View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by University of New Orleans

University of New Orleans

ScholarWorks@UNO

University of New Orleans Theses and Dissertations

Dissertations and Theses

12-19-2003

Best Linear Unbiased Estimation Fusion with Constraints

Keshu Zhang

University of New Orleans

Follow this and additional works at: https://scholarworks.uno.edu/td

Recommended Citation

Zhang, Keshu, "Best Linear Unbiased Estimation Fusion with Constraints" (2003). *University of New Orleans Theses and Dissertations*. 69.

https://scholarworks.uno.edu/td/69

This Dissertation is protected by copyright and/or related rights. It has been brought to you by ScholarWorks@UNO with permission from the rights-holder(s). You are free to use this Dissertation in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you need to obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/or on the work itself.

This Dissertation has been accepted for inclusion in University of New Orleans Theses and Dissertations by an authorized administrator of ScholarWorks@UNO. For more information, please contact scholarworks@uno.edu.

BEST LINEAR UNBIASED ESTIMATION FUSION WITH CONSTRAINTS

A Dissertation

Submitted to the Graduate Faculty of the

University of New Orleans

in partial fulfillment of the

requirement for the degree of

Doctor of Philosophy

in

The Department of Electrical Engineering

by

Keshu Zhang

B.S., Sichuan University, 1998

M.S., University of New Orleans, 2001

Ph.D., Sichuan University, 2002

December 2003

Acknowledgment

I would like to express sincere gratitude to my major advisor, Dr. X. Rong Li, for his invaluable inspiration and guidance, which made it possible for me to complete the Ph.D. dissertation in the Department of Electrical Engineering. Discussions with him were always very delightful, and I gained considerable knowledge and insights into problems and techniques in information fusion, estimation and decision theory, signal processing , etc. I am also indebted to my second major advisor, Professor Yunmin Zhu, for his many fruitful discussions and guidance.

I wish to thank the members in my thesis committee, Dr. X. Rong Li (chair), Dr. Vesselin Jilkov, Dr. Huimin Chen, Dr. Tumulesh K.S. Solanky and Dr. Zhide Fang for their pertinent discussions and comments and helpful suggestions.

I would also like to acknowledge the profitable discussions with the members of the information and systems laboratory at the University of New Orleans, with whom I had the pleasure of carrying out research in an excellent environment. In particular, the discussions with Jifeng Ru, Ming Yang, Peng Zhang, Zhanlue Zhao, Trang Minh Nguyen and Ryan Pitre were very inspiring.

Finally, I can hardly find the suitable words but simply express my deep appreciation to my husband, Haifeng Li, for his love and understanding, and to my parents for their support.

Contents

1 Introduction and Preliminaries **1**

1.1 Estimation Fusion 2

1.1.1 Basic Estimation Fusion Architectures 2

1.1.2 Estimation Fusion Rules 4

1.2 BLUE Fusion 5

1.2.1 Fusion by BLUE with Prior 6

1.2.2 Fusion by BLUE without Prior 6

1.2.3 Effect of Prior Information 7

1.2.4 Recursive BLUE Estimation 10

1.3 Optimality and Efficiency of the Optimal Distributed Fusion 11

1.4 Background and Motivation 12

2 Previous Works **14**

2.1 Data Compression 15

2.1.1 Dimension Reduction 15

2.1.2 Vector Quantization 21

2.2 Out-of-Sequence Measurement Update 28

2.2.1 Single-Lag OOSM Update 29
2.2.2 Multi-Lag OOSM Update 30
2.2.3 OOSM Update in Clutter 31

3 Optimal Sensor Data Dimension Reduction for BLUE Fusion 32

3.1 Problem Formulation 34
3.2 Optimal Compression of a Single-Sensor Data for Estimation 37
3.3 Optimal Data Compression for Estimation Fusion 44
3.3.1 Uncorrelated sensors 44
3.3.2 Linear sensor observation model with uncoupled noises 45
3.3.3 Sensor dimensional requirement larger than rank of $C'_{xz}C_{xz}C_z^+$ 47
3.3.4 Arbitrary sensor dimensional requirement 48
3.4 Simulation 54
3.4.1 Single Sensor 56
3.4.2 Multi-Sensor Fusion System 57
3.5 Summary 59

4 Optimal Sensor Data Quantization for BLUE Fusion 61

4.1 Introduction 61
4.2 Problem Formulation 63
4.3 Optimal Quantizer Design for BLUE Fusion 64
4.4 Dynamic Processing for State Estimation Update 69
4.4.1 General Dynamic System 70
4.4.2 Linear Gaussian Dynamic System 72

4.5 Simulation Results . . . 75
4.5.1 Estimation Fusion for Static Case . . . 76
4.5.2 State Estimation Fusion for Dynamic System . . . 79
4.6 Summary . . . 81

5 Optimal Update with Out-of-Sequence Measurements 82

5.1 Introduction . . . 82
5.2 Problem Formulation . . . 84
5.3 Optimal Update with Available Information . . . 85
5.4 Optimal Update Algorithms . . . 87
5.4.1 Algorithm I — Globally Optimal Update . . . 87
5.4.2 Comparison of Globally Optimal Update Algorithms . . . 97
5.4.3 Algorithm II — Constrained Optimal Update . . . 99
5.4.4 Update with Arbitrarily Delayed OOSMs . . . 109
5.5 OOSMs Update for Linear Gaussian Systems under Nonsingularity Conditions 115
5.5.1 Algorithm I — Globally Optimal Update (ALG-I) . . . 115
5.5.2 Algorithm II — Constrained Optimal Update (ALG-II) . . . 118
5.5.3 Update with Arbitrarily Delayed OOSMs . . . 120
5.6 Numerical Examples . . . 121
5.6.1 Nonsingular $F_{k,d}$ . . . 122
5.6.2 Singular $F_{k,d}$ . . . 125
5.7 Summary . . . 128

6 Multi-Sensor Multi-Target Tracking with OOSMs **129**

6.1 Introduction 129

6.2 Problem Formulation 131

6.3 OOSM Update in Clutter 133

6.3.1 OOSM Update: PDA with ALG-I 133

6.3.2 OOSM Update: PDA with ALG-II 134

6.3.3 OOSM Update: Multiple Target Case 137

6.4 Simulations 138

6.4.1 Scenario 138

6.4.2 Results for OOSMs with Good Accuracy 140

6.4.3 Results for OOSMs with Moderate Accuracy 141

6.5 Summary 141

7 Conclusions and Future Work **144**

Bibliography **147**

Vita **159**

Abstract

Estimation fusion, or data fusion for estimation, is the problem of how to best utilize useful information contained in multiple data sets for the purpose of estimating an unknown quantity — a parameter or a process. Estimation fusion with constraints gives rise to challenging theoretical problems given the observations from multiple geometrically dispersed sensors:

- Under dimensionality constraints, how to preprocess data at each local sensor to achieve the best estimation accuracy at the fusion center?
- Under communication bandwidth constraints, how to quantize local sensor data to minimize the estimation error at the fusion center?
- Under constraints on storage, how to optimally update state estimates at the fusion center with out-of-sequence measurements?
- Under constraints on storage, how to apply the out-of-sequence measurements (OOSM) update algorithm to multi-sensor multi-target tracking in clutter?

The present work is devoted to the above topics by applying the best linear unbiased estimation (BLUE) fusion. We propose optimal data compression by reducing sensor data from a higher dimension to a lower dimension with minimal or no performance loss at the fusion center. For single-sensor and some particular multiple-sensor systems, we obtain the explicit optimal compression rule. For a multisensor system with a general dimensionality requirement, we propose the Gauss-Seidel iterative algorithm to search for the optimal compression rule. Another way to accomplish sensor data compression is to find an optimal

sensor quantizer. Using BLUE fusion rules, we develop optimal sensor data quantization schemes according to the bit rate constraints in communication between each sensor and the fusion center. For a dynamic system, how to perform the state estimation and sensor quantization update simultaneously is also established, along with a closed form of a recursion for a linear system with additive white Gaussian noise. A globally optimal OOSM update algorithm and a constrained optimal update algorithm are derived to solve one-lag as well as multi-lag OOSM update problems. In order to extend the OOSM update algorithms to multisensor multitarget tracking in clutter, we also study the performance of OOSM update associated with the Probabilistic Data Association (PDA) algorithm.

Chapter 1

Introduction and Preliminaries

Data fusion is the collective name given to techniques that are used to combine the measurements from more than one sensor into a single quantity. The motivation behind using multiple sensors is often two-fold: either to reduce error and uncertainty in the measurement, or to obtain the estimate using the measured quantity from multiple sensors that would not be accessible using a single sensor. Data fusion builds on many years of experiences in the established techniques. These techniques are incorporated into a framework by transforming the data into a common format, producing verdicts on the correctness of the various sources, and allowing reliable estimation of the parameters or states of a problem. Data fusion is an emerging technology. Numerous techniques have been promoted and adopted. In recent years, the applications of data fusion or information fusion techniques have significantly increased, such as in target tracking, image processing, economic data analysis.

Roughly, data fusion studies the optimal information processing in distributed environments through intelligent integration of the data from multiple sensors. As a specialized branch of data fusion, estimation fusion focuses on estimating the parameters or states from

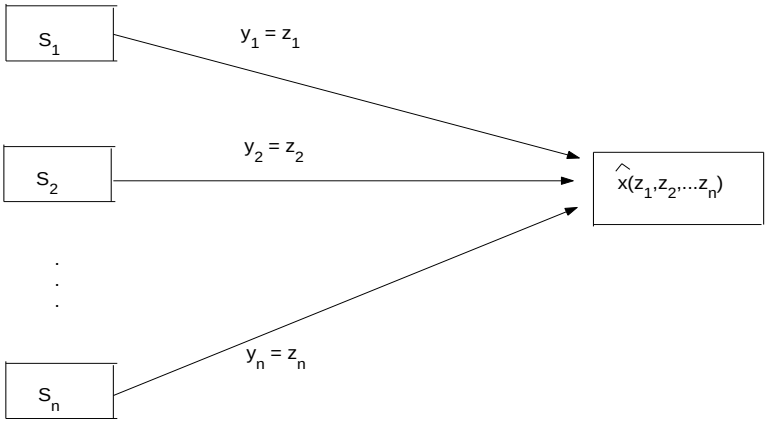
the multisource measurements. The task is to improve the estimation accuracy. Estimation fusion is an exciting area of endeavor in the field of estimation theory. There are numerous publications and conferences devoted to this area.

1.1 Estimation Fusion

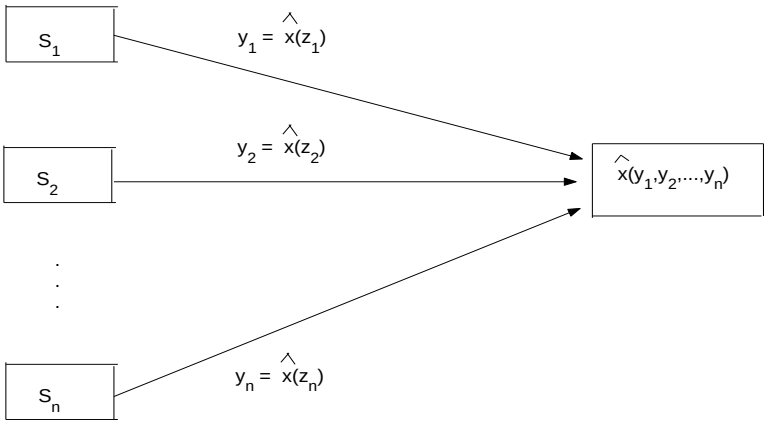
Estimation fusion, or data fusion for estimation, is the problem of how to best utilize useful information contained in multiple sets of data for the purpose of estimating an unknown quantity — a parameter or process. These observations may be of different types or include conflicting information. The multiple sets of observations are usually but not necessarily obtained from multiple sources (e.g., multiple sensors). Even if the observations coming from single source, we can artificially treat these observations coming from different locations and view it as the fusion problem. In this sense, estimation itself is fusion by fusing the prior and posterior information, and filtering is fusion by fusing the prediction and current observation. Estimation fusion is important in many application areas, such as target tracking, sensor informatics, etc.

1.1.1 Basic Estimation Fusion Architectures

There are two basic estimation fusion architectures: centralized and decentralized / distributed (also referred to as measurement fusion and track fusion in target tracking, respectively), depending on whether the raw measurements are sent to the fusion center or not. For centralized fusion, all raw measurements are sent to the fusion center. For distributed fusion, each sensor only sends the processed data to the fusion center. These two approaches

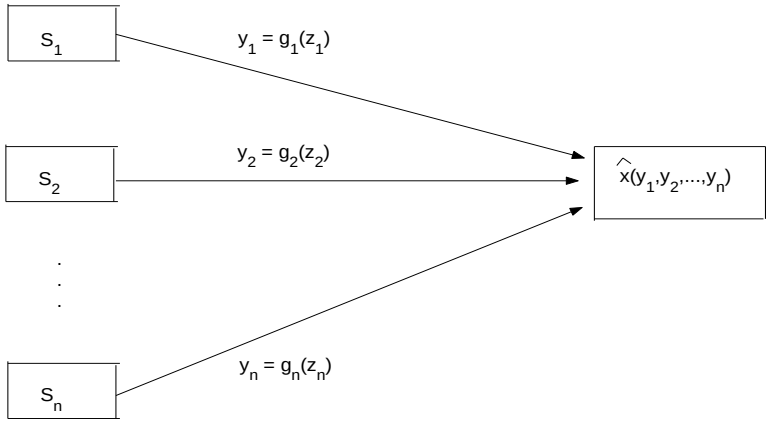


Measurement (Centralized) Fusion



Standard Distributed Fusion

have pros and cons in terms of survivability, autonomy, communication requirements, etc. Centralized fusion is nothing but a conventional estimation problem with distributed data. Distributed fusion is more challenging and has been a focal point of fusion research for many years.

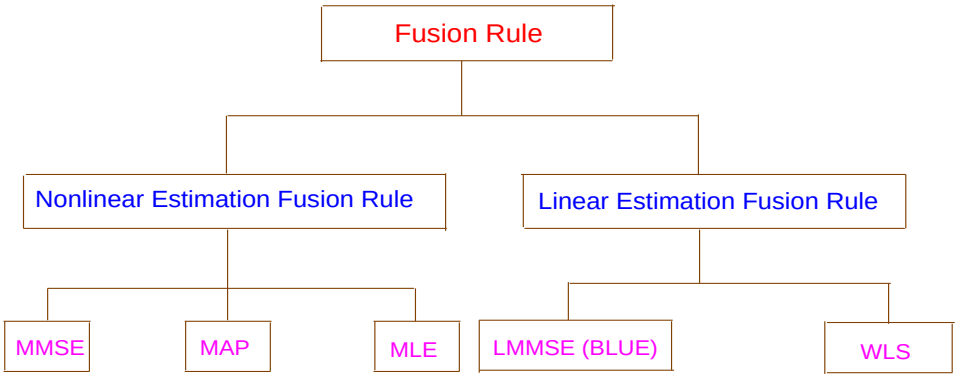


General (Distributed) Fusion

1.1.2 Estimation Fusion Rules

Estimation fusion is to perform estimation by satisfying certain optimality criterion at the fusion center based on information available from multiple sensors. There are two basic classes of estimation fusion rules: linear and nonlinear rules. For the class of nonlinear fusion rules, we generally focus on the minimum mean square error estimation (MMSE) fusion rule, maximum posterior estimation (MAP) fusion rule, maximum likelihood estimation (MLE) fusion rules. For linear rules, we mainly focus on optimal (linear) weighted least squares (WLS) fusion rules and the best linear unbiased estimation (BLUE) fusion rules. The latter is also known as the linear minimum mean-square error estimation (LMMSE) fusion rules, linear minimum variance (LMS) fusion rules, or linear unbiased minimum variance (LUMS) estimation fusion rules. Among all these estimation fusion rules, the commonly used optimality criterion is to minimize the mean square errors (MSE). Within the unbiased estimation fusion, MMSE fusion rule is to minimize the MSE over all estimators, and the BLUE fusion rule is to minimize MSE within the linear class.

This dissertation is based largely on the best linear unbiased estimation fusion. In the



Estimation Fusion Rule

following, we present the BLUE fusion in a general, systematic, and unified setting, largely following [40].

1.2 BLUE Fusion

Under the Gaussian case, the BLUE fusion is identical to the MMSE fusion. Under a non-Gaussian case, BLUE fusion is optimal within the linear class. This framework is flexible. There is no additional assumption made on the correlation of sensor observation errors or local estimation errors. It does not require a common local observation model and synchronization of sensor observations. The BLUE fusion rule does not impose any constraint on the network topology or information pattern.

Denote by y the available information at the fusion center and x the estimatee (i.e., quantity to be estimated). For the observation model, by prior information, we mean the first and second order moments related to the estimatee: $x - \bar{x} = E(x)$, C_x =var(x) and C_{xy} =cov(x, y); with complete prior information we mean that $\bar{x}$, C_x and C_{xy} are known exactly; without prior information we mean none of $\bar{x}$, C_x and C_{xy} are known or exist.

1.2.1 Fusion by BLUE with Prior

The BLUE with prior is given by

$$\hat{x} = \underset{\hat{x}=A+By}{\arg\min} E[(x-\hat{x})(x-\hat{x})'|y] = E^*(x|y) = \bar{x} + C_{xy}C_y^{-1}(y-\bar{y})$$

$$P = \mathrm{cov}(x-\hat{x}) = C_x - C_{xy}C_y^{-1}C_{xy}'$$

where A and B are matrices to be chosen such that the MSE of $\hat{x}$ is minimized. If the inverse C_y^{-1} does not exist, it can be simply replaced with the unique *Moore-Penrose pseudoinverse* (MP inverse in short) C_y^+. The BLUE fusion is the linear estimation fusion rule that only requires the knowledge of the first two order of statistics related to the estimatee.

1.2.2 Fusion by BLUE without Prior

For a linear observation model

$$y = Hx + v \tag{1.1}$$

where for the centralized fusion, y, H, and v are the stacked observations, the measurement matrix, and observation noise, respectively; for (standard) distributed fusion, they are the stacked local estimates $[\hat{x}_1', \ldots, \hat{x}_n']'$, $[I, \ldots, I]'$, and stacked local estimation errors, respectively from all sensors.

Without knowledge of the prior mean $\bar{x}$ and covariance C_x of the estimatee x, the linear unbiased estimator based on the available information y exists if and only if H has full column rank (i.e., $H^+ = (H'H)^{-1}H$). If exists, it is unique (almost surely) and called the

BLUE without prior which is given by [53]

$$\hat{x} = \tilde{K}(y - \bar{v})$$

$$P = \text{cov}(x - \hat{x}) = \tilde{K} R \tilde{K}'$$

where R =cov(v), $\tilde{K} = H^{+}[I - R(TRT)^{+}]$, $T = I - HH^{+}$.

1.2.3 Effect of Prior Information

What is the relationship between BLUE fusion with and without prior? Can one be converted to the other one? These two questions are answered in [49].

For the linear data model (1) with known $\bar{v}$ and R, the BLUE with complete prior information $\bar{x}$, C_x, and C_{xv} can always be converted to the BLUE without prior information by treating the prior mean $\bar{x}$ as extra data in the linear model: $\bar{x} = x + (\bar{x} - x)$. More specifically, BLUE with complete prior information ($\bar{x}$, C_x, and C_{xv}) for the linear data model (1) with known $\bar{v}$ and R always coincides (almost surely) with BLUE without prior information for the linear data model $\tilde{y} = \tilde{H}x + \tilde{v}$ with

$$\tilde{y} = \begin{bmatrix} \bar{x} \\ \\ y \end{bmatrix}, \qquad \tilde{H} = \begin{bmatrix} I \\ \\ H \end{bmatrix}$$

$$E[\tilde{v}] = \begin{bmatrix} 0 \\ \\ \bar{v} \end{bmatrix}, \qquad \tilde{R} = \begin{bmatrix} C_x & -C_{xv} \\ \\ -C'_{xv} & R \end{bmatrix}$$

which is given by

$$\hat{x} = \tilde{K}(\tilde{y} - E[\tilde{v}]), \quad P = \tilde{K}\tilde{R}\tilde{K}', \quad \tilde{K} = [I - KH, K]$$

where K is the gain matrix of the BLUE fuser with complete prior.

It shows that the prior information can always be completely embedded into the linear data model with prior mean as data. This is the foundation of the optimal linear update with out-of-sequence measurements. In the following, we will also show that the BLUE without prior can be converted to BLUE with complete prior.

For the same linear data model, the BLUE fuser without prior clearly can never have a smaller MSE matrix than the BLUE fuser with complete prior information. The questions left to be answered are: (a) how much worse? (b) can the MSE be the same? Clearly, the MSEs of the two fusers are the same if and only if the prior information is redundant given the data for the problem. The following results answer these questions.

Lemma 1: (Redundancy Condition of Prior for BLUE) *[49] Let $\hat{x}_1$ and $\hat{x}_2 = K_2(y - \bar{v})$ be BLUE with complete prior and without prior information, respectively, using the same data y with known mean of error $\bar{v}$. Then a necessary and sufficient condition for $\hat{x}_1 = \hat{x}_2$ almost surely is $K_2 C_y = C_{xy}$, where C_y =cov(y) and C_{xy} =cov(x, y).* □

Note that this theorem is valid regardless if y is linear or nonlinear in the estimatee x provided the BLUE without prior has the form $\hat{x}_2 = K_2(y - \bar{v})$. For linear data, we have the following stronger results.

Theorem 1: (Redundancy Conditions of Prior for BLUE) *[49] Consider the linear data model (1) with known $\bar{v}$ and C. Let $\hat{x}_1$ and $\hat{x}_2$ be BLUE with complete prior and without prior information, respectively, using the same data y. Then the following statements are equivalent. (a) $\hat{x}_1 = \hat{x}_2$ almost surely (i.e., prior information is redundant for the BLUE) (b) The gain matrix K_2 of $\hat{x}_2$ satisfies $K_2 C_y = C_{xy}$. (c) The gain matrix K_2 of $\hat{x}_2$ satisfies $K_2 C_{vy} = 0$. (d) $(I - C_{vy} C_{vy}^{+})H$ has full column rank; that is, $[(I - C_{vy} C_{vy}^{+})H]^{+}[(I - C_{vy} C_{vy}^{+})H] = I$ where C_{vy}^{+} is the MP inverse of C_{vy} =cov(v, y). (e) The following condition*

holds

$$[I - C_{xy}C_y^+H][I - C_{xy}C_y^+H]^+[I - C_{xy}C_y^+H] = [I - C_{xy}C_y^+H]$$

□

Note that the BLUE without prior exists if and only if H has full column rank. The redundancy conditions are general — they hold whenever the BLUE without prior exists for the model (1) with known $\bar{v}$ and R. With additional assumptions, we have stronger results, as stated by the following corollaries.

Corollary 1: (Sufficient Condition for Redundancy of Prior for BLUE) *[49] Consider linear data model (1) with known $\bar{v}$ and R. Then prior information is redundant for BLUE if rank$[C_Y, H]$=rank(C_y) and $I - C_{xy}C_y^+H = 0$.* □

Corollary 2: (Contribution of Prior to BLUE) *[49] Consider the linear data model (1) with known $\bar{v}$ and R. Then the contribution of the prior information ($\bar{x}$, C_x, and C_{xv}) to BLUE fusion in the sense of Fisher information matrix is given by*

$$P_1^{-1} - P_2^{-1} = (I + C_{xv}R^{-1}H)'(C_x - C_{xv}C^{-1}C'_{xv})^{-1}(I + C_{xv}C^{-1}H)$$

where P_1 and P_2 are the MSE matrices of BLUE fusers with complete and without prior information, respectively. In particular, the prior information is redundant for BLUE fusion - BLUE fuser with complete prior information is (almost surely) identical to BLUE fuser without prior information - if and only if

$$I + C_{xv}R^{-1}H = 0$$

□

Corollary 3: *Consider the linear data model (1.1) with known $\bar{v}$ and R. Assume $R > 0$, $C_x > 0$, and $C_{xv} = 0$. Then the prior information ($\bar{x}$, C_x) is useful for BLUE*

fusion - BLUE fuser with complete prior information has a smaller MSE matrix than the BLUE fuser without prior information - and the contribution of the prior information is $P_1^{-1} - P_2^{-1} = C_x^{-1}$. □

This corollary indicates that in the usual case ($C > 0$, $C_x > 0$, and $C_{xv} = 0$), the optimal use of the prior mean and covariance does improve the performance of the BLUE fusion; the prior information and the data information are additive because they are uncorrelated.

1.2.4 Recursive BLUE Estimation

Recursive forms of the BLUE estimation are of major importance in practice for their computational efficiency. The BLUE estimator using $y = [y_1', y_2']'$ with two observations y_1 and y_2 always has the following quasi-recursive forms [53]:

$$E^*[x|y_1,y_2] = E^*[x|\tilde{y}_1] + E^*[x - \bar{x}|\tilde{y}_{2|1}] = \hat{x}_1 + C_{x\tilde{y}_{2|1}} C^+_{\tilde{y}_{2|1}} \tilde{y}_{2|1}$$

$$P_{\tilde{x}} = C_{\tilde{x}_1} - C_{x\tilde{y}_{2|1}} C^+_{\tilde{y}_{2|1}} C'_{x\tilde{y}_{2|1}}$$

where $\tilde{x}_1 = x - \hat{x}_1 = x - E^*[x|y_1]$ has zero mean and the innovation $\tilde{y}_{2|1} = y_2 - E^*[y_2|y_1]$, and $C_{\tilde{x}_1} = \text{cov}(\tilde{x}_1) = C_x - C_{xy_1} C^+_{y_1} C'_{xy_1}$, $C_{\tilde{y}_{2|1}} = \text{cov}(\tilde{y}_{2|1}) = C_{y_2} - C_{y_2y_1} C^+_{y_1} C'_{y_2y_1}$, $C_{x\tilde{y}_{2|1}} = C_{xy_2} - C_{xy_1} C^+_{y_1} C'_{y_2y_1}$.

When

$$E^*[x|\hat{x}_1, \hat{y}_{2|1}] = E^*[x|\hat{x}_1, y_2]$$

the quasi-recursive BLUE becomes truly recursive. We call it the BLUE filter. The famous Kalman filter is a special case of the recursive BLUE filtering under the corresponding assumption.

1.3 Optimality and Efficiency of the Optimal Distributed Fusion

Distributed fusion has certain advantages over centralized fusion in terms of survivability, autonomy, communication requirements, etc.

An important issue in distributed fusion is its performance relative to that of the centralized fusion using the same data. It is well known that under linear-Gaussian assumption (i.e., linear measurements with additive white Gaussian noise), optimal distributed fusion is equivalent to the centralized fusion. However, in many applications, this is not the case. When there exists some correlation across sensors, what is the performance of the distributed fusion? In general, distributed fusion can never outperform the optimal measurement fusion. So, two natural questions arise:

- Under what conditions do they perform the same?
- How much performance degradation does the distributed fusion suffer?

Centralized and distributed estimation fusion architectures can be considered as a general framework. As a basis, the BLUE, optimal WLS, and optimal generalized WLS fusion rules for an arbitrary number of sensors have been presented.

The optimal centralized and distributed fusers are algebraically equivalent when the observation noises are uncorrelated across sensors. When the observation noises are correlated across sensors, they have different performance in general. Thus there should exist necessary and sufficient conditions for the optimal centralized and distributed fusers to have the same performance. In my master thesis [48], several necessary and sufficient conditions for the

optimal distributed and centralized BLUE (or WLS) fusion and the optimal WLS fusion to have identical performance were obtained, which can be easily verified. Several measures of efficiency of distributed fusion relative to centralized fusion were proposed, which include the ratio of the mean-square errors (MSER) and generalized error variance ratio (GEVR) of centralized fusion and distributed fusion. These measures quantify the performance degradation of the distributed fusion to the optimal centralized fusion.

1.4 Background and Motivation

Since there is information loss in terms of MSE for most general distributed fusion systems, without any constraints, all raw measurements from each local sensor may directly be sent to the fusion center for a multisensor fusion system in order to achieve the best performance. However, for multisensor systems, measurements are conveyed through the communication with limited capacity, and the processing power at the fusion center is also limited. Thus, certain constraints need to be considered.

The work is motivated by the applications of practical target tracking and sensor fusion systems. The discussion in this dissertation deals with multi-sensor distributed estimation fusion under various constraints. How do we optimally utilize local information at the sensor level in order to have the best estimation accuracy at the fusion center? The problem can be formulated in terms of constrained optimization. For different constrains, we try to find the corresponding optimal solutions. Under the dimensionality constraint, we will address the problem of how to compress the local sensor data in order to achieve the optimal BLUE fusion at the center. Under communication bandwidth constraint, we try to build the

BLUE estimation scheme based on the limited sensor data quantization levels. Under the information storage constraint, we will present the optimal out-of-sequence update, together with its applications to multi-sensor multi-target tracking under the measurement origin uncertainty.

The main topics in the dissertation are estimation fusion under different constraints, such as the dimensionality constraint, the communication bandwidth constraint, and the information storage constraint. These constraints have been studied in some other areas. In the next chapter, we will give a brief survey of previous works related with the dissertation to show that our problems in estimation fusion need special treatments.

The dissertation is organized as follows. We start in Chapter 1 with an introduction to the basic structure of estimation fusion, several fusion rules, and properties of the BLUE fusion rule. Chapter 2 gives a concise description of the previous works on data compression and estimation fusion related to the dissertation. Optimal sensor data dimension reduction for linear estimation fusion is described in Chapter 3. Specific issues in the optimal estimation fusion with limited communication bandwidth, resulting in an estimation fusion scheme based on sensor data quantization accuracy, and both estimation and quantization update in a dynamic system along with a closed form of a recursion for a linear dynamic system with white Gaussian noise, are studied in some detail in Chapter 4. Optimal update with out-of-sequence measurements and its application to multi-sensor multi-target tracking in clutter are described in Chapters 5 and 6 respectively, along with numerical results obtained using target tracking scenarios. Finally, Chapter 7 summarizes the work and presents a few future directions.

Chapter 2

Previous Works

Estimation fusion has been investigated for more than two decades. Target Tracking and Motion Analysis demand practical fusion systems for both military and non-military applications. Most of research efforts on estimation fusion focus on the optimal fusion rule [4, 16, 54, 53, 50, 48, 49, 85] at the fusion center. With the advances of modern sensor technology such as the wireless network connecting the MEMS microsensors, a lot of new issues in estimation fusion area arise. In general, estimation fusion can occur at any level when the data processor has access to measurements from multiple sensors regarding the same target state. Therefore it is very natural to consider communication, computation and power constraints in the fusion system. In a real situation, both the optimal sensor data processing and the optimal fusion rule need to be taken into consideration since the information is shared by the whole fusion system with various constraints.

With the communication constraints between each sensor and the fusion center, in this dissertation, we will mainly consider how to compress data at the sensor level to achieve the best estimation accuracy at the fusion center. Out-of-sequence measurement update problem

is a special topic in target tracking area. Recently, a lot of research effort is focused on how to build optimal update algorithms. Many approaches have been proposed. In the following, we give a survey for the existing data compression and OOSM update techniques. In Chapter 3, 4, 5 and 6, we consider each problems in more detail in the data fusion framework.

2.1 Data Compression

Data compression is an old and important problem, which has been the subject of a large volume of literature ranging from estimation, detection, information theory to statistical pattern recognition and data mining. Most methods proposed only concern the data compression within each local sensor. There is little research on data compression with distributed estimation fusion. The popular techniques in these areas do not capture the relationship between the local sensors and the fusion center. That is why we need to investigate new data compression techniques particularly useful for fusion systems. Dimension reduction and quantization are two popular techniques to perform data compression in the pattern recognition and communication areas. Sensor data compression for estimation fusion can also be formulated as sensor data dimension reduction and the sensor quantization problems. In the following, we give a brief description on the existing data compression methods.

2.1.1 Dimension Reduction

In pattern recognition, the input data is usually represented by a set of d features or measurements, called a d-dimensional feature vector. In most intelligent data analysis, the dimension of the measurement space is usually very large in order for the measurements to carry as

much information as possible about the physical objects of interest. This high-dimensionality is a major cause of the practical limitations of many sophisticated techniques because the complexity of many algorithms increases very fast with the dimension.

Besides, it has been observed that using a large number of features may actually degrade the performance of classifiers if the number of the input data is small relative to the number of features [30, 69, 70]. This fact, referred to as the "peaking phenomenon", is caused by the "curse of dimensionality" [7]. Actually, the number of parameters in a d-dimensional distribution usually grows much faster than $\mathrm{O}(d)$ unless one makes a strong assumption that the features are independent [10]. For instance, given that the features are not independent, the normal distributions have $O(d^2)$ parameters and the binary distributions have $O(2^d)$ parameters. In other words, the complexity of a distribution increases rapidly when the dimension increases.

Therefore, dimension reduction is essential to many engineering applications. In the past several decades, many dimension reduction techniques have been proposed, see [31] for a complete survey. There is a tradeoff between the complexity of classifiers and that of dimension reduction techniques. Thus, linear dimension reduction methods are usually used in most practical applications. The most well-known linear dimension reduction methods are probably Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA).

Principal Component Analysis

PCA (also known as the Karhunen-Loève transformation in communication theory) [20, 32] is possibly the dimension reduction technique most widely used in practice, perhaps due to its conceptual simplicity and to the fact that relatively efficient algorithms exist for its

computation. PCA is a linear dimension reduction technique based on the mean squared error (MSE) criterion:

$$W = \arg\min_{W} E\left[(z - Wz^*)(z - Wz^*)'\right] \tag{2.1}$$

where z denotes a D -dimensional pattern represented by a lower d-dimensional random vector $z^* = W'z$. The PCA method computes the largest eigenvectors of the covariance matrix $S = E\{(z - \bar{z})(z - \bar{z})'\}$ (or correlation matrix R) of n samples. The eigenvectors are called the principal components. These first d largest principal components contain the most of the variance of data. By expanding data on these orthogonal principal components, we have the minimal reconstruction error on the original data z.

Geometrically, the hyperplane spanned by the first d principal components is the regression hyperplane that minimizes the orthogonal distances to the data. In this sense, PCA is a symmetric regression approach, as opposed to standard linear regression, which points one component as response variable and the rest as predictors.

The key property of principal component analysis is that it attains the best linear map $z \in \mathbf{R}^D \mapsto z^* \in \mathbf{R}^d$ in the senses of:

- Least squared sum of errors of the reconstructed data.
- Maximum mutual information (assuming the data vector z distributed normally) between the original vector z and its projection z^*: $I(z; z^*) = \frac{1}{2}\ln((2\pi e)^d \lambda_1, \ldots, \lambda_d)$ are the first d largest eigenvalues of the covariance matrix.

Certainly, for estimation fusion, we can still rely on PCA to perform sensor data dimension deduction. At the fusion center, the optimal estimation fusion rule can be applied

based on the compressed measurements and certain accuracy can be achieved. But the decorrelation and high measure of statistical significance provided by the first few principal components of the sensor observation cannot guarantee to reveal the necessary information for the estimation at the fusion center. The estimation accuracy could suffer significantly. That is why we need to consider a more fundamental formulation for the problem. The sensor data dimension reduction in linear estimation fusion can be formulated as

$$W = \arg\min_{W} E\{(x - \hat{x}(W'z))(x - \hat{x}(W'z))'\} \tag{2.2}$$

where x is the state to be estimated; with BLUE fusion rule, the linear estimator is $\hat{x}(W'z) = KW'z$. Although the optimality criteria for (2.1) and (2.2) are both MSE, they have different bases: PCA is in the observation space while the optimal sensor data dimension reduction for linear estimation fusion is in the parameter (state) space. The major differences between our problem with PCA are: The goal of sensor data dimension reduction in a fusion system is to provide a good estimate of the state or process at the fusion center, rather than to use a lower dimensional data to represent the sensor measurements; even if we use linear dimension reduction technique, the transformation matrix W for all measurements at the fusion system is a block diagonal matrix with each block corresponding to the transformation at each local sensor. How to get the optimal W with this structural constraint is much harder than solving the full unconstrained matrix W in the pattern recognition area. With our research results, for state estimation fusion in static case, we achieved the optimal linear sensor data dimension reduction based on BLUE fusion rule [80], and its generalization to state estimation fusion in dynamic system [17] .

Linear Discriminant Analysis

Linear discriminant analysis (also called Fisher's Linear Discriminant) is another popular linear dimension reduction method. In many applications, LDA has proven to be very powerful. Fisher originally introduced LDA for two classes [22], and Rao generalized LDA to handle multi-class cases [68]. LDA is given by a linear transformation matrix $W \in \mathcal{R}^{D\times d}$ maximizing the so-called Fisher criterion (a kind of *Rayleigh* coefficient)

$$J(W) = \mathrm{tr}[W'\mathbf{S}_b W(W'\mathbf{S}_w W)^{-1}] \tag{2.3}$$

where $\mathbf{S}_b = \sum_{i=1}^{c} p_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T$ is the between-class scatter matrix; $\mathbf{S}_w = \sum_{i=1}^{c} p_i \mathbf{S}_i$ is the within-class scatter matrix; c is the number of classes; $\mathbf{m}_i$ and p_i are the mean vector and *a priori* probability of class i, respectively; $\mathbf{m} = \sum_{i=1}^{c} p_i \mathbf{m}_i$ is the overall mean vector; $\mathbf{S}_i$ is the scatter matrix of class i; D and d are the dimensions of the data before and after the transformation, respectively. To maximize (2.3), the transformation matrix W must be constituted by the largest eigenvectors of $\mathbf{S}_w^{-1}\mathbf{S}_b$. The purpose of LDA is to maximize the between-class scatter while simultaneously minimizing the within-class scatter. The two-class LDA has a close connection to optimal linear Bayes classifiers. In the two-class case, the transformation matrix W is just a vector, which is in the same direction as the discriminant in the corresponding optimal Bayes classifier. However, it has been shown that LDA is suboptimal for multi-class problems [57].

A major drawback of LDA is that it cannot be applied when $\mathbf{S}_w$ is singular due to the *small sample size problem* [24]. The small sample size problem arises whenever the number of samples is smaller than the dimension of samples. For example, a 64×64 image in a face recognition system has 4096 dimensions, which requires more than 4096 training data to

ensure that $\mathbf{S}_w$ is nonsingular. So, LDA is not a stable method in practice when the training data are scarce. In recent years, many researchers have noticed this problem and tried to overcome the computational difficulty with LDA. Tian *et al.* [77] used the pseudo-inverse matrix $\mathbf{S}_w^+$ instead of the inverse matrix $\mathbf{S}_w^{-1}$. For the same purpose, Hong and Yang [29] tried to add a singular value perturbation to $\mathbf{S}_w$ to make it nonsingular. Neither of these methods are theoretically sound because Fisher's criterion is not valid when $\mathbf{S}_w$ is singular. When $\mathbf{S}_w$ is singular, any positive $\mathbf{S}_b$ makes Fisher's criterion infinitely large. Thus, these naive attempts to calculate the (pseudo or approximate) inverse of $\mathbf{S}_w$ may lead to arbitrary (meaningless) results. Besides, it is also known that an eigenvector could be very sensitive to small perturbation if its corresponding eigenvalue is close to another eigenvalue of the same matrix [75]. In 1992, Liu *et al.* [55] modified Fisher's criterion by using the total scatter matrix $\mathbf{S}_t = \mathbf{S}_b + \mathbf{S}_w$ as the denominator instead of $\mathbf{S}_w$. It has been proven that the modified criterion is exactly equivalent to Fisher's criterion. However, when $\mathbf{S}_w$ is singular, the modified criterion reaches the maximum value (*i.e.*, 1) no matter what the transformation $\mathbf{W}$ is. Such an arbitrary transformation cannot guarantee the maximum class separability unless $\mathbf{W}^T\mathbf{S}_b\mathbf{W}$ is maximized. In 2000, Chen *et al.* [18] proposed the LDA+PCA method. When $\mathbf{S}_w$ is of full rank, the LDA+PCA method just calculates the maximum eigenvectors of $\mathbf{S}_t^{-1}\mathbf{S}_b$ to form the transformation matrix. Otherwise, a two-stage procedure is employed. First, the data are transformed into the null space $\mathcal{V}_0$ of $\mathbf{S}_w$. Second, it tries to maximize the between-class scatter in $\mathcal{V}_0$, which is accomplished by performing principal component analysis (PCA) on the between-class scatter matrix in $\mathcal{V}_0$. Although this method solves the small sample size problem, it is obviously suboptimal because it maximizes the between-class scatter in the null space of $\mathbf{S}_w$ instead of the original input space.

While solving the optimal sensor data dimension reduction problem under BLUE fusion rule, (2.2) can be written as

$$W = \arg\min\{C_x - C_{xz}W(W'C_zW)^{-1}W'C'_{xz}\} \tag{2.4}$$
$$= \arg\max\{W'C'_{xz}C_{xz}W(W'C_zW)^{-1}\}$$

If we treat $C'_{xz}C_{xz} = \mathbf{S}_b$ and $C_z = \mathbf{S}_w$, when C_z is nonsingular, the optimal W without block diagonal structure constraint can be directly borrow the solution from LDA. However, we derived a more general solution for (2.4) which can handle the case that C_z is singular, also W could be block diagonal. Our results provide a direct solution for the *small sample size problem* in pattern recognition area by applying LDA with singular $\mathbf{S}_w$. The full analysis and results are submitted for publication.

2.1.2 Vector Quantization

Vector quantization (VQ) (or block or multidimensional quantization) is a technique that exploits the underlying structure of input vectors for the purpose of data compression. Specifically, an input space is divided into a number of distinct subsets, and for each subset, a reconstruction vector is defined. When the quantizer is presented as a new input vector, the region in which the vector lies is first determined, and then represented by the reproduction vector for that subset. Thus, by using an encoded version of this reproduction vector for storage or transmission in place of the original input vector, considerable savings in storage or transmission bandwidth can be realized, at the expense of some distortion. The collection of possible reproduction vectors is called the *code book* of the quantizer, and its members are called *code words*.

Measurement Quantization

More generally, we can define a quantizer as consisting of a set of cells $S = \{S_i;\ i \in I\}$, where the index set I is ordinarily a collection of consecutive integers beginning with 0 or 1, together with a set of reproduction values or points or levels $C = \{y_i;\ i \in I\}$, so that the overall quantizer q is defined by $q(z) = y_i$ for $z \in S_i$, which can be expressed concisely as

$$q(z) = \sum_i y_i 1_{s_i}(z)$$

where the indicator function $1_s(z)$ is 1 if $z \in S$ and 0 otherwise.

The quality of a quantizer can be measured by the goodness of the resulting reproduction in comparison to the original. One way of accomplishing this is to define a distortion measure $d(z, \hat{z})$ that quantifies cost or distortion resulting from reproducing z as $\hat{z}$ and to consider the average distortion as a measure of the quality of a system, with smaller average distortion meaning higher quality. The most common distortion measure is the squared error $d(z, \hat{z}) = ||z - \hat{z}||^2$. In practice, the average will be a sample average when the quantizer is applied to a sequence of real data, but the theory views the data as sharing a common probability density function (pdf) $f(z)$ corresponding to a generic random variable x and the average distortion becomes an expectation

$$D(q) = E[d(z, q(z))] = \sum_i \int_{S_i} d(z, y_i) f(z) dz$$

If a distortion is measured by squared error, $D(q)$ becomes the MSE.

It is desirable to have the average distortion as small as possible. There is a cost in terms of the number of bits required to describe the quantizer output to a decoder, however, and arbitrarily reliable reproduction will not be possible for digital storage and communication

media with finite capacity. A simple method for quantifying the cost for communications or storage is to assume that the quantizer "codes" an input z into a binary representation. If there are N possible levels and all of the binary representations or binary codewords have equal length, the binary vectors will need $\log N$ components or bits. Thus one definition of the rate of the code in bits per input sample is

$$R(q) = \log N$$

In summary, the goal of quantization is to encode the data from a source, characterized by its probability density function, into as few bits as possible (i.e., with low rate) in such a way that a reproduction may be recovered from the bits with as high quality as possible (i.e., with small average distortion). Clearly, there is a tradeoff between the two primary performance measures: average distortion and rate. There are also many alternative quantization techniques that permit a better tradeoff of distortion and rate; e.g., less distortion for the same rate, or vice versa.

The history of the theory and practice of quantization dates back to 1948. Two complementary approaches dominate the history and present state of the theory: Rate-distortion theory — Shannon's information - theoretic approach to source coding providing the foundations of information theory [72, 73]; high resolution (or high-rate or asymptotic) quantization theory [65, 8, 67]. In contrast to these two asymptotic theories, there is also a small but important collection of results that are not asymptotic in nature. The most important nonasymptotic results, however, are the basic optimality conditions and iterative-descent algorithms for quantizer design, by considering the problem with a squared-error distortion

measure:

$$\sum_i \int_{S_i} m(z)||z - y_i||^2 dz \tag{2.5}$$

where $m(z)$ is a density function defined on Euclidean space. What partition S and collection of vector C minimizes (2.5), the sum of the moments of inertia of the cells about the associated vectors? The solution is first derived by Steinhaus (1956) [74], then by Lloyd (1957), and later popularized by Max (1960) [56]. What we now consider to be the Lloyd-Max optimality conditions (centroid and nearest neighbor mapping) from fundamental principles (without variational techniques), proved the existence of a solution, and described the iterative descent algorithm for finding a good partition and vector collection. A complete survey [26] provides a historical tour of the development of the theory and practice of quantization over the past fifty year, a period encompassing almost the entire literature on the subject. Many of the efforts are described in a general way as "data-compression", "redundancy reduction", and "bandwidth-compression", and most of them rely on quantization and subsequent reconstruction of data.

Estimation with Quantized Measurement

Estimation with quantized measurements is of prime interest to designers of digital communication systems. In most problems in estimation theory, the implicit assumption is that the estimator has direct access to a sequence of possibly noise-corrupted measurements. However, in certain situations there is a constraint on the amount of information available to the estimator. In particular, if the estimator is not at the same location as the measurement sensor and receives information via a digital communication channel with finite capacity, then the measurement must be mapped to a finite set of symbols before transmission to the

estimator, i.e., the vector quantization is performed for the sensor measurements. The task is to have the efficient estimation on quantizated measurements. The results from classical estimation theory need to be extended to this case.

Given two random variables, x as the state to be estimated, and z as the measurement, the joint probability-density function is $p_{x,z}(\xi,\zeta)$. Given that z lies in a region denoted by A, then

$$p_{x,z|z\in A}(\xi,\zeta) = \begin{cases} 0, & \zeta \notin A \\ \frac{p_{x,z}(\xi,\zeta)}{P(z\in A)} & \zeta \in A \end{cases}$$

where $P(z \in A) = \int_A d\zeta \int d\xi p_{x,z}(\xi,\zeta)$. The classical estimation is based on the probability-density function $p_{x,z}(\xi,\zeta)$, while the estimation with quantized measurement will be based on $p_{x,z|z\in A}(\xi,\zeta)$.

For Bayesian estimation, Curry [19] derived the result that for some function $f(x)$

$$E[f(x)|z \in A] = E\{E[f(x)|z]|z \in A\}$$

This indicates that the estimation problem with quantized measurements may now be considered to consist of two operations: finding $E[f(x)|z]$, the conditional mean of $f(x)$ given a measurement (this function of z is the usual goal of the estimation problem without quantized measurements), and averaging this function of z conditioned on $z \in A$. This solution to the estimation problem is most beneficial in the sense that $E[x|z]$ is the MMSE estimator which has been computed for a wide variety of problems. All that remains, then, is to average this function over $z \in A$ when the measurements are quantized.

The natural questions to ask for estimation with quantized measurement are, what is the best way to perform the quantization so as to minimize the estimation error, and how much

capacity is needed to achieve a specified estimation accuracy. The goal of quantization in this case is to provide a good estimation accuracy, rather than to reconstruct the measurements.

There are some existing results solving the state estimation problem with the quantized measurements, such as Curry's MMSE estimator with quantized measurement [19], Zhang's research on how the unbiased estimator depends on the communication rates [83] and Wong's analysis on the performance of estimation algorithms with the data rate [78]. Also, for quantizer design, there are some works given by Ephraim and Gray [21], by Ayanoglu [1], and by Nair [64] on coding and estimating scheme for a Gauss-Markov process, etc.

Estimation Fusion with Communication Constraints

Mutlisensor estimation fusion is more complicated than classical estimation theory. It involves choosing the optimal estimation criteria at the fusion center, considering data processing at each local sensor and characterizing communication. Estimation fusion with communication constraints can be considered the case where the communication from each sensor to the fusion center is constrained to within certain bits. There are major research directions:

- Determine an optimal estimation fusion scheme which can achieve a pre-set estimation accuracy with the smallest number of sensors;
- Determine the sensor quantization and center estimation fusion scheme given a fixed fusion system structure.

The most recent research work given by Luo [58] belongs to the first direction which addresses the decentralized estimation in a bandwidth constrained sensor network. He studied

the parameter estimation problem with a linear sensor observation model

$$z_k = x + v_k \quad k = 1, \ldots, N$$

where noise terms are i.i.d., zero mean, and bounded to, say, $[-U, U]$ with a probability density function $f(x)$. Consider the case where the communication from each sensor to the fusion center is constrained to one binary bit. A linear estimator was considered

$$\hat{x} = \sum_{k=1}^{N} \alpha_k 1_S(z_k)$$

His main results show that whether the noise pdf is known or unknown, $O(\frac{U^2}{\epsilon})$ sensors are necessary and sufficient in order for the sensors and the fusion center to jointly estimate the unknown parameter with MSE($\hat{x}$)$\leq \epsilon$. This asymptotic result provides a good theoretical foundation for fusion system design.

Following the second research direction, multi-sensor quantization for estimation fusion has been studied by Lam and Reibman [33] and Gubner [27]. In [33], the MMSE and the maximum-a-posteriori probability (MAP) were considered as theoretical foundation for design of quantizers. Given $h(q_1(z_1), \ldots, q_N(z_N))$ as the estimation function at the fusion center, where $q_k(z_k) = \sum_i y_{ki} 1_{s_{ki}}(z_k)$, the distortion function corresponding to the MMSE criterion is given by

$$d[x, h(q_1(z_1), \ldots, q_N(z_N))] = [x - h(q_1(z_1), \ldots, q_N(z_N))]^2$$

And the distortion function corresponding to the MAP criterion is given by

$$d[x, h(q_1(z_1), \ldots, q_N(z_N))] = \begin{cases} 0 & \text{if } x = h(q_1(z_1), \ldots, q_N(z_N)) \\ 1 & \text{if } x \neq h(q_1(z_1), \ldots, q_N(z_N)) \end{cases}$$

Comparing the optimal designs using the MMSE and MAP criterion, the computation for the MAP fusion center is more difficult than computing the optimal MSE fusion center. On the other hand, computing the optimal partition regions for the local quantizers is simpler for the MAP criterion than the MMSE criterion. Therefore, it claimed that the choice of the two criteria depends mainly on the nature of the problem. Lam's quantizer design and estimation requires the knowledge of the entire joint distribution of state and all sensor observations. However, [27] proposed an linear estimation fusion, i.e., $h(q_1(z_1), \ldots, q_N(z_N)) = \sum_{k=1}^{N} q_k(z_k)$, where sensor quantization uses only bivariate probability distributions. But the algorithm itself involves solving several multi-variable equations to get y_{ki}.

All of above results are mainly confined to the estimation in a scalar case. In our research, based on BLUE fusion rule, we build the general state estimation fusion scheme by optimal quantizing the local sensor observations (Chapter 4).

2.2 Out-of-Sequence Measurement Update

In multisensor target tracking systems, measurements from the same target can arrive out of sequence. Such "out-of-sequence" measurement (OOSM) arrivals can occur even in the absence of communication time delays. The resulting problem — how to update the current state estimate with an "older" measurement — is a nonstandard estimation problem. Although OOSM update is a small problem from theoretical point of view, solving this problem has a significant impact in target tracking and many other engineering applications.

Suppose there is no constraint. Then the OOSM update problem can be simply solved by rerun the Kalman filter with all data from the OOSM occurrence time to its arrival time.

But we do have limited storage, how do we perform an efficient OOSM update with limited storage?

Many OOSM update algorithms have been proposed. They can be classified into two classes in terms of the delayed time of this OOSM: single-lag and multi-lag. Suppose the delayed measurement is z_d, where the sampling time is t_d ($t_{k-l} < t_d < t_k$). When $l = 1$, z_d is a one-step delayed (single-lag) OOSM; when $l > 1$, z_d is a multi-step delayed (multi-lag) OOSM.

2.2.1 Single-Lag OOSM Update

For the one-step delayed OOSM update problem, the initial work of [28, 9] presented an approximate solution to the OOSM problem. The proposed update algorithm only uses the information with the current state estimate $\hat{x}_{k|k} = E^*(x_k|z_k)$ and the delayed measurement z_d. According to the BLUE fusion theory, this solution for OOSM update is

$$\hat{x}_{k|k,d} = E^*(x_k|\hat{x}_{k|k}, z_d)$$

An optimal OOSM update algorithm for one-step delay problem is first derived in [2] where the information used includes $\hat{x}_{k|k}$, z_d and the state prediction $\hat{x}_{k|k-1}$. This a global optimal algorithm in the sense that the state estimation update can achieve the same performance as that of using all measurements from beginning up to now. Also, according to BLUE fusion theory, this update can be viewed as

$$\hat{x}_{k|k,d} = E^*(x_k|z^k, z_d)$$

where $z^k = [z_1, z_2, \ldots, z_k]'$.

2.2.2 Multi-Lag OOSM Update

The OOSM update for multi-step delayed problem is much harder than the one-step delay case. The main reason is that we have limited information storage. For one step globally optimal update, the information of z_k can be characterized through $\hat{x}_{k|k}$ and $\hat{x}_{k|k-1}$, so we can have an OOSM update algorithm having equivalent performance with in-sequence update using the information $\hat{x}_{k|k-1}$, z_d, z_k. By generalizing to the multi-step delayed OOSM update problem, we can not based only on current information such as $\hat{x}_{k|k}$ and $\hat{x}_{k|k-1}$ to characterize all observations from OOSM occurrence time to its arrival time. There is a tradeoff in terms of optimality and storage requirement. Several techniques have been proposed. "Algorithm Al1" and "Algorithm Bl1" proposed in [2] give up optimality by doing certain approximation in the algorithm and try to show that the performance does not suffer a lot for tracking scenarios. [14, 15, 12] presented multi-step OOSM update using augmented state smoothing. This is, let

$$X_k = \begin{bmatrix} x_{k-l+1} \\ \vdots \\ x_k \end{bmatrix}$$

be the new state vector. Then the OOSM z_d in terms of this new state X_k becomes the in-sequence measurement. The update can be handled directly by the state augmented Kalman filter. The idea has merit, but the computation and storage are intensive which limited the implementation of the algorithms. The suboptimal OOSM update of [59] gives an algorithm that can achieve

$$\hat{x}_{k|k,d} = E^*(x_k|\hat{x}_{k|k}, z_d)$$

2.2.3 OOSM Update in Clutter

The previous results for OOSM update are formulated for a Kalman filter to update the state. In this problem, the measurement at each sampling time is assumed to be target originated and no clutter or interference from other targets is considered. However, in real world multi-sensor multi-target tracking problem, measurements received at the fusion center can originate from targets or clutter, i.e., false alarms. The filter handles the measurement origin uncertainty via the so-called data association.

Considering solving the multi-sensor OOSM update problem in a cluttered environment, [60, 66, 14, 13] discussed several issues by applying their OOSM update algorithms in the case of measurement origin uncertainty. The existing optimality criterion for the OOSM update within the Kalman filter framework is no longer valid for the target tracking problem with measurement origin uncertainty. So, the optimality criterion for OOSM update in clutter is complicated. Most existing works are to study the performance of their OOSM update algorithm combined with certain data association algorithms.

Filtering is a special case of fusion. So our research for this OOSM update problem is based on the theoretical foundation of BLUE fusion with and without prior information. We formulate the OOSM update problem as a constrained estimation fusion problem. Based on different storage constraints, we develop a optimal OOSM updates [81, 61]. We also extend our results for OOSM update in a cluttered environment [79] and mainly study the performance of our algorithms combined with data association algorithms.

Chapter 3

Optimal Sensor Data Dimension Reduction for BLUE Fusion

Estimation fusion has been investigated for more than two decades. Most of existing results focus on building the optimal fusion rules for distributed fusion. In target tracking, the most commonly used distributed architecture, standard distributed fusion [53], only allows local estimates to be available at the fusion center. However, for much wider applications, distributed fusion has more general architecture than this configuration. In most standard distributed fusion, each local sensor provides linearly or nonlinearly processed measurements to the fusion center. How to define these local mappings is still an open problem. The reason is that in the fusion center, the fusion rule can be easily defined by optimizing certain criterion. For example, MMSE, BLUE, and WLS fusion rules are to minimize MSE covariance matrix, MSE covariance within the class of linear rules, and the weighted data fitting error, respectively. However, within a single local sensor, there is no concrete criterion to define the best mapping for processing the sensor observations because our final goal is to achieve

the optimal estimation at the fusion center rather than at the local sensor. For the standard estimation fusion architecture, local sensor uses the same optimality criterion as the fusion center and obtains the rule that maps local observations to the optimal local estimate. The benefit of this fusion structure is that local sensor can also have their own local optimal estimate. Limitations of this local sensor rule are obvious. The local sensor measurement mapping does not

- optimize estimates at the fusion center.
- consider the communication capacity between the local sensors and the fusion center.
- consider the processing ability at the fusion center.

An optimal sensor compression rule should yield the optimal fusion at the center with the constraint on the communication bandwidth between the fusion center and the local sensors, and the processing capability of the fusion center. The problem of data compression is thus a constrained optimization problem. Without constraints, the problem is trivial: the local sensors' observations can be directly sent to the fusion center, and then in the fusion center, a globally optimal solution is guaranteed. Note that the size of the raw data is often large, which has a high demand on communication bandwidth and the fusion center must have good computation capability and large memory. For example, if we consider a uniform quantization for every dimension of the data, the larger the data size, the more bits we need to send. Thus it is crucial to consider data compression when creating the local sensor rule. In this paper, we consider the BLUE fusion as the fusion rule. In the Gaussian case, BLUE is equivalent with MMSE. Without loss of generality, we assume the dimension of

each single observation is fixed. We consider compressing the local raw measurement to a lower dimension. Since it is very difficult to discuss general nonlinear transformations, our discussion is limited to linear rules. A linear rule is optimal if the fusion center achieves the optimal estimation under the constraints.

3.1 Problem Formulation

Consider a distributed system with a fusion center and n sensors (local stations), each connected to the fusion center.

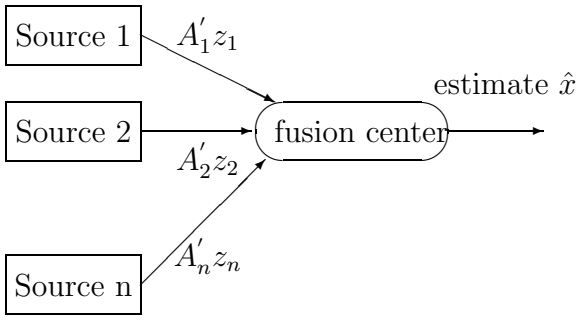


Figure 3.1: framework of estimation fusion

Denote by z_i (an n_i - dimensional vector) the observations of the ith sensor of the estimatee (i.e., the quantity to be estimated) x. For a distributed fusion system, only data-processed observations are sent to the fusion center, that is, a non-trivial mapping g_i $(i = 1, \ldots, n)$ is applied on local sensor measurements:

$$g_i : z_i \rightarrow y_i$$

After sensor data processing, the available data at the fusion center is $y = \{y_1, \ldots, y_n\}$. If

$y_i = z_i$, it is known as the centralized fusion, central-level fusion, or measurement fusion. If $y_i = \hat{x}_i$, that is, only local estimates (based on z_i) and its MSE matrix P_i are available at the fusion center, it is the standard distributed fusion. In general, the sensor rule $g_i(.)$ could be any linear or nonlinear mapping. In this chapter, only linear rules are considered, that is,

$$g_i : g_i(z_i) = A_i' z_i : R^{n_i} \to R^{d_i}$$

where $d_i \leq n_i$ and d_i is less than or equal to the dimensional requirement for the ith sensor due to communication or processing constraints of the system. After data processing, sensor i sends y_i to the fusion center, which has a reduced dimension. Since the original data size is reduced with the local processing, in the sequel, we call this linear transformation the local sensor data compression.

At the fusion center, only linear unbiased estimation fusion is considered; that is, we consider the most commonly used linear estimation method. With our sensor data compression, estimation fusion in center can only base on the compressed data $y = \{y_1, \ldots, y_n\}$, which is

$$y = A'z$$

where

$$A = \text{diag}[A_1, \ldots, A_n]$$

$$z = [z_1, \cdots, z_n]'$$

Note that $C_{xy} = C_{xz}A$ and $C_y = A'C_zA$, and then

$$\hat{x} = \bar{x} + C_{xz}A(A'C_zA)^{-1}A'(z - \bar{z}) \tag{3.1}$$

$$\text{MSE}(\hat{x}) = C_x - C_{xz}A(A'C_zA)^{-1}A'C_{xz}'$$

where

$$C_{xz} = \mathrm{cov}(x, z) = [C_{xz_1}, \cdots, C_{xz_n}]'$$

$$C_z = \mathrm{cov}(z) = \begin{bmatrix} C_{z_1} & C_{z_1 z_2} & \cdots & C_{z_1 z_n} \\ C_{z_2 z_1} & C_{z_2} & & C_{z_2 z_n} \\ \vdots & & \ddots & \vdots \\ C_{z_n z_1} & & & C_{z_n} \end{bmatrix}$$

Then optimal sensor data compression, i.e., finding the optimal matrix $\tilde{A}$ which satisfies the dimensional requirement, is to solve the following constrained optimization problem:

$$\begin{aligned} A &= \arg\min\{\mathrm{MSE}(\hat{x})\} \\ &= \arg\min\{C_x - C_{xz}A(A'C_zA)^{-1}A'C'_{xz}\} \\ &= \arg\min\{\mathrm{tr}[C_x - C_{xz}A(A'C_zA)^{-1}A'C'_{xz}]\} \\ &= \arg\max_A J(A) \\ s.t.\ A &= \mathrm{diag}[A_1, \ldots, A_n] \text{ and } \dim(A_i) = n_i \times d_i \end{aligned} \tag{3.2}$$

where

$$J(A) = \mathrm{tr}[C_{xz}A(A'C_zA)^{-1}A'C'_{xz}] \tag{3.3}$$

The last two equality of (3.2) holds because of the uniqueness of optimal estimation $\hat{x}$ corresponding to the optimal $\tilde{A}$ (We will clearly see this in the following section). If $A = I$, $y = \{z_1, z_2, \ldots, z_n\}$, raw data are sent to the fusion center. This is the centralized fusion and the estimation is globally optimal based on all measurements observed from local sensors. Generally

$$J(A) \leq J(I) = \mathrm{tr}[C_{xz}C_z^+C'_{xz}] \tag{3.4}$$

The equality (3.4) holds when the sensor data compression has no performance loss for BLUE fusion.

3.2 Optimal Compression of a Single-Sensor Data for Estimation

Assume that estimation is only based on the compressed data from a single sensor. In this system, the local sensor collects the observations and processes the data. The compressed data are sent to the center for estimation. We first discuss this case because it is a good starting point to derive the optimal solution.

In this case, $n = 1$, $z = z_1$ and $A' = A'_1$ and thus $y = y_1 = A'_1 z_1$ compresses the data from dimension n_1 to d_1 ($d_1 < n_1$). According to (3.2), the optimal compression is the solution of the following constrained optimization problem

$$A = \arg\max_A J(A) = \arg\max_A \text{tr}[C_{xz} A (A' C_z A)^{-1} A' C'_{xz}] \tag{3.5}$$

$$s.t. \;\; \dim(A_1) = n_1 \times d_1$$

As before, $(A'C_zA)^{-1}$ can be replaced with the MP inverse $(A'C_zA)^+$ if the inverse does not exist. In general, however, we can always find some A such that $A'C_zA$ is nonsingular. Because when $A'C_zA$ is not invertible, we can always find a new $\bar{A}$ with a lower dimension such that $\bar{A}'C_z\bar{A}$ becomes nonsingular as show next.

Lemma 2: *If $A'C_zA$ is singular, we can find a lower dimensional matrix $\bar{A}$ such that $\bar{A}'C_{z_1}\bar{A}$ is invertible and satisfies*

$$J(\bar{A}) = J(A)$$

□

Proof. Because $A'C_zA$ is a positive semi-definite matrix, suppose the rank of $A'C_zA$ is N. There exists a unitary matrix S ($S' = S^{-1}$) such that

$$S'A'C_zAS = \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix}$$

where Λ =diag$[\lambda_1, \ldots, \lambda_n]$, λ_i $(i = 1, \ldots, N)$ are the non-zero eigenvalues of $A'C_zA$. Now with $D = [I_N, 0]$,

$$D'S'A'C_zASD = \Lambda$$

Let $\bar{A} = ASD$. Then $\bar{A}$ has a lower dimension and $\bar{A}'C_z\bar{A} = \Lambda$ is nonsingular. Also,

$$\begin{aligned} J(\bar{A}) &= \text{tr}[C_{xz}\bar{A}(\bar{A}'C_z\bar{A})^{-1}\bar{A}'C_{xz}'] \\ &= \text{tr}[C_{xz}\bar{A}\Lambda^{-1}\bar{A}'C_{xz}'] \\ &= \text{tr}[\bar{A}'C_{xz}'C_{xz}\bar{A}\Lambda^{-1}] \\ &= \text{tr}[D'S'A'C_{xz}'C_{xz}ASD\Lambda^{-1}] \\ &= \text{tr}[(S'A'C_{xz}'C_{xz}AS)(D\Lambda^{-1}D')] \\ &= \text{tr}[(S'A'C_{xz}'C_{xz}AS)\begin{bmatrix} \Lambda^{-1} & 0 \\ 0 & 0 \end{bmatrix}] \end{aligned}$$

According to the property of the MP inverse: $(PAQ)^+ = Q'A^+P'$ for unitary matrices P

and Q,

$$\begin{aligned} J(SA) &= \text{tr}[C_{xz}AS(S'A'C_zAS)^+S'A'C'_{xz}] \\ &= \text{tr}[C_{xz}ASS'(A'C_zA)^+SS'A'C'_{xz}] \\ &= \text{tr}[C_{xz}A(A'C_zA)^+A'C'_{xz}] \\ &= J(A) \end{aligned}$$

On the other hand,

$$\begin{aligned} J(SA) &= \text{tr}[C_{xz}AS(\begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix})^+S'A'C'_{xz}] \\ &= \text{tr}[C_{xz}AS\begin{bmatrix} \Lambda^{-1} & 0 \\ 0 & 0 \end{bmatrix}S'A'C'_{xz}] \\ &= \text{tr}[(S'A'C'_{xz}C_{xz}AS)\begin{bmatrix} \Lambda^{-1} & 0 \\ 0 & 0 \end{bmatrix}] \end{aligned}$$

Thus

$$J(A) = J(SA) = J(\bar{A})$$

■

Lemma 3: *For any nonsingular matrix D, we have*

$$J(A) = J(AD)$$

□

Proof. Based on Lemma 2, since $A'C_zA$ is invertible,

$$(D'A'C_zAD)^{-1} = D^{-1}(A'C_zA)^{-1}(D')^{-1}$$

and thus

$$\begin{aligned} J(AD) &= \mathrm{tr}[C_{xz}AD(D'A'C_zAD)^{-1}D'A'C'_{xz}] \\ &= \mathrm{tr}[C_{xz}ADD^{-1}(A'C_zA)^{-1}(D')^{-1}D'A'C'_{xz}] \\ &= \mathrm{tr}[C_{xz}A(A'C_zA)^{-1}A'C'_{xz}] \\ &= J(A) \end{aligned}$$

■

According to Lemma 2, in the following, we only consider the case that $A'C_zA$ is invertible.

Theorem 2: *The optimal solution $\tilde{A}$ of (3.5) is*

$$\tilde{A} = C_z^+ \bar{K} \tag{3.6}$$

where the column vectors of $\bar{K}$ are the eigenvectors corresponding to the d_1 largest eigenvalues of $(C'_{xz}C_{xz}C_z^+)$, and J is the sum of these d_1 eigenvalues. □

Proof. This proof follows from the one in [24]. Since $J(A)$ is a continuous function, the maximum point of $J(A)$ should satisfy

$$\frac{d}{dA}[J(A)] = 0$$

i.e.

$$\begin{aligned} &\frac{d}{dA}\mathrm{tr}[C_{xz}A(A'C_zA)^{-1}A'C'_{xz}] \\ &= \frac{d}{dA}\mathrm{tr}[A'C'_{xz}C_{xz}A(A'C_zA)^{-1}] \\ &= -2C_zA(A'C_zA)^{-1}(A'C'_{xz}C_{xz}A)(A'C_zA)^{-1} + 2C'_{xz}C_{xz}A(A'C_zA)^{-1} \\ &= 0 \end{aligned}$$

So

$$C'_{xz}C_{xz}A = C_zA(A'C_zA)^{-1}(A^TC'_{xz}C_{xz}A)$$

Two symmetric matrices $(A'C_{xz}^TC_{xz}A)$ and $(A'C_zA > 0)$ can be diagonalize simultaneously into Λ_m =diag$[\lambda_1, \ldots, \lambda_m]$ and I_m:

$$B'(A'C'_{xz}C_{xz}A)B = \Lambda_m$$

$$B'(A'C_zA)B = I_m$$

where B is an $m \times m$ nonsingular square matrix and B^{-1} exists. It can be shown that

$$\begin{aligned} &C_zA(A'C_zA)^{-1}(A'C_{xz}^TC_{xz}A) \\ &= C_zA[(B^{-1})'B^{-1}]^{-1}[(B^{-1})'\Lambda_mB^{-1}] \\ &= C_zAB\Lambda_mB^{-1} \end{aligned}$$

So

$$C'_{xz}C_{xz}A = C_zAB\Lambda_mB^{-1}$$

$$C'_{xz}C_{xz}AB = C_zAB\Lambda_m$$

and because of the identity $C_{xz}C_z^+C_z = C_{xz}$, we have

$$C'_{xz}C_{xz}C_z^+C_zAB = C_zAB\Lambda_m \tag{3.7}$$

Let

$$\bar{K} = C_zAB$$

Then (3.7) can be expressed as

$$(C'_{xz}C_{xz}C_z^+)\bar{K} = \bar{K}\Lambda_m$$

This is the eigen equations for $(C'_{xz}C_{xz}C_z^+)$. Thus the components of Λ_m and the column vectors of $\bar{K}$ are the m eigenvalues and eigenvectors of $(C'_{xz}C_{xz}C_z^+)$. Note that

$$\begin{aligned} J(A) &= J(AB) \\ &= \mathrm{tr}[B'A'C'_{xz}C_{xz}AB(B'A'C_zAB)^{-1}] \\ &= \mathrm{tr}[\Lambda_m I_m^{-1}] \\ &= \lambda_1 + \lambda_2 + \cdots + \lambda_{m_1} \end{aligned}$$

where λ_i are also the eigenvalues of $C'_{xz}C_{xz}C_z^+$. Thus, we can maximize $J(A)$ by selecting the largest m eigenvalues. The transformation is by projecting z onto these m eigenvector. We can form an m- dimensional subspace which is spanned by these m eigenvectors. Then, J is the summation of the corresponding m eigenvalues. Further application of any $m \times m$ nonsingular linear transformation could not change the value of J, i.e. $J(A) = J(AD)$ for any nonsingular matrix D. Also $J(C_z^+C_zA) = J(A)$, because

$$\begin{aligned} &J(C_z^+C_zA) \\ &= \mathrm{tr}[C_{xz}C_z^+C_zA(A'C_zC_z^+C_zC_z^+C_zA)^+A'C_zC_z^+C'_{xz}] \\ &= \mathrm{tr}[C_{xz}A(A'C_zA)^+A'C'_{xz}] = J(A) \end{aligned}$$

Since

$$C_z^+\bar{K} = C_z^+C_zAB$$

The optimal solution of $\tilde{A}$ is

$$\tilde{A} = C_z^+\bar{K}$$

■

From Lemma 3, the optimal solution of (3.5) is not unique [24], because if $\tilde{A}$ is an optimal solution, $\tilde{A}D$ is another optimal solution for any nonsingular matrix D. However different optimal solutions $\tilde{A}$ and $\tilde{A}D$ correspond to the same estimator (3.5):

$$E^*(x|\tilde{A}z) = E^*(x|\tilde{A}Dz) \tag{3.8}$$

This can be shown easily from the definition of BLUE estimator (3.1). As shown in the proof of Theorem 2, the optimal solution set Ω contains all elements $A \in \Omega$ with the relation $A = \tilde{A}B$ for any nonsingular matrix B. So (3.8) verifies the uniqueness of the optimal estimator $\hat{x}$ corresponding to Ω.

Suppose $\text{rank}(C'_{xz}C_{xz}C_z^+) = m$, $m \leq \min\{n, n_x\}$, where n is the dimension of observation z and n_x is the dimension of the estimatee x. So $C'_{xz}C_{xz}C_z^+$ has only m nonzero eigenvalues. This means that if the sensor is allowed to send a vector of a dimension larger than m, we can always project the observation into a subspace of the lower dimension m, in that we can choose the optimal A such that $d_1 = m$. It should be realized that there is no information loss with this data compression. Also, $m = n_1$ means the observation dimension is low, not larger than the dimension of the estimatee. Since $J(A)$ is the sum of d-largest eigenvalues, and $J(I)$ is the sum of all nonzero n eigenvalues of $(C'_{xz}C_{xz}C_z^+)$. It indicates that if we compress the observations with $d < n$, then $J(A) < J(I)$. In this situation, we can not get the data compressed observation without information loss (performance deterioration). If $d < m$, there is information loss even for the optimal compression $y = A'z$, that is, the fusion center can not achieve the globally optimal estimation using the optimally compressed data:

$$J(A) < J(I)$$

With the optimal sensor data compression, we compress the raw measurements and extract the most important information for estimating x at the fusion center.

3.3 Optimal Data Compression for Estimation Fusion

For multiple-sensor estimation fusion ($n > 1$), $z = [z_1, z_2, \ldots, z_n]'$ and A =diag$[A_1, \ldots, A_n]$ which build the linear mapping $y = A'z$ compressing each local sensor measurement from higher dimension n_i to d_i ($d_i < n_i$), $i = 1, \ldots, n$. In this section, we first discuss three special cases which have simple optimal solutions for multiple sensor data compression. The most general and difficult case is discussed last.

3.3.1 Uncorrelated sensors

In this case, $C_{z_i z_j} = 0$ for any $i \neq j$ where $i, j = 1, \ldots, n$. Since C_z and A are all block diagonal matrices,

$$A'C_zA = \text{diag}[A_1'C_{z_1}A_1, \ldots, A_n'C_{z_n}A_n]$$

so

$$(A'C_zA)^{-1} = \text{diag}[(A_1'C_{z_1}A_1)^{-1}, \ldots, (A_n'C_{z_n}A_n)^{-1}]$$

also

$$A'C_{xz}'C_{xz}A = \begin{bmatrix} A_1'C_{xz_1}'C_{xz_1}A_1 & \cdots & A_1'C_{xz_1}'C_{xz_n}A_n \\ \vdots & \ddots & \vdots \\ A_n'C_{xz_n}'C_{xz_1}A_1 & \cdots & A_n'C_{xz_n}'C_{xz_n}A_n \end{bmatrix}$$

according to the property of trace, the objective function (3.3) can be rewritten as

$$\begin{aligned} J(A) &= \mathrm{tr}[A'C'_{xz}C_{xz}A(A'C_zA)^{-1}] \\ &= \sum_{i=1}^{n} \mathrm{tr}[A'_iC'_{xz_i}C_{xz_i}A_i(A'_iC_{z_i}A_i)^{-1}] \\ &= \sum_{i=1}^{n} J_i(A_i) \end{aligned}$$

where

$$J_i(A_i) = \mathrm{tr}[A'_iC'_{xz_i}C_{xz_i}A_i(A'_iC_{z_i}A_i)^{-1}]$$

Now the constrained optimization problem (3.2) concerning $J(A)$ can be divided into the following n individual constrained optimization problems

$$A_i = \arg\max_{A_i} J_i(A_i),\ i = 1, 2, \ldots, n \tag{3.9}$$

$$s.t.\ \dim(A_i) = n_i \times d_i$$

For each A_i, the optimization problem is the same as the single sensor case, so we have the optimal solution $\tilde{A}_i = C^+_{z_i}\bar{K}_i$, where the column vectors of $\bar{K}_i$ are the eigenvectors corresponding to the d_i largest eigenvalues of $C'_{xz_i}C_{xz_i}C^+_{z_i}$ or $\tilde{A}D$ with any nonsingular matrix D, and J_i is the sum of the d_i eigenvalues.

3.3.2 Linear sensor observation model with uncoupled noises

All sensor observation model are linear, that is

$$z_i = H_ix + v_i \quad i = 1, \ldots, n$$

where $C_{v_i} = R_i$ and $C_{v_i v_j} = 0$ for any $i \neq j$ where $i, j = 1, \ldots, n$. Let

$$H = [H_1', H_2', \ldots, H_n']'$$

$$v = [v_1', v_2', \ldots, v_n']'$$

then

$$z = Hx + v$$

and

$$C_z = HC_xH' + R$$

$$C_{xz} = C_xH'$$

With sensor data compression, the available information at fusion center is $y = A'z$, the estimation error covariance at fusion center will be

$$\begin{aligned}
\text{MSE}(\hat{x}) &= C_x - C_{xz}A(A'C_zA)^{-1}A'C_{xz}' \\
&= C_x - C_xH'A(A'HC_xH'A + A'RA)^{-1}A'HC_x \\
&= [C_x^{-1} + H'A(A'RA)^{-1}A'H]^{-1}
\end{aligned}$$

So the optimal A is the solution of following problem

$$\begin{aligned}
A &= \arg\min\{\text{MSE}(\hat{x})\} \\
&= \arg\min\{[C_x^{-1} + H'A(A'RA)^{-1}A'H]^{-1}\} \\
&= \arg\max\{C_x^{-1} + H'A(A'RA)^{-1}A'H\} \\
&= \arg\max\{\text{tr}[H'A(A'RA)^{-1}A'H]\}
\end{aligned}$$

since

$$\mathrm{tr}[H'A(A'RA)^{-1}A'H] = \sum_{i=1}^{n} \mathrm{tr}[A_i'H_iH_i'A_i(A_i'R_iA_i)^{-1}] = \sum_{i=1}^{n} J_i(A_i)$$

thus

$$A_i = \arg\max_{A_i} J_i(A_i),\ i = 1, 2, \ldots, n \tag{3.10}$$

$$s.t.\ \dim(A_i) = n_i \times d_i$$

Now for each A_i, the optimization problem is the same as the single sensor case, so we have the optimal solution $\tilde{A}_i$, where the column vectors of $\tilde{A}_i$ are the eigenvectors corresponding to the d_i largest eigenvalues of $R_i^{-1}H_iH_i'$ or $\tilde{A}_iD$ with any nonsingular matrix D, and J_i is the sum of the d_i eigenvalues.

3.3.3 Sensor dimensional requirement larger than rank of $C_{xz}'C_{xz}C_z^+$

When the dimensional requirement for each sensor is larger than m =rank$(C_{xz}'C_{xz}C_z^+)$, we can simply create the sensor data compression $y_i = A_i'z_i$ such that $d_i = m$ $(i = 1, \ldots, n)$. By concerning the estimation in fusion center, we will see there is no information loss for the estimation fusion in center with this local sensor data compression.

Since all A_i have the same number of columns, we can define

$$A_{\mathrm{new}} = [A_1', A_2', \ldots, A_n']'$$

Then the objective function (3.3) is equivalent to

$$J(A_{\mathrm{new}}) = \mathrm{tr}[A_{\mathrm{new}}'C_{xz}'C_{xz}A_{\mathrm{new}}(A_{\mathrm{new}}'C_zA_{\mathrm{new}})^{-1}]$$

Now the constrained optimization problem for this case is the same as that of the single

sensor case, that is,

$$A_{\text{new}} = \arg\max_{A} J(A_{\text{new}})$$
$$s.t.\ \dim(A_{\text{new}}) = (\sum_{i=1}^{n} n_i) \times m$$

Then the optimal solution is $\tilde{A}_{\text{new}} = C_z^+ \bar{K}$, where the m column vectors of $\bar{K}$ are the eigenvectors corresponding to the m nonzero eigenvalues of $C'_{xz} C_{xz} C_z^+$ or $\tilde{A}D$ with any non-singular matrix D, and J is the sum of the m eigenvalues. Obviously $J(A_{\text{new}}) = \text{tr}(C'_{xz} C_{xz} C_z^+) = J(I)$, so there is no information loss with this sensor data compression.

Observations may have a higher dimension than that of the estimatee. So there is no estimation accuracy degradation if all sensors compress their observations to the same dimension as estimatee. However this data compression needs to consider correlation between sensors. This means that if a sensor compresses its observations by only considering its local information, generally, there is information loss. This is related to the fact that the standard distributed fusion can not achieve the same performance as the centralized fusion in many cases.

3.3.4 Arbitrary sensor dimensional requirement

An arbitrary sensor dimensional requirement for a distributed system is the general case. Each sensor has different dimensional requirement according to the system restriction, and some of sensors have a demanding dimensional requirement of $d_i < m = \text{rank}(C'_{xz} C_{xz} C_z^+)$. So we can not construct the same sensor data compression as above, and data compression generally has information loss.

According to (3.2), the optimal sensor compression is the solution of the following con-

strained optimization problem:

$$A = \arg\max_{A} J(A)$$

$$A = \text{diag}[A_1, \ldots, A_n] \text{ and } \dim(A_i) = n_i \times d_i$$

where

$$J(A) = \text{tr}[C_{xz} A (A' C_z A)^{-1} A' C'_{xz}]$$

Although the objective function $J(A)$ has the same form as the single-sensor case, we can not directly borrow the solution there, because there is one more constraint for the matrix A which requires A to be block diagonal. In the single-sensor case, if we treat the objective function $J(A_1)$ as a single variable function of A_1, the objective function $J(A_1, A_2, \ldots, A_n)$ for the n-sensor case should be a function of multiple variables. Now the problem become a multivariable optimization problem. It is almost impossible to directly get an explicit solution of this particular problem . In the following, we present an algorithm based on the Gauss-Seidel iteration to search for the optimal solution.

For each $i = 1, \ldots, n$, we denote

$$y_{(i)} = [y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_n]'$$

$$z_{(i)} = [z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n]'$$

$$A_{(i)} = \text{diag}[A_1, \ldots, A_{i-1}, A_i, \ldots, A_n]$$

At the fusion center, based on the BLUE fusion and the recursive BLUE, we have

$$\hat{x}^{\text{BLUE}} = E^*(x|y) = E^*(x|y_{(i)}, y_i)$$

$$= E^*(x|y_{(i)}) + C_{x\tilde{y}_{i|y_{(i)}}} C^{-1}_{\tilde{y}_{i|y_{(i)}}} \tilde{y}_{i|y_{(i)}}$$

$$\text{MSE}(\hat{x}^{\text{BLUE}}) = C_{x|y_{(i)}} - C_{x\tilde{y}_{i|y_{(i)}}} C^{-1}_{\tilde{y}_{i|y_{(i)}}} C'_{x\tilde{y}_{i|y_{(i)}}}$$

where with $\tilde{y}_{(i)} = y_{(i)} - \bar{y}_{(i)}$ and $\tilde{z}_{(i)} = z_{(i)} - \bar{z}_{(i)}$

$$\begin{aligned}\tilde{y}_{i|y_{(i)}} &= y_i - E^*(y_i|y_{(i)}) \\ &= y_i - \bar{y}_i - C_{y_i y_{(i)}} C_{y_{(i)}}^{-1} \tilde{y}_{(i)} \\ &= A_i'(z_i - \bar{z}_i) - A_i' C_{z_i z_{(i)}} A_{(i)} (A_{(i)}' C_{z_{(i)}} A_{(i)})^{-1} A_{(i)}' \tilde{z}_{(i)}\end{aligned}$$

So

$$\mathrm{MSE}(\hat{x}^{\mathrm{BLUE}}) = C_{x|y_{(i)}} - C_{x\tilde{z}_{i|y_{(i)}}} A_i [A_i' C_{\tilde{z}_{i|y_{(i)}}} A_i]^{-1} A_i' C_{x\tilde{z}_{i|y_{(i)}}}'$$

where

$$\begin{aligned}C_{x|y_{(i)}} &= C_x - C_{xz_{(i)}} A_{(i)} (A_{(i)}' C_{z_{(i)}} A_{(i)})^{-1} A_{(i)}' C_{xz_{(i)}}' \\ C_{x\tilde{z}_{i|y_{(i)}}} &= C_{xz_i} - C_{xz_{(i)}} A_{(i)} (A_{(i)}' C_{z_{(i)}} A_{(i)})^{-1} A_{(i)}' C_{z_i z_{(i)}}' \\ C_{\tilde{z}_{i|y_{(i)}}} &= C_{z_i} - C_{z_i z_{(i)}} A_{(i)} (A_{(i)}' C_{z_{(i)}} A_{(i)})^{-1} A_{(i)}' C_{z_i z_{(i)}}'\end{aligned}$$

Denote

$$J_i(A_i) = \mathrm{tr}[C_{x\tilde{z}_{i|y_{(i)}}} A_i (A_i' C_{\tilde{z}_{i|y_{(i)}}} A_i)^{-1} A_i' C_{x\tilde{z}_{i|y_{(i)}}}']$$

Then

$$\begin{aligned}J(A) &= J(A_{(i)}, A_i) \\ &= \mathrm{tr}[C_{xz_{(i)}} A_{(i)} (A_{(i)}' C_{z_{(i)}} A_{(i)})^{-1} A_{(i)}' C_{xz_{(i)}}'] + J_i(A_i)\end{aligned}$$

The objective function J is a multivariable function of $A_1, \ldots, A_n$. According to the necessary condition for multi-variable extreme point, A_i should be the optimal solution of the following optimization problem

$$A_i = \arg\max_{A_i} J_i(A_i)$$

$$s.t.\ \dim(A_i) = n_i \times d_i$$

Now it becomes the same problem as optimal sensor data compression for the single-sensor case. Then the optimal solution is

$$\tilde{A}_i = C^+_{\tilde{z}_{i|y_{(i)}}} \bar{K}_i$$

where the column vector of $\bar{K}_i$ is the d_i eigenvectors corresponding to the d_i largest eigenvalues of $C'_{x\tilde{z}_{i|y_{(i)}}} C_{x\tilde{z}_{i|y_{(i)}}} C^+_{\tilde{z}_{i|y_{(i)}}}$. Note, however, that the optimal solution $\tilde{A}_i$ depends on the value of $A_{(i)}$ through $\tilde{z}_{i|y_{(i)}}$. It is not easy to give an explicit solution for each A_i. In the following, we give an iterative algorithm to search for the optimal $\tilde{A}$.

Define the operator $\Gamma = (\Gamma_1, \ldots, \Gamma_n)$, for $i = 1, \ldots, n$

$$\Gamma_i(A_1, \ldots, A_{i-1}, A_{i+1}, \ldots, A_n) = C^+_{\tilde{z}_{i|y_{(i)}}} \bar{K}_i$$

Then we can construct a Gauss-Seidel iteration to search for the optimal solution $(A_1, \ldots, A_n)$. Suppose the nonzero initial value is $(A_1^{(0)}, \ldots, A_n^{(0)})$, at each iteration $k = 1, 2, \ldots$

$$\begin{aligned} A_1^{(k+1)} &= \Gamma_1(A_2^{(k)}, \ldots, A_n^{(k)}) \\ A_2^{(k+1)} &= \Gamma_2(A_1^{(k+1)}, A_3^{(k)}, \ldots, A_n^{(k)}) \\ &\vdots \\ A_n^{(k+1)} &= \Gamma_n(A_1^{(k+1)}, \ldots, A_{n-1}^{(k+1)}) \end{aligned} \tag{3.11}$$

After each iteration (3.11), we have

$$A^{(k)} = \mathrm{diag}[A_1^{(k)}, \ldots, A_n^{(k)}]$$

The iteration stops once the objective function $J(A)$ satisfies

$$J(A^{(k+1)}) - J(A^{(k)}) < \epsilon$$

where ϵ is some predetermined small number.

As shown in Theorem 2, in the Gauss-Seidel iteration, for each $i = 1, \ldots, n$, it is obvious that

$$J(A_1^{(k+1)}, \ldots, A_i^{(k+1)}, A_{i+1}^{(k)}, \ldots, A_n^{(k)})$$
$$\geq J(A_1^{(k+1)}, \ldots, A_{i-1}^{(k+1)}, A_i^{(k)}, \ldots, A_n^{(k)})$$

i.e.,

$$J(A^{(k)}) \geq J(A^{(k-1)})$$

The equality holds if and only if $A_i^{(k+1)} = A_i^{(k)} D_i$, where D_i is any nonsingular matrix. Thus the series $\{J(A^{(k)})\}$ is monotonically nondecreasing.

Theorem 3: *Function $J(A)$ has an upper bound.* □

Proof. Because

$$C_{xz}A(A'C_zA)^{-1}A'C'_{xz}$$
$$= C'_{xz}C_z^+C_z^{\frac{1}{2}}C_z^{\frac{1}{2}}A(A'C_z^{\frac{1}{2}}C_z^{\frac{1}{2}}A)^{-1}A'C_z^{\frac{1}{2}}C_z^{\frac{1}{2}}C_z^+C_{xz}$$
$$= [(A'C_z^{\frac{1}{2}})(C_z^{\frac{1}{2}}C_z^+C_{xz})]'[(A'C_z^{\frac{1}{2}})(A'C_z^{\frac{1}{2}})']^{-1}[(A'C_z^{\frac{1}{2}})(C_z^{\frac{1}{2}}C_z^+C_{xz})]$$

according to Cauchy-Schwarz inequality:

$$C_{xz}A(A'C_zA)^{-1}A'C'_{xz} \leq (C_z^{\frac{1}{2}}C_z^+C_{xz})'(C_z^{\frac{1}{2}}C_z^+C_{xz})$$
$$= C'_{xz}C_z^+C_{xz}$$

so

$$J(A) = \mathrm{tr}[C_{xz}A(A'C_zA)^{-1}A'C'_{xz}] \leq \mathrm{tr}[C'_{xz}C_z^+C_{xz}]$$

therefore the theory holds. ■

According to Theorem 3, series $\{J(A^{(k)})\}$ is the monotonically increasing and has an upper bound, and so it has a limit ζ:

$$\lim_{k\to\infty} J(A^{(k)}) = \zeta$$

Combined with the continuity of function $J(A^{(k)})$, there exists $\tilde{A}$ such that

$$\lim_{k\to\infty} J(A^{(k)}) = J(\tilde{A})$$

So $\tilde{A}$ is a limit of series $\{A^{(k)}\}$.

Theorem 4: *The limit point $\tilde{A}$ in Theorem 3 is a stationary point of the objective function, i.e., $\frac{\partial}{\partial A_i} J(A)|_{A=\tilde{A}} = 0$ for $i = 1, \ldots, n$.* □

Proof.

$$J(A^{(k)}) = J(A^{(k-1)})$$

means

$$\begin{aligned} &J(A_1^{(k)}, \ldots, A_i^{(k)}, A_{i+1}^{(k-1)}, \ldots, A_n^{(k-1)}) \\ &= J(A_1^{(k)}, \ldots, A_{i-1}^{(k)}, A_i^{(k-1)}, \ldots, A_n^{(k-1)}), \quad i = 1, \ldots n \end{aligned}$$

i.e.

$$J_i(A_i^{(k)}) = J_i(A_i^{(k-1)}),\ i = 1, \ldots n$$

Thus $A_i^{(k)}$ and $A_i^{(k-1)}$ should be in the same eigen-space, and there exists a nonsingular matrix B_i such that

$$A_i^{(k)} = B_i A_i^{(k-1)}$$

Also because

$$J'_{A_i}(A_1^{(k)}, \ldots, A_i^{(k)}, A_{i+1}^{(k-1)}, \ldots, A_n^{(k-1)}) = 0$$

for $i = 1, \ldots n$, we have

$$J'_{A_i}(B_1 A_1^{(k-1)}, \ldots, B_i A_i^{(k-1)}, A_{i+1}^{(k-1)}, \ldots, A_n^{(k-1)}) = 0$$

It is easy to get that

$$J'_{A_i}(B_1 A_1^{(k-1)}, \ldots, B_i A_i^{(k-1)}, A_{i+1}^{(k-1)}, \ldots, A_n^{(k-1)})$$

$$= J'_{A_i}(A^{(k-1)}) = 0 \qquad i = 1, \ldots n$$

So $\tilde{A} = A^{(k-1)}$ is the stationary point. ■

Theorem 4 is important. It implies that when the iteration ends, the solution is a critical point of the objective function J. It may be a maximizer because extreme points are also critical points. Unfortunately we are not able to provide further theoretical results concerning the convergence of $\{A^{(k)}\}$ to the globally optimal solution at this stage, because the objective function $J(A)$ is too complex to analysis for block diagonal matrix A. However, we found from simulation that almost often we achieve the globally optimal solution, which means starting from different initial points, the iteration will end with the same value of J and the same estimator $\hat{x}$.

3.4 Simulation

Several simple numerical examples are given in this section to verify formulas presented and the optimality of our Gauss-Seidel iteration based search solutions. All examples are for the following multi-sensor target tracking system:

We consider a constant-velocity moving target in 2 dimensional $x-y$ space. The estimatee $x(t)$ is the state process, consisting of position and velocity components: $x = [\mathsf{x}, \dot{\mathsf{x}}, \mathsf{y}, \dot{\mathsf{y}}]'$. We

set up 6 observation stations S_i, $i = 1, ..., 6$, with 3 different types. The first two stations have a linear observation model: $z_i(t) = Hx(t) + v_i(t)$, where $H = I_4$. The third and fourth stations have the same linear observation model but with $H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$, that is, they only observe position of the target. The last 2 stations have a nonlinear observation model:

$$z_i(t) = \begin{bmatrix} \sqrt{\mathsf{x}(t)^2 + \mathsf{y}(t)^2} \\ \tan^{-1}(\frac{\mathsf{x}(t)}{\mathsf{y}(t)}) \end{bmatrix} + v_i(t)$$

$v_i(t)$ is zero-mean white noise with covariance R_i =cov$(v_i(t))$, $i = 1, .., 6$.

$$R_1 = 10 \begin{bmatrix} 1000 & 20 & 10 & 5 \\ 20 & 100 & 5 & 10 \\ 10 & 5 & 4000 & 50 \\ 5 & 10 & 50 & 300 \end{bmatrix},$$

$$R_2 = 1000 \begin{bmatrix} 40 & 2 & 0 & 0 \\ 2 & 6 & 0 & 0 \\ 0 & 0 & 10 & 1 \\ 0 & 0 & 1 & 5 \end{bmatrix}$$

$$R_3 = 1000 \begin{bmatrix} 20 & 1 \\ 1 & 30 \end{bmatrix}, \; R_4 = 1000 \begin{bmatrix} 15 & 4 \\ 4 & 10 \end{bmatrix}$$

$$R_5 = \begin{bmatrix} 0.004 & 0 \\ 0 & 90000 \end{bmatrix}, \; R_6 = \begin{bmatrix} 1 & 0 \\ 0 & 1.6 \times 10^8 \end{bmatrix}$$

In this setting, the estimatee is the 4-dimensional state and the observations of the six sensors at each sampling interval T_i $(i = 1, \ldots, 6)$ have the dimensions 4, 4, 2, 2, 2, 2,

respectively. It is not realistic that each sensor's data transmitting rate is the same as sampling rate according to the channel capability. The most often case is that each sensor transmits information to the fusion center every N_i sampling intervals. Then the stacked observations sent by each sensor have dimensions $4N_1, 4N_2, 2N_3, 2N_4, 2N_5, 2N_6$, respectively [50]. We use Monte Calro method to calculate the covariance matrices C_{xz} and C_z in order to approach a more realistic scenario, since we use nonlinear observation models for last two sensors. In the following, we try to get the optimal compression $y = A'z$ for several fusion systems and compare the mean-square error $\mathbf{mse}(\hat{x}(y))$ =tr$(C_x - C_{xz}A(A'C_zA)^{-1}A'C'_{xz})$ with the minimum possible $\mathbf{mse}(\hat{x}(z))$ =tr$(C_x - C_{xz}C_z^{-1}C'_{xz})$, in order to test if the compression loss information or not.

3.4.1 Single Sensor

In this case, we only use sensor 1, 3 or 5. Here $N_1 = N_3 = N_5 = 10$, which means every 10 observations are stacked in each sensor to be compressed. At the fusion center, we use the compressed data to estimate the state. By (5.19) we can get the optimal compression for each system with different data dimensional requirement d. In Table 3.1, we compare the **mse** for all cases: The estimation accuracy increases if data of a higher dimension are allowed to be sent. Since rank$(C'_{xz}C_{xz}C_z^{-1}) = 4$, our analysis states that $\mathbf{mse}(\hat{x}(y))$ =$\mathbf{mse}(\hat{x}(z))$ if and only if $d \geq 4$. This is verified by Table 3.1.

Table 3.1: **mse** of estimation using compressed data from a single sensor

$\text{mse}(\hat{x}(y))$	with S_1	with S_3	with S_5
$d = 1$	506.1702	511.8113	603.4013
$d = 2$	483.3577	498.4204	596.2738
$d = 3$	483.3381	498.4161	596.2643
$d = 4$	483.3308	498.4141	596.2609
$\text{mse}(\hat{x}(z))$	483.3308	498.4141	596.2609

3.4.2 Multi-Sensor Fusion System

In this case, we construct the fusion system by using any combination of the 6 sensors. Vector (d_1, d_2, d_3, d_4, d_5, d_6) denote dimensional requirement for the sensors. So $d_i = 0$ means sensor i does not send out any information, i.e., sensor i has no action in the fusion system. Here $N_i = 10$, that is each sensor stacks 10 observations. The Gauss-Seidel iteration was used to get the optimal solution. In Table 3.2, we compare the **mse** for all cases, where $\hat{x}(z)$ denotes the centralized fusion by using all observations from all active sensors. We also label the required iteration steps for searching for the optimal solution with the proposed Gauss-Seidel algorithm to reach $\epsilon \leq 0.0001$.

For the case with (d_1,d_2,d_3,d_4,d_5,d_6)=(1,2,3,4,1,3), in Fig. 2, we use **ratio**= $\frac{\text{mse}(\hat{x}(z))}{\text{mse}(\hat{x}(y))}$, which is the ratio of the mean-square error of the centralized fusion to that of the distributed fusion, for each search iteration k to show the convergence rate and optimality of the search method. It is in the interval of $(0, 1]$. The larger the **ratio** is, the better the solution is. The three lines are for three different initializations $A^{(0)}$.

Table 3.2: **mse** for multi-sensor system

$(d_1, d_2, d_3, d_4, d_5, d_6)$	Gauss-Seidel Solution:mse($\hat{x}(y)$)	mse($\hat{x}(z)$)
$(1, 1, 1, 1, 0, 0)$	369.6550 (step=11)	291.1871
$(2, 2, 2, 2, 0, 0)$	291.2524 (step=9)	291.1871
$(3, 3, 3, 3, 0, 0)$	291.2082 (step=9)	291.1871
$(1, 2, 3, 4, 0, 0)$	308.1658 (step=9)	291.1871
$(4, 4, 4, 4, 0, 0)$	291.1871 (step=8)	291.1871
$(1, 1, 1, 1, 1, 1)$	348.9785 (step=12)	265.0382
$(2, 2, 2, 2, 2, 2)$	265.1296 (step=13)	265.0382
$(3, 3, 3, 3, 3, 3)$	265.0646 (step=13)	265.0382
$(1, 2, 3, 4, 1, 3)$	289.2232 (step=12)	265.0382
$(4, 4, 4, 4, 4, 4)$	265.0382 (step=12)	265.0382

From Table 3.2 and Figure 3.2, we see that the Gaussian-Seidel iteration for searching for the optimal local sensor data compression is efficient and yields the optimal solution, because it always converges to the same estimator for any initialization $A^{(0)}$, and the convergence rate is high.

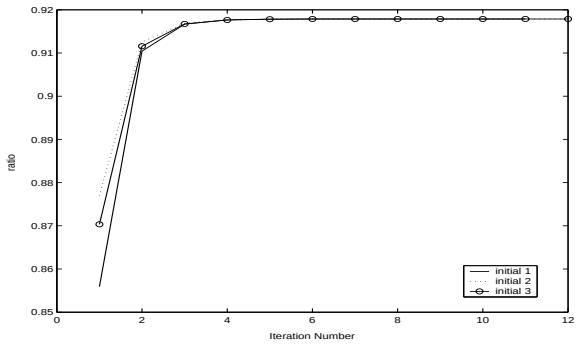


Figure 3.2: **ratio** for different initialization $A^{(0)}$

3.5 Summary

In this Chapter, we propose that in a multi-sensor distributed estimation fusion system, the local sensor measurements processing should based on the fusion rule used in center and the communication channel capability between sensors and fusion center and the processing capability of the fusion center. We formulate the system restriction as the local sensor data dimensional requirement. Based on BLUE fusion, we present the sensor compression within the class of the linear transform which reduces the local data size. The explicit solution for local sensor data compression is given for single sensor system and some particular multiple sensor system. An algorithm with Gauss-Seidel iteration is provided for searching the optimal sensor data compression for most general multiple sensor system with demanding sensor dimensional requirement.

Theoretically, we analysis that there is no estimation accuracy degradation if the dimensional requirement for optimal sensor data compression is larger than the dimension of the estimatee. But the BLUE fusion given by standard estimation fusion system generally will suffer its accuracy when local sensors are correlated. Because the sensor data compression in standard estimation fusion system is not optimal since it does not use the correlation

between each sensors.

It is shown in simulation results that the algorithm for searching the optimal sensor compression rule for most general multiple sensor system with Gauss-Seidel iteration is efficient. In future work, we would like to provide further theoretical support for the optimality of the searching algorithm.

Chapter 4

Optimal Sensor Data Quantization for BLUE Fusion

4.1 Introduction

Dimension reduction is one way to accomplish sensor data compression in fusion system. It provides a linear data compression rule and is not adequate for some problems. Vector quantization is a nonlinear data compression technique which is very popular in communication area. How to design the sensor data quantizer is the focus of this chapter.

In both centralized fusion and distributed fusion, the fusion center is generally not at the same location as the local sensors, and the information sent from the sensors needs to pass via digital communication channels with finite bit rate. Then observations from the sensor need to be quantized into certain levels before transmission to the fusion center in order to satisfy the communication constrains. The estimation at the fusion center can only be based on quantized measurements which can be treated as the compressed measurements from the

local sensors. The goal of sensor quantization in a distributed fusion system is to provide a good estimate of the state, rather than to reconstruct the sensor measurements [23]. A natural question is how to quantize local sensor data so as to minimize the estimation error at the fusion center.

In this chapter, based on the BLUE fusion rule, we build a general state estimation fusion scheme with optimal quantizing the local sensor measurements. The state to be estimated can be a scalar as well as a vector. For state estimation in a dynamic system with communication constraints, the update of sensor quantization needs to be considered. We update sensor quantization by quantizing the measurement residual. When the system is linear Gaussian, we have an explicit formula for the state estimation and sensor quantization update. A new filter with communication bandwidth constraint is then established.

The rest of the chapter is organized as follows. Section 4.2 presents distributed BLUE estimation fusion with limited channel capacity requirements. Given the communication constraints, the problem is formulated as finding partitions of the observation space and representative values for each sensor with fixed quantization levels. A systematic solution for optimal sensor quantization in distributed fusion system is established in Section 4.3. Section 4.4 presents the generalized results for the state estimation in a dynamic system under the communication constraints. A new algorithm for both state estimation and sensor quantization update is developed for linear Gaussian dynamic system. Simulation results and discussions are given in Section 4.5. Section 4.6 provides a summary and briefly discussion about the future work.

4.2 Problem Formulation

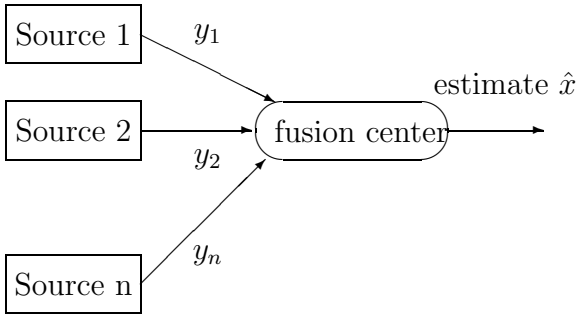


Figure 4.1: framework of distributed estimation fusion

Figure 4.1 shows the framework of distributed estimation fusion where n distributed sensors send measurements to a fusion center. Random observation vectors z_i $(i = 1, \ldots, n)$ are collected by the sensors. x is the state to be estimated (or estimatee). Suppose the fusion center is at a distant location connected to the sensors by digital channels with a capacity of r_i $(i = 1, \ldots, n)$ bits without error. Hence the sensor observations can be vector quantized into $m_i = 2^{r_i}$ levels. The sensor i quantizes observation z_i $(i = 1, \ldots, n)$ by mapping it into a discrete vector y_i $(i = 1, \ldots, n)$ which is transmitted to the fusion center. For sensor k, let Ω_{k1}, ..., Ω_{km_k} be a partition of the observation space, and y_{k1}, ..., y_{km_k} be the representative values of the partition. Then

$$y_k \triangleq \sum_{i=1}^{m_k} y_{ki} I_{\Omega_{ki}}(z_k)$$

where $I_\Omega(z)$ denotes the indicator function, i.e., $I_\Omega(z) = 1$ if $z \in \Omega$ and $I_\Omega(z) = 0$ otherwise. The fusion center then makes an estimate $\hat{x}$ for x based on y_i $(i = 1, \ldots, n)$.

At the fusion center, only the linear unbiased estimation fusion is considered; that is, we consider the most commonly used linear estimation method: best linear unbiased estimation.

With our sensor data quantization, estimation fusion at the center can only be based on the compressed data $y = [y_1', \ldots, y_n']'$.

The problem of optimal sensor data quantization, i.e., finding the optimal partition $\{\Omega_{k1}, \ldots, \Omega_{km_k}\}$ $(k = 1, \ldots, n)$ of the sensor observation space and the corresponding representative values $\{y_{k1}, \ldots, y_{km_k}\}$ $(k = 1, \ldots, n)$, is to solve the following optimization problem:

$$\begin{aligned} &(\{\Omega_{k1}, \ldots, \Omega_{km_k}\}, \{y_{k1}, \ldots, y_{km_k}\})_{k=1,\ldots,n} \\ &= \arg\min\{\mathrm{MSE}(\hat{x})\} \\ &= \arg\min\{C_x - C_{xy}(C_y)^{-1}C_{xy}'\} \\ y_k &\triangleq \sum_{i=1}^{m_k} y_{ki} I_{\Omega_{ki}}(z_k) \end{aligned} \tag{4.1}$$

4.3 Optimal Quantizer Design for BLUE Fusion

The optimal quantizer is designed by solving optimization problem (4.1). For each $k = 1, \ldots, n$, we denote

$$\begin{aligned} y_{(k)} &= \left[y_1', \ldots, y_{k-1}', y_{k+1}', \ldots, y_n'\right]' \\ z_{(k)} &= \left[z_1', \ldots, z_{k-1}', z_{k+1}', \ldots, z_n'\right]' \end{aligned}$$

At the fusion center, based on the BLUE fusion and the recursive BLUE [53], we have

$$\begin{aligned} \hat{x} = E^*(x|y) &= E^*(x|y_{(k)}, y_k) \\ &= E^*(x|y_{(k)}) + C_{x\tilde{y}_{k|y_{(k)}}} C^+_{\tilde{y}_{k|y_{(k)}}} \tilde{y}_{k|y_{(k)}} \end{aligned} \tag{4.2}$$

$$\mathrm{MSE}(\hat{x}) = C_{\hat{x}_{y_{(k)}}} - C_{x\tilde{y}_{k|y_{(k)}}} C^+_{\tilde{y}_{k|y_{(k)}}} C'_{x\tilde{y}_{k|y_{(k)}}} \tag{4.3}$$

where $\tilde{y}_{(k)} = y_{(k)} - \bar{y}_{(k)}$, $\tilde{y}_k = y_k - \bar{y}_k$ with $\bar{y}_{(k)} = E[y_{(k)}]$, $\bar{y}_k = E[y_k]$, and

$$\tilde{y}_{k|y_{(k)}} = y_k - E^*(y_k|y_{(k)}) = \tilde{y}_k - C_{y_k y_{(k)}} C^+_{y_{(k)}} \tilde{y}_{(k)}$$

The covariance matrices in (4.2) and (4.3) are

$$C_{\tilde{y}_{k|y_{(k)}}} = C_{y_k} - C_{y_k y_{(k)}} C^+_{y_{(k)}} C'_{y_k y_{(k)}} \tag{4.4}$$

$$C_{x\tilde{y}_{k|y_{(k)}}} = C_{xy_k} - C_{xy_{(k)}} C^+_{y_{(k)}} C'_{y_k y_{(k)}} \tag{4.5}$$

$$C_{y_{(k)}} = \left[C_{y_i y_j}\right]_{i \neq k,\, j \neq k}$$

As defined above, $y_k \triangleq \sum_{i=1}^{m_k} y_{ki} I_{\Omega_{ki}}(z_k)$, so

$$\begin{aligned}
\bar{y}_k &= \sum_{i=1}^{m_k} y_{ki} P(z_k \in \Omega_{ki}) \\
C_{y_k} &= \sum_{i=1}^{m_k} [y_{ki} - \bar{y}_k][y_{ki} - \bar{y}_k]' P(z_k \in \Omega_{ki}) \\
C_{y_k y_l} &= \sum_{i=1}^{m_k} \sum_{j=1}^{m_l} [y_{ki} - \bar{y}_k][y_{lj} - \bar{y}_l]' P(z_k \in \Omega_{ki}, z_l \in \Omega_{lj}) \\
&= \sum_{i=1}^{m_k} [y_{ki} - \bar{y}_k] P(z_k \in \Omega_{ki}) \left(\sum_{j=1}^{m_l} [y_{lj} - \bar{y}_l]' P(z_l \in \Omega_{lj} | z_k \in \Omega_{ki}) \right) \\
&= \sum_{i=1}^{m_k} [y_{ki} - \bar{y}_k] E(y_l - \bar{y}_l | z_k \in \Omega_{ki}) P(z_k \in \Omega_{ki}) \\
C_{xy_k} &= \sum_{i=1}^{m_k} E[(x - \bar{x})(y_k - \bar{y}_k)'] \\
&= \sum_{i=1}^{m_k} E[(x - \bar{x})(y_k - \bar{y}_k)' | y_k = y_{ki}] P(y_k = y_{ki}) \\
&= \sum_{i=1}^{m_k} [E(x | z_k \in \Omega_{ki}) - \bar{x}][y_{ki} - \bar{y}_k]' P(z_k \in \Omega_{ki})
\end{aligned}$$

We use following notation to simplify the expressions.

$$D_k = [D_{k1}, \ldots, D_{km_k}]'$$

$$A_k = [A_{k1}, \ldots, A_{km_k}]'$$

$$B_{k,l} = [B_{k,l}(1), \ldots, B_{k,l}(m_k)]'$$

and the vectors D_{ki}, A_{ki} and $B_{k,l}(i)$ are given by

$$D_{ki} = [y_{ki} - \bar{y}_k]\sqrt{P(z_k \in \Omega_{ki})}$$

$$A_{ki} = [E(x|z_k \in \Omega_{ki}) - \bar{x}]\sqrt{P(z_k \in \Omega_{ki})}$$

$$B_{k,l}(i) = E(y_l - \bar{y}_l|z_k \in \Omega_{ki})\sqrt{P(z_k \in \Omega_{ki})}$$

Now

$$C_{y_k} = (D_k)'D_k$$

$$C_{y_k y_l} = (D_k)'B_{k,l}$$

$$C_{xy_k} = (A_k)'D_k$$

Let

$$B_{(k)} = [B_{k,1}, \ldots, B_{k,k-1}, B_{k,k+1}, \ldots, B_{k,n}]'$$

$$D_{(k)} = \text{diag}[D_1, \ldots, D_{k-1}, D_{k+1}, \ldots, D_n]$$

$$A_{(k)} = [(A_k)', \ldots, (A_{k-1})', (A_{k+1})', \ldots, (A_n)']'$$

The matrices $C_{y_k y_{(k)}}$ and $C_{xy_{(k)}}$ have

$$C_{y_k y_{(k)}} = D_k{}'B'_{(k)}$$

$$C_{xy_{(k)}} = A'_{(k)}D_{(k)}$$

Then (4.4) and (4.5) can be written as

$$C_{\tilde{y}_{k|y_{(k)}}} = D_k{}'D_k - D_k{}'B'_{(k)}C^+_{y_{(k)}}B_{(k)}D_k$$

$$C_{x\tilde{y}_{k|y_{(k)}}} = A'_kD_k - A'_{(k)}D_{(k)}C^+_{y_{(k)}}B_{(k)}D_k$$

and the MSE matrix (4.3) is

$$\mathrm{MSE}(\hat{x}) = C_{\hat{x}_{y_{(k)}}} - \Lambda_k D_k[D_k{}'\Sigma_k D_k]^+ D'_k\Lambda'_k \tag{4.6}$$

where

$$\Lambda_k = A'_k - A'_{(k)}D_{(k)}C^+_{y_{(k)}}B_{(k)}$$

$$\Sigma_k = I - B'_{(k)}C^+_{y_{(k)}}B_{(k)}$$

According to the derivation in Chapter 3, we can choose $D_k = I$ such that there is no information loss regarding the MSE (4.6) at the fusion center. Thus, we have

$$(y_{ki} - \bar{y}_k)\sqrt{P(z_k \in \Omega_{ki})} = e_i$$

where e_i is the unitary vector with $e_i(l) = \delta_{l-i}$. By now, we finish the procedure of finding representative values for the partition members of the measurement space for sensor k, i.e.,

$$y_{ki} - \bar{y}_k = \frac{e_i}{\sqrt{P(z_k \in \Omega_{ki})}} \tag{4.7}$$

We now discuss how to find the best partition $\{\Omega_{ki}\}_{i=1}^{m_k}$ for sensor k based on fixed $\{\Omega_{li}\}_{i=1}^{m_l}$ for $l \neq k$. According to (4.7), obviously, D_k and $D_{(k)}$ are $m_k \times m_k$ and $(\sum_{i\neq k} m_i) \times (\sum_{i\neq k} m_i)$ identity matrices respectively. Also matrix $B_{k,l}$ has components

$$B_{k,l}(i,j) = \frac{P(z_k \in \Omega_{ki}, z_l \in \Omega_{il})}{\sqrt{P(z_k \in \Omega_{ki})}\sqrt{P(z_l \in \Omega_{lj})}}$$

and

$$C_{\tilde{y}_{k|y_{(k)}}} = I - B'_{(k)} C^{+}_{y_{(k)}} B_{(k)}$$

$$C_{x\tilde{y}_{k|y_{(k)}}} = A'_k - A'_{(k)} C^{+}_{y_{(k)}} B_{(k)}$$

It is natural to ask how the best partition is characterized. For this purpose, we fix $k = 1, \ldots, n$, and write

$$\begin{aligned} \mathrm{MSE}(\hat{x}) &= E[(\tilde{x}_{y_{(k)}} - \mathbf{K}\tilde{y}_k)(\tilde{x}_{y_{(k)}} - \mathbf{K}\tilde{y}_k)'] \\ &= E(\tilde{x}_{y_{(k)}} \tilde{x}'_{y_{(k)}}) - E(\tilde{x}_{y_{(k)}} \tilde{y}'_k)\mathbf{K}' - \mathbf{K}E(\tilde{y}_k \tilde{x}'_{y_{(k)}}) + \mathbf{K}E(\tilde{y}_k \tilde{y}'_k)\mathbf{K}' \end{aligned} \tag{4.8}$$

where

$$\mathbf{K} = C_{x\tilde{y}_{k|y_{(k)}}} C^{+}_{\tilde{y}_{k|y_{(k)}}} = (A'_k - A'_{(k)} C^{+}_{y_{(k)}} B_{(k)})(I - B'_{(k)} C^{+}_{y_{(k)}} B_{(k)})^{+}$$

$$\tilde{x}_{y_{(k)}} = x - \hat{x}_{y_{(k)}} = \tilde{x} - A'_{(k)} C^{+}_{y_{(k)}} \tilde{y}_{(k)}$$

(4.8) can then be written as

$$\mathrm{MSE}(\hat{x}) = J_k + E(\tilde{x}_{y_{(k)}} \tilde{x}'_{y_{(k)}}) - E[E(\tilde{x}_{y_{(k)}}|z_k) E(\tilde{x}_{y_{(k)}}|z_k)']$$

where

$$J_k = \sum_{i=1}^{m_k} \int_{\Omega_{ki}} [E(\tilde{x}_{y_{(k)}}|z_k) - \mathbf{K}(y_{ki} - \bar{y}_k)][E(\tilde{x}_{y_{(k)}}|z_k) - \mathbf{K}(y_{ki} - \bar{y}_k)]' p(z_k) dz_k$$

Now the best partition for sensor k should satisfy

$$\begin{aligned} \{\Omega_{ki}\}_{i=1}^{m_k} &= \arg\min \mathrm{MSE}(\hat{x}) \\ &= \arg\min J_k(\Omega_{k1}, \ldots, \Omega_{km_k}) \end{aligned}$$

Denote by

$$r_k(z_k) = E(\tilde{x}_{y_{(k)}}|z_k) = E(\tilde{x}|z_k) - A'_{(k)}C^+_{y_{(k)}}E(\tilde{y}_{(k)}|z_k)$$

$$c_{ki} = \mathbf{K}(y_{ki} - \bar{y}_k) = \frac{\mathbf{K}e_k}{\sqrt{P(z_k \in \Omega_{ki})}}$$

Thus

$$\Omega_{ki} = \{z_k : (r_k(z_k) - c_{ki})(r_k(z_k) - c_{ki})' \leq (r_k(z_k) - c_{kj})(r_k(z_k) - c_{kj})' \quad \forall j \neq i\} \tag{4.9}$$

The inequality "$\leq$" in terms of matrix may not always meaningful, so in real implementation, we use

$$\Omega_{ki} = \{z_k : (r_k(z_k) - c_{ki})'(r_k(z_k) - c_{ki}) \leq (r_k(z_k) - c_{kj})'(r_k(z_k) - c_{kj}) \quad \forall j \neq i\}$$

which is the trace on both sides of inequality in (4.9). Note that the best partition satisfies the nearest neighbor rule in the transformed space of z_k with center c_{ki} $(i = 1, \ldots, m_k)$.

When the state x is a scalar, if we assume that $c_{k1} < \cdots < c_{km_k}$, then (4.9) is reduces to

$$\Omega_{ki} = \{z_k : \frac{c_{k,i-1} + c_{ki}}{2} < r_k(z_k) \leq \frac{c_{ki} + c_{k,i+1}}{2}\} \tag{4.10}$$

(The choice of $<$ and $\leq$ is arbitrary and is made so that the $\{\Omega_{ki}\}_{i=1}^{m_k}$ are disjoint.) Generally, Ω_{ki} in (4.10) is not an interval, but the inverse image of an interval. It is also important to note that to compute $r_k(z_k)$ and c_{ki} $(i = 1, \ldots, m_k)$ we only require knowledge of the pairwise joint distributions $p(x, z_k)$ and $p(z_k, z_l)$ for all $l = 1, \ldots, n$ instead of the joint distribution $p(x, z_1, \ldots, z_n)$. So if the number of sensor increases, the computation load increases linearly.

4.4 Dynamic Processing for State Estimation Update

In the previous section, we have discussed a static estimation problem where the state is a random variable. In this section, we will consider a dynamic system where the state is a

random process. In this case, when the system dynamics and the measurement model are linear with additive Gaussian noise, the Kalman filter for update of the state estimation is optimal in the MMSE sense if all raw measurements can directly be sent to the fusion center. With the communication constraints, the fusion center can not directly access all raw measurements from all sensors. The measurements need to be quantized and the state update can only use the quantized measurements. In order to have the best performance at the fusion center, we need to consider the update of the state and the update of the sensor quantizer simultaneously.

4.4.1 General Dynamic System

In order to achieve the globally optimal update, the best sensor quantization at time k should use the joint distribution $p(x, z_k)$ and $p(z_k, z_l)$ for all $l = 1, \ldots, k-1$. The reason is given below:

$$\hat{x}_{k|k} = E^*(x_k|y_1, \ldots, y_k) = \hat{x}_{k|k-1} + C_{\tilde{x}_{k|k-1}y_k} C^+_{\tilde{y}_{k|k-1}} \tilde{y}_{k|k-1} \tag{4.11}$$

$$P_{k|k} = \mathrm{MSE}(\hat{x}_{k|k}) = P_{k|k-1} - C_{\tilde{x}_{k|k-1}y_k} C^+_{\tilde{y}_{k|k-1}} C'_{\tilde{x}_{k|k-1}y_k} \tag{4.12}$$

where y_i $(i = 1, \ldots, k)$ is the quantized version of the measurement $\tilde{z}_{i|i-1} = z_i - E^*(z_i|y_1, \ldots, y_{i-1})$ and

$$\hat{x}_{k|k-1} = E^*(x_k|y_1, \ldots, y_{k-1})$$

$$\tilde{y}_{k|k-1} = y_k - E^*(y_k|y_1, \ldots, y_{k-1})$$

According to the BLUE, the measurement residual $\tilde{z}_{k|k-1}$ is orthogonal to $y_1, \cdots, y_{k-1}$. However the quantized measurement y_k does not have such a nice property any more, e.g.,

$E^*(y_k|y_1,\ldots,y_{k-1}) \neq \bar{y}_k$, because quantization itself is not a linear transform. In order to have a recursive formula for quantization update, in the following, we assume

$$E^*(y_k|y_1,\ldots,y_{k-1}) = \bar{y}_k$$

Then (4.11) and (4.12) can be rewritten as

$$\hat{x}_{k|k} = E^*(x_k|y_1,\ldots,y_k) = \hat{x}_{k|k-1} + C_{\tilde{x}_{k|k-1}y_k} C_{y_k}^{+} \tilde{y}_k$$

$$P_{k|k} = \mathrm{MSE}(\hat{x}_{k|k}) = P_{k|k-1} - C_{\tilde{x}_{k|k-1}y_k} C_{y_k}^{+} C'_{\tilde{x}_{k|k-1}y_k}$$

Now

$$\begin{aligned}
\bar{y}_k &= \sum_{j=1}^{m_k} y_{kj} P(\tilde{z}_{k|k-1} \in \Omega_{kj}) \\
C_{y_k} &= \sum_{j=1}^{m_k} [y_{kj} - \bar{y}_k][y_{kj} - \bar{y}_k]' P(\tilde{z}_{k|k-1} \in \Omega_{kj}) = (D_k)' D_k \\
C_{\tilde{x}_{k|k-1}y_k} &= \sum_{j=1}^{m_k} [E(\tilde{x}_{k|k-1}|\tilde{z}_{k|k-1} \in \Omega_{kj})][y_{kj} - \bar{y}_k]' P(\tilde{z}_{k|k-1} \in \Omega_{kj}) = (A_k)' D_k
\end{aligned}$$

where

$$\begin{aligned}
D_{kj} &= [y_{kj} - \bar{y}_k] \sqrt{P(\tilde{z}_{k|k-1} \in \Omega_{kj})} \\
D_k &= [D_{k1}, \ldots, D_{km_k}]' \\
A_{kj} &= [E(\tilde{x}_{k|k-1}|\tilde{z}_{k|k-1} \in \Omega_{kj})] \sqrt{P(\tilde{z}_{k|k-1} \in \Omega_{kj})} \\
A_k &= [A_{k1}, \ldots, A_{km_k}]'
\end{aligned}$$

Therefore

$$C_{\tilde{x}_{k|k-1}y_k} C_{y_k}^{+} C'_{\tilde{x}_{k|k-1}y_k} = A_k' D_k (D_k' D_k)^{+} D_k' A_k$$

Following the derivation the same as Section 4.3, to minimize the MSE, the best representative values and partition are

$$y_{ki} - \bar{y}_k = \frac{e_i}{\sqrt{P(\tilde{z}_{k|k-1} \in \Omega_{ki})}} \tag{4.13}$$

and

$$\Omega_{ki} = \{z_k : (r_k(z_k) - c_{ki})(r_k(z_k) - c_{ki})' \leq (r_k(z_k) - c_{kj})(r_k(z_k) - c_{kj})' \quad \text{for all } j \neq i\} \tag{4.14}$$

where

$$\begin{cases} r_k(z_k) = E(\tilde{x}_{k|k-1}|\tilde{z}_{k|k-1}) \\ c_{ki} = E(\tilde{x}_{k|k-1}|\tilde{z}_{k|k-1} \in \Omega_{ki}) \end{cases}$$

When the state x is a scalar, if we assume that $c_{k1} < \cdots < c_{km_k}$, then (4.14) reduces to

$$\Omega_{ki} = \{z_k : \frac{c_{k,i-1} + c_{ki}}{2} < r_k(z_k) \leq \frac{c_{ki} + c_{k,i+1}}{2}\}$$

Now the update of the sate at the fusion center with the communication constraint is

$$\boxed{\begin{array}{l} \hat{x}_{k|k} = \hat{x}_{k|k-1} + E(\tilde{x}_{k|k-1}|\tilde{z}_{k|k-1} \in \Omega_{kj}) \ \text{ if } \ \tilde{z}_{k|k-1} \in \Omega_{kj} \\ P_{k|k} = P_{k|k-1} - (A_k)'(A_k) \end{array}} \tag{4.15}$$

4.4.2 Linear Gaussian Dynamic System

In the following, we limit our discussion to the linear Gaussian dynamic systems. The state and measurement equations are

$$x_k = F_{k,k-1}x_{k-1} + w_{k,k-1}$$

$$z_k = H_k x_k + v_k$$

where the process noise $w_{k,k-1}$ from time $k-1$ to k, and measurement noise v_k are white Gaussian sequence with covariance $Q_{k,k-1}$ and R_k respectively.

Update for Sensor Quantization

For the sensor quantization update, the main quantities need to be evaluated are

$$E(\tilde{x}_{k+1|k}|\tilde{z}_{k+1|k}) = \frac{\int_{-\infty}^{\infty} \tilde{x}_{k+1|k} p(\tilde{z}_{k+1|k}|\tilde{x}_{k+1|k}) p(\tilde{x}_{k+1|k}) d\tilde{x}_{k+1|k}}{\int_{-\infty}^{\infty} p(\tilde{z}_{k+1|k}|\tilde{x}_{k+1|k}) p(\tilde{x}_{k+1|k}) d\tilde{x}_{k+1|k}} \tag{4.16}$$

$$E(\tilde{x}_{k+1|k}|\tilde{z}_{k+1|k} \in \Omega_{k+1,j}) = \frac{\int_{-\infty}^{\infty} \int_{\tilde{z}_{k+1|k} \in \Omega_{k+1,j}} \tilde{x}_{k+1|k} p(\tilde{z}_{k+1|k}|\tilde{x}_{k+1|k}) p(\tilde{x}_{k+1|k}) d\tilde{z}_{k+1|k} d\tilde{x}_{k+1|k}}{\int_{-\infty}^{\infty} \int_{\tilde{z}_{k+1|k} \in \Omega_{k+1,j}} p(\tilde{z}_{k+1|k}|\tilde{x}_{k+1|k}) p(\tilde{x}_{k+1|k}) d\tilde{z}_{k+1|k} d\tilde{x}_{k+1|k}} \tag{4.17}$$

In both (4.16) and (4.17), probability density functions $p(\tilde{z}_{k+1|k}|\tilde{x}_{k+1|k})$ and $p(\tilde{x}_{k+1|k})$ are needed. Based on the Gaussian assumption:

$$\tilde{z}_{k+1|k} - H_{k+1}\tilde{x}_{k+1|k} \sim N(0, R_k)$$

$p(\tilde{z}_{k+1|k}|\tilde{x}_{k+1|k})$ is Gaussian distributed. Note that $p(\tilde{x}_{k+1|k})$ can be obtained from $p(\tilde{x}_{k|k})$ which can be calculated by

$$\begin{aligned} p(\tilde{x}_{k|k}) &= \sum_{j=1}^{m_k} p(\tilde{x}_{k|k}|\tilde{y}_k = y_{kj} - \bar{y}_k) p(\tilde{y}_k = y_{kj} - \bar{y}_k) \\ &= \sum_{j=1}^{m_k} p(\tilde{x}_{k|k}|\tilde{z}_{k|k-1} \in \Omega_{kj}) p(\tilde{z}_{k|k-1} \in \Omega_{kj}) \end{aligned}$$

where the density function $p(\tilde{x}_{k|k}|\tilde{z}_{k|k-1} \in \Omega_{kj})$ can be obtained directly, because $\tilde{x}_{k|k} - E(\tilde{x}_{k|k-1}|\tilde{z}_{k|k-1} \in \Omega_{kj})$ has the distribution as $p(\tilde{x}_{k|k-1})$. Now we can see that even if the initial point has a Gaussian distribution, $p(\tilde{x}_{k+1|k})$ is a Gaussian mixture. In some cases, if we want to have a simplified results for sensor quantization update, we can use a single Gaussian distribution to approximate the Gaussian mixture for $p(\tilde{x}_{k+1|k})$, i.e., $\tilde{x}_{k+1|k} \sim N(0, P_{k+1|k})$. With this approximation, we have $\tilde{z}_{k+1|k} \sim N(0, S_{k+1})$ and

$$\boxed{r_k(z_k) = E(\tilde{x}_{k+1|k}|\tilde{z}_{k+1|k}) = K_{k+1}\tilde{z}_{k+1|k}}$$

and with [19]

$$\boxed{\begin{aligned} c_{kj} &= E(E(\tilde{x}_{k+1|k}|\tilde{z}_{k+1|k})|\tilde{z}_{k+1|k} \in \Omega_{k+1j}) \\ &= K_{k+1}E(\tilde{z}_{k+1|k}|\tilde{z}_{k+1|k} \in \Omega_{k+1j}) \end{aligned}} \tag{4.18}$$

Now it is easy to apply (4.14) and update the sensor quantization. Note that

$$K_{k+1}\tilde{z}_{k+1|k} \sim N(0, K_{k+1}S_{k+1}K'_{k+1})$$

Then $U_k = (K_{k+1}S_{k+1}K'_{k+1})^{-\frac{1}{2}}K_{k+1}\tilde{z}_{k+1|k}$ will have

$$U_k \sim N(0, I) \tag{4.19}$$

Given the number of quantization levels m_k for time k, we can use the Lloyd-Max [56] quantizer for the standard Gaussian distribution, i.e., the partition $\{\Omega_{01}, \ldots, \Omega_{0m_k}\}$ for standard Gaussian distribution. The rest is to map $\tilde{z}_{k+1|k}$ into U_{k+1} to find the region it belongs to. Now, we do not need to search for the best partition at each time; the quantization update becomes very simple. What we need is only the communication for the state prediction $\hat{x}_{k+1|k}$ from the center to each sensor.

State Estimation Update

According to (4.18) and (4.19), the update for state estimation (4.15) has an explicit formula as follows:

$$\text{Initialization } \hat{x}_{0|0} = \bar{x}_0, \;\; P_{0|0} = P_0$$

$$U_k = (K_k S_k K_k')^{-\frac{1}{2}} K_k \tilde{z}_{k+1|k}$$

$$P_k = \sum_{j=1}^{m_k} E(U_k|U_k \in \Omega_{0j}) E(U_k|U_k \in \Omega_{0j})' P(U_k \in \Omega_{0j})$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + (K_k S_k K_k')^{\frac{1}{2}} E(U_k|U_k \in \Omega_{0j}) \quad \text{if } U_k \in \Omega_{0j}$$

$$P_{k|k} = P_{k|k-1} - (K_k S_k K_k')^{\frac{1}{2}} P_k (K_k S_k K_k')^{\frac{1}{2}}$$

$$\hat{x}_{k+1|k} = F_{k+1,k} \hat{x}_{k|k}$$

$$P_{k+1|k} = F_{k+1,k} P_{k|k} F_{k+1,k}' + Q_{k,k-1}$$

$$\tilde{z}_{k+1|k} = z_{k+1} - H_{k+1} \hat{x}_{k+1|k}$$

$$S_{k+1} = \text{cov}(\tilde{z}_{k+1|k}) = H_{k+1} P_{k+1|k} H_{k+1}' + R_{k+1}$$

$$K_{k+1} = P_{k|k-1} H_k' S_{k+1}^{-1}$$

with the probability-density function $p(x_0)$. It is nice that the new filter is the same as Kalman filter except that the state and estimation error covariance update parts are slightly different.

In the next section, we will study the estimation performance of the update scheme for a linear Gaussian system.

4.5 Simulation Results

Several simple numerical examples are given in this section to verify formulas presented in the previous subsection and the sensor quantization update for a linear Gaussian dynamic system.

4.5.1 Estimation Fusion for Static Case

Consider a 2-sensor estimation fusion problem. The observation model for each sensor i is

$$z_{i,k} = x + w_{i,k} \qquad i = 1, 2$$

where x, $w_{i,k}$ are statistically independent. Let x have density [27]

$$p(x) = \begin{cases} \frac{d}{b}[\frac{5}{4} - \cos(\frac{3\pi x}{2b})] & |x| \leq b \\ 0 & \text{otherwise} \end{cases}$$

where $d \approx 0.3419$ is a normalization constant and $b = 2$. Consider w_{1k}, and w_{2k} having the same density except that $b = 1$. The sensor quantization requirement is $m_i = 8$ (3-bit quantization).

One Dimensional Observation

In this case, each sensor has one dimensional observation. The Lloyd-Max partitions in quantizer design for the random variables y_i, $i = 1, 2$, are identical and given by

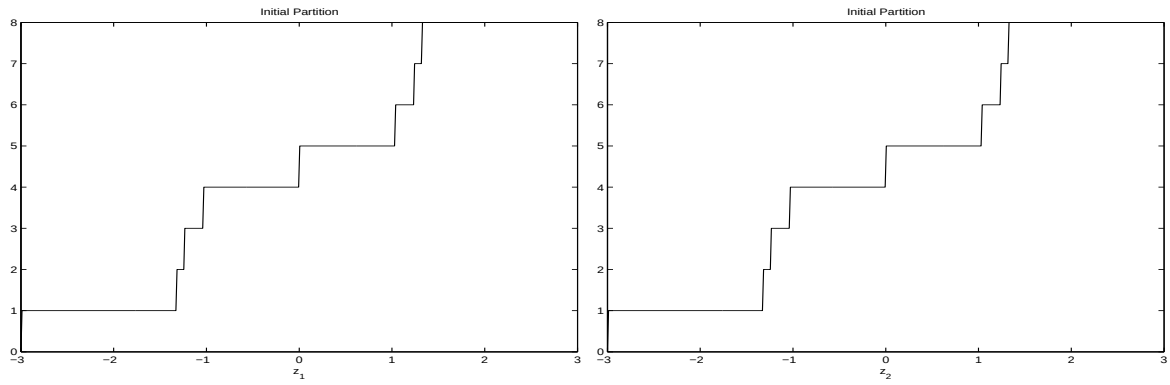


Fig 4.2 **Initial Partition**

After three iterations, the change of MSE for estimator is less than $\epsilon = 0.005$, and the partitions are In **Table 4.1.**, we compare the MSE of MMSE, BLUE results for centralized

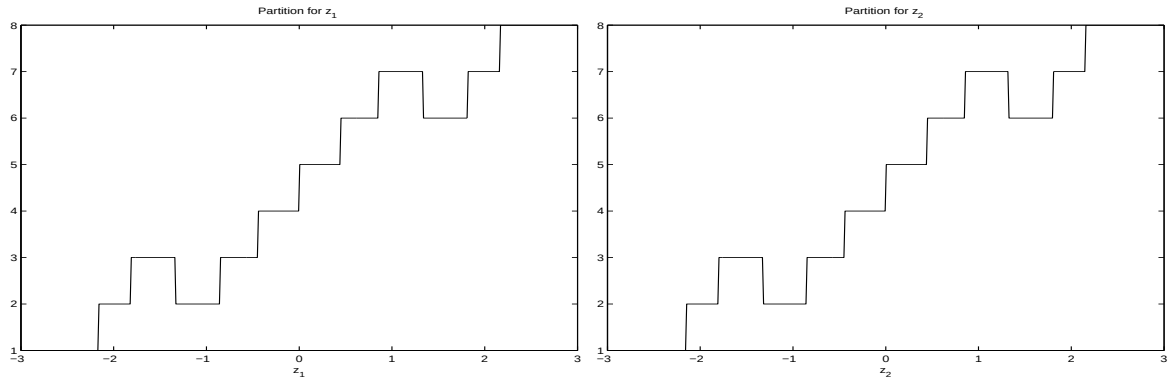


Fig. 4.3 **Partition after Iteration III**

fusion, and the distributed BLUE fusion for each iteration.

Table 4.1: **MSE** Comparison for One Dimensional Sensor Observation

	$\hat{x}(z_1, z_2)$ MMSE	$\hat{x}(z_1, z_2)$ BLUE	$\hat{x}(y_1, y_2)$ (Intinal)	$\hat{x}(y_1, y_2)$ (Iteration I)	$\hat{x}(y_1, y_2)$ (Iteration II)	$\hat{x}(y_1, y_2)$ (Iteration III)
MSE($\hat{x}$)	0.1101	0.1854	0.1747	0.1299	0.1230	0.1217

Two Dimensional Observation

In this case, each sensor has two dimensional observation. We use uniform partitions in quantizer design for the random variables y_i, $i = 1, 2$, are given by

After two iterations, the change of MSE for estimator is less than $\epsilon = 0.005$, and the partitions are In **Table 4.2.**, we compare the MSE of MMSE, BLUE result for centralized fusion results, and the distributed BLUE fusion for each iterations.

From **Table 4.1.** and **4.2.**, you can observe that:

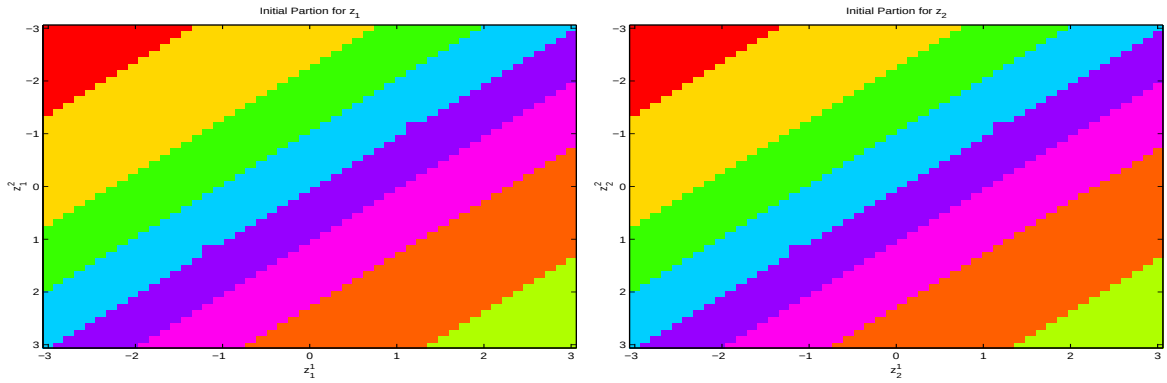


Fig. 4.4 **Initial Partition**

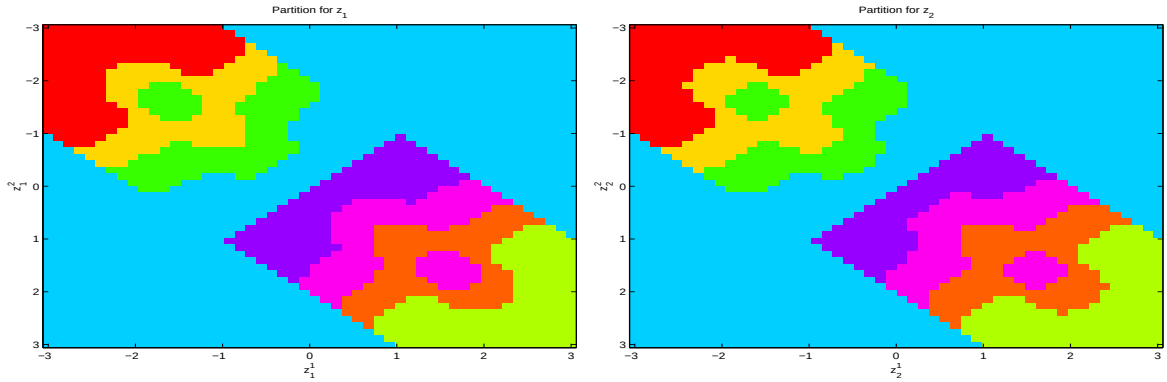


Fig. 4.5 **Partition after Iteration II**

Remark 1: By passing the first iteration, the MSE reduces significantly. But after the first iteration, the MSE does not improve a lot.

Remark 2: The centralized MMSE fusion has the best performance in terms of MSE.

Remark 3: The distributed BLUE fusion with quantized measurement can achieve better estimation accuracy than centralized BLUE fusion. This is not counterintuitive, because the optimal centralized BLUE fusion is only a linear estimator of the sensor observations, however the distributed BLUE fusion with quantized measurement is in fact a nonlinear function of the sensor observations.

Table 4.2: **MSE** Comparison for Two Dimensional Sensor Observation

	$\hat{x}(z_1, z_2)$ (MMSE)	$\hat{x}(z_1, z_2)$ (BLUE)	$\hat{x}(y_1, y_2)$ (Intinal)	$\hat{x}(y_1, y_2)$ (Iteration I)	$\hat{x}(y_1, y_2)$ (Iteration II)
MSE($\hat{x}$)	0.013	0.0982	0.0572	0.0355	0.0354

4.5.2 State Estimation Fusion for Dynamic System

Now, we discuss state estimation update for a dynamic system.

$$x_k = F_{k,k-1}x_{k-1} + w_{k,k-1}$$

$$z_k = H_k x_k + v_k$$

where $x_j = \left[x_j^{(1)}, x_j^{(2)}\right]'$, and w_j and v_j are zero mean white Gaussian noise. Consider a discretized continuous time kinematic system driven by white noise with power spectral density q, called constant velocity model or white-noise acceleration model which is extensively used in many tracking algorithm comparisons, described by

$$F_j = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$$

$$H_j = [1, 0]$$

$$C_{w_j} = Q = \begin{bmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{bmatrix} q$$

$$C_{v_j} = R = 1$$

where T is the sampling interval. The prior information is

$$\hat{x}_{0|0} = \bar{x} = [200\ Km, 0.5\ Km/\sec]',$$

$$P_{0|0} = \begin{bmatrix} R & R/T \\ R/T & 2R/T^2 \end{bmatrix}$$

and the maneuver index is $\lambda = \sqrt{qT^3/R} = 0.707\,11$.

In the results, we use **mser**= $\frac{\text{trace}[P_{k|k}(\text{Kalman})]}{\text{trace}[P_{k|k}(\text{Algorithm})]}$, which is the ratio of the mean-square error of the globally optimal Kalman filter to that of the algorithm under consideration to show the efficiency of an algorithm. It is in the interval of $(0, 1]$. The larger the **mser** is, the more efficient the algorithm is. In Fig 5.5, we draw the theoretical and sample **mser** (over 5000 monte Carlo runs) by using three different communication requirement $m = 2$ (1-bit quantization), 4 (2-bit quantization) and 8 (3-bit quantization) respectively.

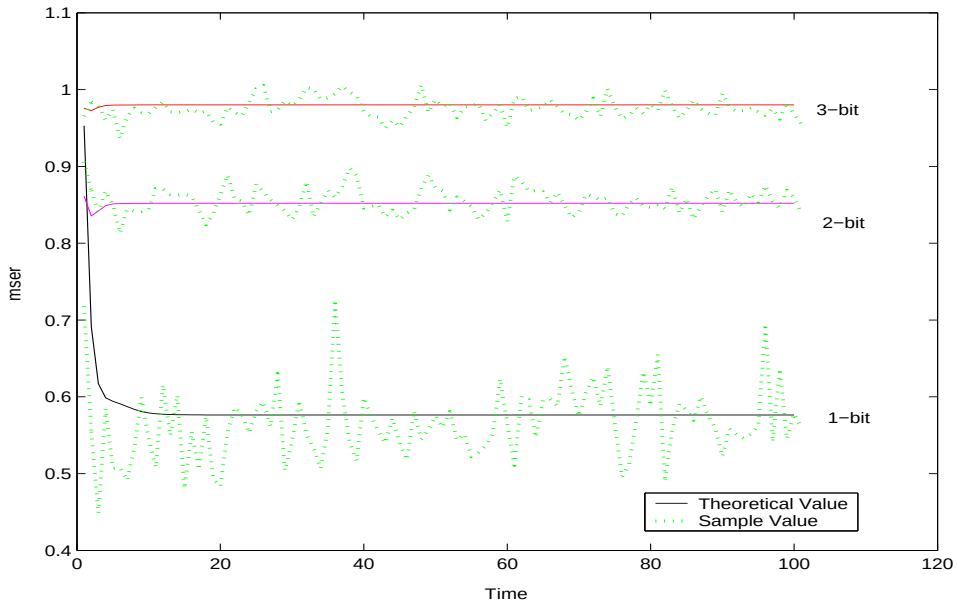


Fig. 4.6 **mser** of State Estimation Update with Communication Constraint

Fig. 4.6 shows that the larger quantization level allowed, the better the performance

is. When 3-bit quantization allowed, the state estimation update by using the quantized measurement only suffers little accuracy compared with the Kalman filter using all raw measurements.

4.6 Summary

We have presented the optimal quantizer design for distributed estimation systems using BLUE fusion. A general framework of how to perform state estimation update and sensor quantization update is introduced a new filter for linear Gaussian dynamic systems is established. The new filter differs slightly from the Kaman filter only in the state and estimation error covariance update parts. At the same time, the quantization update can be easily handled by performing mapping other than searching for new partitions.

The new BLUE fusion structure with the optimal sensor quantization can achieve better performance than traditional centralized BLUE fusion in some cases. Although at the center level, this new BLUE fusion is still linear, it is nonlinear with respect to the whole system because of the nonlinearity of the sensor quantization step. While the work is inspired by the estimation fusion, the ideas can be applied in many other areas for constructing the nonlinear estimator.

In our presented results, sensor quantization update in a dynamic system needs knowledge of state prediction which is not available at the sensor level. In order to perform the task, we need also to consider the communication from the center to the sensors, i.e., feedback. How to formulate the communication constraints for feedback and how to perform update still need further effort.

Chapter 5

Optimal Update with Out-of-Sequence Measurements

5.1 Introduction

In a distributed multiple-sensor tracking system, observations produced by the sensors typically arrive at a fusion center with a random time delay due to communication delay. The state equations are usually obtained in continuous time and then discretized. The sensor may provide a "time stamp" with each measurement. In centralized multi-sensor tracking systems, all these measurements are sent to the fusion center for processing. There are usually different time delays in transmitting data to the fusion center. This can lead to situations where measurements from the same target arrive out of sequence. In this case, a measurement produced at time t_k is received at the fusion center and is used to produce an updated track state estimate and covariance matrix for that time t_k. Then, a delayed observation z_d produced at a prior time t_d ($t_{k-l} \leq t_d < t_{k-l+1}$, $l = 1, 2, \cdots$) is received at

the fusion center. This could occur if the observation produced at time t_d was subject to a longer transmission delay than the delay associated with the observation produced at the later time t_k.

The problem is how to use the "older" measurement from time t_d to update the current state at t_k. There are some methods for updating the state estimate globally optimally with an out-of-sequence measurement (OOSM) within one-step time delay (i.e., $l = 1$, referred to as one-step update) for a system with nonsingular state transition matrix [2], and multi-step OOSM updating using augmented state smoothing [14][15][12]. Also, one-step suboptimal updating algorithms using stored information have been proposed for systems with invertible state transition matrix [76][28][9][2]; multi-step update was discussed in [59] without any discussion of the optimality. These algorithms are shown in this chapter to be special cases of our proposed update algorithms. The globally optimal update algorithm and the optimal update algorithm with limited information are optimal in the linear minimum mean-square error (LMMSE) sense. When the required condition holds, our optimal update algorithm given limited information reduces to the suboptimal algorithms of [2][59], which provides a simple proof of the optimality of these generally suboptimal algorithms.

In this chapter, we first present a discussion concerning what the minimum storage at the current time is to guarantee a globally optimal update. We derive our first algorithm to give a globally optimal LMMSE update by storing all necessary information. It is general and systematic. We consider three cases of prior information about the OOSM. In each case, we try to get the minimum storage. A comparison with existing globally update algorithms in computation and storage is also discussed. Our second algorithm gives the LMMSE update by only using the information available at the current time. Although not guaranteed to be

globally optimal, it is optimal for the information given. As for the first algorithm, we also consider three cases of information storage for the second algorithm. Further, we extend the above single-OOSM update algorithms to the case of arbitrarily delayed multiple OOSMs. For linear Gaussian dynamic system, Algorithm I and II will have a simplified version.

The results presented in this chapter demonstrate how the "static" estimation fusion formulas presented in [53] can be applied to dynamic state estimation and fusion.

5.2 Problem Formulation

The dynamics and measurement models assumed are given by

$$x_j = F_{j,j-1}x_{j-1} + w_{j,j-1} \tag{5.1}$$

$$z_j = H_j x_j + v_j \tag{5.2}$$

where $F_{j,j-1}$ is the state transition matrix from time t_{j-1} to t_j and $w_{j,j-1}$ is (the cumulative effect of) the process noise for the interval $[t_{j-1}, t_j]$. The process noise $w_{j,j-1}$ and the measurement noise v_j are white and have zero mean and covariances

$$C_{w_{j,j-1}} = \mathrm{cov}(w_{j,j-1}) = Q_{j,j-1}, \;\; C_{v_j} = \mathrm{cov}(v_j) = R_j$$

Suppose time t_d is in the sampling interval $t_{k-l} \leq t_d < t_{k-l+1}$, where $l = 1, 2, \cdots$. Which means that the OOSM z_d is l lags behind.

Similar to (5.1), we have

$$x_k = F_{k,d}x_d + w_{k,d}$$

$$z_d = H_d x_d + v_d$$

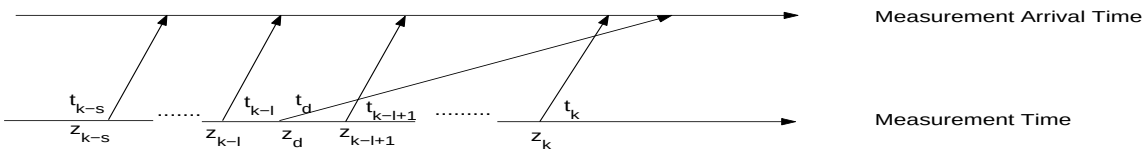


Fig. 5.2.1: The OOSM z_d arrives after the last processed measurement z_k.

The problem is as follows: At time t_k the LMMSE estimator is

$$\hat{x}_{k|k} \triangleq E^*(x_k|z^k) = \underset{\hat{x}_{k|k}=Az^k+B}{\arg\min} P_{k|k}, \quad P_{k|k} \triangleq \text{MSE}[\hat{x}_{k|k}]$$

where $\tilde{x} = x - \hat{x}$. Letting C_{xz} =cov(x, z) and C_z =var(z),

$$\hat{x} = E^*(x|z) = \bar{x} + C_{xz}C_z^{-1}(z - \bar{z}), \quad \text{MSE}(\hat{x}) = E(\tilde{x}\tilde{x}')$$

In the above, $z^k \triangleq \{z_i\}_{i=1}^k$ is the measurement sequence through t_k. If the inverse C_z^{-1} does not exist, it can be simply replaced with the unique *Moore-Penrose pseudoinverse* (MP inverse in short) $C_z^\dagger$. Subsequently, an earlier measurement at time t_d arrives after the state estimate $\hat{x}_{k|k}$ and error covariance $P_{k|k}$ have been calculated. We want to update this estimate with the earlier measurement z_d, that is, to calculate the LMMSE estimator

$$\hat{x}_{k|k,d} = E^*(x_k|\Omega_k, z_d), \quad P_{k|k,d} = \text{MSE}[\hat{x}_{k|k,d}]$$

where Ω_k stands for the information available for update with the OOSM z_d .

5.3 Optimal Update with Available Information

In general, we want to have the globally optimal updated estimate

$$\hat{x}_{k|k,d} = E^*(x_k|z^k, z_d)$$

And generally

$$E^*(x_k|z^k, z_d) = E^*(x_k|\hat{x}_{d|k-l}, z^{d,k})$$

Therefore, if we want to guarantee a globally optimal update, the information stored at each time t_m $(k-l+1 \leq m \leq k)$ should include at least

$$\Omega_m = \{\hat{x}_{k-l|k-l}, P_{k-l|k-l}, z^{k-l+1,m}\}$$

or its equivalent. Otherwise, no guarantee that any update is globally optimal in general. In some special cases, however, information from a smaller storage is sufficient for global optimality. Thus, all measurements, state estimates, and error covariances from the occurrence time of OOSM to its arrival time need to be saved to guarantee a globally optimal update.

In practice, an OOSM z_d has a random time delay (e.g., l is random). But it may not be too far before time t_k. It is not reasonable to store the observation at each time in order to get the optimal updated estimate $E^*(x_k|z^k, z_d)$. In fact, in each step, it is often the case that only $\hat{x}_{j|j}$ and the associated error covariance $P_{j|j}$ are stored. So at time t_k, the available information stored is $\Omega_k = \left\{\hat{x}_{k|k}, P_{k|k}\right\}$. Thus the optimal update is better done based on this Ω_k and OOSM. Now the update in general can be done by using formulas for BLUE fusion without prior [53], because the prior for update includes the prior information $\bar{x}_{0|0}, P_{0|0}$, which may not be available for update with OOSM. This update is not globally optimal in general, but it is optimal for the information given.

Both algorithms presented in the next section are optimal in the LMMSE sense. They differ in that different Ω_k are used and thus they are optimal for different available information Ω_k.

5.4 Optimal Update Algorithms

5.4.1 Algorithm I — Globally Optimal Update

Based on the linear dynamic model, according to recursive LMMSE, the globally optimal update can be expressed as

$$\hat{x}_{k|k,d} = E^*(x_k|z^k, z_d) = \hat{x}_{k|k} + K_d(z_d - H_d\hat{x}_{d|k}) = \hat{x}_{k|k} + K_d\tilde{z}_{d|k} \tag{5.3}$$

$$\text{MSE}(\hat{x}_{k|k,d}) = P_{k|k} - K_d S_d K_d' \tag{5.4}$$

where

$$K_d = U_{k,d}H_d'S_d^{-1},\ \ S_d = H_dP_{d|k}H_d' + R_d,\ \ U_{k,d} = C_{x_k,\tilde{x}_{d|k}} = \text{cov}(x_k, \tilde{x}_{d|k})$$

In the above, if the inverse S_d^{-1} does not exist, we can simply replace it with $S_d^{\dagger}$, the MP inverse of S_d =cov($\tilde{z}_{d|k}$). $\hat{x}_{k|k}$ and $P_{k|k}$ are available in the Kalman filter. In the following we focus on other necessary information $\{\hat{x}_{d|k}, P_{d|k}, U_{k,d}\}$, which in fact exists in a recursive form (non-standard smoothing):

Let

$$\hat{x}_{d|n} = E^*(x_d|z^n),\ \ P_{d|n} = \text{MSE}(\hat{x}_{d|n}),\ \ U_{n,d} = C_{x_n,\tilde{x}_{d|n}}$$

Theorem 5: *Starting from $n = k - l + 1$, we have the recursion*

$$\hat{x}_{d|n+1} = \hat{x}_{d|n} + U_{n,d}'F_{n+1,n}'H_{n+1}'S_{n+1}^{-1}\tilde{z}_{n+1|n} \tag{5.5}$$

$$P_{d|n+1} = P_{d|n} - U_{n,d}'F_{n+1,n}'H_{n+1}'S_{n+1}^{-1}H_{n+1}F_{n+1,n}U_{n,d}$$

$$U_{n+1,d} = (I - K_{n+1}H_{n+1})F_{n+1,n}U_{n,d}$$

with initial value

$$\begin{aligned}
\hat{x}_{d|k-l+1} &= \hat{x}_{d|k-l} + P_{d|k-l}F'_{k-l+1,d}H'_{k-l+1}S^{-1}_{k-l+1}\tilde{z}_{k-l+1|k-l} \\
P_{d|k-l+1} &= P_{d|k-l} - P_{d|k-l}F'_{k-l+1,d}H'_{k-l+1}S^{-1}_{k-l+1}H_{k-l+1}F_{k-l+1,d}P_{d|k-l} \\
U_{k-l+1,d} &= (I - K_{k-l+1}H_{k-l+1})F_{k-l+1,d}P_{d|k-l}
\end{aligned} \tag{5.6}$$

where

$$\hat{x}_{d|k-l} = F_{d,k-l}\hat{x}_{k-l|k-l} \tag{5.7}$$

$$P_{d|k-l} = F_{d,k-l}P_{k-l|k-l}F'_{d,k-l} + Q_{d,k-l} \tag{5.8}$$

□

Proof. The recursion for $\{\hat{x}_{d|n}, P_{d|n}, U_{n,d}\}$ is generated as follows. For $n \geq k-l+1$

$$\begin{aligned}
\hat{x}_{d|n+1} &= E^*(x_d|z^{n+1}) = \hat{x}_{d|n} + C_{x_d,\tilde{x}_{n|n}}F'_{n+1,n}H'_{n+1}S^{\dagger}_{n+1}\tilde{z}_{n+1|n} \\
P_{d|n+1} &= P_{d|n} - C_{x_d,\tilde{x}_{n|n}}F'_{n+1,n}H'_{n+1}S^{\dagger}_{n+1}H_{n+1}F_{n+1,n}C'_{x_d,\tilde{x}_{n|n}} \\
U_{n+1,d} &= C_{x_{n+1},\tilde{x}_{d|n+1}} = C_{x_{n+1},\tilde{x}_{d|n} - C_{x_d,\tilde{x}_{n|n}}F'_{n+1,n}H'_{n+1}S^{\dagger}_{n+1}\tilde{z}_{n+1|n}} \\
&= C_{x_{n+1},\tilde{x}_{d|n}} - C_{x_{n+1},\tilde{z}_{n+1|n}}S^{\dagger}_{n+1}H_{n+1}F'_{n+1,n}C'_{x_d,\tilde{x}_{n|n}} \\
&= F_{n+1,n}U_{n,d} - K_{n+1}H_{n+1}F_{n+1,n}C'_{x_d,\tilde{x}_{n|n}}
\end{aligned} \tag{5.9}$$

And

$$C_{x_n,\tilde{x}_{d|n}} = C_{x_n-\hat{x}_{n|n},x_d-\hat{x}_{d|n}} = C_{x_n-\hat{x}_{n|n},x_d} = C_{\tilde{x}_{n|n},x_d} = C'_{x_d,\tilde{x}_{n|n}}$$

Thus $C_{x_n,\tilde{x}_{d|n}} = C'_{x_d,\tilde{x}_{n|n}}$ holds. So the recursion (5.9) can be simplified as

$$\hat{x}_{d|n+1} = \hat{x}_{d|n} + U'_{n,d}F'_{n+1,n}H'_{n+1}S^{\dagger}_{n+1}\tilde{z}_{n+1|n}$$

$$P_{d|n+1} = P_{d|n} - U'_{n,d}F'_{n+1,n}H'_{n+1}S^{\dagger}_{n+1}H_{n+1}F_{n+1,n}U'_{n,d}$$

$$U_{n+1,d} = (I - K_{n+1}H_{n+1})F_{n+1,n}U'_{n,d}$$

with initial value

$$\hat{x}_{d|k-l+1} = \hat{x}_{d|k-l} + P_{d|k-l}F'_{k-l+1,d}H'_{k-l+1}S^{\dagger}_{k-l+1}\tilde{z}_{k-l+1|k-l}$$

$$P_{d|k-l+1} = P_{d|k-l} - P_{d|k-l}F'_{k-l+1,d}H'_{k-l+1}S^{\dagger}_{k-l+1}H_{k-l+1}F_{k-l+1,d}P_{d|k-l}$$

$$U_{k-l+1,d} = (I - K_{k-l+1}H_{k-l+1})F_{k-l+1,d}P_{d|k-l}$$

■

Based on the above recursion, it is easy to get that $\{\hat{x}_{d|k}, P_{d|k}, U_{k,d}\}$ are highly related with the OOSM occurrence time t_d through $\{\hat{x}_{d|k-l}, P_{d|k-l}\}$ which are highly related with the state estimate of x_d at that time. The key to achieve global optimality for the update lies in when and how to initialize the recursion.

Depending on different prior information about t_d, we consider three cases.

Case I: Perfect Knowledge about t_d at the Next Sampling Time t_{k-l+1}

In this case, we know the exact sampling time at which each observation is made and supposed to arrive. Suppose z_d made at t_d ($t_{k-l} \leq t_d < t_{k-l+1}$) has not arrived by t_{k-l+1} (so we know we have an OOSM); instead, it arrives during $[t_k, t_{k+1})$ with a time stamp t_d. Then at the time at which z_d is supposed to arrive, we can still run the Kalman filter to get prediction $\{\hat{x}_{d|k-l}, P_{d|k-l}\}$; the only difference is that there is no state update with z_d. Then at the next time t_{k-l+1}, we can initialize by (5.6) and run our recursion (5.5) until

the OOSM arrives. This filter is an extension of the traditional Kalman filter by adding $\{\hat{x}_{k|n}, P_{k|n}, U_{k,n}\}$ at each recursion n $(k-l+1 \leq n \leq k)$. After receiving the OOSM, the OOSM update algorithm is globally optimal. The complete algorithm is the Kalman filter associated with the OOSM update, which is referred to as *KF-OOSM*. The *KF-OOSM* for Case I is shown in **Fig.5.4.1**.

Since the traditional Kalman filter stores $\{\hat{x}_{n|n}, P_{n|n}\}$ at each recursion, the information stored in our *KF-OOSM* at each recursion n $(k-l+1 \leq n \leq k)$ is

$$\Omega_n = \{\hat{x}_{n|n}, P_{n|n}, \hat{x}_{d|n}, P_{d|n}, U_{n,d}\}$$

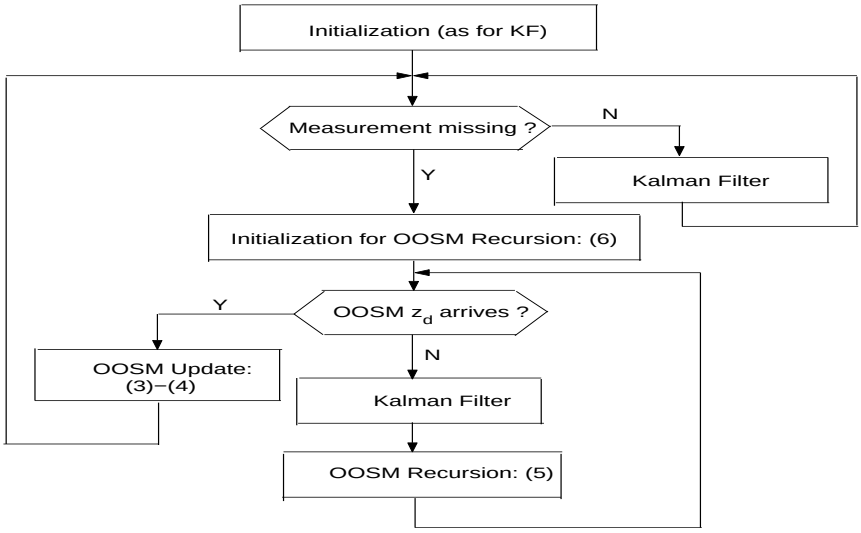


Fig.5.4.1: Algorithm I for Case I

In this case, the storage is fixed as the delay l increases.

Case II: Knowing $t_{k-l} < t_d < t_{k-l+1}$ at Time t_{k-l+1}

In this case, we know exactly the sampling interval over which each observation is made and supposed to arrive; supposed z_d made at t_d $(t_{k-l} \leq t_d < t_{k-l+1})$ has not arrived by t_{k-l+1} (so we know we have an OOSM); instead, it arrives during $[t_k, t_{k+1})$ with a time stamp t_d.

Then, at time t_{k-l+1} we can not use $\{\hat{x}_{d|k-l+1}, P_{d|k-l+1}, U_{k-l+1,d}\}$ directly to initialize our *KF-OOSM* because they are all related with the state x_d. Without receiving the OOSM z_d at time t_{k-l+1}, the necessary state information x_d is not available at that time, but we can initialize our *KF-OOSM* using the replacement $\{y_{k-l+1}, B_{k-l+1}, U_{k-l+1}\}$, defined by

$$y_{k-l+1} = H'_{k-l+1}S^{-1}_{k-l+1}\tilde{z}_{k-l+1|k-l}, \;\; B_{k-l+1} = H'_{k-l+1}S^{-1}_{k-l+1}H_{k-l+1}, \;\; U_{k-l+1} = I - K_{k-l+1}H_{k-l+1} \tag{5.10}$$

None of them are related with state x_d and they are generated using the information available in the traditional Kalman filter at that time. We can define the recursion for $\{y_n, B_n, U_n\}$ with $k-l+1 \leq n \leq k$ as

$$\begin{aligned} y_{n+1} &= y_n + U'_n F'_{n+1,n} H'_{n+1} S^{-1}_{n+1} \tilde{z}_{n+1|n} \\ B_{n+1} &= B_n + U'_n F'_{n+1,n} H'_{n+1} S^{-1}_{n+1} H_{n+1} F_{n+1,n} U_n \\ U_{n+1} &= (I - K_{n+1}H_{n+1}) F_{n+1,n} U_n \end{aligned} \tag{5.11}$$

Then $\{\hat{x}_{d|k}, P_{d|k}, U_{k,d}\}$ can be obtained by renewing $\{y_k, B_k, U_k\}$ once the OOSM z_d arrives.

Theorem 6:

$$\begin{aligned} \hat{x}_{d|k} &= P_{d|k-l} F'_{k-l+1,d} y_k + \hat{x}_{d|k-l} \\ P_{d|k} &= P_{d|k-l} - P_{d|k-l} F'_{k-l+1,d} B_k F_{k-l+1,d} P_{d|k-l} \\ U_{k,d} &= U_k F_{k-l+1,d} P_{d|k-l} \end{aligned} \tag{5.12}$$

□

Proof. Since

$$\hat{x}_{d|k} = \hat{x}_{d|k-1} + U'_{k-1,d}F'_{k,k-1}H'_kS_k^{-1}\tilde{z}_{k|k-1}$$

$$P_{d|k} = P_{d|k-1} - U'_{k-1,d}F'_{k,k-1}H'_kS_k^{-1}H_kF_{k,k-1}U_{k-1,d}$$

$$U_{k,d} = (I - K_kH_k)F_{k,k-1}U_{k-1,d}$$

$\vdots$

$$\hat{x}_{d|k} = \hat{x}_{d|k-l} + U'_{k-l,d}F'_{k-l+1,k-l}H'_{k-l+1}S_{k-l+1}^{-1}\tilde{z}_{k-l+1|k-l} + \ldots + U'_{k-1,d}F'_{k,k-1}H'_kS_k^{-1}\tilde{z}_{k|k-1}$$

$$P_{d|k} = P_{d|k-l} - U'_{k-l,d}F'_{k-l+1,k-l}H'_{k-l+1}S_{k-l+1}^{-1}H_{k-l+1}F_{k-l+1,k-l}U_{k-l,d} - \ldots - U'_{k-1,d}F'_{k,k-1}H'_kS_k^{-1}$$

$$H_kF_{k,k-1}U_{k-1,d}$$

$$U_{k,d} = [(I - K_kH_k)F_{k,k-1}] \times \ldots \times [(I - K_{k-l+1}H_{k-l+1})F_{k-l+1,k-l}]U_{k-l,d}$$

also

$$y_k = y_{k-1} + U'_{k-1}F'_{k,k-1}H'_kS_k^{-1}\tilde{z}_{k|k-1}$$

$$B_k = B_{k-1} + U'_{k-1}F'_{k,k-1}H'_kS_k^{-1}H_kF_{k,k-1}U_{k-1}$$

$$U_k = (I - K_kH_k)F_{k,k-1}U_{k-1}$$

$\vdots$

$$y_k = y_{k-l} + U'_{k-l}F'_{k-l+1,k-l}H'_{k-l}S_{k-l}^{-1}\tilde{z}_{k-l+1|k-l} + \ldots + U'_{k-1}F'_{k,k-1}H'_kS_k^{-1}\tilde{z}_{k|k-1}$$

$$B_k = B_{k-l} + U'_{k-l}F'_{k-l+1,k-l}H'_{k-l}S_{k-l}^{-1}H_{k-l+1}F_{k-l+1,k-l}U_{k-l} + \ldots + U'_{k-1}F'_{k,k-1}H'_kS_k^{-1}H_kF_{k,k-1}U_k$$

$$U_k = [(I - K_kH_k)F_{k,k-1}] \times \ldots \times [(I - K_{k-l+1}H_{k-l+1})F_{k-l+1,k-l}]U_{k-l}$$

and for any $m \geq k - l + 1$

$$U_{m,d} = [(I - K_mH_m)F_{m,m-1}] \times \ldots \times [(I - K_{k-l+2}H_{k-l+2})F_{k-l+2,k-l+1}]U_{k-l+1,d}$$

$$U_m = [(I - K_mH_m)F_{m,m-1}] \times \ldots \times [(I - K_{k-l+2}H_{k-l+2})F_{k-l+2,k-l+1}]U_{k-l+1}$$

By comparing the initial value $\{\hat{x}_{d|k-l+1}, P_{d|k-l+1}, U_{k-l+1,d}\}$ and $\{y_{k-l+1}, B_{k-l+1}, U_{k-l+1}\}$, we have

$$U_{m,d} = U_m F_{k-l+1,d} P_{d|k-l}$$

and

$$P_{d|k-l} F'_{k-l+1,d} y_k = P_{d|k-l} F'_{k-l+1,d} y_{k-l+1} + \hat{x}_{d|k} - \hat{x}_{d|k-l+1}$$

$$= P_{d|k-l} F'_{k-l+1,d} H'_{k-l+1} S^{-1}_{k-l+1} \tilde{z}_{k-l+1|k-l} + \hat{x}_{d|k} - \hat{x}_{d|k-l+1}$$

$$= P_{d|k-l} F'_{k-l+1,d} H'_{k-l+1} S^{-1}_{k-l+1} \tilde{z}_{k-l+1|k-l} + \hat{x}_{d|k} - \hat{x}_{d|k-l} - P_{d|k-l} F'_{k-l+1,d} H'_{k-l+1} S^{+}_{k-l+1} \tilde{z}_{k-l+1|k-l}$$

$$= \hat{x}_{d|k} - \hat{x}_{d|k-l}$$

$$P_{d|k-l} F'_{k-l+1,d} B_k F_{k-l+1,d} P_{d|k-l} = P_{d|k-l} F'_{k-l+1,d} B_{k-l+1} F_{k-l+1,d} P_{d|k-l} - P_{d|k} + P_{d|k-l+1}$$

$$= P_{d|k-l} F'_{k-l+!,d} H'_{k-l+1} S^{-1}_{k-l+1} H_{k-l+1} F_{k-l+1,d} P_{d|k-l} - P_{d|k} + P_{d|k-l} - P_{d|k-l} F'_{k-l+1,d} H'_{k-l+1} S^{-1}_{k-l+1}$$

$$H_{k-l+1}\ F_{k-l+1,d} P_{d|k-l} = P_{d|k-l} - P_{d|k}$$

Thus

$$\hat{x}_{d|k} = P_{d|k-l} F'_{k-l+1,d} y_k + \hat{x}_{d|k-l}$$

$$P_{d|k} = P_{d|k-l} - P_{d|k-l} F'_{k-l+1,d} B_k F_{k-l+1,d} P_{d|k-l}$$

$$U_{k,d} = U_k F_{k-l+1,d} P_{d|k-l}$$

■

The *KF-OOSM* for Case II is shown in **Fig.5.4.2.**

The information needed to be stored for our *KF-OOSM* at each recursion n ($k-l+1 \leq n \leq k$) in this case is

$$\Omega_n = \{\hat{x}_{n|n}, P_{n|n}, y_n, B_n, U_n, \hat{x}_{k-l|k-l}, P_{k-l|k-l}\}$$

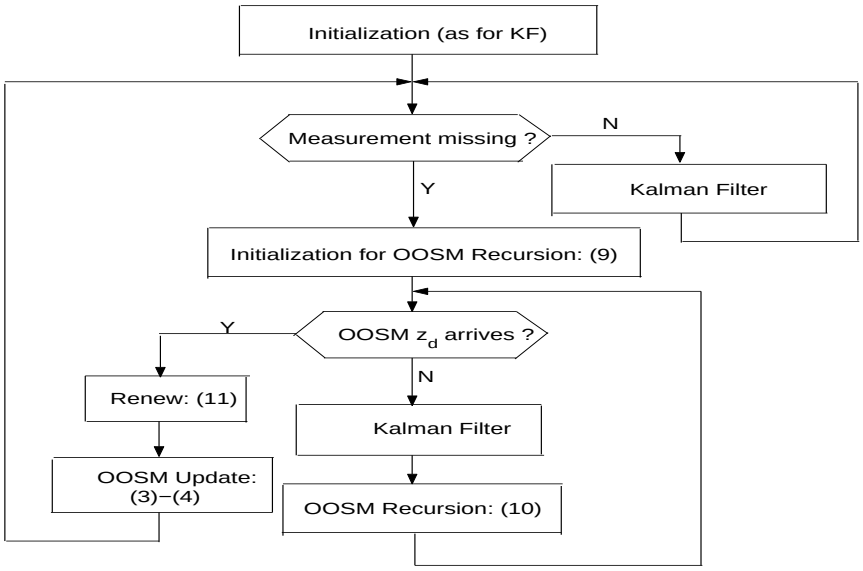


Fig.5.4.2: Algorithm I for Case II

Remark If $l = 1$ (i.e., one-step update), there is only one recursion in our *KF-OOSM*; so the information needed to be stored is simply

$$\Omega_k = \{\hat{x}_{k|k}, P_{k|k}, y_k, B_k, \hat{x}_{k-1|k-1}, P_{k-1|k-1}\}$$

and

$$U_{k,d} = [I + (F_{k,d}P_{d|k-1}F'_{k,d} + Q_{k,d})B_k]F_{k,d}P_{d|k-1}$$

In this case, the storage is also fixed as the delay l increases.

Case III: Knowing Maximum Delay s of OOSM

In this case, there is not any prior information about the OOSM z_d occurrence time t_d before it arrives, but we know the maximum delay s for the OOSM, i.e., $t_{k-s} \leq t_{k-l} \leq t_d < t_{k-l+1} \leq t_k$ with an unknown l.

In this case, we do not know when to initialize our *KF-OOSM*, but we can treat each

discrete time in the time window $[t_{k-s}, t_k)$ as the possible initialization point, such as

$$y_n^{(n)} = H_n' S_n^{-1} \tilde{z}_{n|n-1}, \;\; B_n^{(n)} = H_n' S_n^{-1} H_n, \;\; U_n^{(n)} = I - K_n H_n \tag{5.13}$$

and apply the algorithm in Case II to achieve the optimal update after the OOSM is received. The recursion for $\{y_{n+1}^{(m)}, B_{n+1}^{(m)}, U_{n+1}^{(m)}\}$ $(n > k-l+1,\, n-s < m < n)$ is

$$\begin{aligned} y_{n+1}^{(m)} &= y_n^{(m)} + U_n^{(m)\prime} F_{n+1,n}' H_{n+1}' S_{n+1}^{-1} \tilde{z}_{n+1|n} \\ B_{n+1}^{(m)} &= B_n^{(m)} + U_n^{(m)\prime} F_{n+1,n}' H_{n+1}' S_{n+1}^{-1} H_{n+1} F_{n+1,n} U_n^{(m)} \\ U_{n+1}^{(m)} &= (I - K_{n+1} H_{n+1}) F_{n+1,n} U_n^{(m)} \end{aligned} \tag{5.14}$$

Then $\{\hat{x}_{d|k}, P_{d|k}, U_{k,d}\}$ can be obtained by renewing $\{y_k^{(k-l+1)}, B_k^{(k-l+1)}, U_k^{(k-l+1)}\}$ once the OOSM z_d arrives

$$\begin{aligned} \hat{x}_{d|k} &= P_{d|k-l} F_{k-l+1,d}' y_k^{(k-l+1)} + \hat{x}_{d|k-l} \\ P_{d|k} &= P_{d|k-l} - P_{d|k-l} F_{k-l+1,d}' B_k^{(k-l+1)} F_{k-l+1,d} P_{d|k-l} \\ U_{k,d} &= U_k^{(k-l+1)} F_{k-l+1,d} P_{d|k-l} \end{aligned} \tag{5.15}$$

The *KF-OOSM* for Case III is shown in **Fig.5.4.3**.

The information storage in our *KF-OOSM* at each recursion n in this case increases linearly as time increases from t_{k-s+1} to t_k, which is as follows:

$$\Omega_n = \{\hat{x}_{k-s|k-s}, P_{k-s|k-s}, \cdots, \hat{x}_{n|n}, P_{n|n}, y_n^{(k-s+1)}, B_n^{(k-s+1)}, U_n^{(k-s+1)}, \cdots, y_n^{(n)}, B_n^{(n)}, U_n^{(n)}\}$$

Remark 1 If $s = 1$ (i.e., one-step update), it is just the problem considered in Case II with $l = 1$.

Remark 2 If $t_d = t_{k-l}$, the OOSM was made exactly at a previous sampling time. With

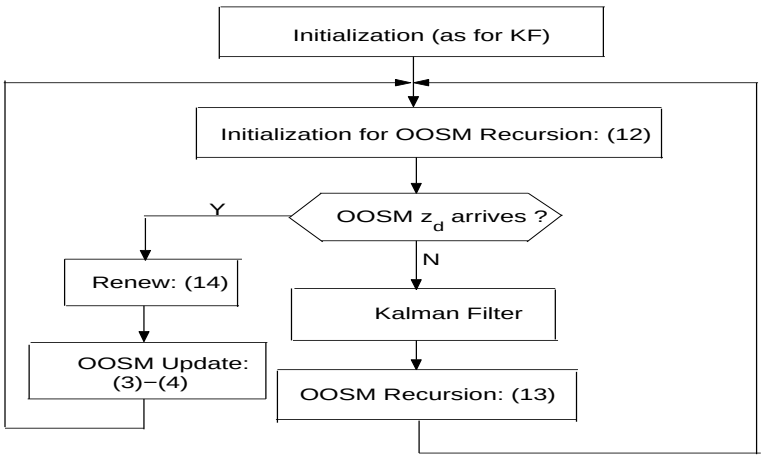


Fig.5.4.3: Algorithm I for Case III

the algorithm slightly changed, the memory can be saved comparing with $t_{k-l} < t_d < t_{k-l+1}$. In this case, the flowchart of *KF-OOSM* for this case has the same structure as above, but the OOSM initialization and renewing part are respectively replaced with

$$y_n^{(n)} = \hat{x}_{n|n} + P_{n|n}F'_{n,n-1}H'_nS_n^{-1}\tilde{z}_{n|n-1}$$

$$B_n^{(n)} = P_{n|n} - P_{n|n}F'_{n,n-1}H'_nS_n^{-1}H_nF_{n,n-1}P_{n|n}$$

$$U_n^{(n)} = (I - K_nH_n)F_{n,n-1}P_{n|n}$$

and

$$\hat{x}_{d|k} = y_k^{(k-l+1)},\ P_{d|k} = B_k^{(k-l+1)},\ U_{k,d} = U_k^{(k-l+1)}$$

The information needed at each recursion n $(k - s < n \leq k)$ in our *KF-OOSM* in this case is

$$\Omega_n = \{\hat{x}_{n|n}, P_{n|n}, y_n^{(k-s+1)}, B_n^{(k-s+1)}, U_n^{(k-s+1)}, \cdots, y_n^{(n)}, B_n^{(n)}, U_n^{(n)}\}$$

Depending on the uncertainty with the OOSM occurrence time, we have considered three cases of the *KF-OOSM* and the associated efficient memory structure for Algorithm I. The more uncertain of the OOSM occurrence time, the more storage we need. Cases I and II

have a fixed storage from time t_{k-l} to t_k. The storage in case III increases linearly with the length of the time interval. None of these algorithms for the three cases generally have any non-singularity requirement on the transition matrix $F_{k,d}$.

5.4.2 Comparison of Globally Optimal Update Algorithms

[2, 14] present algorithms to achieve globally optimal update. [2] deals with the single-step update problem. There are two major steps: retrodiction from current time to OOSM occurring time and update the estimate with the OOSM. [14] is suitable for our Case III assuming $t_d = t_{k-l}$. It is based on a method of non-standard smoothing by augmenting the state vector to include all "states" $x_{k-s}, x_{k-s+1}, \ldots, x_k$. It is conceptually elegant, but not attractive computationally. It seems impossible to have a globally optimal update with the OOSM when $t_d \neq t_{k-l}$ (i.e., t_d is not exactly some sampling time instant) within this framework of state augmentation. A technique was suggested in [14] to handle the problem with $t_{k-l} < t_d < t_{k-l+1}$ by approximating t_d to the nearest time t_{k-l} or t_{k-l+1}. The approximation makes the estimation not globally optimal. The error is small when the sampling intervals are small. However, this requires more lags (i.e., large l) in the augmented state, and hence increases computational load to cover the same maximum time delay. This is a dilemma when one wants to have small errors and efficient computation simultaneously.

The algorithm presented here do not need to retrodict to the previous state or go back by smoothing. It is the traditional Kalman filter with a few more terms in the recursion. It requires more storage than the Kalman filter, but the extra storage used is not large. In Cases I and II, the storage is fixed at each recursion and even in Case III, the storage

increases only linearly as the delay l increases.

Let us simply compare the storage and computational load of our OOSM update algorithm with those of [14] in Case III assuming $t_d = t_{k-l}$. In the following, we consider the same maximum delay s for the OOSM and focus on the total storage and computational burden within a certain time window. The algorithm of [14] is referred as ALG-S and our globally optimal algorithm I for Case III as ALG-I in the following:

ALG-S (Storage):

$$\begin{bmatrix} \hat{x}_{k-s|k} \\ \vdots \\ \hat{x}_{k|k} \end{bmatrix}, \quad \begin{bmatrix} P_{k-s|k} & \cdots & C_{\tilde{x}_{k-s|k},\tilde{x}_{k|k}} \\ \vdots & \ddots & \vdots \\ C_{\tilde{x}_{k|k},\tilde{x}_{k-s|k}} & \cdots & P_{k|k} \end{bmatrix}$$

ALG-I (Storage):

$$\begin{bmatrix} y_k^{(k-s+1)} \\ \vdots \\ y_k^{k-1} \\ \hat{x}_{k|k} \end{bmatrix}, \quad \begin{bmatrix} B_k^{(k-s+1)} & U_k^{(k-s+1)} \\ \vdots & \vdots \\ B_k^{(k-2)} & U_k^{(k-2)} \\ B_k^{(k-1)} & U_k^{(k-1)} \end{bmatrix}$$

Obviously, the dimension of each term of the stacked estimates and the corresponding covariances in the two algorithms is the same, which is the same as x_k or $P_{k|k}$. Let the dimension of x_k be p, and the dimension of $P_{k|k}$ be $p \times p$. The total storages for the two algorithms are:

$$\text{ALG-S} \qquad sp + s^2(p \times p)$$

$$\text{ALG-I} \quad (s-1)p + 2s(p \times p)$$

Although the storage of ALG-S can be as small as $sp + (s^2 + s)(p \times p)/2$ because of the symmetry of the covariance matrix, the storage is still quadratic in s. The storage of the

ALG-I is linear in s, which means as the maximum delay s increases, the storage of ALG-S will be much larger than that of ALG-I. The maximum delay s could be quite large in the case of small sampling interval or large computational delay. In practice, $t_d \neq t_{k-l}$, to have good performance for ALG-S, the sampling interval must be small and thus s is large. Consequently, ALG-S should have significantly larger computational complexity than ALG-I.

By analysis and comparison, we can conclude that our proposed globally optimal update algorithm has (1) an efficient memory structure; and (2) an efficient computational structure to solve the problem by storing the necessary information instead of retrodiction or augmenting the state. Also, it is globally optimal for $t_{k-l} < t_d < t_{k-l+1}$ as well as $t_d = t_{k-l}$, whereas ALG-S is globally optimal only for $t_d = t_{k-l}$. On the other hand, ALG-S is conceptually clearer and simpler than ALG-I.

5.4.3 Algorithm II — Constrained Optimal Update

Only based on information $\hat{x}_{k|k}$ and z_d at the time when OOSM z_d arrives, the OOSM update is the LMMSE estimation $E^*(x_k|\hat{x}_{k|k}, z_d)$. It is in general not globally optimal [i.e., $E^*(x_k|\hat{x}_{k|k}, z_d) \neq E^*(x_k|z^k, z_d)$] because the measurements z^k and z_d of state x_k have correlated measurement noise. Of course, under some conditions, $E^*(x_k|\hat{x}_{k|k}, z_d) = E^*(x_k|z^k, z_d)$ holds. Also

$$E^*(x_k|\hat{x}_{k|k}, z_d) = \hat{x}_{k|k} + C_{x_k,\tilde{z}_{d|\hat{x}_{k|k}}} C^{-1}_{\tilde{z}_{d|\hat{x}_{k|k}}} \tilde{z}_{d|\hat{x}_{k|k}}$$

where

$$\tilde{z}_{d|\hat{x}_{k|k}} = z_d - \bar{z}_d - C_{z_d,\hat{x}_{k|k}} C^{-1}_{\hat{x}_{k|k}} (\hat{x}_{k|k} - \bar{x}_k)$$

Because $\Omega_k = \{\hat{x}_{k|k}, P_{k|k}\}$ does not sum up all prior information for this case, the LMMSE update with prior involves the prior information $\bar{x}_d, C_{x_d}$, which generally are not stored in the Kalman filter. So if we want to get the LMMSE with prior update, the information storage should increase to include the prior information. Now, not increasing our information storage in the Kalman filer, we present the LMMSE update without prior, which is derived as follows.

Let

$$z = \begin{bmatrix} \hat{x}_{k|k} \\ \\ z_d \end{bmatrix}$$

and treat z as the observation of x_k. Then the LMMSE estimator $\hat{x}_{k|k,d}$ of x_k must be a linear combination of $\hat{x}_{k|k}$ and z_d, i.e., a linear function of z

$$\hat{x}_{k|k,d} = Kz + b$$

We obtain the optimal K and b by satisfying the unbiasedness assumption and minimizing the MSE matrix. According to the unbiasedness $E(x_k) = E(\hat{x}_{k|k,d})$ requirement, we have, by (5.1)-(5.2),

$$F_{k,d}\bar{x}_d = KH\bar{x}_d + b$$

where

$$H = \begin{bmatrix} F_{k,d} \\ \\ H_d \end{bmatrix}$$

i.e.

$$(KH - F_{k,d})\bar{x}_d + b = 0$$

Since the prior information is not known, this equation must be satisfied for every $\bar{x}_d$, and

so

$$KH = F_{k,d},\ b = 0 \tag{5.16}$$

A solution of (5.16) always exists, because $KH = F_{k,d}$ holds at least for $K = [I, 0]$. The next step is to obtain the optimal K by minimizing the MSE matrix under linear constraint $KH = F_{k,d}$:

$$K = \arg\min_K \text{MSE}(\hat{x}_{k|k,d}) = \arg\min_K \{(K-\Gamma)R(K-\Gamma)'\} \tag{5.17}$$

$$s.t.\ \ KH = F_{k,d}$$

and

$$\Gamma = \begin{bmatrix} F_{k,d}U'_{k,d} + Q_{k,d} - P_{k|k} & 0 \end{bmatrix} R^{\dagger}$$

$$R = \begin{bmatrix} F_{k,d}U'_{k,d} + U_{k,d}F'_{k,d} + Q_{k,d} - P_{k|k} & 0 \\ 0 & R_d \end{bmatrix}$$

where the last equality in (5.17) follows from a tedious derivation, see below

$$MSE(\hat{x}_{k|k,d}) = \text{cov}(x_k - \hat{x}_{k|k,d}) = \text{cov}(x_k - Kz) = \text{cov}(F_{k,d}x_d + w_{k,d} - K \begin{bmatrix} \hat{x}_{k|k} \\ z_d \end{bmatrix})$$

$$= \text{cov}(F_{k,d}x_d + w_{k,d} - K \begin{bmatrix} x_k - \tilde{x}_{k|k} \\ H_d x_d + v_d \end{bmatrix}) = \text{cov}(F_{k,d}x_d + w_{k,d} - K \begin{bmatrix} F_{k,d}x_\tau + w_{k,d} - \tilde{x}_{k|k} \\ H_d x_d + v_d \end{bmatrix})$$

$$= \text{cov}\{(F_{k,d} - KH)x_d + w_{k,d} - K \begin{bmatrix} w_{k,d} - \tilde{x}_{k|k} \\ v_d \end{bmatrix}\} = \text{cov}(w_{k,d} - K \begin{bmatrix} w_{k,d} - \tilde{x}_{k|k} \\ v_d \end{bmatrix})$$

$$= Q_{k,d} - \begin{bmatrix} C_{w_{k,d}, w_{k,d} - \tilde{x}_{k|k}} & 0 \end{bmatrix} K' - K \begin{bmatrix} C'_{w_{k,d}, w_{k,d} - \tilde{x}_{k|k}} \\ 0 \end{bmatrix} + K \begin{bmatrix} C_{w_{k,d} - \tilde{x}_{k|k}} & 0 \\ 0 & R_d \end{bmatrix} K'$$

$$= (K-\Gamma)R(K-\Gamma)' + Q_{k,d} - \Gamma R \Gamma'$$

where

$$\Gamma = \begin{bmatrix} F_{k,d}U'_{k,d} + Q_{k,d} - P_{k|k} & 0 \end{bmatrix} R^{\dagger}$$

$$R = \begin{bmatrix} F_{k,d}U'_{k,d} + U_{k,d}F'_{k,d} + Q_{k,d} - P_{k|k} & 0 \\ 0 & R_d \end{bmatrix}$$

thus the optimization problem of K is

$$K = \arg\min_{K} \text{MSE}(\hat{x}_{k|k,d}) = \arg\min_{K} \{(K-\Gamma)R(K-\Gamma)'\}$$

$$s.t. \;\; KH = F_{k,d}$$

The general solution, expressed in terms of the MP-inverse, is given by:

$$K = \tilde{K} + \xi T$$

where

$$\tilde{K} = F_{k,d}H^{\dagger} + (\Gamma - F_{k,d}H^{+})R(TRT)^{\dagger}, \;\; T = I - HH^{\dagger}$$

and ξ is any matrix satisfying $\xi T R^{1/2} = 0$. So the estimate of x_k is

$$\hat{x}_{k|k,d} = Kz$$

$$P_{k|k,d} = \text{MSE}(\hat{x}_{k|k,d}) = (K-\Gamma)R(K-\Gamma)' + Q_{k,d} - \Gamma R \Gamma'$$

Note that

$$E(\xi T z) = E[\xi T(Hx_d - v)] = E(\xi T v) = 0$$

$$\text{cov}(\xi T z) = \xi T R T \xi = 0$$

Although K is not unique, the estimate of x_k is unique, given by

$$\hat{x}_{k|k,d} = \tilde{K}z$$

$$P_{k|k,d} = (\tilde{K}-\Gamma)R(\tilde{K}-\Gamma)' + Q_{k,d} - \Gamma R \Gamma' \tag{5.18}$$

Remark 1 When R matrix is non-singular, according to [49], we have

$$H^{\dagger}[I - R(TRT)^{\dagger}] = (H'R^{-1}H)^{\dagger}H'R^{-1}$$

Then

$$\tilde{K} = F_{k,d}(H'R^{-1}H)^{\dagger}H'R^{-1} + \Gamma R(TRT)^{\dagger}$$

Remark 2 For invertible $F_{k,d}$, the LMMSE estimate of x_k without prior is given by the following theorem.

Theorem 7: *For non-singular $F_{k,d}$, we have*

$$\hat{x}_k|k,d = \hat{x}_k|k,d^c, \;\; P_k|k,d = P_k|k,d^c$$

where

$$\hat{x}_k|k,d^c = \tilde{K}^c z, \;\; P_k|k,d^c = \tilde{K}^c R^c \tilde{K}^c$$

and

$$\tilde{K}^c = H^c + [I - R^c(T^c R^c T^c)^{+}], \;\; T^c = I - H^c(H^c)^{\dagger}$$

$$R^c = \begin{bmatrix} P_k|k & (P_k|kF_k, d^-1\prime - U_k, d)H_d' \\ H_d(F_k, d^-1P_k|k - U_k, d') & R_d + H_dF_k, d^-1Q_k, dF_k, d^-1\prime H_d' \end{bmatrix}, \;\; H^c = \begin{bmatrix} I \\ H_dF_k, d^-1 \end{bmatrix}$$

When R^c is invertible, by [49], it becomes

$$\hat{x}_k|k,d^c = (H^c\prime R^c - 1H^c)^-1H^c\prime R^c - 1z \tag{5.19}$$

$$P_k|k,d^c = (H^c\prime R^c - 1H^c)^-1$$

□

Proof. The LMMSE estimate in this case is

$$\hat{x}^c_{k|k,d} = K^c z + b^c$$

according to unbiasedness requirement, we have

$$\bar{x}_k = K^c H^c \bar{x}_k + b^c, \text{ i.e., } (K^c H^c - I)\bar{x}_k + b^c = 0$$

Without knowing the prior information $\bar{x}_k$ is equivalent to without knowing the prior information $\bar{x}_d$ because $F_{k,d}$ is invertible. The equation must be satisfied for every $\bar{x}_k$, and so

$$K^c H^c = I, \;\; b^c = 0$$

A solution always exists, because at least we can choose $K^c = [I, 0]$ to make $K^c H^c = I$ hold. The optimal K^c is the solution of the following optimization problem with a linear constraint:

$$K^c = \arg\min_K \text{MSE}(\hat{x}^c_{k|k,d}) = \arg\min_K K^c R^c K^{c\prime}$$
$$s.t. \;\; K^c H^c = I$$

By [53], we have $T^c = I - H^c (H^c)^\dagger$ and $\tilde{K}^c = (H^c)^\dagger [I - R^c (T^c R^c T^c)^\dagger]$. On the other hand, obviously we have $H^c F_{k,d} = H$, and in view of $K^c H^c = I$, we have

$$K^c H^c F_{k,d} = F_{k,d}, \text{ i.e., } K^c H = F_{k,d}$$

Based on this, the MSE of $\hat{x}^c_{k|k,d}$ has another form

$$\text{MSE}(\hat{x}^c_{k|k,d}) = (K^c - \Gamma) R (K^c - \Gamma)' + Q_{k,d} - \Gamma R \Gamma'$$

So

$$K^c = \arg\min_K (K^c - \Gamma) R (K^c - \Gamma)', \;\; s.t. \;\; K^c H = F_{k,d}$$

Because the linear constrained optimization problem for K^c is the same as that of K, we have

$$\hat{x}^c_{k|k,d} = \hat{x}_{k|k,d}, \;\; P^c_{k|k,d} = P_{k|k,d}$$

■

Obviously, for invertible $F_{k,d}$ and R^c, if the update is only within one step, (5.19) is the solution given by [28, 2]. In the multi-step update case, (5.19) is consistent with [59]. Thus we can say that these algorithms for update with $\Omega_k = \{\hat{x}_{k|k}, P_{k|k}\}$ and OOSM are optimal in the LMMSE sense. As such, we have proven the optimality of these existing algorithms. In Theorem 4.4.2.1, we have shown that $\{\hat{x}^c_{k|k,d}, P^c_{k|k,d}\}$ is a special case of our general LMMSE estimator $\{\hat{x}_{k|k,d}, P_{k|k,d}\}$ when $F_{k,d}$ is nonsingular.

In Algorithm II, the estimator contains a term $U_{k,d}$. According to Algorithm I, $U_{k,d}$ has the following recursion.

At each recursion n $(n \geq k-l+1)$

$$U_{n+1,d} = (I - K_{n+1}H_{n+1})F_{n+1,n}U_{n,d} \tag{5.20}$$

with initial value

$$U_{k-l+1,d} = (I - K_{k-l+1}H_{k-l+1})F_{k-l+1,d}P_{d|k-l} \tag{5.21}$$

where

$$P_{d|k-l} = F_{d,k-l}P_{k-l|k-l}F'_{d,k-l} + Q_{d,k-l}$$

Based on this recursion, it is easy to get that $U_{k,d}$ is highly related with the occurrence time of OOSM through $P_{d|k-l}$, which is highly related with the state estimation error covariance of x_k at the OOSM occurrence time. Again, the key to achieve optimality for the update lies in when and how to initialize the recursion.

According to the uncertainty of OOSM occurrence time, we also consider above three cases of *KF-OOSM* and associated information storage as Algorithm I.

Case I: Perfect Knowledge about t_d at the Next Sampling Time t_{k-l+1}

Similar to Algorithm I for Case I, the *KF-OOSM* adds a recursion for $U_{n,d}$ to the traditional Kalman filter. The flowchart of *KF-OOSM* for this case has the same structure as Algorithm I for Case I, where the OOSM initialization is given by (5.21), OOSM recursion is given by (5.20), and OOSM update by (5.18). The update part has the form of Remark 1 or 2 if the condition is satisfied. Since the Kalman stores $\{\hat{x}_{n|n}, P_{n|n}\}$ at each recursion, the information stored at our *KF-OOSM* at each recursion n $(k-l+1 \leq n \leq k)$ is

$$\Omega_n = \{\hat{x}_{n|n}, P_{n|n}, U_{n,d}\}$$

In this case, the storage is fixed as the delay l increases.

Case II: Knowing $t_{k-l} < t_d < t_{k-l+1}$ at Time t_{k-l+1}

In this case, we can initialize our *KF-OOSM* using the replacement U_{k-l+1} defined by

$$U_{k-l+1} = I - K_{k-l+1}H_{k-l+1} \tag{5.22}$$

and U_n $(n > k-l+1)$ has the recursion

$$U_{n+1} = (I - K_{n+1}H_{n+1})F_{n+1,n}U_n \tag{5.23}$$

so $U_{k,d}$ can be obtained by renewing U_k once the OOSM z_d arrives

$$U_{k,d} = U_k F_{k-l+1,d} P_{d|k-l} \tag{5.24}$$

The flowchart of *KF-OOSM* for this case has the same structure as Algorithm I for Case II, where the OOSM initialization is given by (5.22), OOSM recursion is given by (5.23), renew by (5.24) and OOSM update by (5.18). The *KF-OOSM* has the following information storage structure for each n $(k-l+1 \leq n \leq k)$:

$$\Omega_n = \{\hat{x}_{n|n}, P_{n|n}, U_n, P_{k-l|k-l}\}$$

As the delay l increases, the storage is fixed.

Case III: Knowing Maximum Delay s of OOSM

The method is to treat all time from t_{k-s+1} to t_k as a possible initialization point

$$U_n^{(n)} = (I - K_n H_n) \tag{5.25}$$

and applying the algorithm in Case II to achieve the optimal update with the OOSM. The recursion for $U_n^{(m)}$ $(n > k - l + 1,\ n - s < m < n)$ is

$$U_{n+1}^{(m)} = (I - K_{n+1} H_{n+1}) F_{n+1,n} U_n^{(m)} \tag{5.26}$$

so $U_{k,d}$ can be obtained by renewing $U_k^{(k-l+1)}$ once the OOSM z_d arrives, such as

$$U_{k,d} = U_k^{(k-l+1)} F_{k-l+1,d} P_{d|k-l} \tag{5.27}$$

The flowchart of *KF-OOSM* for this case has the same structure as Algorithm I for Case III, where the OOSM initialization is given by (5.25), OOSM recursion is given by (5.26), renew by (5.27), and OOSM update by (5.18). The information needed to store in our *KF-OOSM* at each recursion n $(k - s < n \leq k)$ in this case increases linearly from time t_{k-s+1} to t_{k-1},

$$\Omega_n = \{\hat{x}_{n|n}, P_{k-s|k-s}, \cdots, P_{n|n}, U_n^{(k-s+1)}, \cdots, U_n^{(n)}\}$$

Remark If the OOSM was made exactly at a previous sampling time. With the algorithm slightly changed, the memory can be saved comparing with $t_{k-l} < t_d < t_{k-l+1}$. In this case, *KF-OOSM* flowchart structure is the same as above, except the OOSM initialization is given by

$$U_n^{(n)} = (I - K_n H_n) F_{n,n-1} P_{n|n}$$

and OOSM renew by

$$U_{k,d} = U_k^{(k-l)}$$

The memory structure of our *KF-OOSM* at each recursion n $(k-s < n \leq k)$ in this case is

$$\Omega_n = \{\hat{x}_{n|n}, P_{n|n}, U_n^{(k-s+1)}, \cdots, U_n^{(n)}\}$$

Algorithm II can always give the optimal update based on the information given. Algorithm II is general, and does not have any invertibility requirement of matrix $F_{k,d}$. If $F_{k,d}$ is non-singular, the expression of the solution can be simplified. Most often R and R^c are invertible, which leads to even more simplified results. The algorithms of [28, 2] are special cases of Algorithm II with invertible $F_{k,d}$ and R^c for the one-step update case. The algorithm of [59] solves multi-step update based on invertible matrices $F_{k,d}$ and R^c. Therefore, we have proven that these existing algorithms are optimal in the LMMSE sense.

The algorithm C of [2] is also a special case of Algorithm II with $w_{k,d} = 0$ and invertible $F_{k,d}$. But by setting some terms to zero in the algorithms, the new estimates may or may not be the minimizer of the original problem, i.e., MSE($\hat{x}^*_{k|k,d}$) $\leq$MSE($\hat{x}_{k|k}$) may or may not hold, where $\hat{x}^*_{k|k,d}$ is the update estimation by applying Algorithm $*$.

It can be seen from the deviation of Algorithm II that the update Algorithm actually needs more information than $\{\hat{x}_{k|k}, P_{k|k}\}$ as provided by the Kalman filter, and the OOSM z_d. Although at the beginning we hope to update based on the current observations, i.e., the optimal linear combination of $\hat{x}_{k|k}$ and z_d, we need their correlation to build the linear combination weight which include both $P_{k|k}$ and $U_{k,d}$. It tells us that the valuable update which will improve the current state estimation in general can not only based on Kalman filer.

5.4.4 Update with Arbitrarily Delayed OOSMs

In all cases discussed before, we only consider the single-OOSM update problem. But there exists arbitrarily delayed multiple OOSMs for update. The OOSMs can be the measurements of the same state or different states; the OOSMs can arrive at the same or different time. The case that any OOSM arrives before the next OOSM occurrence time belongs to the single-OOSM update problem. We can solve it by sequentially applying the single-OOSM update algorithms discussed above. But in some cases, during the period between the occurrence time and arrival time of one OOSM, other OOSMs may occur. We now consider the optimal update problem in such cases, referred to as arbitrarily delayed OOSM update. In the following, we only consider the problem of update with two OOSMs. Generalization to update with more than two OOSMs is straight forward.

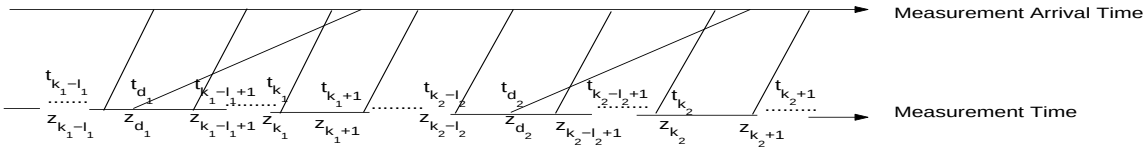


Fig. 5.4.4 The OOSMs within the maximum delay period

Suppose z_{d_1} and z_{d_2} are two OOSMs observed at $t_{k-l_i} \leq t_{d_i} < t_{k-l_i+1}$ with $1 \leq l_i < s$, $i = 1, 2$, and arrived during the time period $[t_{k_i}, t_{k_i+1})$. If z_{d_1} arrives before t_{d_2} (see Fig. 5.4.3), the state update with z_{d_2} at its arrival time is just the single-OOSM update problem as before. z_{d_1} had been used to update the state when it arrived. At z_{d_2} occurrence time, there is not any other OOSMs except z_{d_2}. So we can directly apply Algorithm I or II for updating with the single-OOSM z_{d_2} at its arrival time. If both of them arrive at the same

time, although we can update the state estimate with them stacked together, computationally and operationally, it is better to update with the OOSM one by one sequentially.

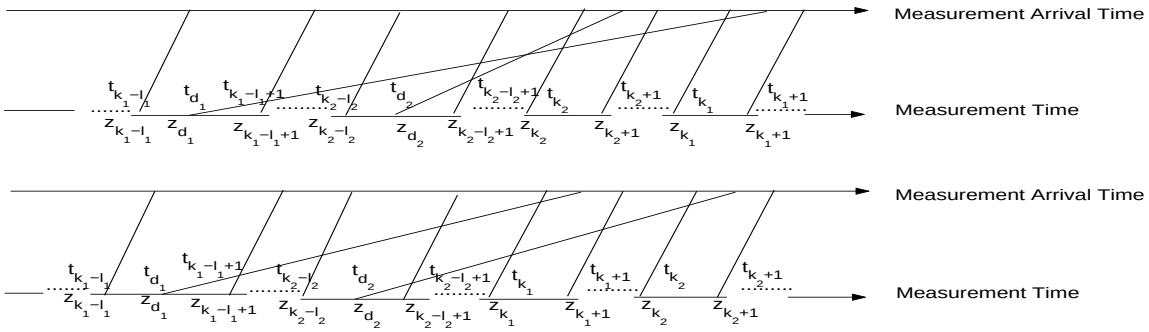


Fig. 5.4.5 The OOSMs within the maximum delay period

In the following, we will consider the case that z_{d_1} arrives after t_{d_2} (see Fig. 5.4.5). Suppose z_{d_2} arrives before z_{d_1}, or we process z_{d_2} before z_{d_1} if both of them arrive at the same time. According to Algorithm I and II for single-OOSM update, we need to update the state estimation $\hat{x}_{k_2|k_2}$ and $P_{k_2|k_2}$ with z_{d_2} when it arrived. At the same time, we also need to update other quantities, such as $\{\hat{x}_{d_1|k_2}, P_{d_1|k_2}, U_{k_2,d_1}\}$ for AlG-I, $\{U_{k_2,d_1}\}$ for ALG-II with z_{d_2}. Because z_{d_1} has not arrived yet, the quantities are necessary for updating the state estimate with later arrived z_{d_1}. Based on different update procedures at z_{d_2} arrival time, we consider two LMMSE optimal updates cases: (a) globally optimal and (b) constrained optimal.

Globally Optimal Update

In this globally optimal update case, when z_{d_2} arrives, we need to update not only $\{\hat{x}_{k_2|k_2}, P_{k_2|k_2}\}$ with z_{d_2}, but also $\{\hat{x}_{d_1|k_2}, P_{d_1|k_2}, U_{k_2,d_1}\}$ used to update with the next z_{d_1}. Denote the updated quantities as $\{\hat{x}_{k_2|k_2,d_2}, P_{k_2|k_2,d_2}\}$ and $\{\hat{x}_{d_1|k_2,d_2}, P_{d_1|k_2,d_2}, U^*_{k_2,d_1}\}$. Update from $\{\hat{x}_{k_2|k_2}, P_{k_2|k_2}\}$

to $\{\hat{x}_{k_2|k_2,d_2}, P_{k_2|k_2,d_2}\}$ is trivial, it can be done by directly applying single-OOSM globally optimal update algorithm. Here we focus on the update from $\{\hat{x}_{d_1|k_2}, P_{d_1|k_2}, U_{k_2,d_1}\}$ to $\{\hat{x}_{d_1|k_2,d_2}, P_{d_1|k_2,d_2}, U^*_{k_2,d_1}\}$.

By definition

$$\hat{x}_{d_1|k_2,d_2} = E^*(x_{d_1}|z^{k_2}, z_{d_2}),\ \ P_{d_1|k_2,d_2} = \mathrm{MSE}(\hat{x}_{d_1|k_2,d_2}),\ \ U^*_{k_2,d_1} = C_{x_{k_2},\tilde{x}_{d_1|k_2,d_2}}$$

According to the recursive LMMSE, we have

$$\hat{x}_{d_1|k_2,d_2} = \hat{x}_{d_1|k_2} + C^{k_2}_{d_2,d_1} H'_{d_2} (H_{d_2} P_{d_2|k_2} H'_{d_2})^{-1} \tilde{z}_{d_2|k_2}$$

$$P_{d_1|k_2,d_2} = P_{d_1|k_2} - C^{k_2}_{d_2,d_1} H'_{d_2} (H_{d_2} P_{d_2|k_2} H'_{d_2})^{-1} H_{d_2} (C^{k_2}_{d_2,d_1})'$$

$$U^*_{k_2,d_1} = U_{k_2,d_1} + U_{k_2,d_2} H'_{d_2} (H_{d_2} P_{d_2|k_2} H'_{d_2})^{-1} H_{d_2} (C^{k_2}_{d_2,d_1})'$$

where

$$\tilde{z}_{d_2|k_2} = z_{d_2} - H_{d_2} \hat{x}_{d_2|k_2},\ \ C^{k_2}_{d_2,d_1} = C_{x_{d_1},\tilde{x}_{d_2|k_2}}$$

Let

$$C^n_{d_2,d_1} = C_{x_{d_1},\tilde{x}_{d_2|n}}$$

Theorem 8: $C^n_{d_1,d_2} = (C^n_{d_2,d_1})'$. □

Proof.

$$\begin{aligned} C^n_{d_1,d_2} &= \mathrm{cov}(x_{d_2}, x_{d_1} - \hat{x}_{d_1|n}) \\ &= \mathrm{cov}(x_{d_2} - \hat{x}_{d_2|n}, x_{d_1} - \hat{x}_{d_1|n}) \\ &= \mathrm{cov}(x_{d_2} - \hat{x}_{d_2|n}, x_{d_1}) \\ &= \mathrm{cov}(x_{d_1}, x_{d_2} - \hat{x}_{d_2|n})' \\ &= (C^n_{d_2,d_1})' \end{aligned}$$

■

Theorem 9: *$C^n_{d_2,d_1}$ has a recursion for $k_2 - l_2 + 1 \leq n \leq k_2$, which has the form*

$$C^n_{d_2,d_1} = C^{n-1}_{d_2,d_1} - U'_{n-1,d_1} F'_{n,n-1} H'_n S_n^{-1} H_n F_{n,n-1} U_{n-1,d_2} \tag{5.28}$$

with initial value

$$C^{k_2-l_2}_{d_2,d_1} = F_{d_2,k_2-l_2} U_{k_2-l_2,d_1}$$

□

Proof. When $n \geq k_2 - l_2$

$$\begin{aligned}
C^n_{d_2,d_1} &= \mathrm{cov}(x_{d_1}, x_{d_2} - \hat{x}_{d_2|n}) \\
&= \mathrm{cov}(x_{d_1}, x_{d_2} - \hat{x}_{d_2|n-1} - U'_{n-1,d_2} F'_{n,n-1} H'_n S_n^{\dagger} \tilde{z}_{n|n-1}) \\
&= \mathrm{cov}(x_{d_1}, x_{d_2} - \hat{x}_{d_2|n-1}) - \mathrm{cov}(x_{d_1}, \tilde{x}_{n-1|n-1}) F'_{n,n-1} H'_n S_n^{\dagger} H_n F_{n,n-1} U_{d_2,n-1} \\
&= C^{n-1}_{d_2,d_1} - U'_{n-1,d_1} F'_{n,n-1} H'_n S_n^{\dagger} H_n F_{n,n-1} U_{n-1,d_2}
\end{aligned}$$

with initial value

$$C^{k-l_2}_{d_2,d_1} = F_{d_1,k_2-l_2} U_{k_2-l_2,d_1}$$

■

After this update procedure, we will have $\{\hat{x}_{k_2|k_2,d_2},\ P_{k_2|k_2,d_2}\}$ and $\{\hat{x}_{d_1|k_2,d_2},\ P_{d_1|k_2,d_2},\ U^*_{k_2,d_1}\}$. Through *KF-OOSM*, at z_{d_1} arrival time, we will get $\{\hat{x}_{k_1|k_1,d_2},\ P_{k_1|k_1,d_2}\}$ and $\{\hat{x}_{d_1|k_1,d_2},\ P_{d_1|k_1,d_2},\ U^*_{k_1,d_1}\}$, where $U^*_{k_1,d_1} = C_{x_{k_1},\tilde{x}_{d_1|k_1,d_2}}$. Now, the single-OOSM globally optimal update Algorithm I can be directly applied to obtain $\{\hat{x}_{k_1|k_1,d_2,d_1},\ P_{k_1|k_1,d_2,d_1}\}$.

It is easy to see that this OOSM update is just a sequential application of the single-OOSM globally optimal update algorithm except that at each OOSM arrival point, we need update not only the state estimate but also some other necessary quantities prepared to

update other OOSMs which are not arrived yet. This update contains a new term $C^{k_2}_{d_2,d_1}$ (i.e., the correlation between two OOSMs) which fortunately has a recursive form. So very similar to Algorithm I for single OOSM update case, we can have *KF-OOSM* for the three different cases considered before.

Constrained Optimal Update

In this constrained optimal update case, when z_{d_2} arrives, we need to update not only $\{\hat{x}_{k_2|k_2}, P_{k_2|k_2}\}$ with z_{d_2}, but also $\{U_{k_2,d_1}\}$. Denote the updated quantities as $\{\hat{x}_{k_2|k_2,d_2}, P_{k_2|k_2,d_2}\}$ and $\{\bar{U}_{k_2,d_1}\}$. Update from $\{\hat{x}_{k_2|k_2}, P_{k_2|k_2}\}$ to $\{\hat{x}_{k_2|k_2,d_2}, P_{k_2|k_2,d_2}\}$ is trivial by directly applying single-OOSM constrained optimal update algorithm. Now, we focus on how to update from $\{U_{k_2,d_1}\}$ to $\{\bar{U}_{k_2,d_1}\}$.

Theorem 10:

$$\bar{U}_{k_2,d_1} = C_{x_{d_1},\tilde{x}_{k_2|k_2,d_2}} = \begin{bmatrix} U_{k_2,d_1} & 0 \end{bmatrix} \tilde{K}' \tag{5.29}$$

where

$$\tilde{K} = F_{k_2,d_2}H^\dagger + (\Gamma - F_{k_2,d_2}H^\dagger)R(TRT)^{-1},\ T = I - HH^\dagger$$

$$\Gamma = \begin{bmatrix} F_{k_2,d_2}\bar{U}_{k_2,d_2} + Q_{k_2,d_2} - P_{k_2|k_2,d_2} & 0 \end{bmatrix} R^\dagger$$

$$R = \begin{bmatrix} F_{k_2,d_2}\bar{U}_{k_2,d_2} + \bar{U}_{k_2,d_2}F'_{k_2,d_2} + & 0 \\ Q_{k_2,d_2} - P_{k_2|k_2,d_2} & \\ 0 & R_{d_2} \end{bmatrix},\ H = \begin{bmatrix} F_{k_2,d_2} \\ H_{d_2} \end{bmatrix}$$

□

Proof.

$$
\begin{aligned}
\bar{U}_{k_2,d_1} &= C_{x_{d_1},\tilde{x}_{k_2|k_2,d_2}} \\
&= cov(x_{d_1}, x_{k_2} - \hat{x}_{k_2|k_2,d_2}) \\
&= cov(x_{d_1}, x_{k_2} - \tilde{K}_{(2)} \begin{bmatrix} \hat{x}_{k_2|k_2} \\ z_{d_2} \end{bmatrix}) \\
&= cov(x_{d_1}, x_{k_2} - \tilde{K}_{(2)} \begin{bmatrix} x_{k_2} - \tilde{x}_{k_2|k_2} \\ z_{d_2} \end{bmatrix}) \\
&= cov(x_{d_1}, F_{k_2,d_2} x_{d_2} + w_{k_2,d_2} - \tilde{K}_{(2)} H_{(2)} x_{d_2} - \tilde{K}_{(2)} \begin{bmatrix} w_{k_2 d_2} - \tilde{x}_{k_2|k_2} \\ z_{d_2} \end{bmatrix}) \\
&= cov(x_{d_1}, w_{k_2,d_2} - \tilde{K}_{(2)} \begin{bmatrix} w_{k_2,d_2} - \tilde{x}_{k_2|k_2} \\ v_{d_2} \end{bmatrix}) \\
&= \begin{bmatrix} U_{k_2,d_1} & 0 \end{bmatrix} \tilde{K}'_{(2)}
\end{aligned}
$$

where $\tilde{K}_{(2)}$ is the gain matrix for update with z_{d_2} at its arrival time. ■

In fact, $\tilde{K}$ is the gain matrix for update with z_{d_2} when it arrives. Thus after this update procedure, we have $\{\hat{x}_{k_2|k_2,d_2}, P_{k_2|k_2,d_2}\}$ and $\{\bar{U}_{k_2,d_1}\}$. Through recursion, at z_{d_1} arrival time, we can have $\{\hat{x}_{k_1|k_1,d_2}, P_{k_1|k_1,d_2}\}$ and $\{U^*_{k_1,d_1}\}$, where $U^*_{k_1,d_1} = C_{x_{d_1},\tilde{x}_{k_1|k_1,d_2}}$. Now, the single-OOSM constrained optimal update Algorithm II can be directly applied to obtain state estimate $\hat{x}_{k_1|k_1,d_2,d_1}$ and $P_{k_1|k_1,d_2,d_1}$.

This constrained OOSM update is just the sequential single-OOSM constrained optimal update procedure. So all previous conditions that will simplify the estimation formulas can be derived directly here. The three cases for different uncertainty of the OOSMs occurrence time can be considered in the same way as the single-OOSM case.

5.5 OOSMs Update for Linear Gaussian Systems under Nonsingularity Conditions

In a linear Gaussian dynamic system, the dynamic and measurement models for a single target are given by

$$x_j = F_{j,j-1}x_{j-1} + w_{j,j-1} \tag{5.30}$$

$$z_j = H_j x_j + v_j \tag{5.31}$$

where $F_{j,j-1}$ is the state transition matrix from time t_{j-1} to t_j and $w_{j,j-1}$ is (the cumulative effect of) the process noise for this interval. The process noise $w_{j,j-1}$ and the measurement noise v_j are white, mutually uncorrelated, with zero mean and variances $\text{var}(w_{j,j-1}) = Q_{j,j-1}$, $\text{var}(v_j) = R_j$.

5.5.1 Algorithm I — Globally Optimal Update (ALG-I)

Based on the linear dynamic model, it follows from recursive LMMSE estimation that the globally optimal update can be written as

$$\hat{x}_{k|k,d} = E^*[x_k|z^k, z_d] = \hat{x}_{k|k} + K_d(z_d - H_d\hat{x}_{d|k}) = \hat{x}_{k|k} + K_d\tilde{z}_{d|k} \tag{5.32}$$

$$P_{k|k,d} = P_{k|k} - K_d S_d K_d' \tag{5.33}$$

where

$$K_d = U_{k,d}H_d'S_d^{-1},\ \ S_d = H_d P_{d|k}H_d' + R_d,\ \ U_{k,d} = C_{x_k,\tilde{x}_{d|k}}$$

Let

$$\hat{x}_{d|n} = E^*(x_d|z^n),\ \ P_{d|n} = \text{MSE}(\hat{x}_{d|n}),\ \ U_{n,d} = C_{x_n,\tilde{x}_{d|n}}$$

Theorem 11: *When $P_{n+1|n+1}$, $P_{n+1|n}$ and R_{n+1} are nonsingular (which hold for most target tracking problems), we have*

$$\begin{aligned} H'_{n+1}S^{-1}_{n+1}\tilde{z}_{n+1|n} &= P^{-1}_{n+1|n}(\hat{x}_{n+1|n+1} - \hat{x}_{n+1|n}) \\ H'_{n+1}S^{-1}_{n+1}H_{n+1} &= P^{-1}_{n+1|n}(P_{n+1|n} - P_{n+1|n+1})P^{-1}_{n+1|n} \\ (I - K_{n+1}H_{n+1}) &= P_{n+1|n+1}P^{-1}_{n+1|n} \end{aligned} \tag{5.34}$$

□

Proof.

$$\begin{aligned} S^{-1}_{n+1} &= (H_{n+1}P_{n+1|n}H'_{n+1} + R_{n+1})^{-1} \\ &= R^{-1}_{n+1} - R^{-1}_{n+1}H'_{n+1}(P^{-1}_{n+1|n} + H'_{n+1}R^{-1}_{n+1}H_{n+1})^{-1}H_{n+1}R^{-1}_{n+1} \end{aligned}$$

$$\begin{aligned} H'_{n+1}S^{-1}_{n+1}\tilde{z}_{n+1|n} &= H'_{n+1}R^{-1}_{n+1}\tilde{z}_{n+1|n} - H'_{n+1}R^{-1}_{n+1}H'_{n+1}(P^{-1}_{n+1|n} + H'_{n+1}R^{-1}_{n+1}H_{n+1})^{-1}H_{n+1}R^{-1}_{n+1}\tilde{z}_{n+1|n} \\ &= P^{-1}_{n+1|n}(P^{-1}_{n+1|n} + H'_{n+1}R^{-1}_{n+1}H_{n+1})^{-1}H_{n+1}R^{-1}_{n+1}\tilde{z}_{n+1|n} \\ &= P^{-1}_{n+1|n}(P^{-1}_{n+1|n} + P^{-1}_{n+1|n+1} - P^{-1}_{n+1|n})^{-1}(P^{-1}_{n+1|n+1}\hat{x}_{n+1|n+1} - P^{-1}_{n+1|n}\hat{x}_{n+1|n} \\ &\quad - P^{-1}_{n+1|n+1}\hat{x}_{n+1|n} + P^{-1}_{n+1|n}\hat{x}_{n+1|n}) \\ &= P^{-1}_{n+1|n}P_{n+1|n+1}(P^{-1}_{n+1|n+1}\hat{x}_{n+1|n+1} - P^{-1}_{n+1|n+1}\hat{x}_{n+1|n}) \\ &= P^{-1}_{n+1|n}(\hat{x}_{n+1|n+1} - \hat{x}_{n+1|n}) \end{aligned}$$

$$\begin{aligned} (I - K_{n+1}H_{n+1}) &= I - P_{n+1|n}H'_{n+1}S^{-1}_{n+1}H_{n+1} \\ &= I - P_{n+1|n}[P^{-1}_{n+1|n}(P_{n+1|n} - P_{n+1|n+1})P^{-1}_{n+1|n}] \\ &= (P_{n+1|n} - P_{n+1|n} + P_{n+1|n+1})P^{-1}_{n+1|n} \\ &= P_{n+1|n+1}P^{-1}_{n+1|n} \end{aligned}$$

■

Based on the recursion for $\{\hat{x}_{d|k},\, P_{d|k},\, U_{k,d}\}$ derived in [81] and Theorem 11, the recursion for $\{\hat{x}_{d|k}, P_{d|k}, U_{k,d}\}$ starting from $n = k - l + 1$ can be rewritten as

$$\begin{aligned}
\hat{x}_{d|n+1} &= \hat{x}_{d|n} + U'_{n,d}F'_{n+1,n}P^{-1}_{n+1|n}(\hat{x}_{n+1|n+1} - \hat{x}_{n+1|n}) \\
P_{d|n+1} &= P_{d|n} - U'_{n,d}F'_{n+1,n}P^{-1}_{n+1|n}(P_{n+1|n} - P_{n+1|n+1})P^{-1}_{n+1|n}F_{n+1,n}U_{n,d} \\
U_{n+1,d} &= P_{n+1|n+1}P^{-1}_{n+1|n}F_{n+1,n}U_{n,d}
\end{aligned} \tag{5.35}$$

with the initial conditions given by

$$\begin{aligned}
\hat{x}_{d|k-l+1} &= \hat{x}_{d|k-l} + P_{d|k-l}F'_{k-l+1,d}P^{-1}_{k-l+1|k-l}(\hat{x}_{k-l+1|k-l+1} - \hat{x}_{k-l+1|k-l}) \\
P_{d|k-l+1} &= P_{d|k-l} - P_{d|k-l}F'_{k-l+1,d}P^{-1}_{k-l+1|k-l}(P_{k-l+1|k-l} - P_{k-l+1|k-l+1})P^{-1}_{k-l+1|k-l}F_{k-l+1,d}P_{d|k-l}
\end{aligned} \tag{5.36}$$

$$U_{k-l+1,d} = P_{k-l+1|k-l+1}P^{-1}_{k-l+1|k-l}F_{k-l+1,d}P_{d|k-l}$$

where

$$\hat{x}_{d|k-l} = F_{d,k-l}\hat{x}_{k-l|k-l} \tag{5.37}$$

$$P_{d|k-l} = F_{d,k-l}P_{k-l|k-l}F'_{d,k-l} + Q_{d,k-l} \tag{5.38}$$

From (5.35)-(5.36), we can see that the necessary information in order to update $\{\hat{x}_{d|k}, P_{d|k}, U_{k,d}\}$ includes

$$\Omega_k = \{\hat{x}_{k-l|k-l}, P_{k-l|k-l}, \cdots, \hat{x}_{k|k}, P_{k|k}\}$$

For this situation, we do not have any prior information about the OOSM z_d occurrence time t_d, and what we know is the maximum delay s for the OOSM, i.e., $t_{k-s} \le t_{k-l} \le t_d < t_{k-l+1} \le t_k$. In order to save all necessary information for the update, we should have

$$\Omega_k = \{\hat{x}_{k-s|k-s}, P_{k-s|k-s}, \cdots, \hat{x}_{k|k}, P_{k|k}\}$$

and use (5.35)-(5.36) together with (5.32)-(5.33) for the OOSM update algorithm. The OOSM filter update uses the traditional Kalman filter with an additional OOSM update, which is shown in **Fig5.5.1.**

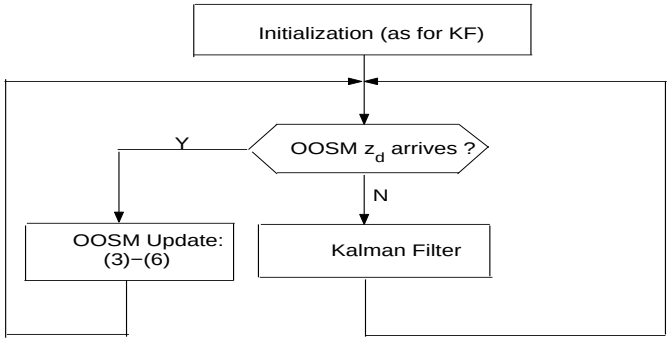


Fig5.5.1: Flowchart for Algorithm I

Algorithm I presented above is the globally optimal update [81] with less storage requirement since $P_{n+1|n+1}$, $P_{n+1|n}$ and R_{n+1} are all nonsingular. The storage requirement increases linearly with maximum delay s. However, the necessary information only includes the state estimates in the concerned time interval. The storage is the same as Algorithm $Al1$ proposed in [6], but can achieve the best performance within the linear class.

We conclude that Algorithm I has (1) an efficient memory structure; and (2) an efficient computational structure to solve the problem by storing the necessary information instead of retrodiction or augmenting the state [76, 28, 9, 14]. Also, it is globally optimal for $t_{k-l} < t_d < t_{k-l+1}$ as well as $t_d = t_{k-l+1}$.

5.5.2 Algorithm II — Constrained Optimal Update (ALG-II)

Based only on the information $\hat{x}_{k|k}$ and z_d at time when OOSM z_d arrives, the optimal OOSM update is the LMMSE estimator $E^*(x_k|\hat{x}_{k|k}, z_d)$ without prior information [53]. It

is in general not globally optimal [i.e., $E^*(x_k|\hat{x}_{k|k}, z_d) \neq E^*(x_k|z^k, z_d)$], but it is optimal conditioned on the information available. As presented in [81], the LMMSE update without prior is given by

$$\begin{aligned} \hat{x}_{k|k,d} &= \tilde{K} z \\ P_{k|k,d} &= (\tilde{K} - \Gamma)\bar{R}(\tilde{K} - \Gamma)' + Q_{k,d} - \Gamma \bar{R} \Gamma' \end{aligned} \tag{5.39}$$

where $z = \begin{bmatrix} \hat{x}_{k|k} \\ z_d \end{bmatrix}$ and

$$\begin{aligned} \tilde{K} &= F_{k,d}H^+ + (\Gamma - F_{k,d}H^+)\bar{R}(T\bar{R}T)^+ \\ T &= I - HH^+, \;\; H = \begin{bmatrix} F_{k,d} \\ H_d \end{bmatrix} \\ \Gamma &= \begin{bmatrix} F_{k,d}U'_{k,d} + Q_{k,d} - P_{k|k} & 0 \end{bmatrix} \bar{R}^+ \\ \bar{R} &= \begin{bmatrix} F_{k,d}U'_{k,d} + U_{k,d}F'_{k,d} + Q_{k,d} - P_{k|k} & 0 \\ 0 & R_d \end{bmatrix} \end{aligned} \tag{5.40}$$

where $[\cdot]^+$ is the MP inverse of $[\cdot]$. Algorithm II always gives the optimal update based on the available information. We have the property that, for nonsingular $F_{k,d}$ and R^c, if the update is only one lag, (5.39) is the solution given by [28, 2]. In the multi-step update case, (5.39) is consistent with [59]. In Algorithm II, the estimator contains the term $U_{k,d}$. As for Algorithm I, when $P_{n+1|n+1}$, $P_{n+1|n}$ and R_{n+1} are nonsingular, we have

$$(I - K_{n+1}H_{n+1}) = P_{n+1|n+1}P^{-1}_{n+1|n}$$

So the recursion for $\{U_{k,d}\}$ starting from $n = k - l + 1$ can also be rewritten as

$$U_{n+1,d} = P_{n+1|n+1}P^{-1}_{n+1|n}F_{n+1,n}U_{n,d} \tag{5.41}$$

with the initial value

$$U_{k-l+1,d} = P_{k-l+1|k-l+1} P^{-1}_{k-l+1|k-l} F_{k-l+1,d} P_{d|k-l} \tag{5.42}$$

If the maximum delay for the OOSM is s. We should require

$$\Omega_k = \{P_{k-s|k-s}, \cdots, P_{k|k}\}$$

Similar to Algorithm I, this OOSM update algorithm can be implemented at the arrival time of OOSM z_d. The OOSM update is the traditional Kalman filter with the OOSM update using equations (5.39)-(5.42).

Algorithm II always gives the optimal update based on the available information when $P_{n+1|n+1}$, $P_{n+1|n}$ and R_{n+1} are all nonsingular. If $F_{k,d}$ is nonsingular, the expression of (5.40) can be simplified (See **Remarks 1** in [81]). Most often $\bar{R}$ are nonsingular, which leads to even more simplified results (See **Remarks 2** in [49]). The information storage increases linearly with the maximum delay s. The storage is the same as Algorithm $Bl1$ of [6], it can achieve better performance in terms of the MSE errors.

5.5.3 Update with Arbitrarily Delayed OOSMs

In the case of arbitrarily delayed multiple OOSMs, if we consider the case that z_{d_1} arrives after t_{d_2}, we can not simply apply the single OOSM update algorithm twice. Suppose z_{d_2} arrives before z_{d_1}, or we process z_{d_2} before z_{d_1} if both of them arrive at the same time. According to ALG-I and ALG-II for single-OOSM update, we need to update $\hat{x}_{k_2|k_2}$ and $P_{k_2|k_2}$ with z_{d_2} when it arrived. By using a Kalman filter, until time t_{k_1}, we have the state estimate sequence $\{\hat{x}_{k_2|k_2,d_2}, P_{k_2|k_2,d_2}, \ldots, \hat{x}_{k_1|k_1,d_2}, P_{k_1|k_1,d_2}\}$. At the time when OOSM z_{d_1}

arrives, in order to update $\{\hat{x}_{k_1|k_1,d_2},\ P_{k_1|k_1,d_2}\}$, the necessary information for Algorithm I or II includes $\{\hat{x}_{k_2-l_2|k_2-l_2,d_2},\ P_{k_2-l_2|k_2-l_2,d_2},\ \ldots,\ \hat{x}_{k_2-1|k_2-1,d_2},\ P_{k_2-1|k_2-1,d_2}\}$ or $\{P_{k_2-l_2|k_2-l_2,d_2},\ \ldots,\ P_{k_2-1|k_2-1,d_2}\}$. Therefore, at the time when the first OOSM z_{d_2} arrives, we need to update not only update the current state $\{\hat{x}_{k_2|k_2},\ P_{k_2|k_2}\}$, but also $\{\hat{x}_{k_2-l_2|k_2-l_2},\ P_{k_2-l_2|k_2-l_2},\ \ldots,\ \hat{x}_{k_2-1|k_2-1},\ P_{k_2-1|k_2-1}\}$ for Algorithm I or $\{P_{k_2-l_2|k_2-l_2},\ \ldots,\ P_{k_2-1|k_2-1}\}$ for Algorithm II between OOSM z_{d_2} occurrence time and its arrival time. The update for $\{\hat{x}_{k_2-i|k_2-i},\ P_{k_2-i|k_2-i}\}$ or just $\{P_{k_2-i|k_2-i}\}$ with $i = 1, \ldots, l_2$ is quite simple. It can be implemented with the same procedure for$\{\hat{x}_{k_2|k_2},\ P_{k_2|k_2}\}$ by treating the arrival time of OOSM z_{d_2} as in the time interval $[t_{k-i}, t_{k-i+1}]$ with $i = 1, \ldots, l_2$.

5.6 Numerical Examples

Several simple numerical examples are given in this section to verify the formulas presented and the existence of the optimal solution in the case where the state transition matrix is not invertible. All these examples are for the following linear system

$$x_j = F_{j-1}x_{j-1} + w_{j-1}$$

$$z_j = H_j x_j + v_j$$

where $x_j = \left[x_j^{(1)}, x_j^{(2)}\right]'$, and w_j and v_j are zero mean white Gaussian noise. In order to consider multi-lag delay as well as single-lag delay update, we choose a series of OOSMs z_d, these OOSM occurred at $d = (l+1)n$ and arrived at $(l+1)n + l$ with $n = 1, 2, \ldots$, which corresponding to l-lag delayed OOSMs, where $l = 1, 2, \ldots$. For example, suppose the in-sequence observation series is $\{z_1, z_2, z_3, \ldots\}$, then the observation series with OOSMs for $l = 1$ is $\{z_1, z_3, z_2, z_5, z_4, \ldots\}$ and the updated states are $x_3, x_5, x_7, \ldots$; the observation

series with OOSMs for $l = 2$ is $\{z_1, z_2, z_4, z_3, z_5, z_7, z_8, z_6, \ldots\}$ and the updated states are $x_4, x_8, x_{11,} \ldots$; and so on. Globally optimal estimates were obtained by the Kalman filter using all observations (including OOSMs) in the right time sequence.

In the result, we use **mser**= $\frac{\text{trace}[P_{k|k,d}(\text{KF})]}{\text{trace}[P_{k|k,d}(\text{Algorithm})]}$, which is the ratio of the mean-square error of the globally optimal Kalman filter to that of the algorithm under consideration. It shows the efficiency of the algorithm. It is in the interval of $(0, 1]$. The larger the **mser** is, the better the algorithm is.

5.6.1 Nonsingular $F_{k,d}$

Consider a discretized continuous time kinematic system driven by white noise with power spectral density q, known as constant velocity model or white-noise acceleration model in target tracking, described by

$$F_j = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad H_j = [1, 0]$$

$$C_{w_j} = Q = \begin{bmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{bmatrix} q, \quad C_{v_j} = R = 1$$

where T is the sampling interval. The prior information is

$$\hat{x}_{0|0} = \bar{x} = [200\ Km, 0.5\ Km/\sec]', \quad P_{0|0} = \begin{bmatrix} R & R/T \\ R/T & 2R/T^2 \end{bmatrix}$$

and the maneuver index is $\lambda = \sqrt{qT^3/R}$.

Single-Step Update ($l = 1$)

In this example, we first apply our globally optimal update algorithm in Case I and II. Then we apply Algorithm B of [2], referred to as ALG-B, to compare with our optimal update Algorithm II with limited information, referred to as ALG-II. Although there are two outputs (filtering output or smoothed output) possible from the framework of [14], smoothed output algorithm needs use future observations to estimate the current state. It is unfair to compare the smoothed output update result with that of other algorithms by update only with the observations up to current time. In order to have a fair comparison, we only apply the filtering output Algorithm of [14], referred to as ALG-S. For the case $t_{k-1} < t_d < t_k$, in order to apply ALG-S, we approximate t_d to the nearest sampling time t_{k-1}. In Table I, we present **mser** of all algorithms at time $j = 3$. We adopt the same values of (λ, q) as in [2], [14], [59].

Table 5.1: **mser**s of algorithms

(λ, q)	KF	KF*	ALG-I	ALG-II	ALG-B	ALG-S
(2,2)	1	0.8786	1	0.9680	0.9680	0.7071
(1,1)	1	0.9121	1	0.9790	0.9790	0.8163
(0.5,0.5)	1	0.8787	1	0.9999	0.9999	0.9283

In Table 1, KF stands for the globally optimal Kalman filter; KF* stands for the Kalman filter without using OOSM. It is easy to see that our Algorithm I in Cases I and II give the globally optimal update. While Algorithm II gives the same update as Algorithm B of [2]. But the update using the algorithm of [14] is not optimal and sometimes can be worse than without using OOSM, i.e., **mser**(ALG-S) $<$ **mser**(KF*) [see the cases $(\lambda, q) = (2, 2)$

and $(\lambda, q) = (1, 1)$]. It is caused by the error arising from the approximation of the OOSM occurrence time t_d. For $(\lambda, q) = (0.5, 0.5)$, the sampling interval T becomes smaller, ALG-S becomes better.

In Figure 5.6.1, we show the theoretical and sample **mser** of ALG II for $(\lambda, q) = (2, 2)$. The two curves match each other very well, which verifies our update formula.

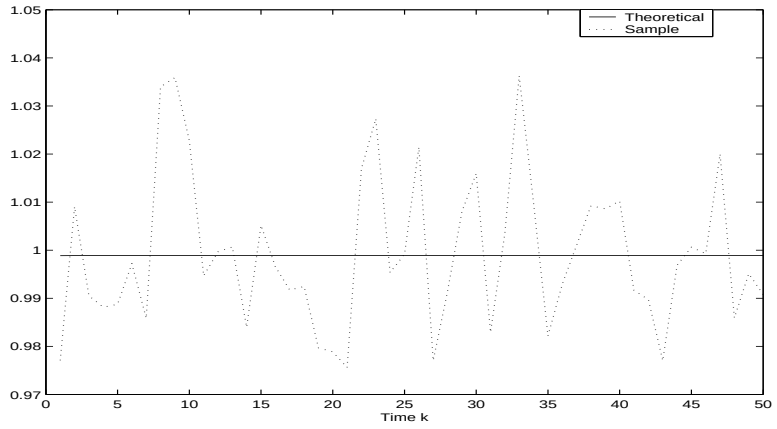


Fig 5.6.1 Theoretical and sample mser

It is reasonable to require that any algorithm which updates $\hat{x}_{k|k}$ with an OOSM z_d to yield $\hat{x}_{k|k,d}$ should satisfy

$$\text{MSE}(\hat{x}^*_{k|k,d}) \leq \text{MSE}(\hat{x}_{k|k}) \tag{5.43}$$

Otherwise update by the algorithm is questionable. All our algorithms satisfy (5.43), because they are optimal estimators that minimize MSE based on the information given, but not Alg-S of [14].

Multi-Step Update

In this example, first consider $l = 2$ with $d = 3n$ and $(\lambda, q) = (2, 2)$, we apply our globally optimal update Algorithm I in all three cases and yield **mser**$= 1$, which verifies the global

optimality of the algorithm. Also, we apply Algorithm II and compare the **mser** with the algorithm of [59], referred to as ALG-M. Figure 5.6.2 shows that the sample **mser** of ALG II matches its theoretical **mser**. ALG-II and ALG-M have the same theoretical **mser**. Meanwhile it shows the benefit of updating by comparing with KF*.

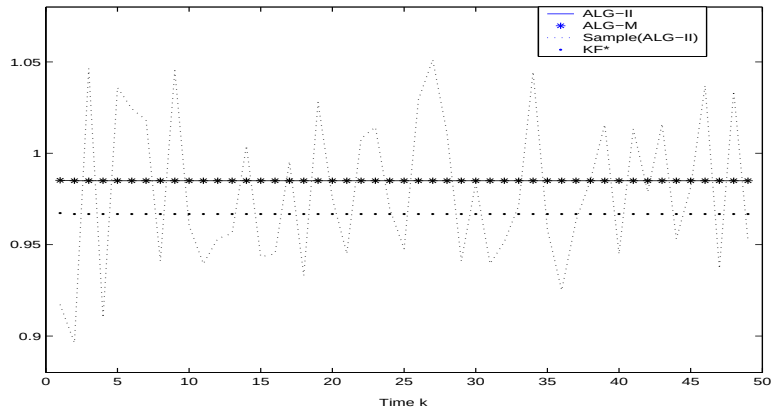


Fig 5.6.2 Comparison of mser

Table 2 shows the benefit of updating when $(\lambda, q) = (2, 2)$, $l = 1, \cdots, 3$ with $d = 2n, 3n, \cdots$ and $6n$ respectively, which shows that as the lag l of OOSM becomes larger, the benefit of updating becomes smaller. Results of KF* for Kalman filter by ignoring OOSM show that the effect of OOSM fades quickly with the target maneuvering behaviors. It provides a strong hint for the maximum delay s to consider, in addition to physical considerations.

5.6.2 Singular $F_{k,d}$

A system with a singular state transition matrix $F_{k,d}$ is rare in practice because most discrete time systems are discretized from continuous systems. However in some cases, when a

Table 5.2: **mser**s of algorithms

mser	$d = 2n$	$d = 3n$	$d = 4n$	$d = 5n$	$d = 6n$
ALG II(ALG M)	0.9680	0.9738	0.9976	0.9999	1
KF*	0.8786	0.9667	0.9972	0.9999	1

practical system is defined directly in discrete time, state transition matrix $F_{k,d}$ may be singular. The corresponding OOSM update problem needs to be considered. Also, allowing $F_{k,d}$ to be singular provides additional flexibility to handle some artificial system models, just like the study of noncausal systems, which is meaningful.

Here we consider a system with $H = [1, 1]$, $Q = 0.2I$, $R = 0.3I$ and prior information $\hat{x}_{0|0} = \bar{x} = [1, 1]'$, $P_{0|0} = 0.001I$.

Single-Step Update ($l = 1$)

In this case, we use

$$F_{2n-1} = \frac{1}{n^2}\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \; F_{2n} = \begin{bmatrix} 1 & -\frac{1}{n^2} \\ -1 & \frac{1}{n^2} \end{bmatrix}$$

Algorithm I in Case I or II can always gets **mser**= 1. As shown in Figure 5.6.3, the theoretical and sample **mser** of ALG II match each other, which verifies the formula.

Multi- step Update ($l = 2$)

In this case, $k = 3n + 2$, we use

$$F_{3n} = \begin{bmatrix} 1 & 1/n \\ -1 & -1/n \end{bmatrix}, \; F_j = \begin{bmatrix} 1 & 1/j \\ 0 & 1 \end{bmatrix} \quad j \neq 3n$$

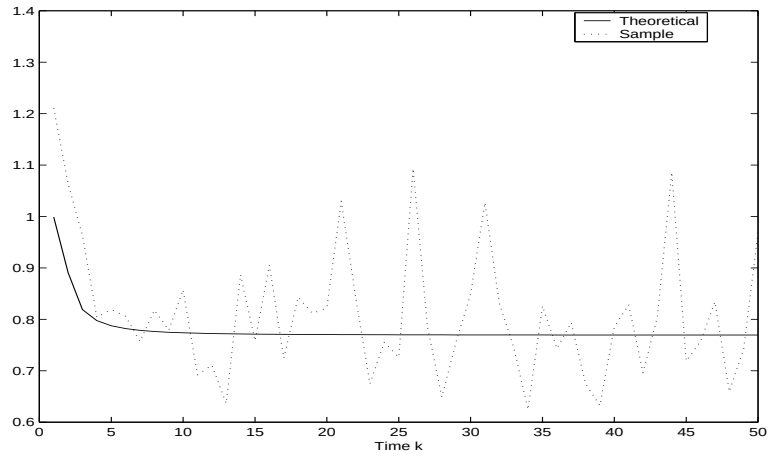


Fig 5.6.3 Theoretical and Sample mser

Algorithm I in all three cases all gave **mser**= 1. As shown in Figure 5.6.4, ALG II can give the benefit of updating, as shown by comparing **mser** with KF*.

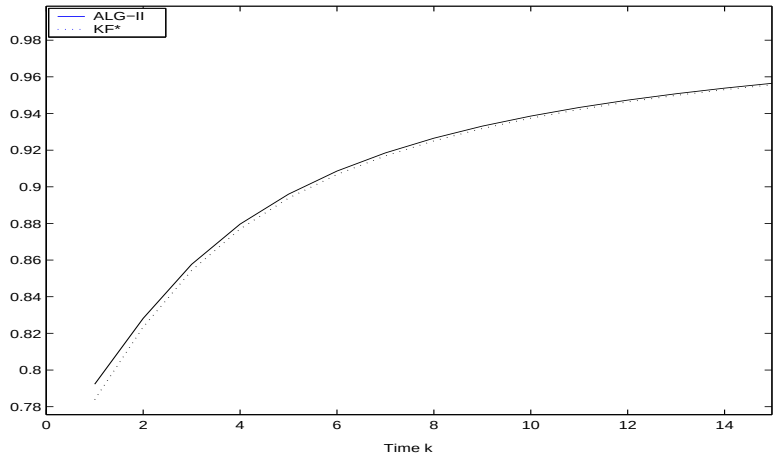


Fig 5.6.4 Comparison of mser

Examples in this section verify both Algorithms I and II. They show that our proposed Algorithms I and II are more general. They can solve single-step as well as multi-step update. Also, for singular state transition matrix $F_{k,d}$, they are still efficient.

5.7 Summary

We have presented two general algorithms with three cases of different information storage for state estimation update with out-of-sequence measurements. Both algorithms are optimal in the LMMSE sense for the information given and are more general than previously available algorithms. In particular, they are optimal for multiple-step as well as single-step update; they do not have any non-singularity requirement on the matrix $F_{k,d}$; they yield the best unbiased estimates among all linear estimation algorithms. Algorithm I is always globally optimal (in the LMMSE sense). Algorithm II is optimal (in the LMMSE sense) for the information given. Under the linear-Gaussian assumption of the Kalman filtering, both algorithms give the conditional mean, and have a simplified version under the assumptions of nonsingularity of certain matrices valid for most tracking applications.

Both algorithms need the smallest storage in Case I, the largest storage in Case III. The storage of Algorithm II for all cases are smaller than Algorithm I. For single-step update, the information stored is even smaller. Each item in the algorithms has a recursive form and can be computed easily, as presented. As illustrated by the simulation results, these variants in information storage complement each other in that they are suitable for different practical situations and yield the same optimal update.

Overall, both Algorithms have (1) an efficient processing structure for information update; (2) an efficient memory structure for storing historical information; (3) an efficient computational structure, and thus (4) an easy generalization for arbitrarily delayed multiple OOSMs.

Chapter 6

Multi-Sensor Multi-Target Tracking with OOSMs

6.1 Introduction

The previous results for OOSM update are formulated for a Kalman filter to update the state at time t_k by using the "older" measurement from time t_d. In this problem, the measurement at each sampling time is assumed to be target originated and no clutter or interference from other targets is considered. We call the above setting an OOSM update problem. There are some optimal methods [2, 14, 81] and suboptimal methods [59, 6, 76, 28, 9] for single-lag as well as multi-lag OOSM update. Two general algorithms ALG-I and ALG-II proposed in [81] can solve the one-lag as well as the multi-lag OOSM update problem globally optimal or suboptimal (optimal with limited information) with little restriction. They are optimal in the LMMSE sense.

However, in real-world multi-sensor multi-target tracking problems, measurements re-

ceived at the fusion center can originate from targets or clutter, i.e., false alarms. The filter handles the measurement origin uncertainty via the so-called data association. The existing optimal criterion for OOSM update within the Kalman filter framework is no longer valid for the target tracking problem with measurement origin uncertainty. In this case, the OOSM update needs to be combined with data association. However, the optimal data association (in the Bayesian sense) relies on all measurements from the beginning up to the current time. With limited storage, for example only based on the state estimate without storing the measurements, it is complicated to have the optimal data association. There exist data association algorithms, such as Probabilistic Data Association (PDA) for a single target in clutter and Joint Probabilistic Data Association (JPDA) or Multiple Hypothesis Tracker (MHT) for multiple targets in clutter, that solve the measurement-to-track association sub-optimally. In this setting, it is hard to propose a meaningful criterion to update the OOSMs optimally for multi-target tracking in clutter. In this chapter, we provide one solution by incorporating the PDA technique with the OOSM update for tracking a single target in clutter. Through performance comparison between the PDA with OOSM update and the in-sequence PDA filter, we find that the performance degradation of the PDA with the OOSM update is relatively small. We also find that the PDA with the OOSM update has better performance than just ignoring OOSMs. The generalization of incorporating the JPDA or MHT with the OOSM filter update for multi-sensor multi-target tracking in clutter is briefly discussed.

6.2 Problem Formulation

For multi-sensor multi-target tracking in the presence of clutter, during each sampling period, there is a set of measurements $\mathbf{z}_j$ arriving at the fusion center. Some of them are target originated, others are false measurements. The existing algorithms for tracking a target in the presence of clutter include Bayesian and non-Bayesian techniques [4]. The PDA and its extension JPDA, belong to Bayesian techniques. PDA and JPDA are target-oriented approach. For a known number of targets, PDA (JPDA) evaluates the measurement-to-target association probabilities and combines them into the corresponding state estimates. The MHT is a measurement-oriented approach. It finds the best hypothesis for a track to be associated with a measurement sequence.

Here we limit the discussion to a single target tracking in clutter and assume a measurement set $\mathbf{z}_d$ produced at previous time t_d arrived at the fusion center after the measurement set $\mathbf{z}_k$ produced at the most recent time t_k, where $t_k > t_d$. Then we can identify that the measurement set produced at t_d contains OOSMs. We will formulate the PDA incorporating OOSM update for a single target tracking. It is easy to analyze and the result can be generalized to other more complicated cases.

The set of validated measurements is denoted as

$$\mathbf{z}_j = \{z_j^i\}_{i=1}^{m_j}$$

where z_j^i is the i-th validated measurement and m_j is the number of measurements in the validated region at time t_j. In view of the assumptions listed, the association events

$$\theta_j^i = \begin{cases} \{z_j^i \text{ is the target originated measurement}\} & i = 1, \ldots, m_j \\ \{\text{None of the measurements is target originated}\} & i = 0 \end{cases}$$

are mutually exclusive and exhaustive for $m_j \geq 1$. The problem is as follows: an earlier set of measurements $\mathbf{z}_d = \{z_d^i\}_{i=1}^{m_d}$ at time t_d arrives after the state estimate $\hat{x}_{k|k}$ and the covariance $P_{k|k}$ have been calculated. Using the total probability theorem, the state estimate using $\mathbf{z}_d$ is

$$\begin{aligned}\hat{x}_{k|k,d} &= E(x_k|\mathbf{z}^k, \mathbf{z}_d) \\ &= \sum_{i=0}^{m_d} E(x_k|\theta_d^i, \mathbf{z}^k, \mathbf{z}_d) P(\theta_d^i|\mathbf{z}^k, \mathbf{z}_d) \\ &= \sum_{i=0}^{m_d} \hat{x}_{k|k,d}^i \beta_d^i\end{aligned}$$

where $\hat{x}_{k|k,d}^i$ for $i = 1, \ldots, m_d$ is the updated state conditioned on the event that the i-th validated measurement z_d^i is target originate and $\beta_d^i = P(\theta_d^i|\mathbf{z}^k, \mathbf{z}_d)$ is the conditional probability of the event — the association probability — and $\hat{x}_{k|k,d}^0 = \hat{x}_{k|k}$, $P_{k|k,d}^0 = P_{k|k}$. Also

$$\begin{aligned}P_{k|k,d} &= E\{[x_k - \hat{x}_{k|k,d}][x_k - \hat{x}_{k|k,d}]'|\mathbf{z}^k, \mathbf{z}_d\} \\ &= \sum_{i=0}^{m_d} E\{[x_k - \hat{x}_{k|k,d}][x_k - \hat{x}_{k|k,d}]'|\theta_d^i, \mathbf{z}^k, \mathbf{z}_d\}\beta_d^i \\ &= \bar{P}_{k|k,d} + \tilde{P}_d\end{aligned}$$

where $\bar{P}_{k|k,d} = \sum_{i=0}^{m_d} \beta_d^i P_{k|k,d}^i$, $\tilde{P}_d = \sum_{i=0}^{m_d} \beta_d^i \hat{x}_{k|k,d}^i (\hat{x}_{k|k,d}^i)' - \hat{x}_{k|k,d}(\hat{x}_{k|k,d})'$. Based on different OOSM update, $\hat{x}_{k|k,d}^i$, $P_{k|k,d}^i$ and β_d^i will have different forms.

6.3 OOSM Update in Clutter

6.3.1 OOSM Update: PDA with ALG-I

It follows from the globally optimal OOSM update Algorithm I of chapter 5, that

$$\hat{x}^i_{k|k,d} = \hat{x}_{k|k} + K_d(z^i_d - H_d\hat{x}_{d|k}) = \hat{x}_{k|k} + K_d\tilde{z}^i_d$$

$$P^i_{k|k,d} = P_{k|k} - K_dS_dK'_d$$

With $\tilde{z}_d = \sum_{i=1}^{m_d} \beta^i_d \tilde{z}^i_d$, we have

$$\hat{x}_{k|k,d} = \sum_{i=0}^{m_d} \hat{x}^i_{k|k,d}\beta^i_d = \hat{x}_{k|k} + K_d\tilde{z}_d$$

Let $P^c_{k|k,d} = P_{k|k} - K_dS_dK'_d$. Then

$$\bar{P}_{k|k,d} = \beta^0_d P_{k|k} + [1 - \beta^0_d]P^c_{k|k,d}$$

and

$$\tilde{P}_d = K_d \left[\sum_{i=0}^{m_d} \beta^i_d \tilde{z}^i_d (\tilde{z}^i_d)' - \tilde{z}_d\tilde{z}'_d \right] K'_d$$

The association probability can be derived in the same way as for the in-sequence PDA filter,

$$\beta^i_d = \begin{cases} \frac{e_i}{b + \sum_{j=1}^{m_d} e_j} & i = 1, \ldots, m_k \\ \frac{b}{b + \sum_{j=1}^{m_d} e_j} & i = 0 \end{cases} \tag{6.1}$$

where $e_i = e^{-\frac{1}{2}(\tilde{z}^i_d)'S_d^{-1}\tilde{z}^i_d}$ and $b = \lambda|2\pi S_d|^{1/2}\frac{1-P_DP_G}{P_D}$ with gate probability P_G and detection probability P_D.

6.3.2 OOSM Update: PDA with ALG-II

It follows from the proposed constrained optimal OOSM update Algorithm II, that

$$\hat{x}^i_{k|k,d} = E^*(x_k|\hat{x}_{k|k}, z^i_d) = \tilde{K} z^*_i$$

$$P^i_{k|k,d} = \text{MSE}(\hat{x}^i_{k|k,d}) = (\tilde{K}-\Gamma)\bar{R}(\tilde{K}-\Gamma)' + Q_{k,d} - \Gamma\bar{R}\Gamma'$$

where $E^*(x_k|\hat{x}_{k|k}, z^i_d)$ is the LMMSE without prior and $z^*_i = \begin{bmatrix} \hat{x}'_{k|k} & (z^i_d)' \end{bmatrix}'$. With $\tilde{z}_d = \sum_{i=1}^{m_d} \beta^i_d z^*_i = \begin{bmatrix} (1-\beta^0_d)\hat{x}'_{k|k} & \sum_{i=1}^{m_d} \beta^i_d (z^i_d)' \end{bmatrix}'$, we have

$$\hat{x}_{k|k,d} = \sum_{i=0}^{m_d} \hat{x}^i_{k|k,d}\beta^i_d = \beta^0_d \hat{x}_{k|k} + \tilde{K}\tilde{z}_d$$

Let $P^c_{k|k,d} = (\tilde{K}-\Gamma)\bar{R}(\tilde{K}-\Gamma)' + Q_{k,d} - \Gamma\bar{R}\Gamma'$. Then

$$\bar{P}_{k|k,d} = \beta^0_d P_{k|k} + [1-\beta^0_d]P^c_{k|k,d}$$

and

$$\tilde{P}_d = \tilde{K}[\sum_{i=0}^{m_d} \beta^i_d z^*_i (z^*_i)' - \tilde{z}_d(\tilde{z}_d)']\tilde{K}' + \beta^0_d[1-\beta^0_d]\hat{x}_{k|k}\hat{x}'_{k|k} - \beta^0_d \hat{x}_{k|k}(\tilde{z}_d)'\tilde{K}' - \beta^0_d \tilde{K}\tilde{z}_d\hat{x}'_{k|k}$$

The association probability has the same form as (6.1) with $e_i = e^{-\frac{1}{2}(\tilde{z}^i_d)' S_d^{-1}\tilde{z}^i_d}$ and $b = \lambda|2\pi S_d|^{1/2}\frac{1-P_D P_G}{P_D}$, where $\tilde{z}^i_d = z^i_d - H_d E^*(x_d|\hat{x}_{k|k})$ and $S_d = H_d P_{d|k} H'_d + R_d$. Therefore, the only task for the PDA to incorporate the constrained optimal OOSM update is to get the LMMSE $\hat{x}_{d|k} = E^*(x_d|\hat{x}_{k|k})$ and $P_{d|k}$ =MSE$[\hat{x}_{d|k}]$. Based on the limited information storage $\Omega_k = \{P_{k-s|k-s}, \ldots, P_{k|k}, \mathbf{z}(d)\}$, we can only do LMMSE estimation without prior.

Theorem 12: *The LMMSE $\hat{x}_{d|k} = E^*(x_d|\hat{x}_{k|k})$ and $P_{d|k}$ =MSE$[\hat{x}_{d|k}]$ without prior is*

$$\hat{x}_{d|k} = F^{-1}_{k,d}\hat{x}_{k|k} \tag{6.2}$$

$$P_{d|k} = F^{-1}_{k,d}\bar{R}_d F^{-1}_{k,d}{}' \tag{6.3}$$

where

$$\bar{R}_d = Q_{k,d} + F_{k,d}U'_{d,k} + U_{d,k}F'_{k,d} - P_{k|k}$$

□

Proof. According to

$$\hat{x}_i(d|k) = K_d\hat{x}_{k|k} + b$$

$$\bar{x}_d = K_dF_{k,d}\bar{x}_d + b$$

i.e.

$$(K_dF_{k,d} - I)\bar{x}_d + b = 0$$

Since the prior information is not known, this equation must be satisfied for every $\bar{x}_k$, and so

$$K_dF_{k,d} = I,\ b = 0$$

$$\begin{aligned}
&\mathrm{MSE}[\hat{x}_i(d|k)] = \mathrm{cov}(x_d - \hat{x}_i(d|k)) \\
&= \mathrm{cov}(x_d - K_d\hat{x}_{k|k}) = \mathrm{cov}[x_d - K_d(x_k - \tilde{x}_{k|k})] \\
&= \mathrm{cov}[x_d - K_dF_{k,d}x_d - K_dw_{k,d} + K_d\tilde{x}_{k|k})] \\
&= \mathrm{cov}[K_dw_{k,d} - K_d\tilde{x}_{k|k})] = K_d\mathrm{cov}[w_{k,d} - \tilde{x}_{k|k})]K'_d \\
&= K_d\bar{R}_dK'_d
\end{aligned}$$

where

$$\bar{R}_d = Q_{k,d} + F_{k,d}U'_{d,k} + U_{d,k}F'_{k,d} - P_{k|k}$$

Now the optimal $\tilde{K}_d$ is the solution of the following optimization problem:

$$\tilde{K}_d = \arg\min_K \text{MSE}[\hat{x}_i(d|k)] = \arg\min_K \{K_d \bar{R}_d K_d'\}$$

$$s.t. \quad \tilde{K}_d F_{k,d} = I$$

The general solution is given by:

$$\tilde{K}_d = F_{k,d}^{-1}$$

So

$$\hat{x}_{d|k} = F_{k,d}^{-1} \hat{x}_{k|k}$$

$$P_{d|k} = F_{k,d}^{-1} \bar{R}_d F_{k,d}^{-1\prime}$$

■

According to Theorem 12, $\tilde{z}_d^i = z_d^i - H_d F_{k,d}^{-1} \hat{x}_{k|k}$ and $S_d = H_d F_{k,d}^{-1} \bar{R}_d F_{k,d}^{-1\prime} H_d' + R_d$.

PDA with OOSM update is suggested for single target tracking in clutter. The OOSM update filter can only handle the state update problem. OOSM update in the presence of measurement origin uncertainty can not be done easily. This means we can not expect the PDA with OOSM update will achieve the same performance as the in-sequence PDA filter. If we want the updated PDA filter to have the same performance as the in-sequence PDA, we need also update the associated probability β_j^i with $j = k - l + 1, \ldots, k$. But β_d^i relies on the observation z_j^i. So in order to update β_j^i, all observations from t_{k-l+1} to t_k are needed. However, the information we have at the OOSMs arrival time is limited, such as $\Omega_k = \{\hat{x}_{k-s|k-s}, P_{k-s|k-s}, \ldots, \hat{x}_{k|k}, P_{k|k}, \mathbf{z}_d\}$ or $\Omega_k = \{P_{k-s|k-s}, \ldots, P_{k|k}, \mathbf{z}_d\}$. Even the PDA with globally optimal OOSM update will have difference in performance with the in-sequence PDA. But we can not be sure that the updated PDA filter will always perform poorer than

the in-sequence PDA since the PDA filter itself is not optimal. There are no fundamental optimal criteria for us to obtain the optimal OOSMs update within the PDA framework.

If the available information for update is $\Omega_k = \{\hat{x}_{k-s|k-s},\, P_{k-s|k-s},\, \ldots,\, \hat{x}_{k|k},\, P_{k|k},\, \mathbf{z}_k,\, \mathbf{z}_d\}$ or $\Omega_k = \{P_{k-s|k-s},\, \ldots,\, P_{k|k},\, \mathbf{z}_k,\, \mathbf{z}_d\}$, we can also update β_k^i with $i = 1, \ldots, m_i$ to obtain a more accurate estimate of x_k. The procedure includes update state $\hat{x}_{k-1|k-1}$ to $\hat{x}_{k-1|k-1,d}$, then applying the in-sequence PDA filter to yield $\hat{x}_{k|k,d}$ by recalculating β_k^i with $\mathbf{z}_k$. The performance will be the same as PDA with $l-1$ lag OOSM update if we treat the original PDA with OOSM update as an l-lag problem.

6.3.3 OOSM Update: Multiple Target Case

In the previous subsections, we have incorporated the PDA with Algorithms I and II for the OOSM update. For multi-target tracking in clutter, we need to consider both data association and OOSM update. For data association via JPDA or its variants e.g., nearest neighbor JPDA, incorporating the OOSM update using Algorithms I and II is straightforward. For each track, when OOSMs are received, the Algorithms I and II operate based on the marginal data associated probability obtained via JPDA and the filter updates are decoupled among different tracks once the marginal data association probabilities for the validated measurements are computed by evaluating the joint events using JPDA. For the multiple hypothesis tracker, the decision on measurement-to-track association is based on measurements from multiple frames and the hypothesis management is often based on a sliding window where the decision on which measurement goes to which track at the end of the sliding window is frozen due to limited storage and computational power. In this setting,

the measurements within the sliding window need to be stored to resolve the measurement origin uncertainty issue. Any OOSMs within the sliding window can be reordered and processed as an in-sequence measurement within the MHT framework. If the OOSM has a time stamp earlier than the frozen time of the sliding window, then Algorithms I and II can be applied to the state estimate at the frozen time for track oriented MHT. All these subtleties are data association issue rather than the OOSM update.

6.4 Simulations

6.4.1 Scenario

Several simple numerical examples are given in this section to verify the proposed algorithms. Consider a discretized continuous time kinematic system driven by white noise with power spectral density q, known as constant velocity model or white-noise acceleration model in target tracking, described by the following linear model

$$x_j = F_{j-1}x_{j-1} + w_{j-1}$$

$$z_j = H_j x_j + v_j$$

where $x_j = \left[x_j^{(1)}, x_j^{(2)}\right]$, w_j and v_j are zero mean white mutually uncorrelated Gaussian noise with

$$F_j = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \qquad H_j = [1, 0]$$

$$\mathrm{var}(w_j) = Q = \begin{bmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{bmatrix} q, \;\; \mathrm{var}(v_j) = R = 1$$

where T is the sampling interval. The prior information is

$$\hat{x}_{0|0} = \bar{x} = [200\text{km}, 0.5\text{km/ sec}]', \quad P_{0|0} = \begin{bmatrix} R & R/T \\ R/T & 2R/T^2 \end{bmatrix}$$

and the maneuver index is $\lambda = \sqrt{qT^3/R}$.

In order to consider multi-lag delay as well as single-lag OOSM update, we choose a series of OOSMs z_d. These OOSMs occurred at $d = (l+1)n$ and arrived at $(l+1)n + l$ with $n = 1, 2, \ldots$, corresponding to l-lag delayed OOSMs, where $l = 1, 2, \ldots$. For example, if the in-sequence observation series is $\{z_1, z_2, z_3, \ldots\}$, then the observation sequence with OOSMs for $l = 1$ is $\{z_1, z_3, z_2, z_5, z_4, \ldots\}$ and the updated states are $x_3, x_5, x_{7,} \ldots$; the observation series with OOSMs for $l = 2$ is $\{z_1, z_2, z_4, z_5, z_3, z_7, z_8, z_6, \ldots\}$ and the updated states are $x_5, x_8, x_{11,} \ldots$; and so on. Ideal estimates were obtained by the Kalman filter using all target originated observations only (including OOSMs) in the right time sequence.

For clutter generation, we use a poisson model with a spatial density λ to get the number of false measurements in the validation region:

$$\mu_F(m) = e^{-\lambda V} \frac{(\lambda V)}{m!}$$

By choosing $\lambda V \in [0, 1]$, we can simulate different clutter densities. We use $P_D = 1$. Residual based $\chi^2(99.9)$ test is used for testing tracking divergence. In simulation results, there is less than 10% track loss.

We show RMS errors over 1000 monte Carlo runs for the OOSM updated states at $(l+1)n + l$ with $n = 1, 2, \ldots$, where KF — in-sequence Kalman filter without clutter; IS-PDAF — in-sequence PDA filter; IG-PDAF — in-sequence PDA filter ignoring the OOSMs; UD-PDAF1 — PDA with globally optimal OOSM Update; UD-PDAF2 — PDA with constrained

optimal OOSM update.

6.4.2 Results for OOSMs with Good Accuracy

We design the OOSM model by choosing $\text{var}(v_j) = R/10$ at $j = (l+1)n$ with $n = 1, 2, \ldots$, which means the OOSMs are more accurate than the in-sequence measurements. In the following, we show the RMS errors for the 1-lag, 2-lag and 4-lag OOSM update problems at time $k = 29$ with $\lambda V \in \{0,\ 0.25,\ 0.5,\ 0.75,\ 1\}$ in **Fig.6.1**, **Fig.6.2** and **Fig.6.3** respectively.

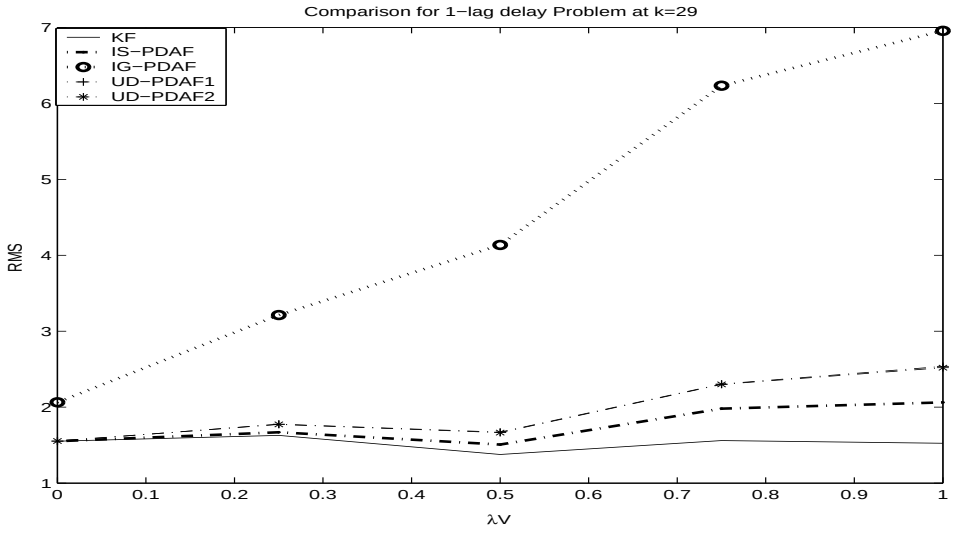


Fig.6.1 RMS errors with 1-lag OOSMs at time $k = 29$ with $\lambda V \in \{0,\ 0.25,\ 0.5,\ 0.75,\ 1\}$

As shown in these figures, when there is no clutter, i.e., $\lambda V = 0$, Algorithm I has the same performance as the KF, while Algorithm II has slightly poorer performance. For target tracking in clutter, the PDA with OOSM update by Algorithm I or II yields RMS errors close to the KF, which indicates that through OOSM update, the performance has significant improvement especially for small-lag OOSMs. For large-lag OOSMs, by ignoring them, the performance does not suffer much even if the OOSMs have much better accuracy. The RMS

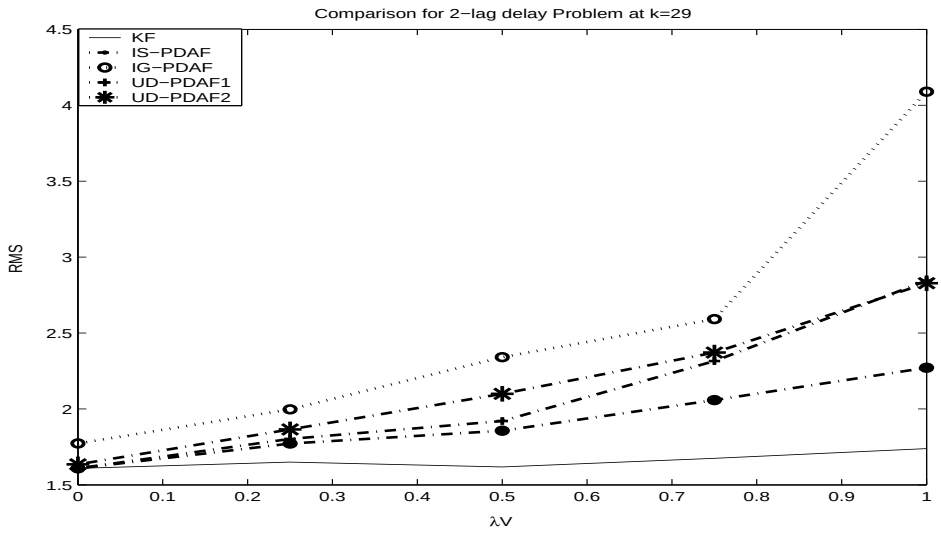


Fig.6.2 RMS errors with 2-lag OOSM at time $k = 29$ with $\lambda V \in \{0, 0.25, 0.5, 0.75, 1\}$

errors of UD-PDAF1 and UD-PDAF2 are very close to that of the in-sequence PDA filter. It also shows that the performance of IS-PDAF, IG-PDAF, UD-PDAF1 and UD-PDAF2 deteriorates as the clutter becomes heavier.

6.4.3 Results for OOSMs with Moderate Accuracy

The OOSMs have $C_{v_j} = R$, at $j = (l+1)n$ with $n = 1, 2, \ldots$, meaning that the in-sequence measurements have the same accuracy as the OOSMs. From **Fig.6.4**, we can clearly see that by ignoring the OOSMs, the performance still suffers a lot. But the PDA with OOSM update improves the performance, which is close to that of the in-sequence PDA filter.

6.5 Summary

In this Chapter, we proposed using PDA with Algorithm I and II for the OOSM update in the presence of clutter. Simulation results show that the PDA with the OOSMs update

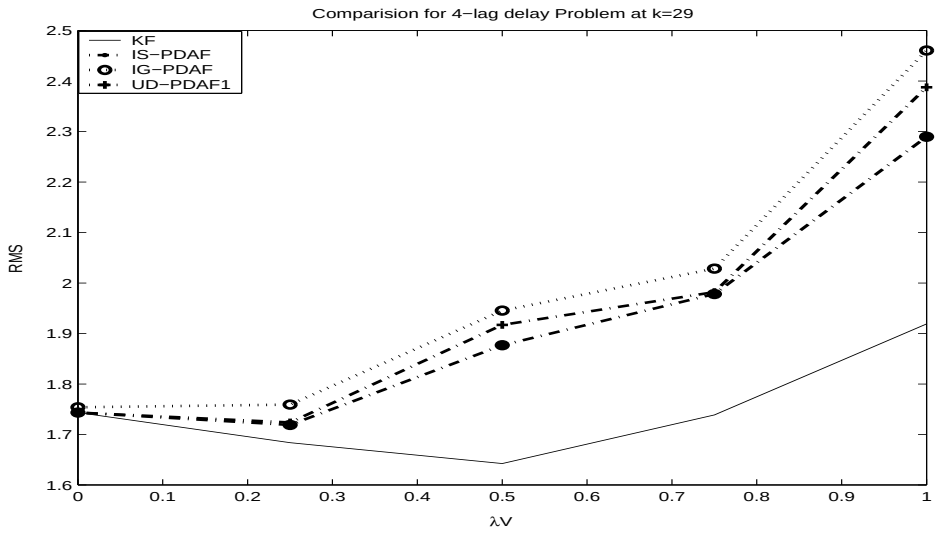


Fig.6.3 RMS errors with 4-lag OOSM at time $k = 29$ with $\lambda V \in \{0, 0.25, 0.5, 0.75, 1\}$

in clutter performs significantly better than ignoring the OOSMs, especially for small-lag OOSMs. Its performance is close to the in-sequence PDA update for OOSMs with various lags and under mild clutter where the PDA filter has less than 10% track loss. In summary, the PDA incorporating the two OOSM update algorithms has (1) an efficient processing structure; (2) an efficient memory structure; (3) an efficient computational structure. A brief discussion was given concerning how to incorporate the OOSM update algorithms with the JPDA or MHT for multi-target tracking in clutter.

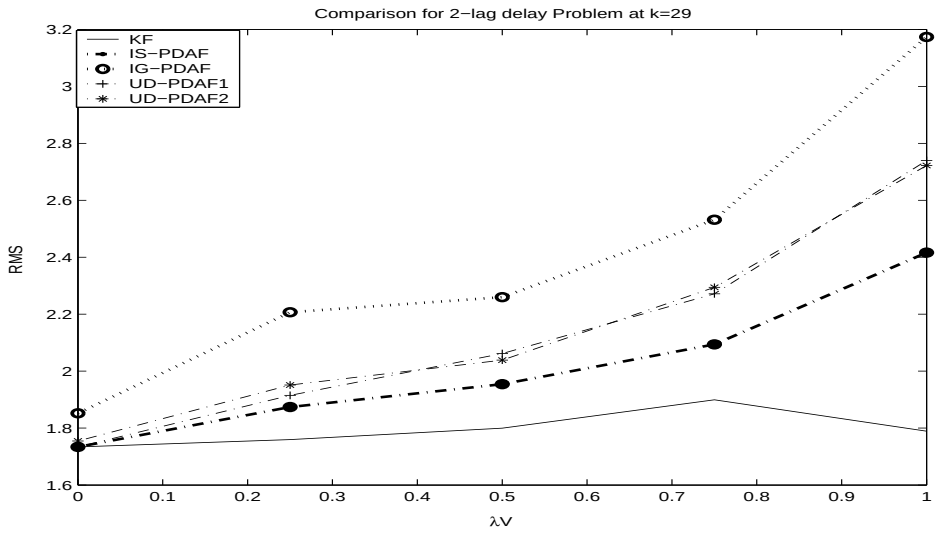


Fig.6.4 RMS errors with 2-lag OOSM at time $k = 29$ with $\lambda V \in \{0, 0.25, 0.5, 0.75, 1\}$

Chapter 7

Conclusions and Future Work

Distributed estimation fusion is useful for surveillance using sensor networks. The measurements of all the sensors among a sensor network are shared directly with the fusion center with constraints on communication, data processing capability at fusion center and information storage.

Due to the limited communication bandwidth and the limited processing capability at the fusion center, however, it is crucial to compress the raw measurements for the final estimation at the fusion center. One way of accomplishing this is to reduce the dimension of the data with minimum or no loss of information. Based on the BLUE fusion developed recently, in this dissertation, we have presented the optimal rules for compressing data at each local sensor to an allowable size (i.e., dimension) such that the fused estimate is optimal. We showed that without any performance deterioration, all sensor data can be compressed to a dimension not larger than that of the estimatee (i.e., the quantity to be estimated). For some simple cases, these optimal compression rules are given analytically; for the general case, they can be found numerically by Gaussian-Seidel iteration algorithm.

Due to the capacity constraints at the communication links, the measurements from the sensors are transmitted at a rate insufficient to convey all observations reliably. Therefore, the observations are vector quantized and the estimation is achieved using compressed measurements. In this dissertation, using the BLUE fusion, we developed optimal sensor quantization schemes which use only bivariate probability distributions of the state and sensor observations. For a dynamic system, it is shown that, under the communication constrains, the state update reduces to quantizing and estimating the current state conditioned on the past quantized measurements. A simple quantization and state estimation update structure for general dynamic system have been presented, and a new filter based on this structure for linear Gaussian system has been derived.

In multi-sensor target tracking systems, measurements from the same target can arrive out of sequence, called the out-of-sequence measurements (OOSMs). A problem is how to update the current state estimates with these "old" measurements. Due to the limited information storage, under BLUE fusion, we have presented two algorithms for updating with OOSM that are optimal for the information available at the time of update. Different minimum storages of information concerning the occurrence time of OOSMs were given for both algorithms. The update algorithms assume perfect target detection and no clutter in the received measurements. The real world has, however, possible missed target detection and random clutter in the possible OOSMs and thus the filter has to handle the measurement origin uncertainty. we have incorporated the probabilistic data association (PDA) into the two OOSM update algorithms with economic storage and efficient computation based on the nonsingularity assumption of some special matrices. The results shows that PDA with the two OOSM update algorithms have compatible RMS errors to the in-sequence PDA filter.

My future research work is to develop novel techniques for data compression in estimation fusion systems. I hope to address the dynamic quantization issue, which includes recursive update for the optimal quantization. A research in which the optimal sensor data compression rule and the optimal fusion rule can both be taken into consideration will thrive prominently.

Meanwhile, I am interested in distributed decision and classification. An intelligent optimal fusion system can be built in which the local processing and communication are taken into account in optimization. Also, this structure can have a very promising application in machine learning, data mining, and statistical inference areas.

Another interesting topic is to develop advanced information fusion frameworks with the regularization theory. Inverse problems of mathematical physics frequently lead to the ill-posed issues which are incorrectly formulated. Regularization theory is a general solution to incorrectly formulated problems. Recently, the regularization method has achieved great success in machine learning area (e.g. regularization networks and support vector machines). I feel confident that an advanced theoretical framework for information fusion with the regularization theory can dramatically improve the fusion performance, especially for those inaccurate model-based fusion frameworks.

In summary, I am interested in the research and technology area of information fusion, and hope my research work can assist in bringing this area to maturity, and in fostering specific applications.

Bibliography

[1] E. Ayanoglu. On optimal quantization of noisy sources. *Proc. IEEE Trans. Inform. Theory*, 36(6):1450–1452, Nov. 1990.

[2] Y. Bar-Shalom. Update with Out-of-Sequence Measurements in Tracking: Exact Solution. In *Proc. 2000 SPIE Conf. on Signal and Data Processing of Small Targets,* vol. 4048, Apr. 2000.

[3] Y. Bar-Shalom and X. R. Li. *Estimation and Tracking: Principles, Techniques, and Software*. Artech House, Boston, MA, 1993. (Reprinted by YBS Publishing, 1998).

[4] Y. Bar-Shalom and X. R. Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS Publishing, Storrs, CT, 1995.

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

[6] Y. Bar-Shalom, M. Mallick, H. Chen, and R. Washburn. One-Step Solution for the General Out-of-Sequence Measurements Problem in Tracking. In *Proc. 1st IEEE Conf. Aerospace*, Big Sky, MT, March 2002.

[7] R. E. Bellman. *Adaptive Control Precesses: A Guided Tour*. Princeton University Press, Princeton, NJ, 1961.

[8] W. R. Bennett. Spectra of quantized signals. *Bell Syst. Tech. J.*, 27:446C–472, July 1948.

[9] S. S. Blackman and R. F. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, Norwood, MA, 1999.

[10] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth & Brooks, Monterey, CA, 1984.

[11] L. Campo, Y. Bar-Shalom, and X. R. Li. Control of Discrete-Time Hybrid Stochastic Systems. In C. T. Leondes, editor, *Control and Dynamic Systems: Advances in Theory and Applications*, volume 76. Academic Press, New York, 1996.

[12] S. Challa, R. Evans, and J. Legg. A Fixed Lag Smoothing Framework for OOSI Problems. *Communications in Information and Systems*, 2(4):327–350, December 2002.

[13] S. Challa, R. Evans, and X. Wang. A Fixed Lag Smoothing Framework for Target Tracking in Clutter Using Out of Sequence Measurements. *Defense Applications of Signal Processing*, October 2001.

[14] S. Challa, J. E. Robin, and X. Wang. A Bayesian Solution and Its Approximations to Out-of-Sequence Measurement Problems. *Journal of Information Fusion*, 4(3):185–199, Sepetember 2003.

[15] S. Challa, X. Wang, and J. Legg. Track-to-Track Fusion using Out-of-Sequence Tracks. In *Proc. International Conf. on Information Fusion*, pages 919–926, Annapolis, MD, July 2002.

[16] K. C. Chang, S. Mori, and C. Y. Chong. Evaluating a Multiple-Hypothesis Multitarget Tracking Algorithm. *IEEE Trans. Aerospace and Electronic Systems*, 30(2):578–590, Apr. 1994.

[17] H. M. Chen, K.-S. Zhang, and X. R. Li. Optimal Data Compression for Multisensor Target Tracking with Communication Constraints. In *International Symposium on Information Processing in Sensor Networks 2004, submitted*, Oct. 2003.

[18] L. F. Chen, H. Y. M. Liao, M. T. Ko, J. C. Lin, and G. J. Yu. A New LDA-based Face Recognition System Which Can Solve the Small Sample Size Problem. *Pattern Recognition*, 33(10):1713–1726, 2000.

[19] R. E. Curry. *Estimation and Control with Quantized Measurements*. The M.I.T. Press, Cambridge, Massachusetts, London, England, 1970.

[20] K. I. Diamantaras and S. Y. Kung. *Principal Component Neural Networks*. Wiley, New York, 1996.

[21] Y. Ephraim and R. M. Gray. A Unified Approach for Encoding Clean and Noisy Sources by Means of Waveform and Autoregressive Model Vector Quantization. *Proc. IEEE Trans. Inform. Theory*, 34:826–834, July 1988.

[22] R. A. Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annual of Eugenics*, 7:179–188, 1936.

[23] T. J. Flynn and R. M. Gray. Encoding of correlated observations. *Proc. IEEE Trans. Inform. Theory*, (33):773–787, Nov. 1987.

[24] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, New York, 2nd edition, 1990.

[25] A. Gelb. *Applied Optimal Estimation.* The MIT Press, 1974.

[26] R. M. Gray and D. L. Neuhoff. Quantization. *Proc. IEEE Trans. Inform. Theory*, 6(44):2325–2383, 1998.

[27] J. A. Gubner. Constrained distributed estimation and Quantization for Distributed Estimation Systems. In *IEEE Int. Symp. Inform. Theory*, pages 32–33, San Diego, CA, Jan. 1990.

[28] R. D. Hilton, D. A. Martin, and W. D. Blair. Tracking with Time-Delayed Data in Multisensor Systems. Technical Report NSWCDD/TR-93/351, Naval Surface Warfare Center, Dalgren, VA, Aug. 1993.

[29] Z. Q. Hong and J. Y. Yang. Optimal Discriminant Plane for a Small Number of Samples and Design Method of Classifier on the Plane. *Pattern Recognition*, 24(4):317–324, 1991.

[30] A. K. Jain and B. Chandrasekaran. Dimensionality and Sample Size Considerations in Pattern Recognition Practice. In P.R. Krishnaiah and L.N. Kanal, editors, *Handbook of Statistics*, volume 2, pages 835–855. Amsterdam, North Holland, 1982.

[31] A. K. Jain, R. P. W. Duin, and J. C. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[32] I. T. Jolliffe. *Principal Component Analysis*. Springer Verlag, New York, 1986.

[33] W. M. Lam and A. R. Reibman. Design of Quantizers for Decentralized Estimation System. *Proc. IEEE Trans. Communications*, 41(11):1602–1605, Nov. 1993.

[34] B.-H. Li, Y.-M. Zhu, and X. R. Li. Fault-Tolerant Interval Estimation Fusion by Dempster-Shafer Theory. In *Proc. 2002 International Conf. on Information Fusion*, pages 1605–1613, Annapolis, MD, USA, July 2002.

[35] X. R. Li. The PDF of Nearest Neighbor Measurement and Probabilistic Nearest Neighbor Filter for Tracking in Clutter. In *Proc. 32nd IEEE Conf. on Decision and Control*, pages 918–923, San Antonio, TX, Dec. 1993.

[36] X. R. Li. Hybrid Estimation Techniques. In C. T. Leondes, editor, *Control and Dynamic Systems: Advances in Theory and Applications*, volume 76, pages 213–287. Academic Press, New York, 1996.

[37] X. R. Li. Data Association Probability and Measurement Density Function of Tracking with Strongest Neighbor Measurements. In *Proc. 1997 SPIE Conf. on Signal and Data Processing of Small Targets,* vol. 3163, San Diego, CA, July 1997.

[38] X. R. Li. *Probability, Random Signals, and Statistics*. CRC Press, Boca Raton, FL, 1999.

[39] X. R. Li. *Applied Estimation and Filtering*. Course Notes, University of New Orleans, 2002.

[40] X. R. Li. *Optimal Linear Estimation, Filtering, and Fusion.* Course Notes, University of New Orleans, 2002.

[41] X. R. Li and Y. Bar-Shalom. Tracking in Clutter with Nearest Neighbor Filters: Analysis and Performance. *IEEE Trans. Aerospace and Electronic Systems*, AES-32(3):995–1010, July 1996.

[42] X. R. Li and V. P. Jilkov. A Survey of Maneuvering Target Tracking: Dynamic Models. In *Proc. 2000 SPIE Conf. on Signal and Data Processing of Small Targets,* vol. 4048, pages 212–236, Orlando, Florida, USA, April 2000.

[43] X. R. Li and V. P. Jilkov. A Survey of Maneuvering Target Tracking—Part II: Ballistic Target Models. In *Proc. 2001 SPIE Conf. on Signal and Data Processing of Small Targets,* vol. 4473, pages 559–581, San Diego, CA, USA, 2001.

[44] X. R. Li and V. P. Jilkov. A Survey of Maneuvering Target Tracking—Part III: Measurement Models. In *Proc. 2001 SPIE Conf. on Signal and Data Processing of Small Targets,* vol. 4473, pages 423–446, San Diego, CA, USA, 2001.

[45] X. R. Li and V. P. Jilkov. A Survey of Maneuvering Target Tracking—Part IV: Decision-Based Methods. In *Proc. 2002 SPIE Conf. on Signal and Data Processing of Small Targets,* vol. 4728, Orlando, Florida, USA, April 2002.

[46] X. R. Li and N. Li. Integrated Real-Time Estimation of Clutter Density for Tracking. *IEEE Trans. Signal Processing*, 48(10):2797–2805, Oct. 2000.

[47] X. R. Li and J. Wang. Unified Optimal Linear Estimation Fusion—Part II: Discussions and Examples. In *Proc. 2000 International Conf. on Information Fusion*, pages MoC2.18–MoC2.25, Paris, France, July 2000.

[48] X. R. Li and K. S. Zhang. Optimal Linear Estimation Fusion—Part IV: Optimality and Efficiency of Distributed Fusion. In *Proc. 2001 International Conf. on Information Fusion*, pages WeB1.19–WeB1.26, Montreal, QC, Canada, Aug. 2001.

[49] X. R. Li, K.-S. Zhang, J. Zhao, and Y.-M. Zhu. Optimal Linear Estimation Fusion–Part V: Relationships. In *Proc. International Conf. on Information Fusion*, pages 497–504, Annapolis, MD, USA, July 2002.

[50] X. R. Li and P. Zhang. Optimal Linear Estimation Fusion—Part III: Cross-Correlation of Local Estimation Errors. In *Proc. 2001 International Conf. on Information Fusion*, pages WeB1.11–WeB1.18, Montreal, QC, Canada, Aug. 2001.

[51] X. R. Li, Z. Zhao, and V. P. Jilkov. Estimator's Credibility and Its Measures. In *Proc. IFAC 15th World Congress*, Barcelona, Spain, July 2002.

[52] X. R. Li and X. Zhi. Probabilistic Strongest Neighbor Filter for Tracking in Clutter. In *Proc. 1996 SPIE Conf. on Signal and Data Processing of Small Targets,* vol. 2759, pages 230–241, Orlando, FL, Apr. 1996.

[53] X. R. Li, Y. M. Zhu, and C. Z. Han. Optimal Linear Estimation Fusion-Part I: Unified Fusion Rules. *IEEE Trans. Information Theory*, 49(9):2192–2208, Sep. 2003.

[54] M. Liggins, C. Y. Chong, I. Kadar, M. G. Alford, V. Vannicola, and S. Thomopoulos. Distributed Fusion Architectures and Algorithms for Target Tracking. *Proceedings of the IEEE*, 85(1):95–107, Jan. 1997.

[55] K. Liu, Y. Cheng, and J. Yang. A Generalized Optimal Set of Discriminant Vectors. *Pattern Recognition*, 25(7):731–739, 1992.

[56] S. P. Lloyd. Least Squares Quantization in PCM. *Proc. IEEE Trans. Inform. Theory*, (28):129–136, Mar. 1982.

[57] M. Loog, R. P. W. Duin, and R. H. Umbach. Multiclass linear dimension reduction by weighted pairwise fisher criteria. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(7):762–766, 2001.

[58] Z. Q. Luo. Universal Decentralized Estimation in a Bandwidth Constrained Sensor Network. *Draft*, 2003.

[59] M. Mallick, S. Coraluppi, and C. Carthel. Advances in Asynchronous and Decentralized Estimation. In *Proc. IEEE Conf. Aerospace*, Big Sky, MT, March 2001.

[60] M. Mallick, SJ. Krant, and Y. Bar-Shalom. Multi-sensor Multi-target Tracking using Out-of-Sequence Measurements. In *Proc. International Conf. on Information Fusion*, pages 135–142, Annapolis, MD, July 2002.

[61] M. Mallick, K. S. Zhang, and X. R. Li. Comparative Analysis of Multiple-lag Out-of-sequence Measurement Filtering Algorithms. In *Proc. International SPIE Conf. on Signal and Data Processing of Small Targets*, Aug. 2003.

[62] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Muller. Fisher Discriminant Analysis with Kernels. In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, pages 41–48. IEEE, 1999.

[63] G. N. Nair and R. J. Evans. State Estimation via a Capacity-Limited Communication Channel. In *IEEE Int. Conf. Decision and Control*, pages 866–871, 1997.

[64] Girish N. Nair and Robin J. Evans. State Estimation under Bit-Rate Constraints. pages 251–256, 1998.

[65] B. M. Oliver, J. Pierce, and C. E. Shannon. The Philosophy of PCM. *Proc. IRE*, 36:1324C–1331, Nov. 1948.

[66] M. Orton and A. Marrs. A Bayesian Approach to Multi-target Tracking Data Fusion with Out-of Sequence Measurements. *Proc. IEEE* (submitted), 2001.

[67] P. F. Panter and W. Dite. Quantizing Distortion in Pulse-Count Modulation with Nonuniform Spacing of Levels. *Proc. IRE*, 39:44C–48, Jan. 1951.

[68] C. R. Rao. The Utilization of Multiple Measurements in Problems of Biological Classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10:159–203, 1948.

[69] S. J. Raudys and A. K. Jain. Small Sample Size Effefcts in Statistical Pattern Recognition: Recommendations for Practitioners. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):252–264, 1991.

[70] S. J. Raudys and V. Pikelis. On Dimensionality, Sample Size, Classification Error, and Complexity of Classification Algorithms in Pattern Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2:243–251, 1980.

[71] B. Scholkopf, A. Smola, and K. R Muller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Compution*, 10:1299–1319, 1998.

[72] C. E. Shannon. A Mathematical Theory of Communication. *Bell Syst. Tech. J.*, pages 379C–423, 623–C656, 1948.

[73] C. E. Shannon. Coding Theorems for a Discrete Source with a Fidelity Criterion,. *IRE Nat. Conv. Rec.*, pages 142–163, 1959.

[74] H. Steinhaus. Sur la Division Des Corp Materiels en Parties. *Bull. Acad. Polon. Sci.*, pages 801–C804, Jan. 1956.

[75] G. W. Stewart. *Introduction to Matrix Computations*. Academic Press, New York, 1973.

[76] S. Thomopoulos and L. Zhang. Distributed Filtering with Random Sampling and Delay. In *Proc. 27th IEEE Conf. Decision and Control*, Austin, TX, Dec. 1988.

[77] Q. Tian, M. Barbero, Z.-H. Gu, and S. H. Lee. Image classification by the Foley-Sammon transform. *Optical Engineering*, 25(7):834–840, 1986.

[78] W. S. Wong and R. W. Brockett. Systems with Finite Communication Bandwidth Constraints: State Estimation Problem. *IEEE Trans. on Automatic Control*, 42(9):1294–1299, 1997.

[79] K. S. Zhang, X. R. Li, and H. M. Chen. Multisensor Multitarget Tracking with Out-of-Sequence Measurements. In *Proc. International Conf. on Information Fusion*, pages 672–679, Caims, Queensland, Australia, July 2003.

[80] K.-S. Zhang, X. R. Li, P. Zhang, and H. F. Li. Optimal Linear Estimation Fusion—Part VI: Sensor Data Compression. In *Proc. International Conf. on Information Fusion*, pages 221–229, Caims, Queensland, Australia, July 2003.

[81] K. S. Zhang, X. R. Li, and Y. M. Zhu. Optimal Update with Out-of-Sequence Measurements for Distributed Filtering. In *Proc. International Conf. on Information Fusion*, pages 1519–1526, Annapolis, MD, USA, July 2002.

[82] Z. Zhang and T. Berger. Estimation via Compressed Data. *IEEE Trans. Info. Theory*, 34(2):198–211, 1989.

[83] Z. Zhang and T. Berger. Estimation via Compressed Information. *Proc. IEEE Trans. Inform. Theory*, 34(2):198–211, 1989.

[84] Z.-L. Zhao, X. R. Li, V. P. Jilkov, and Y.-M. Zhu. Optimal Linear Unbiased Filtering with Polar Measurements for Target Tracking. In *Proc. 2002 International Conf. on Information Fusion*, pages 1527–1534, Annapolis, MD, USA, July 2002.

[85] Y. M. Zhu. *Multisensor Decision and Estimation Fusion*. Kluwer Academic Publishers, 2002.

[86] Y. M. Zhu and X. R. Li. Best Linear Unbiased Estimation Fusion. In *Proc. 1999 International Conf. on Information Fusion*, Sunnyvale, CA, USA, July 1999.

[87] Y. M. Zhu and X. R. Li. Recursive Least Squares with Linear Constraints. In *Proc. 38th IEEE Conf. on Decision and Control*, pages 2414–2419, Phoenix, AZ, USA, Dec. 1999.

[88] Y. M. Zhu, Z. S. You, J. Zhao, K.-S. Zhang, and X. R. Li. The Optimality for the Distributed Kalman Filtering Fusion with Feedback. *Automatica*, 37:1489–1493, 2001.

[89] Y. M. Zhu, K. S. Zhang, and X. R. Li. An SPRT-Type Procedure with Finite Upper Bound on Stopping Time. *IEEE Trans. Information Theory* (submitted), 2002.

[90] Y. M. Zhu, J. Zhao, K. S. Zhang, X. R. Li, and Z. S. You. Performance Analysis for Feedback Track Fusion. In *Proc. the 3rd Chinese World Congress on Intelligent Control and Intelligent Automation*, Hefei, China, June 2000.

VITA

Born in Leshan, Sichuan, China in 1975, Keshu Zhang spent her first twenty-four years in Chengdu, the capital city of Sichuan province, P.R. China. She enrolled in Sichuan University in 1994, which is one of the best universities in China. In 1998, she earned a Bachelor of Science in Mathematics with a minor in Computer Science. After that, she entered her master program in Mathematics Department of Sichuan University, major in Probability and Statistics. She has been doing research on distributed decision, data fusion and stochastic signal processing ever since she pursued her Ph.D. degree in 1999. In 2002, she earned a Ph.D. degree in Probability and Statistics from Mathematics Department of Sichuan University.

Based on mutual agreement between two universities, since August of 2000, she also enrolled in the graduate program in Electrical Engineering Department at University of New Orleans, and worked as a research assistant. Her graduate study is concentrated on multi-sensor estimation, distributed decision, target tracking, data fusion and stochastic signal processing.