predictions

$$c_{ML}\left(\underline{\xi}\right) = \arg\max_{\underline{\xi}} p\left(\underline{x}_k \mid \underline{z}_k\right). \tag{5.11}$$

The ML criterion yields better results for the PF regarding velocity convergence. Tuning the PF using the ML criterion leads to a narrower pdf than the LS criterion since the tuning is done with respect to the most likely estimate. A narrower pdf leads to a more responsive filter, which means that the PF tuned using ML has better maneuver handling than the PF tuned using LS. This explains the superior performance of the former.

### 5.3.2 The Estimators Tested

Table 5.1 shows the estimators tested. Note that only model (5.4) is used as filter process model in our performance evaluation. IMM estimators using the

***Table 5.1:*** *Estimators tested for performance evaluation. All use filter process model* (5.4)*. From Topland and Hallingstad (2007); reprinted with permission of the American Institute of Aeronautics and Astronautics, Inc.*

| Estimator | Maneuver handling | Measurement model | Comments |
|---|---|---|---|
| UKFKF | IMM* | (5.7) & (5.8) | Uses UKF on (5.7), and KF on (5.8) for maneuvers. |
| UKFIMM | IMM* | (5.7) & (5.7) | |
| UKFA | Noise adjustment using (5.9) | (5.7) | |
| UKF | None | (5.7) | |
| PF | Reinitialization using (5.9) | (5.1) | Uses Algorithm C.1 and $w_k = 0$ in (5.1). |
| KFIMM | IMM* | (5.8) & (5.8) | |
| KFA | Noise adjustment using (5.9) | (5.8) | |
| KF | None | (5.8) | |

* The IMM estimators have two parallel filter models where the process models are one standard model (5.4) and one maneuver model (5.4) with larger process noise covariance. The filter models differ in choice of measurement models and tuning.

models (5.2) or (5.3) specialized for level flight produce poor results. The reason is that in order for these to be chosen by the IMM algorithm when measurements confirm level flight, they need to have lower process noise than the constant velocity model for altitude change (5.4). Problems occur because an aircraft that climbs or descends will often produce a couple of measurements at each flight level before moving to the next due to the quantization. When that happens the level flight model is chosen by the IMM algorithm instead of model (5.4). This is logical because both models may get their predictions confirmed. This yields poor performance whenever the aircraft is actually climbing or descending, since the IMM estimator switches models to level flight when it is not appropriate. On the other hand, if the process noise of the level flight model is increased it may never be chosen, as (5.4) fits for all constant velocity motions, including zero vertical velocity.

### 5.3.3 Example: Performance on a Simulated Trajectory

The estimators are tested on a deterministic trajectory in order to show how quickly they respond to maneuvers and how they perform when vertical velocity is zero. This serves as a verification of the filter tuning. The simulated trajectory is a trajectory which all estimators should perform well on. The aircraft climbs at 6 $m/s$ for 23 samples, flies level for 22 samples and descends at 6 $m/s$ for

25 samples. The sampling interval $\Delta t = 4\ s$. The measurements are generated from a deterministic version of (5.1) where $w_k = 0$.

#### 5.3.3.1 Nonlinear Estimators

These estimators take the nonlinear measurement into account. The PF uses (5.1) whereas the UKF uses (5.7). Figure 5.1 shows estimated altitude and velocity of the aircraft, and Figure 5.2 shows one step altitude prediction errors. If we look at the estimated velocity in Figure 5.1 we see that the estimated velocity of the PF eventually converges to a value close to the truth. This is the desired filter behavior, and it is achieved because the PF models the entire pdf. The moment matching of the UKF does not necessarily converge to the true velocity. The estimate will stay at a value close to the truth for a couple of samples, and then it will jump to a new value close to the truth. Typically one would want convergence to the true value, at least when the aircraft is flying at constant altitude. However, due to the measurement quantization, it is impossible to know whether the aircraft is flying level. If the aircraft is moving slowly vertically it will generate several measurements at the same altitude, just like an aircraft which is not moving vertically. So a sequence of measurements at the same altitude fits with several values of constant velocity. The UKF chooses one of these and keeps it until it misses with a prediction, which can be seen in Figure 5.1 and 5.2. All UKF based estimators show this kind of behavior. They differ in maneuver handling, which allows for lower process noise covariance.

#### 5.3.3.2 Estimators Using the KF

The KF based estimators use the linear measurement model (5.8), and the performance of the KF is depicted in Figure 5.1 and 5.2. The KF performs well when the aircraft flies level. It converges to zero vertical velocity because that is when its measurement predictions fit perfectly. However when the vertical velocity is different from zero, the KF is not able to estimate it properly, since the measurement quantization is not modeled well. Maneuver handling separates the performance of KF based estimators.

### 5.3.4 Performance Comparison on Real Measurements

The estimators are tested on 186 real Mode C measurement sequences, the testing data, which do not include the tuning data. When comparing estimators it is important to choose performance metrics appropriate for the problem at hand. We will look at the Root Mean Square Error (RMSE) and the Average Euclidean

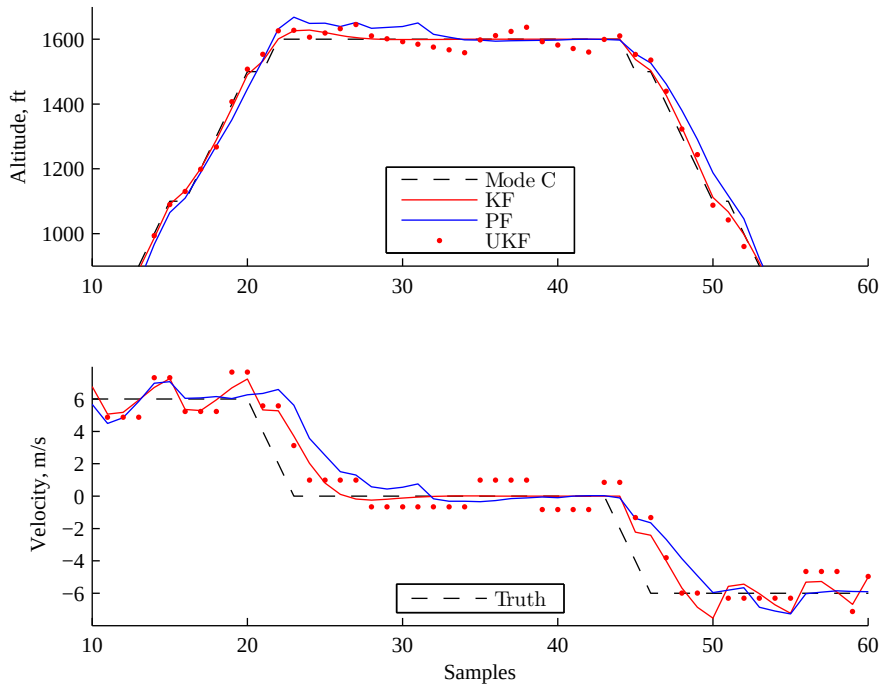



***Figure 5.1:*** *Estimates for simulated trajectory. From Topland and Hallingstad (2007); reprinted with permission of the American Institute of Aeronautics and Astronautics, Inc.*

Error (AEE) (Li and Zhao, 2001). The AEE does not penalize large errors as much as the RMSE, and it has a clear physical interpretation. The AEE is the average Euclidean distance between the prediction and the measurement. The RMSE is a statistical measure, and if the prediction errors are zero mean it equals the prediction error standard deviation. They are defined as

$$\text{RMSE} = \left( \frac{1}{M} \sum_{i=1}^{M} e_i^T e_i \right)^{1/2} \tag{5.12}$$

$$\text{AEE} = \frac{1}{M} \sum_{i=1}^{M} \left( e_i^T e_i \right)^{1/2} \tag{5.13}$$

$$e_i = \frac{1}{N_i} \sum_{k=1}^{N_i} \left( z_k - \bar{z}_k \right) \tag{5.14}$$

where $M$ is the number of measurement sequences, $N_i$ is the number of

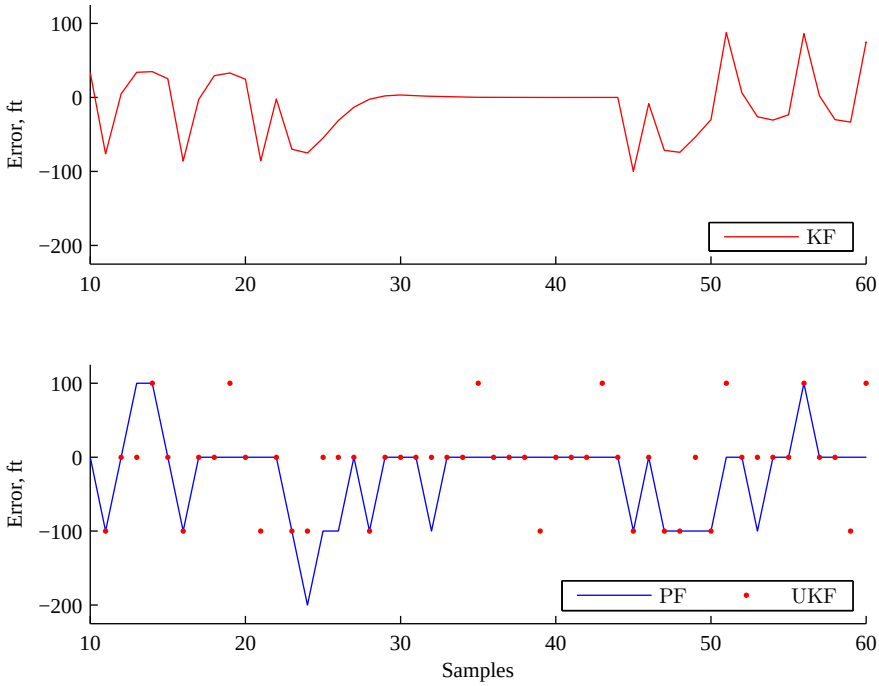



***Figure 5.2:*** *Prediction error for simulated trajectory. From Topland and Hallingstad (2007); reprinted with permission of the American Institute of Aeronautics and Astronautics, Inc.*

samples of a given measurement sequence and $\bar{z}_k$ is the predicted measurement. These metrics are computed as an average across each individual track and across all the measurement sequences.

Figure 5.3 and 5.4 show AEE and RMSE for the estimators applied to the tuning and testing data respectively. Note the large difference in performance on the tuning data versus the testing data. The estimators are tuned to fit the tuning data well, but that is not the only explanation. The testing data include long sequences of level flight. These sequences cause estimated covariance to be small. Hence the estimators have to rely heavily on their maneuver handling ability when the aircraft suddenly maneuvers. This happens more often on the testing data which leads to larger prediction errors.

Overall the PF is the worst performer. It may model the system well, but it is usually slow to converge to good velocity estimates. This is because it has larger

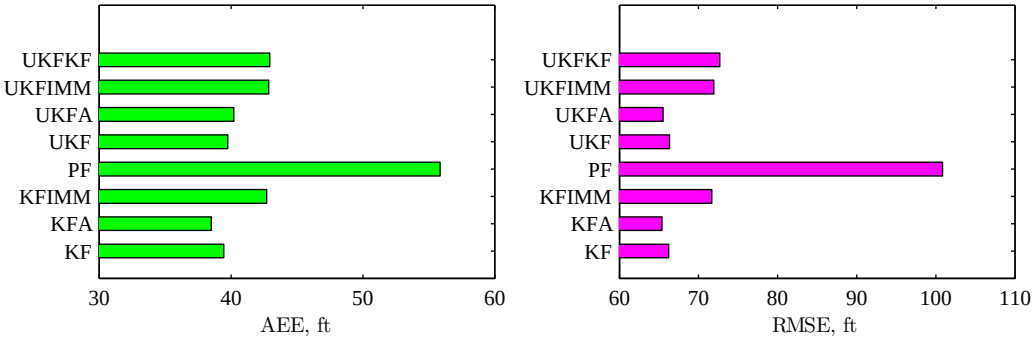



***Figure 5.3:*** *Estimator performance on tuning data for five step predictions (20 s). From Topland and Hallingstad (2007); reprinted with permission of the American Institute of Aeronautics and Astronautics, Inc.*

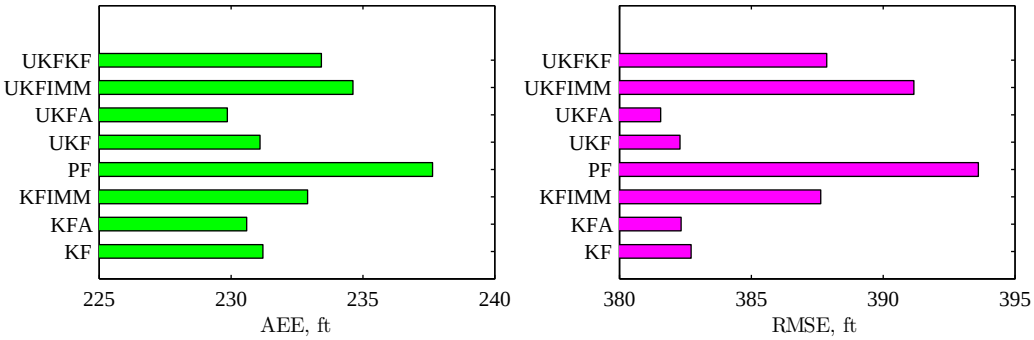



***Figure 5.4:*** *Estimator performance on testing data for five step predictions (20 s). From Topland and Hallingstad (2007); reprinted with permission of the American Institute of Aeronautics and Astronautics, Inc.*

process noise covariance which is necessary to have a good spread of the particles. The IMM estimators (KFIMM, UKFIMM, UKFKF) are the second worst performers. This is probably because they let the maneuver model have too much influence on the estimates. Stronger process noise makes the velocity estimates unstable. The UKFA is the winner regarding five step predictions, and that is explained by its maneuver handling, its accurate model of the measurement (5.7) and low process noise. The latter yields accurate velocity estimates which become more important when long predictions are made. Note that the KFA, UKF and KF are not far behind the UKFA in Figure 5.4.

# 6

# Discussion and Conclusion

This chapter discusses the measurement models in Section 3.2, the estimability analysis in Section 3.4 and the results of Chapter 4 and Chapter 5. Conclusions are drawn, and topics for future work are suggested.

## 6.1 Discussion on Linearization

Measurement models for universal, absolute and relative bias estimation were presented in Section 3.2 for estimation of an alignment bias, a location bias, and a sensor bias. The latter originates from sensor imperfections, while the other two originate from sensor calibration. The coordinate system of the sensor is neither aligned nor located correctly with respect to the assumed coordinate system. The resulting measurement models are nonlinear. Linearization is used in the bias estimators, which makes it possible to use linear analysis in the form of the static Cramer-Rao Lower Bound (CRLB), and to use linear filter algorithms. An alternative would have been to use analysis of the nonlinear models, and to use nonlinear filters. However, nonlinear function analysis is in general a challenging problem, so the simplicity of linearization is preferred.

## 6.2 Estimability Analysis using the Static CRLB

The purpose of the estimability analysis in Section 3.4 is to find when a stochastic bias estimation problem can be solved. This is investigated by plotting the

estimability index of the CRLB covariance matrix, and explaining the plots using analysis and examples.

### 6.2.1 Discussion on The Estimability Index

The CRLB is used to account for the measurement noise in addition to the geometry of the problem. To determine when a bias estimation problem becomes weakly estimable or unestimable, the estimability index (Definition 3.3) is used which is defined using the static CRLB. The estimability index is zero (becomes unestimable) when the measurement does not yield more information about *all* state variables. The goal of the estimability contour plots is to show when this happens in selected scenarios. Estimability indices not close to zero are of less importance, since weak estimability can be compensated for by adding more measurements.

A collection of sensors and targets can be positioned in 3D space in an infinite number of ways. Therefore the scenarios selected show the simplest scenarios where estimability is achieved in some area of the state space. These simple scenarios reveal challenges from the geometry of the problem and sensitivity to increased measurement noise.

### 6.2.2 Conclusions from Analysis

The analysis revealed the following important cases where the measurement biases are unestimable.

*The Alignment Bias* Unestimable in the universal case if all sensors are located along the same line. Thus at least three sensors are required to get universal estimability.

*The Location Bias* Unestimable in the universal case since no points in space are known exactly.

*The Sensor Range Bias* Unestimable in the universal and absolute case at long range due to measurement noise.

*The Alignment and Sensor Bias* Unestimable in the universal case when estimated jointly. However estimability is achieved by removing one of the angular biases roll, pitch or elevation.

The above findings would also have been found investigating observability, except for the sensor range bias case where measurement noise increasing with range makes the problem unestimable. This highlights the key benefit of using estimability instead of observability.

## 6.3 Constant Velocity Simulation Results

### 6.3.1 Discussion

The conclusions regarding estimability in Section 3.4 are used to decide interesting simulation scenarios in Section 4.1. The goal of Section 4.1 is to compare the performance of the Extended Kalman Filters (EKFs) the Universal Bias Estimator (UBE), the Absolute Bias Estimator (ABE) and the Relative Bias Estimator (RBE) on targets moving at constant velocity. Targets moving at constant velocity is one of the most common assumptions in target tracking. The EKF estimators estimate measurement biases and target positions jointly. The performance of these estimators are also compared to standard bias estimators found in the literature, the Standard Universal Bias Estimator (SUBE) and the Standard Absolute Bias Estimator (SABE). Recall that the KFU, KFBU and KFBS are Kalman Filters (KFs) which herein are used as performance bounds. The biases are known in the KFU (lower bound), while they are unestimated but accounted for in the KFBU and KFBS (higher bound). The KFBU and KFBS perform tracking in the sensor indpendent universal coordinate system $\{u\}$ and the master sensor coordinate system $\{s_0\}$ respectively. The best estimators have performance close to the lower bound.

### 6.3.2 Conclusion

#### 6.3.2.1 Sensor Independent Coordinates

If tracking is performed in the sensor independent universal coordinate system $\{u\}$ performance depends on which biases it is necessary to estimate. In the first scenario neither the UBE, the ABE, the RBE, the SUBE nor the SABE yield performance which justify their use with respect to the KFBU. Recall that the KFBU is simpler to implement as it accounts for the biases without estimating them. In this scenario the UBE attempts to estimate three alignment bias angles, a range bias, and an elevation angle bias for each sensor, in addition to target positions. All of these quantities rely on a known sensor location, thus performance suffers when the location bias is not estimated. In the second and third

scenario the UBE is the best performer of the estimators tested. When removing the elevation angle in the second scenario performance is midway between the KFU and the KFBU. In the third scenario where the location bias is removed the UBE performance is close to the KFU both in $\{u\}$ and $\{s_0\}$. Recall that the UBE needs at least three sensors observing the same targets to estimate the biases. If only two sensors are present and target tracking is performed in $\{u\}$, the UBE should be replaced by the SUBE in cases where the SUBE outperforms the KFBU.

#### 6.3.2.2 Master Sensor Coordinates

If tracking is performed in $\{s_0\}$, and performance in $\{u\}$ is of no importance, the best performing bias estimator is the ABE. Its performance in $\{s_0\}$ is close to the KFU in all simulation scenarios, and it is able to estimate successfully the alignment, location and sensor biases. It is the only estimator which accomplishes this. Note that the performance of the RBE is approximately identical to the ABE despite the fact that it does not estimate the master sensor range and elevation bias. This is because the unestimated master sensor range and elevation biases do not have a noticeable effect on tracking performance. Thus the added complexity of the ABE does not yield better performance with respect to the RBE in the scenarios treated herein. This may explain why tracking in the coordinates of an assumed bias free master sensor is popular in the bias estimation literature. Note that the UBE has slightly better performance in $\{s_0\}$ than in $\{u\}$, suggesting that tracking should be done in $\{s_0\}$ if tracking performance is of primary concern. Using the UBE it is possible to convert positions to $\{u\}$ if necessary, provided that its bias estimation performance is good enough.

## 6.4 Bias Estimation Using the Curvature of the Earth

### 6.4.1 Discussion

A drawback of the UBE in the previous section is that it requires at least three sensors observing the same targets in order to determine the alignment bias successfully in sensor independent coordinates. Section 4.2 investigates a way to relax this requirement by assuming that aircraft flies at constant altitude, speed and heading above the Earth. This motion results in curved motion which enhances the estimability of the alignment bias. The single sensor case is discussed

in Section 4.2, and it is shown that either roll $\alpha$ or pitch $\beta$ become estimable because of the curved motion. For an aircraft flying at constant velocity, none of the alignment bias Euler angles are estimable for a single sensor.

However, determining one of the Euler angles in this manner is not sufficient. Two are required to get a UBE which for two sensors can determine the alignment bias successfully. A way to do this is discussed in Section 4.2.4.1. It involves constraining the allowed aircraft motion even further. The simulated model allows several curved motions, but this can be constrained to a specific curved motion such as the motion on a Great Circle. This constrained motion is expected to allow determination of both roll and pitch for one sensor. Yaw becomes observable when a second sensor is added. This is a topic for further research.

### 6.4.2 Conclusion

It is shown that assuming a curved constant altitude motion makes it possible to estimate either roll $\alpha$ or pitch $\beta$ biases for a single sensor observing a single target. Note that Figure 4.11 shows slow convergence of $\alpha$. In fact the aircraft has traveled 80 km when the position error is close to zero. This is a weakness of the proposed method since it requires a long continuous track for the curved motion to become visible for the estimator. Another challenge for this method is that measurement noise will make it more challenging to observe the curved motion. Measurement noise is not included in the simulation herein since the focus is on determining whether such a method could be used in theory. While the answer to that question may be yes, it seems to be challenging to use this method in practice given the slow convergence and measurement noise issues. It could possibly be used for ships instead of aircraft, since a ship's altitude is known. A 2D radar observing ships may also have an alignment bias as discussed in this thesis.

## 6.5 Aircraft Altitude Prediction Using Mode C Measurements

### 6.5.1 Discussion

Several Bayesian estimators are tuned and tested on the problem using real measurements and comparing estimator predictions with actual measurements. Of

particular interest are Interacting Multiple Model (IMM) estimators, but in this case they perform worse than single model estimators with adjustable process noise, and a standard KF. The IMM estimators using specialized models for level flight are often not able to choose the correct model when the alternative is a constant velocity model. The IMM estimators using constant velocity models with different process noise levels perform better, but they tend to give too much weight to the maneuver model during small maneuvers. The KF based estimators have position estimates closer to the measurements because of their linear measurement model. This model mismatch is apparent in the velocity estimates which are inferior to the nonlinear estimators using the Unscented Kalman Filter (UKF) or the Particle Filter (PF). Improved velocity estimates yield better long term predictions, which are important in conflict alarm systems for Air Traffic Control (ATC).

### 6.5.2 Conclusion

The best overall performers are the estimators with adjustable process noise for maneuver handling, in particluar the UKF based UKFA. The IMM estimators can adjust their weighting of the maneuver model, whereas the latter estimators use either the standard model or the maneuver model based on a threshold. This threshold is not exceeded during small maneuvers, which leads to better performance. The IMM estimators tend to put too much weight on the maneuver model. The theoretically most advanced estimator, the PF, is the worst performer since it depends on a larger process noise covariance to spread its particles in state space. Note that the performance of the KF with adjustable process noise (KFA) is only slightly worse than the UKFA. Since this is a linear estimator with a simpler impementation than the UKF, it is a good alternative.

## 6.6 Suggested Topics for Future Research

### 6.6.1 Alternative Bias Models

The mathematical model of a bias is defined in Definition 3.1 which states that a bias is a constant parameter. There are alternative bias models which could be appropriate, for instance that a bias is a parameter which oscillates about a constant value. This could be done using a first order Markov model.

Sensor scale biases (Lin and Bar-Shalom, 2005) is a type of bias which could

be added to the sensor bias model, thus (3.6) would be replaced with

$$\underline{z}_k = F\left(\underline{g}\right)\underline{p}_k + \underline{b} + \underline{w}_k \tag{6.1a}$$

$$\underline{w}_k \sim \mathcal{N}\left(\underline{0}, R\delta_{kl}\right) \tag{6.1b}$$

$$\underline{b} \sim \mathcal{N}\left(\underline{0}, B\right) \tag{6.1c}$$

$$\underline{g} \sim \mathcal{N}\left(\underline{0}, G\right) \tag{6.1d}$$

where

$$\underline{g} = \begin{bmatrix} g_\rho, & g_\theta, & g_\phi \end{bmatrix}^T \tag{6.2a}$$

$$F\left(\underline{g}\right) = \begin{bmatrix} 1+g_\rho & 0 & 0 \\ 0 & 1+g_\theta & 0 \\ 0 & 0 & 1+g_\phi \end{bmatrix}. \tag{6.2b}$$

Adding scale biases to the sensor bias makes the bias estimation problem more challenging, since there are three more biases to estimate per sensor. An estimability study is required to see how this affects estimability of the alignment, location and sensor biases.

Another type of bias which could be investigated using the measurement models herein is a time bias, a systematic error due to time offsets across sensors. Time bias estimation would require a reference clock. In universal bias estimation the clock would be sensor independent, while in absolute/relative bias estimation the clock of the master sensor would be used as reference.

### 6.6.2 Asynchronous Sensors

In a real world scenario sensors will in general send measurements asynchronously. The results herein are for synchronous sensors. There are several ways of dealing with asynchronous sensors. One way is to manipulate the measurements directly as in Lin and Bar-Shalom (2005). Another is to use biased local sensor tracks or tracklets as measurements for the bias estimator. Tracks from a given target are in general correlated, but there are ways to remove this correlation, for instance by using tracklets, which are essentially uncorrelated tracks. Tracks and tracklets are straightforward to propagate to a common time, hence it is possible to use bias estimation methods for synchronous sensors. For more information on fusion of tracks and tracklets from multiple sensors consult the series of papers Drummond (1997a,b, 1995, 1996, 2002) which discusses how to calculate cross correlations between tracks, taking into account the sensor network communication load of several methods.

### 6.6.3 Performance on Real Measurements

The bias estimation results herein are simulated, but it is vital to test the suggested bias estimators on real measurements. The bias estimator would have to handle asynchronous sensors, as treated in the previous section, and maneuvering targets. Although targets that maneuver a lot could be ignored, targets that maneuver a little should be accounted for, since requiring constant velocity motion could be too restrictive. Targets that move in a coordinated turn are of particular interest as they increase bias observability and estimability as in Nabaa and Bishop (1999).

### 6.6.4 Nonlinear Bias Estimation

Although the measurement models herein are nonlinear, the filter models are obtained using linearization for the EKF and matching of moments up to the fourth order for the UKF. Furthermore both of these methods are recursive, and they rely on the Markov property which says that the future states only depend on the present state. This means that only the most recent estimate is required to predict future states.

To improve bias estimator performance it would be interesting to investigate bias estimators that use the nonlinear models directly, such as the PF. If such estimators are found to be too expensive computationally, it would be interesting to see if one could derive a Best Linear Unbiased Estimator (BLUE) (Li et al., 2003) without using linearization.

A recursive algorithm may be efficient, but more accurate bias estimates may be obtained by using batch methods that operate on a measurement set. Using the measurement models herein one could formulate a nonlinear optimization problem, and solve it using a numerical search or a genetic algorithm. See Kragel et al. (2007) for a nonlinear Least Squares (LS) method which estimates the alignment bias. A simpler alternative would be to use a smoothing algorithm (Gelb, 1974).

### 6.6.5 Bias Estimation for 2D Radars and Mode C Measurements

In ATC 2D radar and Mode C measurements are used to measure the position of airborne targets. The bias estimation models herein should be adapted to perform bias estimation with these sensors. This involves using (5.1) with (3.16) to get a

new model for a sensor measurement in Cartesian coordinates. Note that several types of sensors are used in ATC. The modeling herein may be adapted to radars, Primary Surveillance Radars (PSRs) and Secondary Surveillance Radars (SSRs) for airborne aircraft, and Surface Movement Radars (SMRs) for aircraft close to the ground. The 2D position measurement of a SSR relies on transponder messages from aircraft, where the alignment and location bias discussed herein applies, but the sensor bias does not.

### 6.6.6 Bias Estimation for Extended Targets

The mathematical models herein all assume that the target can be modeled as a point in 3D space. This is a valid assumption when the target is an aircraft situated far away from the radar. However, radars are also used for aircraft surveillance on airports, where the targets are close to the radars. In this case the point assumption no longer holds, and biases are introduced because the radars will report different positions since their beams reflect on different parts of the aircraft body. This kind of problem is also present in harbor surveillance of ships. It would be of great interest to adapt the bias models herein to this scenario, where the radars would typically be 2D radars. If the targets are assumed to be on the ground the target altitude would be known, and in the case of ship surveillance at long range, it might be possible to use the curvature of the Earth to aid the bias estimator.

# Bibliography


Filippo Arrichiello, Gianluca Antonelli, Antonio Pedro Aguiar, and Antonio Pascoal. An observability metric for underwater vehicle localization using range measurements. *Sensors (Switzerland)*, 13(12):16191–16215, November 2013. doi: 10.3390/s131216191.

Y. Bar-Shalom. Mobile radar bias estimation using unknown location targets. In *Proceedings of the Third International Conference on Information Fusion*, pages 3–6, Paris, France, 2000.

Yaakov Bar-Shalom and Xiao-Rong Li. *Multitarget-Multisensor Tracking : Principles and Techniques*. Storrs, Connecticut, USA: YBS, 1995.

Yaakov Bar-Shalom, Xiao Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation*. New York, USA: John Wiley & Sons, Inc., 2001.

David Knox Barton. *Modern Radar System Analysis*. Norwood, Massachusetts, USA: Artech House, 1988.

Niclas Bergman. *Recursive Bayesian Estimation: Navigation and Tracking Applications*. PhD thesis, Linköping University, 1999.

Juan A. Besada, Andrés Soto, Gonzalo De Miguel, and Javier Portillo. Design of an a-SMGCS prototype at barajas airport: Airport surveillance sensors bias estimation. In *2005 7th International Conference on Information Fusion, FUSION*, volume 2, pages 1343–1350, Philadelphia, PA, United states, 2005. doi: 10.1109/ICIF.2005.1592012.

J.A. Besada Portas, J. Garcia Herrero, and G. De Miguel Vela. Radar bias correction based on GPS measurements for ATC applications. *IEE Proceedings: Radar, Sonar and Navigation*, 149(3):137–144, June 2002. ISSN 1350-2395. doi: 10.1049/ip-rsn:20020381.

J.A. Besada Portas, J. Garcia Herrero, and G. De Miguel Vela. New approach to online optimal estimation of multisensor biases. *IEE Proceedings: Radar, Sonar and Navigation*, 151(1):31–40, February 2004. ISSN 1350-2395. doi: 10.1049/ip-rsn:20040116.

Samuel S Blackman and Robert Popoli. *Design and Analysis of Modern Tracking Systems*. Boston, USA: Artech House, 1999.

S.S. Blackman and N.D. Banh. Track association using correction for bias and missing data. In *Proceedings of the SPIE - The International Society for Optical Engineering*, volume 2235, pages 529–39, Orlando, FL, USA, 1994.

R. S. Bucy and K. D. Senne. Digital synthesis of non-linear filters. *Automatica*, 7(3):287–298, May 1971. doi: 10.1016/0005-1098(71)90121-X.

S. J. Davey, B. Cheung, and B. Cheung. Asynchronous sensor registration via expectation maximisation. In *2008 International Conference on Radar*, pages 622–6, Adelaide, SA, Australia, 2008. doi: 10.1109/RADAR.2008.4653997.

E.J. Dela Cruz, A.T. Alouani, T.R. Rice, and W.D. Blair. Estimation of sensor bias in multisensor systems. In *Proceedings. IEEE SOUTHEASTCON '92 (Cat. No.92CH3094-0)*, pages 210–14, Birmingham, AL, USA, 1992. doi: 10.1109/SECON.1992.202338.

O.E. Drummond. A hybrid sensor fusion algorithm architecture and tracklets. In *Proceedings of the SPIE - The International Society for Optical Engineering*, volume 3163, pages 485–502, San Diego, CA, USA, 1997a.

O.E. Drummond. Tracklets and a hybrid fusion with process noise. In *Proceedings of the SPIE - The International Society for Optical Engineering*, volume 3163, pages 512–24, San Diego, CA, USA, 1997b.

Oliver E. Drummond. Feedback in track fusion without process noise. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 2561, pages 369–383, San Diego, CA, USA, July 1995. doi: 10.1117/12.217722.

Oliver E. Drummond. Track fusion with feedback. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 2759, pages 342–360, Orlando, FL, USA, April 1996. doi: 10.1117/12.241196.

Oliver E. Drummond. On track and tracklet fusion filtering. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 4728, pages 176–195, Orlando, FL, United States, April 2002. doi: 10.1117/12.478532.

Zhansheng Duan, Chongzhao Han, and X. Rong Li. Comments on "unbiased converted measurements for tracking". *IEEE Transactions on Aerospace and Electronic Systems*, 40(4):1374–1377, October 2004. ISSN 0018-9251. doi: 10.1109/TAES.2004.1386889.

Zhansheng Duan, V. P. Jilkov, and X. R. Li. State estimation with quantized measurements: approximate MMSE approach. In *2008 11th International Conference on Information Fusion (FUSION 2008)*, Cologne, Germany, 2008a. doi: 10.1109/ICIF.2008.4632328.

Zhansheng Duan, Vesselin P. Jilkov, and X. Rong Li. Posterior cramer-rao bounds for state estimation with quantized measurement. In *40th Southeastern Symposium on System Theory, SSST*, pages 376–380, New Orleans, LA, United states, 2008b. doi: 10.1109/SSST.2008.4480258.

P.F. Easthope. Observability of sensor biases using multiple track reports. *Proceedings of SPIE - The International Society for Optical Engineering*, 3809: 332–342, Jul 1999. ISSN 0277-786X. doi: 10.1117/12.364032.

P.F. Easthope. Prototype system for multi-sensor tracking with sensor bias correction. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 4048, pages 390–401, Orlando, FL, USA, Apr 2000. doi: 10.1117/12.391993.

Stefano Fortunati, Alfonso Farina, Fulvio Gini, Antonio Graziano, Maria S. Greco, and Sofia Giompapa. Least squares estimation and cramér-rao type lower bounds for relative sensor registration process. 59(3):1075–1087, March 2011. ISSN 1053587X. doi: 10.1109/TSP.2010.2097258.

Stefano Fortunati, Fulvio Gini, Maria S. Greco, Alfonso Farina, Antonio Graziano, and Sofia Giompapa. Least squares estimation and hybrid cramér-rao lower bound for absolute sensor registration. In *2012 Tyrrhenian Workshop on Advances in Radar and Remote Sensing: From Earth Observation to Homeland Security, TyWRRS 2012*, pages 30–35, Naples, Italy, 2012a. doi: 10.1109/TyWRRS.2012.6381098.

Stefano Fortunati, Fulvio Gini, Maria S. Greco, Alfonso Farina, Antonio Graziano, and Sofia Giompapa. An EM-based approach to the relative sensor registration in multi-target scenarios. In *2012 IEEE Radar Conference: Ubiquitous Radar, RADARCON 2012*, pages 0602–0607, Atlanta, GA, United states, 2012b. doi: 10.1109/RADAR.2012.6212211.

Stefano Fortunati, Fulvio Gini, Maria S. Greco, Alfonso Farina, Antonio Graziano, and Sofia Giompapa. On the identifiability problem in the presence of random nuisance parameters. 92(10):2545–2551, October 2012c. ISSN 01651684. doi: 10.1016/j.sigpro.2012.04.004.

Brita Gade. Bias estimator. Appendix B in "NASAMS IRF - Data Fusion Study Phase 3B Appendices" (Restricted) Doc no. 00TN60184534, Kongsberg Defence & Aerospace AS, Norway, May 2004.

Arthur Gelb. *Applied Optimal Estimation*. Cambridge, Massachusetts, USA: The M.I.T. Press, 1974.

R.E. Helmick and T.R. Rice. Removal of alignment errors in an integrated system of two 3-d sensors. *IEEE Transactions on Aerospace and Electronic Systems*, 29(4):1333–43, October 1993. ISSN 0018-9251. doi: 10.1109/7.259537.

R.E. Helmick, J.E. Conte, S.A. Hoffman, and W.D. Blair. One-step fixed-lag IMM smoothing for alignment of asynchronous sensors. In *Proceedings of the SPIE - The International Society for Optical Engineering*, volume 2235, pages 507–18, Orlando, FL, USA, 1994.

Shawn M. Herman and Aubrey B. Poore. Nonlinear least-squares estimation for sensor and navigation biases. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 6236, Kissimmee, FL, United States, April 2006. doi: 10.1117/12.673524.

A. S. Hodel and J. Y. Hung. A state estimator with reduced sensitivity to sensor quantization. In *IECON'03. 29th Annual Conference of the IEEE Industrial Electronics Society*, pages 586–90, Roanoke, VA, USA, 2003. doi: 10.1109/IECON.2003.1280045.

Simon Julier and Jeffrey K. Uhlmann. A general method for approximating nonlinear transformations of probability distributions. Technical report, Robotics Research Group, Department of Engineering Science, University of Oxford, 1996. URL `http://www.sanbi.ac.za/~konrad/papers/Unscented.ps.gz`. Thorough article on the Unscented filter.

Simon J. Julier. The scaled unscented transformation. In *Proceedings of the American Control Conference*, volume 6, pages 4555–4559, Anchorage, AK, United States, May 2002. doi: 10.1109/ACC.2002.1025369.

S.J. Julier and J.K. Uhlmann. Unscented filtering and nonlinear estimation. In *Proceedings of the IEEE*, volume 92, pages 401–422, 2004. doi: 10.1109/JPROC.2003.823141.

S.J. Julier, J.K. Uhlmann, and H.F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of the American Control Conference, Seattle, Washington, USA*, volume vol.3, pages 1628–32, Dept. of Eng. Sci., Oxford Univ., UK, 1995. American Autom Control Council.

Steven M. Kay. *Fundamentals of Signal Processing*, volume 1. Upper Saddle River, New Jersey, USA: Prentice Hall, 1993.

Yoon Hak Kim. Distributed estimation based on quantized data. 8(10):699–704, 2011. doi: 10.1587/elex.8.699.

Y. Kosuge and T. Okada. Statistical analysis for radar bias error estimation in a data fusion system of 3-dimensional radars. In *Proceedings of 2000 IEEE International Conference on Industrial Electronics, Control and Instrumentation*, pages 2001–6, Nagoya, Japan, 2000. doi: 10.1109/IECON.2000.972583.

Bret D. Kragel, Scott Danford, Shawn M. Herman, and Aubrey B. Poore. Bias estimation using targets of opportunity. In *Proceedings of SPIE - The International Society for Optical Engineering*, 2007. doi: 10.1117/12.738161.

X. Rong Li and Zhanlue Zhao. Measures of performance for evaluation of estimators and filters. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 4473, pages 530–541, San Diego, CA, United States, July 2001. doi: 10.1117/12.492751.

X.R. Li, Yunmin Zhu, Jie Wang, and Chongzhao Han. Optimal linear estimation fusion - part i: Unified fusion rules. *IEEE Transactions on Information Theory*, 49(9):2192–208, September 2003. ISSN 0018-9448. doi: 10.1109/TIT.2003.815774.

Zhenhua Li and Henry Leung. An expectation maximization based simultaneous registration and fusion algorithm for radar networks. In *Canadian Conference on Electrical and Computer Engineering*, 2006.

Xiangdong Lin and Yaakov Bar-Shalom. Multisensor target tracking performance with bias compensation. volume 5913. SPIE Proceedings of Signal and Data Processing of Small Targets, 2005. doi: 10.1117/12.613771.

Xiangdong Lin, Y. Bar-Shalom, and T. Kirubarajan. Exact multisensor dynamic bias estimation with local tracks. *IEEE Transactions on Aerospace and Electronic Systems*, 40(2):576–590, April 2004a. ISSN 0018-9251. doi: 10.1109/TAES.2004.1310006.

Xiangdong Lin, Thiagalingam Kirubarajan, and Yaakov Bar-Shalom. Multisensor bias estimation with local tracks without a priori association. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 5204, pages 334–345, San Diego, CA, United States, August 2004b. doi: 10.1117/12.503715.

Xiangdong Lin, Yaakov Bar-Shalom, and Thiagalingam Kirubarajan. Multisensor-multitarget bias estimation for general asynchronous sensors. *IEEE Transactions on Aerospace and Electronic Systems*, 41(3): 899–921, July 2005. ISSN 0018-9251. doi: 10.1109/TAES.2005.1541438.

Mo Longbin, Song Xiaoquan, Zhou Yiyu, Sun Zhong Kang, and Y. Bar-Shalom. Unbiased converted measurements for tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 34(3):1023–7, July 1998. ISSN 0018-9251. doi: 10.1109/7.705921.

Takashi Matsuzaki, Hiroshi Kameda, Junichi Uchida, and Fumiya Hiroshima. A study of bias error estimation method by KGBE. In *2010 IEEE International Conference on Control Applications, CCA 2010*, pages 452–457, Yokohama, Japan, 2010. doi: 10.1109/CCA.2010.5611306.

Peter S. Maybeck. *Stochastic Models, Estimation, and Control*, volume 1. Arlington, Virginia, USA: Navtech Book & Software Store, 1994.

N. Nabaa and R.H. Bishop. Solution to a multisensor tracking problem with sensor registration errors. *IEEE Transactions on Aerospace and Electronic Systems*, 35(1):354–363, 1999. ISSN 0018-9251. doi: 10.1109/7.745706.

R.Y. Novoselov, S.M. Herman, S.M. Gadaleta, and A.B. Poore. Mitigating the effects of residual biases with schmidt-kalman filtering. In *2005 7th International Conference on Information Fusion (FUSION)*, Philadelphia, PA, USA, 2005.

Takamitsu Okada, Atsushi Okamura, and Yoshio Kosuge. Position estimation of three-dimensional radars with bias errors. *Electronics and Communications in Japan, Part I: Communications (English translation of Denshi Tsushin Gakkai Ronbunshi)*, 87(9):64–74, September 2004. ISSN 8756-6621. doi: 10.1002/ecja.10159.

Nickens N. Okello and Graham W. Pulford. Simultaneous registration and tracking for multiple radars with cluttered measurements. pages 60–63, Cooperative Research Cent, Aust, 1996. IEEE, Los Alamitos, CA, USA. doi: 10.1109/SSAP.1996.534820.

N.N. Okello and S. Challa. Joint sensor registration and track-to-track fusion for distributed trackers. *Aerospace and Electronic Systems, IEEE Transactions on*, 40(3):808–823, 2004. doi: 10.1109/TAES.2004.1337456.

Jiang-Huai Pan, Jia-Zhou He, and Yan-Li Li. New registration algorithm of sensor in ECEF coordinate system. In *2008 5th International Multi-Conference on Systems, Signals and Devices, SSD'08*, Amman, Jordan, 2008. doi: 10.1109/SSD.2008.4632792.

Athanasios Papoulis and S. Unnikrishna Pillai. *Probability, Random Variables and Stochastic Processes*. Singapore: McGraw-Hill, Inc., fourth edition edition, 2002.

Yongqing Qi, Zhongliang Jing, and Shiqiang Hu. General solution for asynchronous sensors bias estimation. In *11th International Conference on Information Fusion, FUSION 2008*, Cologne, Germany, 2008a. doi: 10.1109/ICIF.2008.4632219.

Yongqing Qi, Zhongliang Jing, Shiqiang Hu, and Haitao Zhao. New method for dynamic bias estimation: Gaussian mean shift registration. 47(2):026401–1, February 2008b. ISSN 0091-3286. doi: 10.1117/1.2841054.

A. Rafati, B. Moshiri, K. Salahshoor, and M. Tabatabaei Pour. Asynchronous sensor bias estimation in multisensor-multitarget systems. In *2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, MFI*, pages 402–407, Heidelberg, Germany, 2006. doi: 10.1109/MFI.2006.265654.

Amir Rafati, Behzad Moshiri, and Javad Rezaei. A new algorithm for general asynchronous sensor bias estimation in multisensor-multitarget systems. In *FUSION 2007 - 2007 10th International Conference on Information Fusion*, Quebec, QC, Canada, 2007. doi: 10.1109/ICIF.2007.4408191.

Stephen Rhodes. Real time detection and compensation of registration errors in a network of radars. Number 490, pages 11–16. Institution of Electrical Engineers, 2002. doi: 10.1109/RADAR.2002.1174644.

Branko Ristic, Sanjeev Arulampalam, and Neil Gordon. *Beyond the Kalman Filter*. Boston, USA: Artech House, 2004.

Bret Kragel Scott Danford and Aubrey Poore. Joint map bias estimation and data association: Algorithms. In *Proceedings of SPIE - The International Society for Optical Engineering*, 2007a. doi: 10.1117/12.735202.

Bret Kragel Scott Danford and Aubrey Poore. Joint map bias estimation and data association: Simulations. In *Proceedings of SPIE - The International Society for Optical Engineering*, 2007b. doi: 10.1117/12.735225.

Torsten Soderstrom. *Discrete-time Stochastic Systems*. London, England: Springer Verlag, 2002.

J.J. Sudano. An exact conversion from an earth-centered coordinate system to latitude, longitude and altitude. In *Proceedings of the IEEE 1997 National Aerospace and Electronics Conference. NAECON 1997*, pages 646–50, Dayton, OH, USA, 1997. doi: 10.1109/NAECON.1997.622711.

Egils Sviestins. Bias estimation for multisensor tracking. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 4048, pages 402–413, Orlando, FL, USA, Apr 2000. doi: 10.1117/12.391994.

Egils Sviestins and Torbjorn Wigren. Nonlinear techniques for Mode C climb/descent rate estimation in ATC systems. *IEEE Transactions on Control Systems Technology*, 9(1):163–174, 2001. ISSN 1063-6536. doi: 10.1109/87.896757.

J.H. Taylor. The cramer-rao estimation error lower bound computation for deterministic nonlinear systems. *IEEE Transactions on Automatic Control*, 24(2): 343–4, April 1979. ISSN 0018-9286. doi: 10.1109/TAC.1979.1101979.

Technical Staff. Operational requirements document for eatchip phase III added functions. Technical report, Eurocontrol, 1999.

Petr Tichavsky, Carlos H. Muravchik, and Arye Nehorai. Posterior cramer-rao bounds for discrete-time nonlinear filtering. *IEEE Transactions on Signal Processing*, 46(5):1386–1389, May 1998. ISSN 1053-587X. doi: 10.1109/78.668800.

Morten P. Topland and Oddvar Hallingstad. Aircraft altitude prediction using mode c measurements. In *7th AIAA Aviation Technology, Integration and Operations Conference (ATIO), Belfast, Northern Ireland*, September 2007.

Morten P. Topland and Oddvar Hallingstad. Estimability of sensor alignment biases using the static cramer rao lower bound. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008.

Morten P. Topland, Oddvar Hallingstad, Abhijit Sinha, and Thiagalingam Kirubarajan. Bias estimation for distributed radars in 3D. In *IEEE Aerospace Conference*, 2007.

R. Van Der Merwe and E.A. Wan. The square-root unscented kalman filter for state and parameter-estimation. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, volume 6, pages 3461–3464, Salt Lake, UT, May 2001. doi: 10.1109/ICASSP.2001.940586.

S. B. Vardeman and Chiang-Sheng Lee. Likelihood-based statistical estimation from quantized data. 54(1):409–14, February 2005. ISSN 0018-9456. doi: 10.1109/TIM.2004.838912.

E.A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of Symposium on Adaptive Systems for Signal Processing Communications and Control*, pages 153–8, Lake Louise, Alta., Canada, 2000. doi: 10.1109/ASSPCC.2000.882463.

G.A. Watson and T.R. Rice. Sensor bias estimation and compensation for improved track correlation. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 4365, pages 112–125, Orlando, FL, Apr 2001. doi: 10.1117/12.438039.

Gregory A. Watson and Theodore R. Rice. Assessment metrics for absolute sensor alignment. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 4714, pages 129–142, Orlando, FL, United States, April 2002a. doi: 10.1117/12.472587.

Gregory A. Watson and Theodore R. Rice. Sensor alignment and compensation for composite tracking. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 4728, pages 354–367, Orlando, FL, United States, April 2002b. doi: 10.1117/12.478517.

Gregory A. Watson, Denis H. McCabe, and Theodore R. Rice. Multisensor-multisite composite tracking in the presence of sensor residual bias. *Proceedings of SPIE - The International Society for Optical Engineering*, 3809:370–381, Jul 1999. ISSN 0277-786X. doi: 10.1117/12.364035.

Eric W. Weisstein. Great circle. From MathWorld – A Wolfram Web Resource. http://mathworld.wolfram.com/GreatCircle.html, 2002. URL `http://mathworld.wolfram.com/GreatCircle.html`. Accessed 26.10.2006.

Eric W. Weisstein. Similarity transformation. From MathWorld – A Wolfram Web Resource, 2013. URL `http://mathworld.wolfram.com/SimilarityTransformation.html`. [Online; accessed 29-July-2013].

Man Ying, Jiang-Yuan Huang, and Zhi-Hui Yu. 3d asynchronous multisensor target tracking performance with bias compensation. In *2010 2nd International Conference on Computer Engineering and Technology, ICCET 2010*, volume 5, Chengdu, China, 2010. doi: 10.1109/ICCET.2010.5486094.

Zhanlue Zhao, X.R. Li, V.P. Jilkov, and Yunmin Zhu. Optimal linear unbiased filtering with polar measurements for target tracking. In *Proceedings of Fifth International Conference on Information Fusion*, pages 1527–34, Annapolis, MD, USA, 2002. doi: 10.1109/ICIF.2002.1020998.

Zhanlue Zhao, X. Rong Li, and Vesselin P. Jilkov. Best linear unbiased filtering for target tracking with spherical measurements. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 5204, pages 418–428, San Diego, CA, United States, August 2004a. doi: 10.1117/12.511168.

Zhanlue Zhao, X. Rong Li, and Vesselin P. Jilkov. Best linear unbiased filtering with nonlinear measurements for target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 40(4):1324–1336, 2004b. ISSN 0018-9251. doi: 10.1109/TAES.2004.1386884.

Yifeng Zhou, H. Leung, and P.C. Yip. An exact maximum likelihood registration algorithm for data fusion. *IEEE Transactions on Signal Processing*, 45(6): 1560–73, June 1997. ISSN 1053-587X. doi: 10.1109/78.599998.

Yifeng Zhou, Henry Leung, and Martin Blanchette. Sensor alignment with earth-centered earth-fixed (ECEF) coordinate system. *IEEE Transactions on Aerospace and Electronic Systems*, 35(2):410–418, April 1999. ISSN 0018-9251. doi: 10.1109/7.766925.

Peter H. Zipfel. *Modeling and Simulation of Aerospace Vehicle Dynamics*. AIAA Education Series. American Institute of Aeronautics and Astronautics, Reston, VA, second edition edition, 2007.

# A

# Cramer-Rao Lower Bound

## A.1 Proof of the Static CRLB

**Proof:** To prove Theorem 2.1 we need the relation

$$\frac{\partial p\left(\underline{z}:\underline{x}\right)}{\partial \underline{x}} = \frac{\partial \ln p\left(\underline{z}:\underline{x}\right)}{\partial \underline{x}} p\left(\underline{z}:\underline{x}\right) \tag{A.1}$$

which is proved by looking at differentiation with respect to element $x_i$ where $i \in \{1, 2, \ldots, n_x\}$, and using the chain rule,

$$\begin{aligned}\frac{\partial \ln p\left(\underline{z}:\underline{x}\right)}{\partial x_i} p\left(\underline{z}:\underline{x}\right) &= \frac{\partial \ln p\left(\underline{z}:\underline{x}\right)}{\partial p\left(\underline{z}:\underline{x}\right)} \frac{\partial p\left(\underline{z}:\underline{x}\right)}{\partial x_i} p\left(\underline{z}:\underline{x}\right) \\ &= \frac{1}{p\left(\underline{z}:\underline{x}\right)} \frac{\partial p\left(\underline{z}:\underline{x}\right)}{\partial x_i} p\left(\underline{z}:\underline{x}\right) = \frac{\partial p\left(\underline{z}:\underline{x}\right)}{\partial x_i}. \end{aligned} \tag{A.2}$$

We start by proving (2.2) using that the integral of a probability density function (pdf) equals 1.

$$\int_{-\infty}^{\infty} p\left(\underline{z}:\underline{x}\right) d\underline{z} = 1$$

$$\int_{-\infty}^{\infty} \frac{\partial}{\partial x_i} p\left(\underline{z}:\underline{x}\right) d\underline{z} = \int_{-\infty}^{\infty} \frac{\partial \ln p\left(\underline{z}:\underline{x}\right)}{\partial x_i} p\left(\underline{z}:\underline{x}\right) d\underline{z} = 0$$

$$\frac{\partial}{\partial x_j} \int_{-\infty}^{\infty} \frac{\partial \ln p\left(\underline{z}:\underline{x}\right)}{\partial x_i} p\left(\underline{z}:\underline{x}\right) d\underline{z} = 0$$

$$\int_{-\infty}^{\infty} \frac{\partial^2 \ln p\left(\underline{z}:\underline{x}\right)}{\partial x_i \partial x_j} p\left(\underline{z}:\underline{x}\right) d\underline{z} + \int_{-\infty}^{\infty} \frac{\partial \ln p\left(\underline{z}:\underline{x}\right)}{\partial x_i} \frac{\partial p\left(\underline{z}:\underline{x}\right)}{\partial x_j} d\underline{z} = 0$$

$$\int_{-\infty}^{\infty} \frac{\partial^2 \ln p\left(\underline{z}:\underline{x}\right)}{\partial x_i \partial x_j} p\left(\underline{z}:\underline{x}\right) d\underline{z} + \int_{-\infty}^{\infty} \frac{\partial \ln p\left(\underline{z}:\underline{x}\right)}{\partial x_i} \frac{\partial \ln p\left(\underline{z}:\underline{x}\right)}{\partial x_j} p\left(\underline{z}:\underline{x}\right) d\underline{z} = 0$$

$$E\left\{\frac{\partial^2 \ln p\left(\underline{z}:\underline{x}\right)}{\partial x_i \partial x_j}\right\} + E\left\{\frac{\partial \ln p\left(\underline{z}:\underline{x}\right)}{\partial x_i} \frac{\partial \ln p\left(\underline{z}:\underline{x}\right)}{\partial x_j}\right\} = 0. \tag{A.3}$$

The Fisher Information Matrix (FIM) is by definition (Kay, 1993)

$$J = -E\left\{\frac{\partial^2 \ln p\left(\underline{z}:\underline{x}\right)}{\partial x_i \partial x_j}\right\} = -E\left\{\frac{\partial^2 \ln p\left(\underline{z}:\underline{x}\right)}{\partial \underline{x} \partial \underline{x}^T}\right\}, \tag{A.4}$$

thus (2.2) is proved. Proving (2.1) requires the use of Schwarz's inequality

$$\left|\langle f_1, f_2\rangle\right|^2 \leq \langle f_1, f_1\rangle \langle f_2, f_2\rangle \tag{A.5}$$

where $f_1$ and $f_2$ are real integrable functions, and the inner product is defined by

$$\langle f_1, f_2\rangle = \int_{-\infty}^{\infty} f_1\left(\underline{z}\right) f_2\left(\underline{z}\right) d\underline{z}. \tag{A.6}$$

We prove (2.1) starting with the fact that the estimator is unbiased. The estimate is denoted $\underline{\hat{x}}\left(\underline{z}\right)$.

$$\begin{aligned}
E\left\{\underline{\hat{x}}\left(\underline{z}\right)-\underline{x}\right\}&=0\\
\int_{-\infty}^{\infty}\left(\underline{\hat{x}}\left(\underline{z}\right)-\underline{x}\right)p\left(\underline{z}:\underline{x}\right)d\underline{z}&=0\\
\frac{\partial}{\partial\underline{x}^T}\int_{-\infty}^{\infty}\left(\underline{\hat{x}}\left(\underline{z}\right)-\underline{x}\right)p\left(\underline{z}:\underline{x}\right)d\underline{z}&=0\\
-\int_{-\infty}^{\infty}Ip\left(\underline{z}:\underline{x}\right)d\underline{z}+\int_{-\infty}^{\infty}\left(\underline{\hat{x}}\left(\underline{z}\right)-\underline{x}\right)\frac{\partial p\left(\underline{z}:\underline{x}\right)}{\partial\underline{x}^T}d\underline{z}&=0\\
\int_{-\infty}^{\infty}\left(\underline{\hat{x}}\left(\underline{z}\right)-\underline{x}\right)\frac{\partial p\left(\underline{z}:\underline{x}\right)}{\partial\underline{x}^T}d\underline{z}&=I\\
\int_{-\infty}^{\infty}\left(\underline{\hat{x}}\left(\underline{z}\right)-\underline{x}\right)\frac{\partial \ln p\left(\underline{z}:\underline{x}\right)}{\partial\underline{x}^T}p\left(\underline{z}:\underline{x}\right)d\underline{z}&=I
\end{aligned}\tag{A.7}$$

To use (A.5) we need real scalar functions inside the integral, and not vector functions. Defining two arbitrary vectors $\underline{a}$ and $\underline{b}$, we multiply with $\underline{a}^T$ from the left and $\underline{b}$ from the right.

$$\int_{-\infty}^{\infty}\underbrace{\underline{a}^T\left(\underline{\hat{x}}\left(\underline{z}\right)-\underline{x}\right)\sqrt{p\left(\underline{z}:\underline{x}\right)}}_{f_1(\underline{z})}\underbrace{\frac{\partial \ln p\left(\underline{z}:\underline{x}\right)}{\partial\underline{x}^T}\underline{b}\sqrt{p\left(\underline{z}:\underline{x}\right)}}_{f_2(\underline{z})}d\underline{z}=\underline{a}^T\underline{b}\tag{A.8}$$

Using (A.5) we get

$$\begin{aligned}
\left|\underline{a}^T\underline{b}\right|^2\leq&\int_{-\infty}^{\infty}\underline{a}^T\left(\underline{\hat{x}}\left(\underline{z}\right)-\underline{x}\right)\left(\underline{\hat{x}}\left(\underline{z}\right)-\underline{x}\right)^T\underline{a}\,p\left(\underline{z}:\underline{x}\right)d\underline{z}\\
&\int_{-\infty}^{\infty}\underline{b}^T\frac{\partial \ln p\left(\underline{z}:\underline{x}\right)}{\partial\underline{x}}\frac{\partial \ln p\left(\underline{z}:\underline{x}\right)}{\partial\underline{x}^T}\underline{b}\,p\left(\underline{z}:\underline{x}\right)d\underline{z}.
\end{aligned}\tag{A.9}$$

Assuming that neither $\underline{a}$ nor $\underline{b}$ depend on $\underline{z}$ we get

$$\left|\underline{a}^T\underline{b}\right|^2\leq\underline{a}^TP\underline{a}\,\underline{b}^TJ\underline{b}\tag{A.10}$$

where we recognize the true covariance matrix $P$ of the estimate and the FIM $J$. In order to prove (2.1) we need to find suitable expressions for $\underline{a}$ and $\underline{b}$. □

# B

# Measurement Covariance

## B.1 Measurement Covariance Conversion from Spherical to Cartesian Coordinates

As discussed in Bar-Shalom and Li (1995) it is desirable to track targets in Cartesian coordinates, since the modeling of target motion is practical in these coordinates. If the measurement is in different coordinates like polar or spherical coordinates, the measurement and its covariance can be converted to Cartesian coordinates. The covariance conversion from spherical to Cartesian coordinates follows. The calculation is analogous to the covariance conversion from polar to Cartesian coordinates in Bar-Shalom and Li (1995). The conversion from polar to Cartesian coordinates is also investigated in Longbin et al. (1998), Duan et al. (2004), and Zhao et al. (2002, 2004a,b). Conversion from spherical coordinates to Cartesian coordinates is given in (2.27). The measurement error

$$\underline{\epsilon} = \underline{z} - \underline{x} \tag{B.1}$$

is used to calculate the covariance conversion, where $\underline{z}$ is the measurement and $\underline{x}$ is the true state vector. We will now find expressions for the measurement errors of $x$, $y$ and $z$, and calculate the associated covariance. These expressions are based on linearization about the measurements when the angle errors are assumed small. Note that this linearization may be done about estimates instead of measurements. The measurement error $x_\epsilon$ is

$$x_\epsilon = x_z - x = \rho_z \cos\theta_z \cos\phi_z - \rho \cos\theta \cos\phi \tag{B.2}$$

where $x_\epsilon$, $x_z$ and $x$ are the $x$ components of $\underline{\epsilon}$, $\underline{z}$ and $\underline{x}$ respectively. The linearization leads to $x_\epsilon =$ Term 1 + Term 2 + . . ..
Term 1:

$$(\rho_z - \rho)\cos\theta_z \cos\phi_z = \rho_\epsilon \cos\theta_z \cos\phi_z \tag{B.3}$$

Term 2:

$$\rho_z \cos\phi_z (\cos\theta_z - \cos\theta) = -\theta_\epsilon \rho_z \sin\theta_z \cos\phi_z \tag{B.4}$$

Term 3:

$$\rho_z \cos\theta_z (\cos\phi_z - \cos\phi) = -\phi_\epsilon \rho_z \cos\theta_z \sin\phi_z \tag{B.5}$$

The measurement error $y_\epsilon$ is

$$y_\epsilon = y_z - y = \rho_z \sin\theta_z \cos\phi_z - \rho \sin\theta \cos\phi. \tag{B.6}$$

Term 1:

$$(\rho_z - \rho)\sin\theta_z \cos\phi_z = \rho_\epsilon \sin\theta_z \cos\phi_z \tag{B.7}$$

Term 2:

$$\rho_z \cos\phi_z (\sin\theta_z - \sin\theta) = \theta_\epsilon \rho_z \cos\theta_z \cos\phi_z \tag{B.8}$$

Term 3:

$$\rho_z \sin\theta_z (\cos\phi_z - \cos\phi) = -\phi_\epsilon \rho_z \sin\theta_z \sin\phi_z \tag{B.9}$$

The measurement error $z_\epsilon$ is

$$z_\epsilon = z_z - z = \rho_z \sin\phi_z - \rho \sin\phi. \tag{B.10}$$

The linearized measurement errors become:

$$\begin{aligned} x_\epsilon &\approx \rho_\epsilon \cos\theta_z \cos\phi_z - \theta_\epsilon \rho_z \sin\theta_z \cos\phi_z - \phi_\epsilon \rho_z \cos\theta_z \sin\phi_z \\ y_\epsilon &\approx \rho_\epsilon \sin\theta_z \cos\phi_z + \theta_\epsilon \rho_z \cos\theta_z \cos\phi_z - \phi_\epsilon \rho_z \sin\theta_z \sin\phi_z \\ z_\epsilon &\approx \rho_\epsilon \sin\phi_z + \phi_\epsilon \rho_z \cos\phi_z \end{aligned} \tag{B.11}$$

Assuming

$$E\{x_\epsilon\} = E\{y_\epsilon\} = E\{z_\epsilon\} = 0, \tag{B.12}$$

we calculate the elements of the covariance matrix.

$$\begin{aligned} R_{11} &= E\left\{x_\epsilon^2\right\} - \overbrace{E\left\{x_\epsilon\right\}^2}^{0} \\ &= E\Big\{\rho_\epsilon^2 \cos^2\theta_z \cos^2\phi_z - 2\rho_\epsilon\theta_\epsilon\rho_z \cos\theta_z \sin\theta_z \cos^2\phi_z \\ &- 2\rho_\epsilon\phi_\epsilon\rho_z \cos^2\theta_z \cos\phi_z \sin\phi_z + \theta_\epsilon^2\rho_z^2 \sin^2\theta_z \cos^2\phi_z \\ &+ 2\theta_\epsilon\phi_\epsilon\rho_z \sin\theta_z \cos\theta_z \cos\phi_z \sin\phi_z + \phi_\epsilon^2\rho_z^2 \cos^2\theta_z \sin^2\phi_z\Big\}. \end{aligned} \tag{B.13}$$

Using

$$\begin{aligned} E\left\{\tilde{(\cdot)}\tilde{(\cdot)}\right\} &= \sigma_{(\cdot)}^2 \\ E\left\{\rho_\epsilon\theta_\epsilon\right\} = E\left\{\theta_\epsilon\phi_\epsilon\right\} &= E\left\{\rho_\epsilon\phi_\epsilon\right\} = 0 \end{aligned} \tag{B.14}$$

we get the diagonal elements:

$$R_{11} = \sigma_\rho^2 \cos^2\theta_z \cos^2\phi_z + \sigma_\theta^2\rho_z^2 \sin^2\theta_z \cos^2\phi_z + \sigma_\phi^2\rho_z^2 \cos^2\theta_z \sin^2\phi_z \tag{B.15a}$$

$$R_{22} = \sigma_\rho^2 \sin^2\theta_z \cos^2\phi_z + \sigma_\theta^2\rho_z^2 \cos^2\theta_z \cos^2\phi_z + \sigma_\phi^2\rho_z^2 \sin^2\theta_z \sin^2\phi_z \tag{B.15b}$$

$$R_{33} = \sigma_\rho^2 \sin^2\phi_z + \sigma_\phi^2\rho_z^2 \cos^2\phi_z \tag{B.15c}$$

The off-diagonal elements become:

$$\begin{aligned} R_{12} &= E\left\{x_\epsilon y_\epsilon\right\} - \overbrace{E\left\{x_\epsilon\right\} E\left\{y_\epsilon\right\}}^{0} \\ &= \sigma_\rho^2 \cos\theta_z \sin\theta_z \cos^2\phi_z - \sigma_\theta^2\rho_z^2 \sin\theta_z \cos\theta_z \cos^2\phi_z + \sigma_\phi^2\rho_z^2 \cos\theta_z \sin\theta_z \sin^2\phi_z \end{aligned} \tag{B.16a}$$

$$R_{23} = \sigma_\rho^2 \sin\theta_z \cos\phi_z \sin\phi_z - \sigma_\phi^2\rho_z^2 \sin\theta_z \cos\phi_z \sin\phi_z \tag{B.16b}$$

$$R_{13} = \sigma_\rho^2 \cos\theta_z \cos\phi_z \sin\phi_z - \sigma_\phi^2\rho_z^2 \cos\theta_z \cos\phi_z \sin\phi_z \tag{B.16c}$$

# C

# Filtering Algorithms

## C.1 The Particle Filter

While the Kalman Filter (KF) and the Extended Kalman Filter (EKF) estimate first and second order central moments of a stochastic variable, the Particle Filter (PF) is a filtering algorithm that estimate the entire probability density function (pdf) of a stochastic variable. It can be used for any nonlinear system, but it is computationally expensive. For more details on the PF consult Ristic et al. (2004).

### C.1.1 General Bayesian Filtering

A general discrete time nonlinear system with the state vector $\underline{x}_k$, the input vector $\underline{u}_k$, the process noise vector $\underline{v}_k$, the measurement noise vector $\underline{w}_k$, and the measurement vector $\underline{z}_k$, is given by

$$\underline{x}_{k+1} = \underline{f}_k\left(\underline{x}_k, \underline{u}_k, \underline{v}_k\right) \tag{C.1a}$$

$$\underline{z}_k = \underline{h}_k\left(\underline{x}_k, \underline{w}_k\right) \tag{C.1b}$$

where

$$\underline{x}_0 \sim p_x\left(\underline{x}_0\right), \quad \underline{v}_k \sim p_v\left(\underline{v}_k\right), \quad \underline{w}_k \sim p_w\left(\underline{w}_k\right), \tag{C.2}$$

and $\underline{x}_0$, $\underline{v}_k$ and $\underline{w}_k$ are statistically independent. A general filtering solution exist for the system described by (C.1) and (C.2), and it is given by the general

Bayesian filtering equations

$$p\left(\underline{x}_k \mid Z_{k-1}\right) = \int p\left(\underline{x}_k \mid \underline{x}_{k-1}\right) p\left(\underline{x}_{k-1} \mid Z_{k-1}\right) d\underline{x}_{k-1} \tag{C.3a}$$

$$p\left(\underline{x}_k \mid Z_k\right) = \frac{p\left(\underline{z}_k \mid \underline{x}_k\right) p\left(\underline{x}_k \mid Z_{k-1}\right)}{\int p\left(\underline{z}_k \mid \underline{x}_k\right) p\left(\underline{x}_k \mid Z_{k-1}\right) d\underline{x}_k} \tag{C.3b}$$

which return the conditional probability density function (cpdf) $p\left(\underline{x}_k \mid Z_k\right)$ given the prior cpdf $p\left(\underline{x}_{k-1} \mid Z_{k-1}\right)$, the process cpdf $p\left(\underline{x}_k \mid \underline{x}_{k-1}\right)$, the likelihood function $p\left(\underline{z}_k \mid \underline{x}_k\right)$, the initial pdf $p\left(\underline{x}_0\right)$ and the measurement sequence $Z_k = \{\underline{z}_0, \underline{z}_1, \ldots, \underline{z}_k\}$. In general the integrals in (C.3) can not be solved. Hence they need to be approximated. Popular approximations are modeling the pdfs in a grid with finite resolution (Bucy and Senne, 1971), or using a cloud of particles. These implementations all suffer from the *curse of dimensionality*, which means that the computational load increases exponentially as the state vector grows in dimension.

### C.1.2 The Particle Cloud Approximation

When using the particle cloud approximation the pdfs are approximated using particles that are sampled from the underlying pdf. This approximation yields

$$p\left(\underline{x}_k \mid Z_{k-1}\right) \approx \frac{1}{N} \sum_{i=1}^{N} \delta\left(\underline{x}_k - \underline{x}_k^i\right) \tag{C.4}$$

where the cloud of $N$ particles $\{\underline{x}_k^i\}_{i=1}^N$ is assumed to be independent and identically distributed samples from $p\left(\underline{x}_k \mid Z_{k-1}\right)$. The pdf of an area in the state space is now derived from the density of the particles in that area. Inserting (C.4) into (C.3b) yields

$$\begin{aligned} p\left(\underline{x}_{k+1} \mid Z_k\right) &= \int p\left(\underline{x}_{k+1} \mid \underline{x}_k\right) p\left(\underline{x}_k \mid Z_k\right) d\underline{x}_k \\ &\approx \int p\left(\underline{x}_{k+1} \mid \underline{x}_k\right) \frac{1}{N} \sum_{j=1}^{N} \delta\left(\underline{x}_k - \underline{x}_k^j\right) d\underline{x}_k \\ &= \frac{1}{N} \sum_{j=1}^{N} p\left(\underline{x}_{k+1} \mid \underline{x}_k^j\right) \end{aligned} \tag{C.5}$$

and

$$\begin{aligned} p\left(\underline{x}_k \mid Z_k\right) &= \frac{p\left(\underline{z}_k \mid \underline{x}_k\right) p\left(\underline{x}_k \mid Z_{k-1}\right)}{\int p\left(\underline{z}_k \mid \underline{x}_k\right) p\left(\underline{x}_k \mid Z_{k-1}\right) d\underline{x}_k} \\ &\approx \frac{p\left(\underline{z}_k \mid \underline{x}_k\right) \frac{1}{N} \sum_{i=1}^{N} \delta\left(\underline{x}_k - \underline{x}_k^i\right)}{\int p\left(\underline{z}_k \mid \underline{x}_k\right) \frac{1}{N} \sum_{i=1}^{N} \delta\left(\underline{x}_k - \underline{x}_k^i\right) d\underline{x}_k} \\ &= \frac{p\left(\underline{z}_k \mid \underline{x}_k\right) \sum_{i=1}^{N} \delta\left(\underline{x}_k - \underline{x}_k^i\right)}{\sum_{i=1}^{N} p\left(\underline{z}_k \mid \underline{x}_k^i\right)} \end{aligned} \tag{C.6}$$

where the particle cloud $\{\underline{x}_k^i\}_{i=1}^N$ represents $p\left(\underline{x}_k \mid Z_{k-1}\right)$ and $\left\{\underline{x}_k^j\right\}_{i=1}^N$ represents $p\left(\underline{x}_k \mid Z_k\right)$. There are several algorithms used to implement the PF. One of the most common is known as the *Bayesian Bootstrap* algorithm.

**Algorithm C.1** Particle Filter using Sequential Importance Resampling

1: Given $p_x\left(\underline{x}_k\right)$, generate $\{\underline{x}_k^i\}_{i=1}^N$ through $N$ samples from $p_x\left(\underline{x}_k\right)$.
2: Time increment: $k = k + 1$
3: Predict every particle in the cloud $r = M/N \ \ (M > N)$ times using (C.1a), obtaining $\left\{\underline{x}_k^j\right\}_{i=1}^M$.
4: Compute the probability weights $w_k^j$ using (C.7b).
5: Calculate estimate and covariance using (C.8).
6: Sample $N$ particles from $\left\{\underline{x}_k^j\right\}_{i=1}^M$ where the particle weights are given by $w_k^j$, producing $\{\underline{x}_k^i\}_{i=1}^N$.
7: Go to item 2.

## C.1.3 Bayesian Bootstrap

The Bayesian Bootstrap algorithm is also known as Sequential Importance Resampling (SIR). Consider the particle cloud $\{\underline{x}_k^i\}_{i=1}^N$ which approximates $p\left(\underline{x}_k \mid Z_{k-1}\right)$. When a measurement $\underline{z}_k$ is received a probability weight $w_k^i$ is defined for every particle

$$w_k^i = \gamma_k^{-1} p\left(\underline{z}_k \mid \underline{x}_k^i\right) \tag{C.7a}$$

$$\gamma_k = \sum_{j=1}^{N} p\left(\underline{z}_k \mid \underline{x}_k^i\right) \tag{C.7b}$$

where $\gamma_k$ is a normalizing constant ensuring $\sum_{i=1}^N w_k^i = 1$. The weights $w_k^i$ are proportional to $p\left(\underline{z}_k \mid \underline{x}_k^i\right)$, which means that the most likely (important)

particles are given the largest weights. Estimate and covariance can now be calculated:

$$\hat{\underline{x}}_k = \int \underline{x}_k p\left(\underline{x}_k \mid Z_k\right) d\underline{x}_k \approx \frac{\int \underline{x}_k p\left(\underline{z}_k \mid \underline{x}_k\right) \sum_{i=1}^{N} \delta\left(\underline{x}_k - \underline{x}_k^i\right) d\underline{x}_k}{\sum_{i=1}^{N} p\left(\underline{z}_k \mid \underline{x}_k^i\right)}$$

$$= \frac{\sum_{i=1}^{N} \underline{x}_k^i p\left(\underline{z}_k \mid \underline{x}_k^i\right)}{\sum_{i=1}^{N} p\left(\underline{z}_k \mid \underline{x}_k^i\right)} = \sum_{i=1}^{N} w_k^i \underline{x}_k^i \tag{C.8a}$$

$$\hat{P}_k = \sum_{i=1}^{N} w_k^i \left(\underline{x}_k^i - \hat{\underline{x}}_k\right)\left(\underline{x}_k^i - \hat{\underline{x}}_k\right)^T \tag{C.8b}$$

The recursive SIR algorithm is presented in Algorithm C.1. According to Bergman (1999), the amount of predicted particles $M$ should be about $10$ times more than the cloud of $N$ particles that is propagated. Line 6 in Algorithm C.1 is the resampling step, where a cloud of $M$ particles is reduced to a cloud of $N$ particles. Note that there are cases where a measurement may yield large weights to only a few particles, resulting in a cloud spanning a small region of the state space. If the next measurements occur outside this region, the PF may diverge. This problem is reduced by using the mentioned resampling step, and by increasing the process noise.

## C.2 The Unscented Kalman Filter

The Unscented Kalman Filter (UKF) is an alternative to the EKF. While the EKF uses linearization to handle nonlinearities, the UKF uses the unscented transformation. This transformation can deal with a wider range of nonlinearities, where it in some cases yields better approximations of the mean and covariance. The UKF is discussed in Julier et al. (1995) and Julier and Uhlmann (1996, 2004), and it is also known as the Sigma Point Kalman Filter (Wan and Van Der Merwe, 2000). Next the unscented transform and a UKF implementation are presented.

### C.2.1 The Unscented Transformation

Given a nonlinear function

$$\underline{y} = \underline{g}\left(\underline{x}\right) \tag{C.9}$$

where $\underline{x}$ is a stochastic vector with known mean $\bar{\underline{x}}$ and covariance $P_x$. The unscented transformation is performed through the following steps.

1. Calculate the sigma points $\underline{x}^i$ and the associated weights $w^i$ for $\underline{x}$ using $\bar{\underline{x}}$ and $P_x$.
2. Apply (C.9) to the sigma points, and calculate the sigma points $\underline{y}^i$ of $\underline{y}$.
3. Calculate the mean and covariance of the function using the sigma points $\underline{y}^i$ and the weights $w^i$.

$$\bar{\underline{y}} = \sum_i w^i \underline{y}^i \tag{C.10a}$$

$$P_y = \sum_i w^i \left(\underline{y}^i - \bar{\underline{y}}\right)\left(\underline{y}^i - \bar{\underline{y}}\right)^T \tag{C.10b}$$

There are several ways to calculate the sigma points and their weights, yielding several possible sigma point sets for a given function. A sigma point set must yield correct values for the mean and covariance of $\underline{x}$, and the associated weights must satisfy

$$\sum_i w^i = 1, \tag{C.11}$$

but the weights do not have to be positive. If they are positive, they can be interpreted as a probability mass function.

### C.2.2 The Extended Symmetric Sigma Point Set

The Extended Symmetric Sigma Point Set is the result of a sigma point selection algorithm yielding $2n_x + 1$ sigma points where $n_x$ is the dimension of $\underline{x}$. It is presented in Algorithm C.2. The weight $w^0$ is a tuning parameter, and it can be used to adjust the central fourth order moment of the Gaussian pdf resulting from the sigma points. Assuming that the pdf of $\underline{y}$ is Gaussian, $w^0 = 1/3$. Other sigma point sets can be found in Julier and Uhlmann (1996) and Julier (2002).

### C.2.3 Unscented Filtering

Consider (C.1) where

$$E\left\{\underline{x}_0\right\} = \hat{\underline{x}}_0; \quad E\left\{\left(\underline{x}_0 - \hat{\underline{x}}_0\right)\left(\underline{x}_0 - \bar{\underline{x}}_0\right)^T\right\} = \hat{P}_0 \tag{C.12a}$$

$$E\left\{\underline{v}_k\right\} = \underline{0}; \quad E\left\{\underline{v}_k \underline{v}_l^T\right\} = \delta_{kl} Q_k \tag{C.12b}$$

$$E\left\{\underline{w}_k\right\} = \underline{0}; \quad E\left\{\underline{w}_k \underline{w}_l^T\right\} = \delta_{kl} R_k \tag{C.12c}$$

$$E\left\{\underline{x}_0 \underline{v}_k^T\right\} = 0; \quad E\left\{\underline{x}_0 \underline{w}_k^T\right\} = 0; \quad E\left\{\underline{v}_k \underline{w}_l^T\right\} = 0. \tag{C.12d}$$

**Algorithm C.2** Extended Symmetric Sigma Point Set

1: Compute sigma points:

$$
\begin{aligned}
P &= UU^T \\
U &= \begin{bmatrix} \underline{u}_1 & \underline{u}_2 & \dots & \underline{u}_{n_x} \end{bmatrix} \\
\underline{x}^0 &= \bar{\underline{x}} \\
\underline{x}^i &= \bar{\underline{x}} + \sqrt{\frac{n_x}{1-w^0}}\underline{u}_i \text{ for } i \in \{1, 2, \dots, n_x\} \\
\underline{x}^{i+n_x} &= \bar{\underline{x}} - \sqrt{\frac{n_x}{1-w^0}}\underline{u}_i \text{ for } i \in \{1, 2, \dots, n_x\} \\
w^i &= \frac{1-w^0}{2n_x} \text{ for } i \in \{1, 2, \dots, 2n_x\} \\
\underline{y}^i &= \underline{g}\left(\underline{x}^i\right) \text{ for } i \in \{0, 1, 2, \dots, 2n_x\}
\end{aligned}
$$

2: Calculate mean and covariance:

$$
\begin{aligned}
\bar{\underline{y}} &= \sum_{i=0}^{2n_x} w^i \underline{y}^i \\
P_y &= \sum_{i=0}^{2n_x} w^i \left(\underline{y}^i - \bar{\underline{y}}\right)\left(\underline{y}^i - \bar{\underline{y}}\right)^T
\end{aligned}
$$

The UKF algorithm is presented in Algorithm C.3. To use Algorithm C.2, an augmented state vector $\hat{\underline{x}}^a$ is defined yielding $2n_{x^a} + 1$ sigma points. A square root version of Algorithm C.3 can be found in Van Der Merwe and Wan (2001).

**Algorithm C.3** The Unscented Kalman Filter

1: Augment the state vector to yield the system

$$\underline{x}_k^a = \begin{bmatrix} \underline{\hat{x}}_k \\ \underline{0} \\ \underline{0} \end{bmatrix}; \quad \hat{P}_k^a = \begin{bmatrix} \hat{P}_k & 0 & 0 \\ 0 & Q_k & 0 \\ 0 & 0 & R_{k+1} \end{bmatrix}.$$

2: Compute the sigma points of the augmented system yielding $n_p$ sigma points:

$$\underline{x}_k^{a,i} = \begin{bmatrix} \underline{x}_k^i \\ \underline{v}_k^i \\ \underline{w}_{k+1}^i \end{bmatrix}; \quad i \in \{0, 1, \ldots, n_p\}$$

3: Predict state and covariance:

$$\begin{aligned} \bar{\underline{x}}_{k+1}^i &= \underline{f}\left(\hat{\underline{x}}_k^i, \underline{u}_k, \underline{v}_k^i\right); \quad i \in \{0, 1, \ldots, n_p\} \\ \bar{\underline{x}}_{k+1} &= \sum_{i=0}^{p} w^i \bar{\underline{x}}_{k+1}^i \\ \bar{P}_{k+1} &= \sum_{i=0}^{p} w^i \left(\underline{x}_{k+1}^i - \bar{\underline{x}}_{k+1}\right)\left(\underline{x}_{k+1}^i - \bar{\underline{x}}_{k+1}\right)^T \end{aligned}$$

4: Predict measurement and covariance:

$$\begin{aligned}
\underline{z}_{k+1}^{i} &= \underline{h}\left(\underline{x}_{k+1}^{i}, \underline{w}_{k+1}^{i}\right); \quad i \in \{0, 1, \ldots, n_p\} \\
\bar{\underline{z}}_{k+1} &= \sum_{i=0}^{p} w^i \underline{z}_{k+1}^{i} \\
\bar{P}_{zz}\left(k+1\right) &= \sum_{i=0}^{p} w^i \left(\underline{z}_{k+1}^{i} - \bar{\underline{z}}_{k+1}\right)\left(\underline{z}_{k+1}^{i} - \bar{\underline{z}}_{k+1}\right)^T \\
\bar{P}_{xz}\left(k+1\right) &= \sum_{i=0}^{p} w^i \left(\underline{x}_{k+1}^{i} - \bar{\underline{x}}_{k+1}\right)\left(\underline{z}_{k+1}^{i} - \bar{\underline{z}}_{k+1}\right)^T
\end{aligned}$$

5: Calculate estimated state and covariance:

$$\begin{aligned}
K_{k+1} &= \bar{P}_{xz}\left(k+1\right)\bar{P}_{zz}^{-1}\left(k+1\right) \\
\hat{P}_{k+1} &= \bar{P}_{k+1} - K_{k+1}\bar{P}_{zz}\left(k+1\right)K_{k+1}^{T} \\
\hat{\underline{x}}_{k+1} &= \bar{\underline{x}}_{k+1} + K_{k+1}\left(\underline{z}_{k+1} - \bar{\underline{z}}_{k+1}\right)
\end{aligned}$$

6: Time increment: $k = k + 1$