

DYNAMIC QUANTIZATION FOR TRACK FUSION UNDER  
COMMUNICATION CONSTRAINTS

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

GÖRKEM GÖK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF MASTER OF SCIENCE  
IN  
ELECTRICAL AND ELECTRONICS ENGINEERING

JUNE 2015



Approval of the thesis:

**DYNAMIC QUANTIZATION FOR TRACK FUSION UNDER  
COMMUNICATION CONSTRAINTS**

submitted by **GÖRKEM GÖK** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. M. Gülbin Dural Ünver  
Dean, Graduate School of **Natural and Applied Sciences** \_\_\_\_\_

Prof. Dr. Gönül Turhan Sayan  
Head of Department, **Electrical and Electronics Engineering** \_\_\_\_\_

Assoc. Prof. Dr. Umut Orguner  
Supervisor, **Electrical and Electronics Eng. Dept., METU** \_\_\_\_\_

**Examining Committee Members:**

Prof. Dr. Mübeccel Demirekler  
Electrical and Electronics Engineering Dept., METU \_\_\_\_\_

Assoc. Prof. Dr. Umut Orguner  
Electrical and Electronics Engineering Dept., METU \_\_\_\_\_

Assoc. Prof. Dr. Çağatay Candan  
Electrical and Electronics Engineering Dept., METU \_\_\_\_\_

Assoc. Prof. Dr. Emre Tuna  
Electrical and Electronics Engineering Dept., METU \_\_\_\_\_

Assoc. Prof. Dr. Özgül Salor-Durna  
Electrical and Electronics Engineering Dept., Gazi University \_\_\_\_\_

**Date:** \_\_\_\_\_

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last name : Görkem Gök

Signature :

## **ABSTRACT**

### **DYNAMIC QUANTIZATION FOR TRACK FUSION UNDER COMMUNICATION CONSTRAINTS**

Gök, Görkem

M.S., Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. Umut Orguner

June 2015, 69 pages

Quantization is one of the important problems for track fusion in defense systems. Due to the communication constraints, the track information has to be quantized while it is sent from local agents to the fusion center. In this study, a new quantization approach is proposed for track fusion in fusion systems under communication constraints. The quantization algorithm used in practice for track fusion is a static nearest neighbor approach which selects the closest vector and the covariance in a table to the current track information. The quantization algorithm proposed here involves posing the quantization problem in an optimization framework and solving it by also including the predicted future values of the track into the picture. Since the approach considers the inherent dynamic characteristics of the tracks, the resulting methodology is called as dynamic quantization. The simulation results show that the dynamic quantization is much more advantageous compared to static quantization even under very low bit rates.

Keywords: Track Fusion, Fusion System, Communication Constraint, Quantization

## ÖZ

### İLETİŞİM KISITLARI ALTINDA İZ BİRLEŞTİRME İÇİN DİNAMİK NİCEMLEME KULLANIMI

Gök, Görkem

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Doç. Dr. Umut Orguner

Haziran 2015, 69 sayfa

Savunma sistemlerinde iz birleştirme için nicemleme işlemi önemli problemlerden biridir. İletişim kısıtlarından dolayı iz bilgisi, yerel merkezlerden füzyon merkezine gönderilirken nicemlenmek zorundadır. Bu çalışmada iletişim kısıtları altındaki bir füzyon sistemindeki iz birleştirme için yeni bir nicemleme yöntemi önerilmektedir. Pratikte bu iş için kullanılan statik nicemleme yöntemi belli bir anda bir bilgi kaynağından iz bilgisini gönderirken sabit bir tablodan izin o andaki durum vektörü ve kovaryansına en yakın vektör ve kovaryansı seçmektedir. Önerilen yöntem ise iz bilgisi gönderme problemini izin olası gelecek değerlerini de öngören bir eniyileme problemi olarak modelleyip bu problemi çözmektedir. Elde edilen nicemleme yöntemi izlerin dinamik yapılarını da hesaba kattığından dinamik nicemleme olarak adlandırılmaktadır. Elde edilen benzetim sonuçları dinamik nicemlemenin statik nicemlemeye göre düşük bit sayılarında dahi çok daha avantajlı olduğunu göstermektedir.

Anahtar Kelimeler: İz Birleştirme, Füzyon Sistemi, İletişim Kısıtları, Nicemleme

*To my beloved and my family...*

## **ACKNOWLEDGEMENTS**

It is a pleasure for me to express my sincere gratitude to my thesis supervisor Assoc. Prof. Dr. Umut Orguner for his belief, patience, encouragement and guidance throughout the study.

I would also like to thank my company ASELSAN Inc. for supporting this thesis work and encouraging us to make scientific studies for the problems that we face off during the work.

I also thank Turkish Scientific and Technological Research Council (TÜBİTAK) for their financial support during my study.

I am also grateful to my family for their love, trust and support throughout my life.



## TABLE OF CONTENTS

<b>ABSTRACT .....</b>	<b>V</b>
<b>ÖZ.....</b>	<b>VI</b>
<b>ACKNOWLEDGEMENTS.....</b>	<b>VIII</b>
<b>TABLE OF CONTENTS.....</b>	<b>IX</b>
<b>LIST OF TABLES .....</b>	<b>XI</b>
<b>LIST OF FIGURES .....</b>	<b>XII</b>
<b>CHAPTERS</b>	
<b>1 INTRODUCTION.....</b>	<b>1</b>
1.1. LITERATURE SURVEY ON TRACK ASSOCIATION AND FUSION.....	6
1.2. THESIS CONTENT.....	9
1.3. PUBLICATION .....	11
<b>2 PROBLEM DEFINITION .....</b>	<b>13</b>
2.1. LOCAL CENTER USED IN PRACTICE.....	14
2.2. FUSION CENTER STRUCTURE USED IN PRACTICE .....	15
2.3. PROPOSED LOCAL AND FUSION CENTER STRUCTURES.....	17
2.4. A SIMPLE EXAMPLE .....	21
<b>3 MATHEMATICAL ANALYSIS.....</b>	<b>25</b>
3.1. THE CASE OF KNOWN QUANTIZED COVARIANCES.....	26
3.2. THE GENERAL CASE .....	30
3.3. IMPLEMENTATION ISSUE: SIZE OF THE QUANTIZATION TABLE.....	31
<b>4 SIMULATION RESULTS .....</b>	<b>33</b>
4.1. ARCHITECTURE 1 .....	33
4.1.1. Target Model.....	34

4.1.2. Quantization Parameters.....	34
4.1.3. Methods .....	35
4.1.4 Results .....	36
4.2. ARCHITECTURE 2.....	41
4.2.1. Target Model .....	41
4.2.2. Quantization Parameters.....	42
4.2.3. Methods .....	42
4.2.4. Results .....	43
4.3 ARCHITECTURE 3.....	45
4.3.1. Target Model .....	46
4.3.2. Quantization Parameters.....	47
4.3.3. Methods .....	47
4.3.4. Results .....	47
4.4 EFFECTS OF HORIZON LENGTH .....	48
4.5 COMPUTATION TIME.....	50
<b>5 CONCLUSION AND FUTURE WORK.....</b>	<b>51</b>
<b>REFERENCES .....</b>	<b>53</b>
<b>APPENDICES</b>	
<b>A. DERIVATIONS FOR THE TERMS <math>W</math>, <math>\Gamma A</math> AND <math>\Lambda</math>.....</b>	<b>59</b>
A.1 EXPRESSING THE COST IN MATRIX FORM .....	59
A.2 CALCULATION OF THE EXPECTED VALUES .....	63
A.3 FINAL FORM OF THE COST FUNCTION .....	68

## LIST OF TABLES

### TABLES

<b>Table 4.1:</b> Covariance quantization table .....	35
<b>Table 4.2:</b> RMS errors for Architecture 1 .....	40
<b>Table 4.3:</b> RMS errors for Architecture 2 .....	46
<b>Table 4.4:</b> RMS errors for Architecture 3 .....	47
<b>Table 4.5:</b> RMS errors of quantization methods for different time horizons .....	49
<b>Table 4.6:</b> Computation times for different time horizons .....	50

## LIST OF FIGURES

### FIGURES

<b>Figure 1.1:</b> A target tracking system block diagram .....	2
<b>Figure 1.2:</b> Fusion architectures [12] .....	4
<b>Figure 2.1:</b> The simple fusion architecture used in the chapter .....	13
<b>Figure 2.2:</b> Architecture of the proposed local center. ....	20
<b>Figure 2.3:</b> Unquantized states in the local center and quantized states in the fusion center obtained with static and dynamic quantization. The communicated quantized states between the local center and the fusion center are shown with dots for both static and dynamic quantization methods. ....	23
<b>Figure 4.1:</b> Information graph of architecture 1 .....	33
<b>Figure 4.2:</b> Position estimates in the fusion center with no quantization, static and dynamic quantization when $M = 16$ bits: (a) x-positions (b) y-positions .....	36
<b>Figure 4.3:</b> Position estimates in the fusion center with no quantization, static and dynamic quantization when $M = 16$ bits. ....	37
<b>Figure 4.4:</b> Velocity estimates in the fusion center with no quantization, static and dynamic quantization when $M = 16$ bits: (a) x-velocity (b) y-velocity .....	38
<b>Figure 4.5:</b> RMS errors in the fusion center with static and dynamic quantization when $M = 16$ : (a) RMS position errors (b) RMS velocity errors .....	39
<b>Figure 4.6:</b> RMS errors in the fusion center with static and dynamic quantization when $M = 32$ : (a) RMS error positions (b) RMS error velocities .....	39
<b>Figure 4.7:</b> RMS errors in the fusion center with static and dynamic quantization when $M = 64$ : (a) RMS error positions (b) RMS error velocities .....	40
<b>Figure 4.8:</b> Information graph of Architecture 2 .....	41
<b>Figure 4.9:</b> Position estimates in the fusion center with no quantization, static and dynamic quantization when $M = 16$ bits: (a) x-positions (b) y-positions .....	43

<b>Figure 4.10:</b> Position estimates in the fusion center with no quantization, static and dynamic quantization when $M = 16$ bits.....	44
<b>Figure 4.11:</b> Velocity estimates in the fusion center with no quantization, static and dynamic quantization when $M = 16$ bits: (a) x-velocity estimates (b) y-velocity estimates. ....	44
<b>Figure 4.12:</b> RMS errors in the fusion center with static and dynamic quantization when $M = 16$ bits: (a) RMS position errors (b) RMS velocity errors.....	45

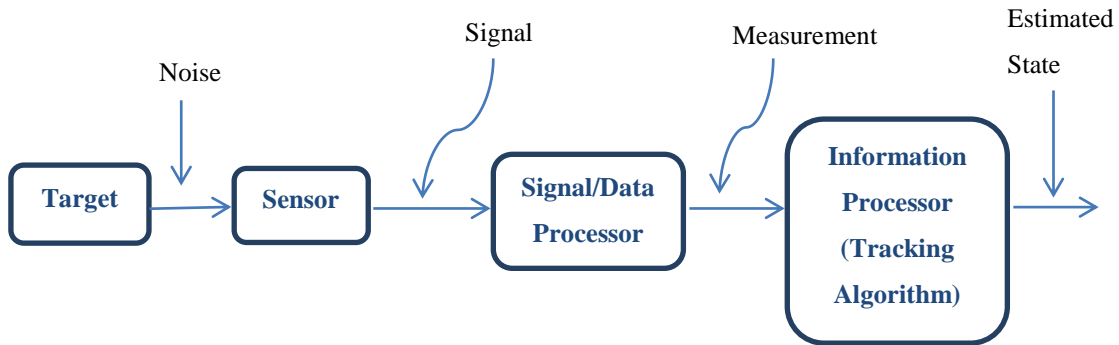


# CHAPTER 1

## INTRODUCTION

Target tracking is one of the important problems of the contemporary defense systems. In a simple sense, target tracking can be defined as extracting kinematic and/or attribute information from one or more than one measurements coming from varying number of sources by using association and state estimation algorithms [1]. The purpose of target tracking is to determine the kinematic quantities, i.e., position, velocity and other characteristics/attributes of the targets, such as their type. In order to collect information from targets, sensors are utilized. A sensor can be any measuring device which collects information from the objects existing in the environment. One of the most widespread sensors used for target tracking in both commercial and defense applications is the radar. A radar utilizes radio waves to measure the range, altitude, direction and speed of detected objects. Sonar, camera, ultrasound, infrared sensors can be given as other examples of the common sensors used in target tracking. Some typical target tracking examples are the radar tracking of aircraft, tracking a person or a vehicle in video surveillance systems, tracking weather balloons in weather monitoring and tracking of lymphocyte cells in cell biology [2].

Target tracking problem is essentially a state estimation problem. Tracking uses the tools of estimation and statistical decision theory [3]. A general target tracking system is shown in Figure 1.1. The figure is composed of a target or targets to be detected, a sensor for measurement, a signal processor and information processor for the state estimation and data association [2].



**Figure 1.1:** A target tracking system block diagram

The most widespread algorithm used for target tracking in the literature is Kalman filter (KF). Kalman filter is a well-known Bayesian algorithm and follows from general Bayesian recursive equation using Gaussian densities. Under linear and Gaussian assumptions, the posterior density of the state of linear system at any time becomes a Gaussian density which can be represented by a mean and a covariance matrix. Therefore, instead of propagating densities, Kalman filter propagates only the sufficient statistics – the mean and the covariance [2]. In literature, [4, 5] describes the Kalman filter for linear Gaussian systems. Because of its ubiquitous use and practical convenience, we will use Kalman filter in our analysis and experiments in this thesis.

Other state estimation algorithms commonly used in target tracking are Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF) and Particle Filter (PF). EKF and UKF are generally used in nonlinear non-Gaussian Bayesian state estimation problems where the nonlinearities are mild. While EKF utilizes a linearized model system [6], UKF uses unscented transform which propagates a number of points in the state-space to represent nonlinear transformation of Gaussian random variables [7, 8]. A particle filter which is a stochastic generalization of UKF can be used in general cases [9]. A general description of PF and relation with other algorithms can be found in [10].

In target tracking literature, a track is defined as the filtered information obtained from targets by using KF or other state estimation algorithms. A track is represented by the estimated state statistics which are the posterior mean and the

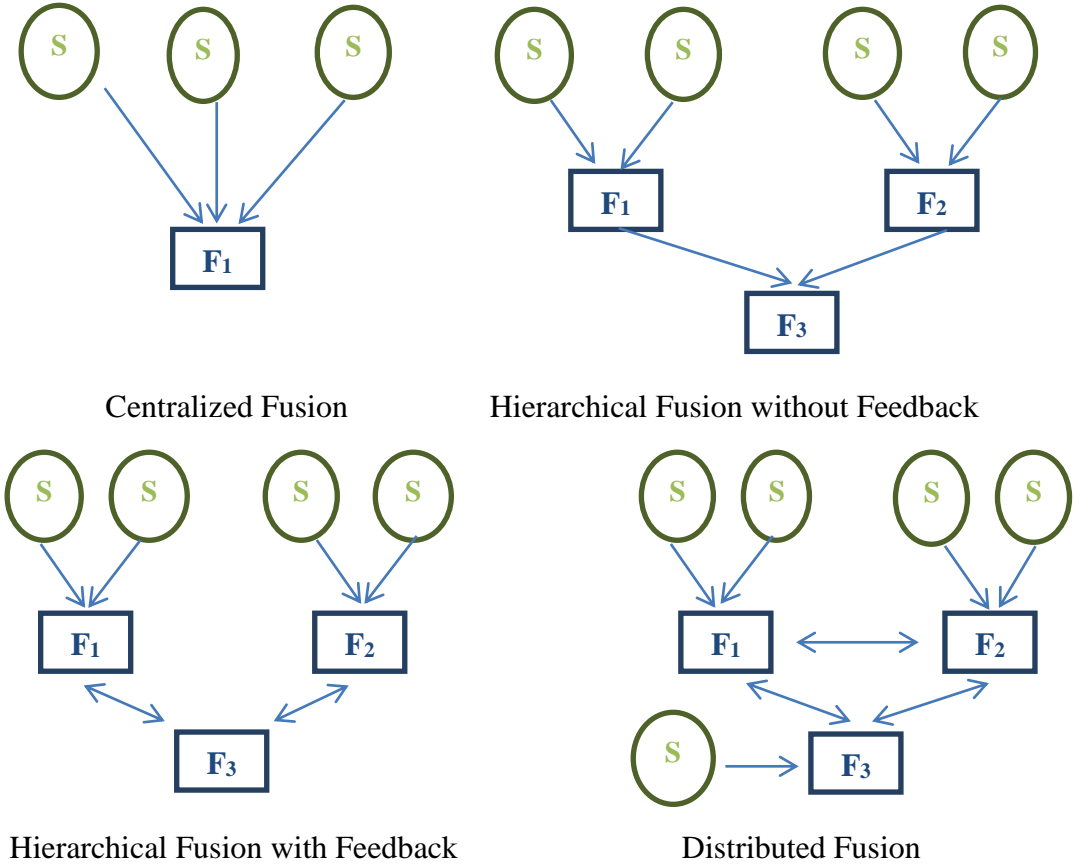


posterior covariance in a Kalman filter. Target tracking can be performed with only one sensor as well as with the help of more than one sensor. Hence, one of the main problems of target tracking is “Multiple Sensor Tracking” which is to obtain a track with information coming from multiple sensors.

There are five important problems of multiple sensor tracking as explained below:

- Registration: Coordinates of different sensors, local and fusion centers may not be aligned in time or space [11]. They must be aligned for tracking. In this thesis we assume that all coordinates of the sensors are already aligned in our experiments.
- Bias: Because of the transformations from/to different coordinates, biases may exist between different sensors. They have to be removed by using bias estimation tools. In this thesis, we assume that there is no bias problem in our system [11].
- Correlation: Though measurements might be conditionally independent across different sensors, due to the common process noise, processed information can be correlated. This effect can be calculated and compensated. In this thesis there exists and we handle correlation in our system [11].
- Rumor Propagation: Because of the correlation, the same information can travel in loops in a sensor network and can result in unrealistic error statistics making the fused system overconfident. Rumor Propagation is more common in systems with memory and feedback [11]. In the thesis, the fusion architectures we use does not involve loops.
- Out of Sequence Measurements: Measurements sometimes arrive to a fusion center later than more recent measurements because of delays in the communication channels and the recent measurement might have been processed already [11]. In this thesis, we assume that there is no communication delay in the fusion system we consider.

Combining information coming from the multiple sources is usually called as “fusion” [1] in the literature. A crucial point in multiple sensor tracking is the fusion architecture. There are mainly 4 types of fusion architectures as shown in Figure 1.2. While the circles show sensors, the squares show fusion or local centers.



**Figure 1.2:** Fusion architectures [12]

The first fusion architecture is the centralized fusion architecture. In centralized fusion, the information taken from the sensors are directly sent to and collected in the fusion center. This information is then processed in the fusion center by using tracking algorithms. The information sent from the sensors is composed of raw measurements. Since the local sensors do not process the raw measurements, sensors always have to send this information to the fusion center. In other words, a piece of information not sent to the fusion center disappears since the local centers related to each sensor do not obtain or keep any processed information. Although the

traditional centralized architecture has optimal performance, it has important drawbacks. It requires high bandwidth communication links between local agents and fusion center. Moreover, lags or saturations can happen in the communication links and these lags/saturations can cause Out-of-Sequence Measurements [13] or data disappearance.

In the hierarchical fusion architecture, there are local centers that take the raw measurements from sensors and process them by tracking algorithms. It is then this processed information which is sent to the higher node which is called as the fusion center. When the fusion center processes information, if the processed data are sent back to the local centers and used for reducing the estimation errors, the architecture is called as hierarchical fusion architecture with feedback. Hierarchical fusion architectures can work on limited communication bandwidths since local centers can keep processed information. In other words, thanks to the capability of the local centers to keep processed information, the communication rates between the local centers and the fusion center can be reduced without any information loss.

The last alternative to traditional centralized architecture is the distributed fusion architecture where there is no superior center in the network. All centers can send and receive processed information among each other. Like hierarchical fusion architectures, distributed fusion architectures can have reduced communication rates between the nodes.

Another advantage of both hierarchical and distributed architectures is that the computational loads can be distributed to different nodes as opposed to a centralized architecture where all of the computation is at the fusion center [12]. Furthermore, hierarchical and decentralized architectures have multiple centers which keep processed information. Therefore, if one or more of the centers are lost in a battle scenario, the rest of the network can still function with valuable (processed) information. On the other hand, centralized systems have a single weak point, which is the fusion center, and the loss of this weak point leads to the loss of all of the useful information.

In reality, the selection of a fusion architecture should be made by considering all of the factors described above. Most of the times, the optimal centralized

architecture has to be traded off for a hierarchical or distributed system due to communication constraints. Another case when a hierarchical or distributed approach has to be used is the case of legacy systems in which sensors might not be able to provide raw information to the fusion center, which might be the case, for example, with radars purchased from foreign countries. In this thesis we use hierarchical fusion architecture without feedback in our experiments.

There are two main tasks in multiple sensor information fusion. The first one is the track association which is the task of deciding whether two tracks coming from different local sensors belong to the same target [14]. The second task in multiple sensor tracking is the track fusion whose aim is to obtain a single fused track from the associated tracks coming from the local sensors [14]. Since the topic of track fusion is the main subject of this thesis, we give a detailed literature survey about it below.

### **1.1. Literature Survey on Track Association and Fusion**

In multisensor target tracking, an important problem is to decide whether two tracks coming from different sensors represent the same target or not. This is called as the track association problem. There are several methods in the literature to overcome the track association problem. One method proposed by [15] is a test for track association for two local agents. When track association is for more than two local agents, multi-dimensional assignment problem can be utilized. Moreover, there is another test in literature given in [16] which decides whether two tracks from different systems represent the same target by comparing the latest estimates of two tracks.

After it is known that two tracks are from the same target, the next question is to combine the tracks, which is named as the track fusion problem [17]. There has been a large amount of research conducted about track fusion since 1970s. Many exact and approximate solutions to the track fusion problem have been proposed and

different evaluation techniques have been applied to these solutions to find the optimum approach.

An optimal fusion solution is proposed by [18], [19] and [20] under arbitrary communication patterns. This solution is optimal provided that the system is deterministic, i.e., no process noise. However, in practice, there is crosscorrelation between local agents due to the common process noise of the target. In 1986, the study [21] is conducted to show the effect of the common process noise on the track fusion for two  $\alpha - \beta$  filters where a fusion formula was proposed. Then, [17] shows that this formula is optimal only in a maximum likelihood sense.

A kinematic state vector fusion algorithm is proposed in [22] for dissimilar sensors by considering the effects of cross-correlation. This cross-correlation has been obtained in terms of the steady-state covariances of Kalman filters. Also, when the cross-correlation is positive, it is observed that the performance of track fusion is increased. [23] has also proposed an efficient algorithm for track fusion by considering the effects of cross-covariance and analyzed the structure of cross-covariance matrix.

[24] and [25] propose a unified linear model as a general optimal solution for centralized, distributed and hybrid fusion architectures. This model is optimal under the presence of cross-correlation for arbitrary number of sensors. For this purpose, the fusion techniques of the best linear unbiased estimation (BLUE) and optimal weighted least-squares (WLS) are presented. Moreover, the formulae are given in [26-29] to obtain cross correlations, filter gains and observation matrices of local agents under linearity assumptions. [30] proposes a different method for obtaining the exact fusion algorithm by calculating the cross-covariances for sequential fusion. Different sequential fusion structures such as track fusion with no feedback, T2TF (Track-to-Track Fusion) with partial feedback and T2TF with full feedback are examined. Furthermore, an approximate implementation is proposed for systems with low communication capacity.

Another good reference is [31] which proposes an optimal solution to the distributed track fusion problem for maneuvering targets when communication rates are irregular. The disadvantage of cross-correlation between local tracks are handled

with the decorrelation of all local tracks. The track decorrelation and optimal fusion are achieved provided that sensor properties are known in each center. Other similar studies were performed by W. Koch in [32, 33]. In addition, [34] and [35] generalize these solutions for the multi-sensor case.

In [36], the analysis of track fusion problem is performed and a methodology is described to compare the performance of several fusion algorithms with information matrix fusion. Also, this study analyzes the hierarchical fusion architecture and concludes that the partial feedback case has better performance than complete and no feedback cases. With the approach given in [36], performance evaluation of track fusion algorithms become much simpler. Also, it is concluded that the quality of the information to be fused is much more important than the quantity.

Another performance evaluation of track fusion is performed for various numbers of sensors in [37]. This study generalizes the result given in [14] to the cases with more than 2 sensors. In this study, exactly the same kinematic model is used for centralized and distributed trackers and an exact algorithm is applied to them by using  $\alpha - \beta$  filters. The study shows that the performance of the distributed architecture gets worse with respect to the centralized architecture as the number of sensors increases. A similar work has also been realized by [38] for only two sensors.

In the light of these references given, we can now classify the track fusion methods. The first proposed method to track fusion is naïve fusion. In this method, common process noise and common priors are neglected and conditional independence is assumed, i.e. no process noise and previous communication. Since this method does not take the correlation between the estimation errors into account, it usually gives very bad results and it is extremely suboptimal.

The second method is Channel Filter (Information Matrix Fusion). This assumes that common prior is dominant and common process noise is negligible. This method is a little bit more complicated than Naïve Fusion. Moreover, this method is almost optimal if process noise is not significant [12].

Maximum A Posteriori Fusion (MAP) is the third method to track fusion. It considers the effect of both common prior and common process noise. Therefore,

MAP fusion is the most accurate and the most complicated solution to the track fusion problem [12]. However, this is not a practical method due to its complexity. The last method is Covariance Intersection (CI) Fusion. It is a general fusion technique that makes no assumption about the independence of estimation errors. In CI Fusion, a convex combination of the local mean and covariance estimates is calculated to avoid overconfidence in the fused estimates [39].

According to [40], channel filter seems to be the most suitable algorithm for track fusion among all other methods in terms of scalability, estimation errors and memory. As a result, we use the Channel Filter method (Information Matrix Fusion) to solve track fusion problem in this thesis work.

## **1.2. Thesis Content**

One of the crucial problems encountered in track fusion is the track quantization problem. Due to the communication constraints between the nodes, the track information has to be quantized while it is sent from local agents to the fusion center. In many applications, this information has to be squeezed into only a few bits. Therefore, the quantization mechanism to be used in track fusion plays an important role in performance of a multisensory target tracking algorithm.

In a local center where Kalman filter is used for state estimation, since the information is kept and sent as a state vector and its covariance, the track quantization has to be applied to these processed state and covariance matrix to be sent to the fusion center. Then, these quantized processed track information is sent to the fusion center from each local agent and fused there in order to obtain a single track. There are only few studies in the literature about the quantization problem for track fusion. Different representations of a covariance matrix for track quantization are analyzed and compared in [41].

The widespread quantization technique used in practice for track fusion is a static nearest neighbor approach which selects the closest state vector and the covariance to the current track information from fixed tables at the local center.

Afterwards, the selected state vector and the corresponding covariance matrix are sent to the fusion center for track fusion. The selected quantized values from the tables depend only on the current state vector and the covariance at the local agent. Therefore, the nearest neighbor approach can be named as *static quantization*. On the other hand, since the Kalman filter operating at the fusion center has a dynamic nature, it is easily understandable that performing quantization by only looking at the current state vector and the covariance (at the local center) is highly myopic. By benefiting from the fact that Kalman filter operating at fusion center is essentially a low-pass filter, a better quantization approach can be developed by using the predicted values of tracks as well in the quantization at the local center.

In this thesis study, based on the idea given above, a new track quantization algorithm is proposed. The proposed algorithm poses the quantization problem as an optimization problem and then solves it by also including the predicted local track information into the picture. The optimization problem is defined over a certain time horizon which takes the low-pass filter nature of the Kalman filter into account. Since this approach considers the inherent dynamic characteristics of the tracks, the resulting methodology is named as *dynamic quantization*. As understood from the proposed approach above, the main purpose of this study is improving quantization performance for track fusion under communication constraints even under very low bit rates by proposing a more advantageous quantization technique named as dynamic quantization.

The outline of this thesis study is as follows. General information about target tracking, fusion architectures, algorithms and track quantization has been given in this chapter along with a detailed literature survey on track fusion. In Chapter 2, the track quantization and the proposed optimization problem will be presented on a simple architecture. Moreover, Chapter 2 will give a brief description of the simple static track quantization method applied in practice. The solution of the optimization problem and the proposed track quantization method are given in Chapter 3 which utilizes the solution of a similar optimization problem used in a different context in the literature. The results obtained by the proposed method are examined and compared to those of the static quantization approach in Chapter 4 using



simulations. Chapter 5 summarizes the results obtained from this study and finalizes the thesis by mentioning about possible future work.

### **1.3. Publication**

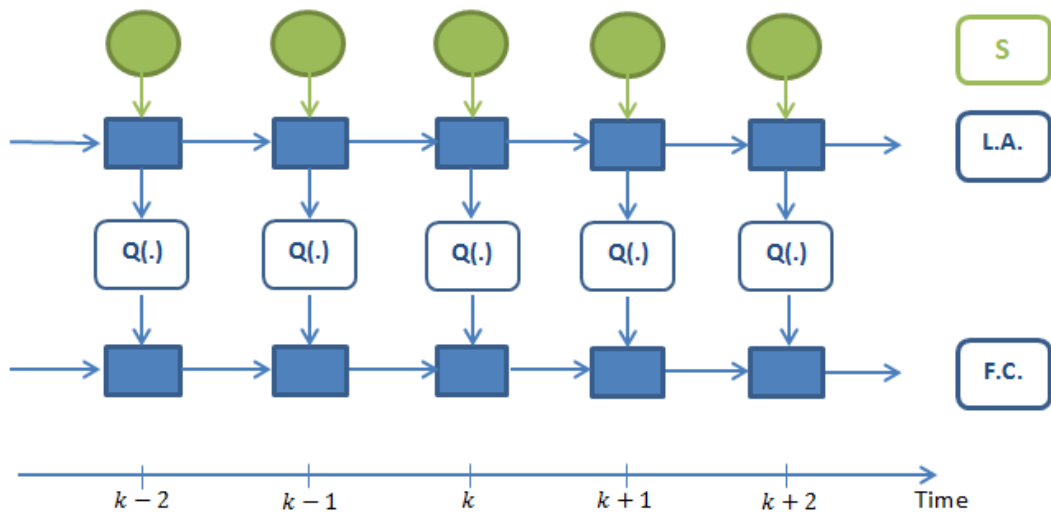
Within the scope of this thesis the conference article [1] was written and presented at the 23<sup>rd</sup> Signal Processing and Communications Applications Conference (SIU'15) on May 18<sup>th</sup>, 2015 in Malatya.



## CHAPTER 2

### PROBLEM DEFINITION

The simple fusion architecture we are working with in this chapter is shown in Figure 2.1. In the fusion system, there are a local center (local agent) and a fusion center which are shown by the blue rectangles for each time  $k$ . The green circles in the figure denote the sensor measurements. The arrows represent data flow and the block labeled as  $Q(\cdot)$  represents the quantization operation. As shown in the Figure 2.1 a single sensor is connected to the local center and the information to the fusion center comes only from the local center after a quantization operation.



**Figure 2.1:** The simple fusion architecture used in the chapter

## 2.1. Local Center Used in Practice

In the local center, the state vector of the target to be estimated is denoted as  $x_k$ . The measurements coming from the sensors belonging to the target are denoted as  $y_k^1$ . The state vector  $x_k \in \mathbb{R}^n$  and the measurement  $y_k^1 \in \mathbb{R}^m$  are modeled with the linear Gaussian state-space representation given below.

$$x_k = Ax_{k-1} + Bw_k, \quad (2.1)$$

$$y_k^1 = Cx_k + v_k, \quad (2.2)$$

where  $A$ ,  $B$  and  $C$  show the state transition matrix, process noise gain matrix and measurement matrix respectively. The quantity  $w_k$  denotes the white process noise of the target which has Gaussian distribution with zero-mean and covariance  $Q$ . The term  $v_k$  is the white measurement noise of the sensor of the local agent which is independent of the process noise and has Gaussian distribution with zero-mean and covariance  $R$ . The initial state  $x_0$  has also a Gaussian distribution with zero-mean and covariance  $P_0^1$ .

By using the model shown in (2.1)-(2.2), the local agent performs state estimation and covariance calculation by running its own Kalman filter. The prediction and the measurement update equations of the Kalman Filter that local agent operates are given in the following.

Prediction Update:

$$\hat{x}_{k|k-1}^1 = A\hat{x}_{k-1|k-1}^1 \quad (2.3)$$

$$P_{k|k-1}^1 = AP_{k-1|k-1}^1A + BQB^T \quad (2.4)$$

Measurement Update:

$$\hat{x}_{k|k}^1 = \hat{x}_{k|k-1}^1 + K_k^1(y_k^1 - \hat{y}_{k|k-1}^1) \quad (2.5)$$

$$P_{k|k}^1 = P_{k|k-1}^1 - K_k^1S_{k|k-1}^1(K_k^1)^T \quad (2.6)$$

where

$$\hat{y}_{k|k-1}^1 = C\hat{x}_{k|k-1}^1 \quad (2.7)$$

$$S_{k|k-1}^1 = CP_{k|k-1}^1C^T + R_1 \quad (2.8)$$

$$K_k^1 = P_{k|k-1}^1C^T(S_{k|k-1}^1)^{-1} \quad (2.9)$$

In (2.5)-(2.6),  $\hat{x}_{k|k}^1$  and  $P_{k|k}^1$  are the posterior state estimate and its covariance obtained by the Kalman filter in the local agent. The quantities  $S_{k|k-1}^1$  and  $K_k^1$  are the measurement/innovation covariance and the Kalman gain respectively. After the local agent obtains state and covariance estimates from its own Kalman Filter, it quantizes the corresponding information and sends the quantized information to the fusion center.

In the fusion systems used in practice, *static quantization* is used for quantizing posterior state estimate and covariance to obtain the data sent from the local agent to the fusion center. The static quantization used in practice, which is denoted as  $Q_s(\cdot)$ , represents a nearest-neighbor selection of the quantized state vector and covariance from predetermined fixed tables. We will show these tables with the symbol ‘‘T’’. The data sent from the local center to the fusion center then becomes the quantized versions of  $\hat{x}_{k|k}^1$  and  $P_{k|k}^1$  given as;

$$\hat{x}_{k|k}^{1 \rightarrow 2}, P_{k|k}^{1 \rightarrow 2} = Q_s(\hat{x}_{k|k}^1, P_{k|k}^1), \quad (2.10)$$

where  $\hat{x}_{k|k}^{1 \rightarrow 2}$  and  $P_{k|k}^{1 \rightarrow 2}$  are the quantized posterior state and its covariance respectively.

## 2.2. Fusion Center Structure Used in Practice

Since the information coming from the local center is correlated along time (since the communicated data is the estimates and covariances of a Kalman filter), the fusion center used in practice applies the so-called Channel Filter to obtain its own estimates  $\hat{x}_{k|k}^2$  and  $P_{k|k}^2$  from  $\hat{x}_{k|k}^{1 \rightarrow 2}$  and  $P_{k|k}^{1 \rightarrow 2}$ . The corresponding Channel Filter equations are given as follows.

$$\begin{aligned} (P_{k|k}^2)^{-1} \hat{x}_{k|k}^2 &= (P_{k|k-1}^2)^{-1} \hat{x}_{k|k-1}^2 + (P_{k|k}^{1 \rightarrow 2})^{-1} \hat{x}_{k|k}^{1 \rightarrow 2} - (P_{k|k-1}^{1 \rightarrow 2})^{-1} \hat{x}_{k|k-1}^{1 \rightarrow 2} \\ (P_{k|k}^2)^{-1} &= (P_{k|k-1}^2)^{-1} + (P_{k|k}^{1 \rightarrow 2})^{-1} - (P_{k|k-1}^{1 \rightarrow 2})^{-1} \end{aligned} \quad (2.11)$$

$$(2.12)$$

The equations given above are simply Kalman filter equations in the so-called information form. The explanations for the terms in (2.11)-(2.12) are given below.

- $\hat{x}_{k|k-1}^2$  and  $P_{k|k-1}^2$  are the predicted state estimate and covariance in the fusion center;
- $\hat{x}_{k|k}^2$  and  $P_{k|k}^2$  are the updated state estimate and covariance in the fusion center;
- $\hat{x}_{k|k-1}^{1\rightarrow 2}$  and  $P_{k|k-1}^{1\rightarrow 2}$  are the predicted versions of the previously communicated information  $\hat{x}_{k-1|k-1}^{1\rightarrow 2}$  and  $P_{k-1|k-1}^{1\rightarrow 2}$  between the local center and the fusion center.

In the expressions (2.11)-(2.12) it is seen that the predicted information in the fusion center  $\hat{x}_{k|k-1}^2$ ,  $P_{k|k-1}^2$  is first combined/summed with the new information  $\hat{x}_{k|k}^{1\rightarrow 2}$ ,  $P_{k|k}^{1\rightarrow 2}$  coming from the local center. Since the new information  $\hat{x}_{k|k}^{1\rightarrow 2}$ ,  $P_{k|k}^{1\rightarrow 2}$  is correlated with the previously communicated information  $\hat{x}_{k-1|k-1}^{1\rightarrow 2}$ ,  $P_{k-1|k-1}^{1\rightarrow 2}$  (which are implicit in the predicted information  $\hat{x}_{k|k-1}^2$ ,  $P_{k|k-1}^2$  of the fusion center) the predicted versions  $\hat{x}_{k|k-1}^{1\rightarrow 2}$ ,  $P_{k|k-1}^{1\rightarrow 2}$  of the previously communicated information  $\hat{x}_{k-1|k-1}^{1\rightarrow 2}$ ,  $P_{k-1|k-1}^{1\rightarrow 2}$  are subtracted from the combined information. In a general fusion architecture, the predicted information  $\hat{x}_{k|k-1}^2$ ,  $P_{k|k-1}^2$  in the fusion center is distinct from the predicted previously communicated information  $\hat{x}_{k|k-1}^{1\rightarrow 2}$ ,  $P_{k|k-1}^{1\rightarrow 2}$ . However, in the simple architecture we consider in Figure 2.1, since the fusion center receives information only from the local center, we have

$$\hat{x}_{k|k-1}^2 = \hat{x}_{k|k-1}^{1\rightarrow 2}, \quad P_{k|k-1}^2 = P_{k|k-1}^{1\rightarrow 2}. \quad (2.13)$$

Hence in the fusion architecture we consider, we get

$$\hat{x}_{k|k}^2 = \hat{x}_{k|k}^{1\rightarrow 2}, \quad P_{k|k}^2 = P_{k|k}^{1\rightarrow 2}. \quad (2.14)$$

That is, the quantized posterior quantities sent from the local agent at each time are the posterior quantities of the fusion center at the corresponding instant. Hence we

see that the estimates obtained in the fusion center used in practice are directly equal to the quantized information coming from the local center.

### 2.3. Proposed Local and Fusion Center Structures

In the proposed fusion center structure there also exists a quantization operation between the local agent and the fusion center. However, unlike the static quantization used in practice, the quantization in the proposed system is performed not only by considering the current values of posterior state estimate and covariance, but also looking at the expected predicted future values of these quantities. The proposed quantization method is called as *dynamic quantization* and shown as  $Q_d(\cdot)$ .

$$\hat{x}_{k|k}^{1 \rightarrow 2}, P_{k|k}^{1 \rightarrow 2} = Q_d(\hat{x}_{k|k}^1, P_{k|k}^1) \quad (2.15)$$

Since the low-pass nature of the Kalman filter will be used in proposed quantization method, the following Kalman filter equations are used in the fusion center.

$$\hat{x}_{k|k}^2 = \hat{x}_{k|k-1}^2 + K_k^2 (\hat{x}_{k|k}^{1 \rightarrow 2} - \hat{x}_{k|k-1}^2) \quad (2.16)$$

$$P_{k|k}^2 = P_{k|k-1}^2 - P_{k|k-1}^2 (S_{k|k-1}^2)^{-1} P_{k|k-1}^2 \quad (2.17)$$

$$S_{k|k-1}^2 = P_{k|k-1}^2 + P_{k|k}^{1 \rightarrow 2} \quad (2.18)$$

$$K_k^2 = P_{k|k-1}^2 (S_{k|k-1}^2)^{-1} \quad (2.19)$$

where  $\hat{x}_{k|k}^2$  and  $P_{k|k}^2$  are the posterior state estimate and covariance of the fusion center respectively. The predicted state estimate and covariance of the fusion center are shown by  $\hat{x}_{k|k-1}^2$  and  $P_{k|k-1}^2$  respectively. The Kalman filter equations above utilize the quantized posterior state  $\hat{x}_{k|k}^{1 \rightarrow 2}$  and the covariance  $P_{k|k}^{1 \rightarrow 2}$  sent from the local agent as the measurement and measurement noise covariance respectively. The quantities  $S_{k|k-1}^2$  and  $K_k^2$  are the measurement/innovation covariance and the Kalman gain respectively in the fusion center.

It is assumed in this study that the Kalman filter structure applied in the fusion center given above is known by the local center. Therefore, the local center can select the quantized information  $\hat{x}_{k|k}^{1 \rightarrow 2}, P_{k|k}^{1 \rightarrow 2}$  sent to the fusion center in a clever

manner to make the fusion center estimate and covariance  $\hat{x}_{k|k}^2, P_{k|k}^2$  as close to the unquantized local center estimate and covariance  $\hat{x}_{k|k}^1, P_{k|k}^1$  as possible. Note that such an approach would solve the issue of correlation between the consecutive communicated estimates automatically.

In this study, we propose that the quantized quantities  $\hat{x}_{k|k}^{1 \rightarrow 2}, P_{k|k}^{1 \rightarrow 2}$  sent to the fusion center should be selected to be the values which solve the following optimization problem.

$$\{\hat{x}_{l|l}^{1 \rightarrow 2}, P_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1} = \arg \min_{\{\hat{x}_{l|l}^{1 \rightarrow 2}, P_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1} \in \mathbb{T}} J(\{\hat{x}_{l|l}^{1 \rightarrow 2}, P_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1}) \quad (2.20)$$

where the cost function  $J(\cdot)$  is defined as

$$J(\{\hat{x}_{l|l}^{1 \rightarrow 2}, P_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1}) = \sum_{l=k}^{k+N-1} E \left\{ \|\hat{x}_{l|l}^2 - \hat{x}_{l|l}^1\|_2^2 + \lambda \|P_{l|l}^2 - P_{l|l}^1\|_F^2 \mid y_{0:k}^1 \right\} \quad (2.21)$$

In (2.21) the notations  $\|\cdot\|_2$  and  $\|\cdot\|_F$  represent  $L_2$  and Frobenius norms respectively, which are defined as

$$\|v\|_2 = \sqrt{\sum_{i=1}^n |v_i|^2}$$

$$\|V\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |v_{ij}|^2} = \sqrt{\text{trace}(V^T V)}$$

where the notation  $v_i$  denotes the  $i$ th element of a vector  $v$  and  $v_{ij}$  denotes the  $ij$ th element of a matrix  $V$ .  $V^T$  denotes the transpose of the matrix  $V$ .

In the optimization problem (2.20), the function  $J(\cdot)$  defined in (2.21) is to be minimized with respect to the optimization variables  $\{\hat{x}_{l|l}^{1 \rightarrow 2}, P_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1}$  which are to be selected from the quantization tables denoted by  $\mathbb{T}$ . The function  $J(\cdot)$  depends on the optimization variables  $\{\hat{x}_{l|l}^{1 \rightarrow 2}, P_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1}$  through the fusion center state estimates and covariances  $\{\hat{x}_{l|l}^2, P_{l|l}^2\}_{l=k}^{k+N-1}$  via the Kalman filter equations in (2.16)-(2.19). The term  $N$  in the summation operator denotes the time horizon in the



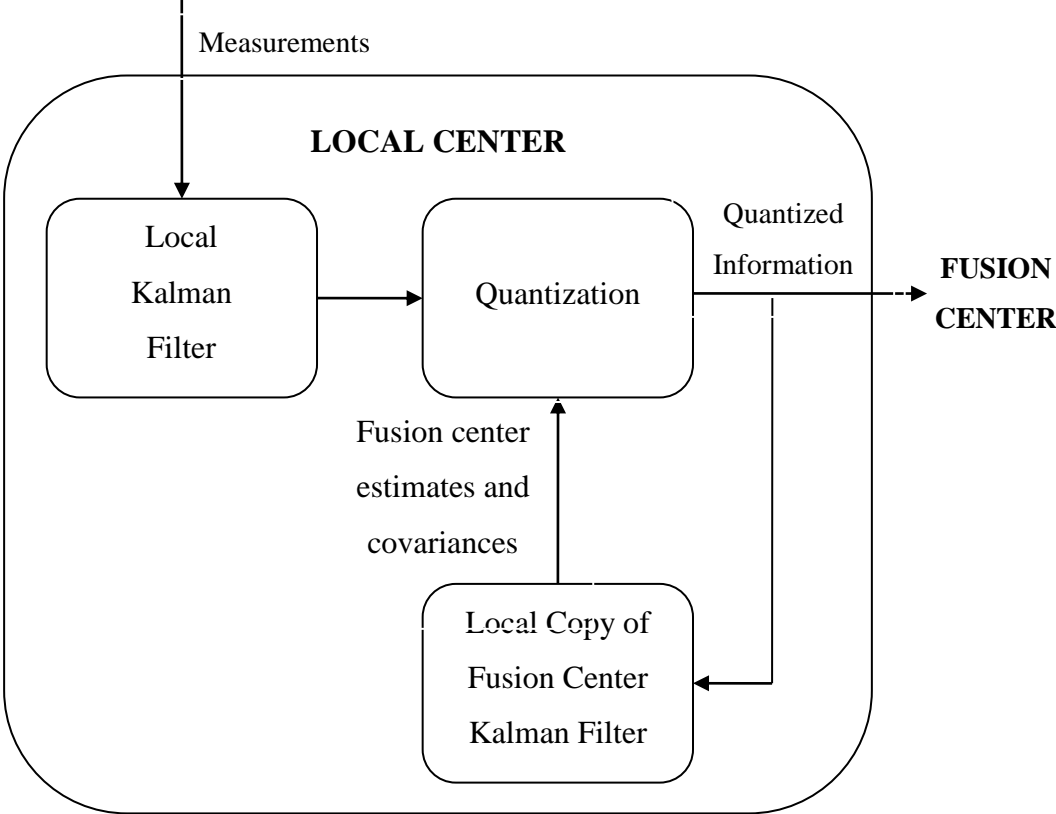
optimization problem over which the squared  $L_2$ -differences of the state estimates  $\hat{x}_{k|k}^1$  and  $\hat{x}_{k|k}^2$  and squared Frobenius-differences of the covariances  $P_{k|k}^1$  and  $P_{k|k}^2$  are summed in the cost function  $J(\cdot)$ . The term  $\lambda$  is a scaling constant determining the weights of the covariances in the cost function. The operator  $E\{\cdot\}$  denotes the statistical expectation and it is used to eliminate the unknown stochastic measurements  $y_{k+1}, y_{k+2}, \dots, y_{k+N-1}$  (unknown at time  $k$ ) which appears in the cost function (in the estimates  $\{\hat{x}_{l|l}^1\}_{l=k+1}^{k+N-1}$ ).

In the optimization problem, the quantized states and covariances  $\{\hat{x}_{l|l}^{1 \rightarrow 2}, P_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1}$  are obtained for  $N$  time instants, namely,  $k, k+1, \dots, k+N-1$ . Afterwards, only the quantized results corresponding to time  $k$ , i.e.,  $\hat{x}_{k|k}^{1 \rightarrow 2}, P_{k|k}^{1 \rightarrow 2}$ , are sent to the fusion center. The operation is repeated for future time instants by shifting the optimization horizon. This type of optimization/control topology is called as *receding horizon control* in the literature [42].

Receding horizon control method has been used in different applications in literature. According to [42], many design problems in control and signal processing such as power conversion problems, audio quantization problems, design of FIR filters with quantized coefficients and equalization of band-limited communication channels etc. can be expressed as optimization problems where decision variables can only take several finite values. By restricting the optimization to a finite horizon and solving the problem in a receding horizon fashion, excellent optimization results can be achieved even with very small horizons. In this study, we will apply the receding horizon control technique explained in [42] to the quantization problem posed in (2.20).

The quantization approach we propose above requires the state estimate of the fusion center state estimates and covariances in order to solve the optimization problem (2.20) at the local center. Since these quantities are the products of the Kalman filter at the fusion center and since it is assumed that the parameters of this Kalman filter is known at the local center, these quantities can be calculated at the local center by running a local copy of the fusion center Kalman filter and feeding it

with the transmitted quantized state and covariance. Therefore, at the local center, two Kalman filters are executed. The first Kalman filter is the original Kalman filter of the local center processing the measurements coming from the sensor. The second Kalman filter is the local copy of the fusion center Kalman filter which is fed by the quantized states and covariances sent to the fusion center. The proposed architecture of the local center is shown in Figure 2.2.



**Figure 2.2:** Architecture of the proposed local center.

## 2.4. A Simple Example

In order to make the proposed quantization/optimization problem more understandable, in this section, we illustrate the main idea of our study on a very simple deterministic example. Consider that we have a local agent and a fusion center in this problem as shown in Figure 2.1. The state evolution in the local agent is deterministic and is given as

$$x_k^1 = 0.95x_{k-1}^1 \quad (2.22)$$

where  $x_k^1$  is a scalar and the initial state is given as  $x_0 = 1$ . The state evolution in (2.22) represents the Kalman filter of the local agent in our original problem. We consider the fusion center in two different parts.

The first fusion center represents the fusion center used in practice and it uses the results of static quantization. We consider only two bits in the quantization table  $T$  of the static quantization. Hence, the information sent to the first fusion center is given as

$$x_k^{1 \rightarrow 2} = Q_s(x_k^1) = \begin{cases} 1, & x_k^1 \geq 0.5 \\ 0, & x_k^1 < 0.5 \end{cases} \quad (2.23)$$

which represents as nearest neighbor type quantization. As in the original problem, the state at the first fusion center is directly equated to the quantized data received from the local center, i.e.,

$$x_k^2 = x_k^{1 \rightarrow 2}. \quad (2.24)$$

The second fusion center represents the fusion center we propose in this study and it uses the dynamic quantization. It is assumed that the second fusion center uses the following state evolution.

$$x_k^2 = 0.9x_{k-1}^2 + 0.1x_k^{1 \rightarrow 2} \quad (2.25)$$

where the quantity  $\hat{x}_k^{1 \rightarrow 2}$  denotes the received data from the local center quantized by dynamic quantization. The state evolution (2.25) represents the Kalman filter

running in the fusion center in the original problem which uses the quantized data as its input. Note that the filter given in (2.25) has a low-pass characteristics similar to a Kalman filter. The optimization problem for the quantization is given as

$$x_k^{1 \rightarrow 2} = \arg \min_{x_k^{1 \rightarrow 2} \in \{0,1\}} (x_k^2 - x_k^1)^2 \quad (2.26)$$

where we selected the time horizon  $N = 1$ . We can write the cost function in (2.26) as follows.

$$J = (x_k^2 - x_k^1)^2 \quad (2.27)$$

$$= (0.9x_{k-1}^2 + 0.1x_k^{1 \rightarrow 2} - x_k^1)^2 \quad (2.28)$$

$$= (0.9x_{k-1}^2 - x_k^1 + 0.1x_k^{1 \rightarrow 2})^2 \quad (2.29)$$

$$= 0.01 \left( x_k^{1 \rightarrow 2} - \frac{x_k^1 - 0.9x_{k-1}^2}{0.1} \right)^2. \quad (2.30)$$

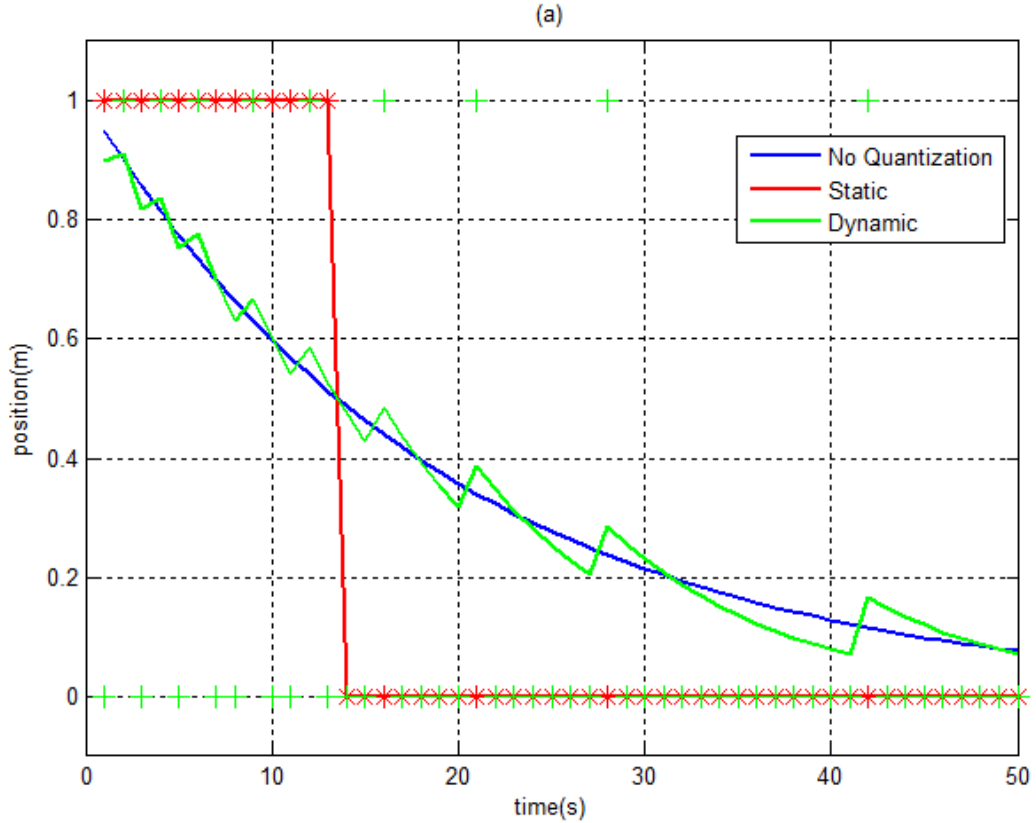
With the form of the cost function in (2.30) it can be seen that the dynamic quantization rule is given as

$$x_k^{1 \rightarrow 2} = Q_d(x_k^1) = \begin{cases} 1, & \frac{x_k^1 - 0.9x_{k-1}^2}{0.1} \geq 0.5 \\ 0, & \frac{x_k^1 - 0.9x_{k-1}^2}{0.1} < 0.5 \end{cases}. \quad (2.31)$$

Note that in the local center, the quantity  $x_k^1$  is known exactly. Similarly, since the local center knows both the state evolution in (2.25) and the quantized data  $\{x_l^{1 \rightarrow 2}\}_{l=0}^{k-1}$  sent to the fusion center, it can calculate the fusion center state  $x_{k-1}^2$ . Therefore, the local center can easily calculate the quantized data  $x_k^{1 \rightarrow 2}$  in (2.31).

There is also an issue we should pay attention to in this example in order to obtain reasonable results from (2.31). The scalar state transition coefficient of the state evolution in the second fusion center must be selected smaller than that of the state evolution in the local center. Hence, the bandwidth of the low-pass filter in the fusion center must be larger than that of the process in the local center. In this example, we have selected the state transition coefficient of the fusion center as 0.9 which is smaller than 0.95.

In order to observe and compare the results of the quantization methods, a simulation is performed for the simple example described above. Figure 2.3 shows the unquantized states  $x_k^1$  (blue) in the local center along with the fusion center states  $x_k^2$  obtained with static (red) and dynamic (green) quantization.



**Figure 2.3:** Unquantized states in the local center and quantized states in the fusion center obtained with static and dynamic quantization. The communicated quantized states between the local center and the fusion center are shown with dots for both static and dynamic quantization methods.

As shown in Figure 2.3, the static quantization (red curve) selects ‘1’ when  $x_k^1$  values in the local agent are higher than or equal to ‘0.5’ and ‘0’ otherwise, as expected. These values then become the states  $x_k^{1 \rightarrow 2}$  in the fusion center which gets the data with static quantization. On the other hand, the fusion center with dynamic quantization (green curve) can get much closer states to the unquantized states. It can do this by selecting occasional ‘0’ values when the unquantized state value is

closer to '1' and occasional '1' values when the unquantized state value is closer to '0'. The low-pass nature of the fusion center smooths out the jumps considerably.

Now we describe the similarities and the differences between the simple example and our original fusion scenario.

- In the simple example, the state process was deterministic. In the original fusion scenario, the state process (i.e., Kalman filter) is stochastic in that it is driven with stochastic measurements of the state.
- In the simple example, the fusion center runs a low-pass filter with the quantized data as its input. The filter run in the fusion center was known by the local center. In the original scenario, the fusion center will run a Kalman filter with low-pass characteristics and this filter will be known by the local center.
- In the simple example, the local center can calculate the state at the fusion center using the information of the low-pass filter (running in the fusion center) and the previously sent quantized data to the fusion center. In the original scenario, the local center will be able to calculate the state (of the Kalman filter) at the fusion center using the information of the Kalman filter (running in the fusion center) and the previously sent quantized data.

## CHAPTER 3

### MATHEMATICAL ANALYSIS

As previously mentioned, there are a local agent and a fusion center in our fusion system (see Figure 2.1). The local agent has its own measurements, runs its own Kalman Filter with these measurements and it has its own estimates. After the local agent obtains the state estimates and their covariances from its own Kalman filter, it quantizes the information and sends these estimates to the fusion center. At the same time, the fusion center operates its own Kalman filter, as well. It uses the quantized track information, coming from the local agent, as measurements; i.e., the fusion center utilizes the quantized state and covariance as measurement and measurement noise covariance respectively. Since our approach considers the quantization problem in an optimization framework and solves the problem by including the predicted future values of the states and covariances, it is named as *dynamic quantization*.

In this chapter, we will make the mathematical analysis of the optimization problem (2.20) and present a solution to it. When the quantized posterior covariance values  $\{P_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1}$  sent from the local center are known in the cost function (2.21), the cost function becomes quadratic in the unknown quantized state values  $\{\hat{x}_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1}$ . For this special case, there is a solution proposed in literature to the optimization problem (2.20) which can be used to find the optimal quantized posterior states  $\{\hat{x}_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1}$ . Hence we will separate this chapter into two main sections. In Section 3.1, we will study the (hypothetical) case in which the quantized

covariance  $\{P_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1}$  are known and find out how quantized posterior states  $\{\hat{x}_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1}$  should be selected. Afterwards we will study the general case in which the quantized covariance values  $\{P_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1}$  are also unknown in Section 3.2. By using the method proposed in Section 3.1, a simple solution will also be suggested for this general and realistic case.

In order to be able to solve the optimization problem we are going to use a theorem taken from the literature. For being able to use the theorem, we are going to bring the cost function (2.21) into a form suitable for the application of the result of the theorem. The cost function will be first separated into two parts, namely state and covariance parts. Then the expected values will be taken and the random terms will be eliminated. The minimum cost and corresponding quantized states and covariances will then be calculated.

### 3.1. The Case of Known Quantized Covariances

In this section, it is assumed that the quantized covariances  $\{P_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1}$  in the cost function (2.21) are already known. In this case, it is going to be shown that the cost function will be quadratic in the unknown quantized states  $\{\hat{x}_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1}$ . The following theorem taken from [42] describes how a quadratic function can be minimized on a finite number of points by utilizing a nearest-neighbor quantization rule.

**Theorem 1 [42]:** Consider the following quadratic function  $J(u)$ .

$$J(u) = u^T W u + 2u^T F x. \quad (3.1)$$

where  $u, x \in \mathbb{R}^n$  and  $W$  and  $F$  are real valued matrices with dimension  $m \times m$ .  $W$  is assumed to be symmetric and positive definite. The aim is to minimize the cost function  $J(\cdot)$  on  $U \triangleq \{u_1, u_2, \dots, u_K\}$  which is a finite set defined as the quantization table. Hence the solution of the following optimization problem is desired.



$$\hat{u} = \arg \min_{u \in U} J(u). \quad (3.2)$$

The solution to this optimization problem can be obtained by a nearest-neighbor quantization rule as below.

$$\hat{u} = W^{-1/2} Q_{NN}(-W^{-1/2} Fx) \quad (3.3)$$

where the function  $Q_{NN}(\cdot)$  represents the nearest-neighbor quantization rule which is defined as follows.

$$Q_{NN}(t) \triangleq \arg \min_{v \in V} \|t - v\|_2 \quad (3.4)$$

where the set  $V \triangleq \{v_1, v_2, \dots, v_K\}$  has been defined as

$$v_i \triangleq W^{1/2} u_i, \quad i = 1, \dots, K. \quad (3.5)$$

The notation  $W^{1/2}$  denotes the positive-definite square-root of the matrix  $W$ . The value of  $Q_{NN}(t)$  in (3.4) is the nearest vector to the vector  $t$  in the set  $V$  with respect to  $L_2$  norm.

In the theorem given above, the optimal unquantized vector  $u$  which minimizes the cost function  $J(\cdot)$  in  $\mathbb{R}^n$  is given as

$$\tilde{u} = -W^{-1} Fx. \quad (3.6)$$

The nearest-neighbor vector given in (3.3) on the set  $U$  is usually different from the optimal unconstrained solution  $\tilde{u}$  in (3.6).

In order to be able to apply Theorem 1 to our quantization problem, we need to have the cost function in (2.21) in the form (3.1). When the quantized covariance values sent from the local center,  $\{P_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1}$ , are known, the cost function in (2.21) can be written as shown below after the expected values are taken.

$$J(\Gamma | \{P_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1}) = \Gamma^T W \Gamma + 2\Gamma_A^T \Gamma + \Lambda \quad (3.7)$$

where the augmented vector  $\Gamma \in \mathbb{R}^{nN}$  includes all quantized state vectors over the time horizon and is defined as

$$\Gamma \triangleq \left[ (\hat{x}_{k|k}^{1 \rightarrow 2})^T (\hat{x}_{k+1|k+1}^{1 \rightarrow 2})^T \cdots (\hat{x}_{k+N-1|k+N-1}^{1 \rightarrow 2})^T \right]^T. \quad (3.8)$$

The matrix  $W$ , the vector  $\Gamma_A$  and the scalar  $\Lambda$  are some complicated functions of the variables  $\{P_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1}$  and  $\hat{x}_{k|k}^1, P_{k|k}^1$  as given below.

$$W \triangleq W \left( \{P_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1} \right) \quad (3.9)$$

$$\Gamma_A \triangleq \Gamma_A \left( \{P_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1}, \hat{x}_{k|k}^1, P_{k|k}^1 \right) \quad (3.10)$$

$$\Lambda \triangleq \Lambda \left( \{P_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1}, \hat{x}_{k|k}^1, P_{k|k}^1 \right) \quad (3.11)$$

The more detailed expressions for  $W$ ,  $\Gamma_A$  and  $\Lambda$  are given below.

$$W \triangleq \sum_{l=0}^{N-1} (\bar{B}_l^2)^T \Omega \bar{B}_l^2 \quad (3.12)$$

$$\Gamma_A \triangleq \left( \sum_{l=0}^{N-1} (\xi_{k-1}^T (A_{k+l:k})^T + \bar{X}^T (\bar{B}_l^1)^T) \Omega \bar{B}_l^2 \right)^T \quad (3.13)$$

$$\Lambda \triangleq \sum_{l=0}^{N-1} \left( \begin{array}{c} \xi_{k-1}^T (A_{k+l:k})^T \Omega A_{k+l:k} \xi_{k-1} + 2 \xi_{k-1}^T (A_{k+l:k})^T \Omega \bar{B}_l^1 \bar{X} \\ + \text{tr}((\bar{B}_l^1)^T \Omega \bar{B}_l^1 \bar{P}) + \text{tr}((\bar{B}_l^1)^T \Omega \bar{B}_l^1 \bar{R}) \end{array} \right) \quad (3.14)$$

where

$$\xi_k \triangleq \begin{bmatrix} \hat{x}_{k|k}^1 \\ \hat{x}_{k|k}^2 \end{bmatrix}$$

$$\Omega \triangleq \begin{bmatrix} I_n & -I_n \\ -I_n & I_n \end{bmatrix}$$

$$\bar{B}_l^1 \triangleq [A_{k+l:k+1} B_k^1 \quad A_{k+l:k+2} B_{k+1}^1 \quad \dots \quad A_{k+l} B_{k+l-1}^1 \quad B_{k+l}^1 \quad 0_m \quad \dots \quad 0_m]$$

$$\bar{B}_l^2 \triangleq [A_{k+l:k+1} B_k^2 \quad A_{k+l:k+2} B_{k+1}^2 \quad \dots \quad A_{k+l} B_{k+l-1}^2 \quad B_{k+l}^2 \quad 0_n \quad \dots \quad 0_n].$$

The matrices  $A_{i:j}$ ,  $B_k^1$ ,  $B_k^2$ ,  $\bar{P}$ ,  $\bar{X}$ , and  $\bar{R}$  are defined as

$$A_{i:j} \triangleq A_i \times \dots \times A_j$$

$$A_k \triangleq \begin{bmatrix} (I - K_k^1 C) A & 0_n \\ 0_n & (I - K_k^2) A \end{bmatrix}$$

$$B_k^1 \triangleq \begin{bmatrix} K_k^1 \\ 0_{n,m} \end{bmatrix}$$

$$B_k^2 \triangleq \begin{bmatrix} 0_n \\ K_k^2 \end{bmatrix}$$

$$\bar{P} \triangleq \begin{bmatrix} y_k^1 (y_k^1)^T & y_k^1 (\hat{y}_{k+1|k}^1)^T & \dots & y_k^1 (\hat{y}_{k+N-1|k}^1)^T \\ \hat{y}_{k+1|k}^1 (y_k^1)^T & P_{1,1} & \dots & P_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{y}_{k+N-1|k}^1 (y_k^1)^T & P_{N-1,1} & \dots & P_{N-1,N-1} \end{bmatrix}$$

$$\bar{X} \triangleq \begin{bmatrix} y_k^1 \\ \hat{y}_{k+1|k}^1 \\ \vdots \\ \hat{y}_{k+l|k}^1 \\ \vdots \\ \hat{y}_{k+N-1|k}^1 \end{bmatrix}$$

$$\bar{R} \triangleq \begin{bmatrix} 0_m & 0_m & \cdots & 0_m \\ 0_m & R & \cdots & 0_m \\ \vdots & \vdots & \ddots & \vdots \\ 0_m & 0_m & \cdots & R \end{bmatrix}$$

where the submatrices  $P_{i,j}$  are defined as

$$P_{i,j} \triangleq C(P_{\min(k+i,k+j)|k}^1 + \hat{x}_{\min(k+i,k+j)|k}^1 \hat{x}_{\min(k+i,k+j)|k}^{1T}) (A^{i-j})^T C^T.$$

The matrices  $0_n$  and  $0_{n,m}$  denote the zero matrices of size  $n \times n$  and  $n \times m$  respectively. The detailed derivations of the expressions (3.12)-(3.14) are made in Appendix A.

At this point, it is appropriate to calculate the unquantized optimal solution for  $\Gamma$  since we will use it in our experiments to reduce number of the elements in the quantization table (See Section 3.3 for details). We can modify the expression for  $J(\cdot)$  in (3.7) as shown below.

$$J\left(\Gamma \mid \{P_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1}\right) = \Gamma^T W \Gamma + 2\Gamma_A^T W W^{-1} \Gamma + (W^{-T} \Gamma_A)^T W (W^{-T} \Gamma_A) - (W^{-T} \Gamma_A)^T W (W^{-T} \Gamma_A) + \Lambda \quad (3.15)$$

$$= \Gamma^T W \Gamma + 2\Gamma_A^T W W^{-1} \Gamma + (W^{-T} \Gamma_A)^T W (W^{-T} \Gamma_A) + \bar{\Lambda} \quad (3.16)$$

where

$$\bar{\Lambda} = -(W^{-T} \Gamma_A)^T W (W^{-T} \Gamma_A) + \Lambda.$$

The expression (3.16) can be written as

$$J = (\Gamma + W^{-T} \Gamma_A)^T W (\Gamma + W^{-T} \Gamma_A) + \bar{\Lambda}. \quad (3.17)$$

Hence the optimal solution (unquantized state vectors) without considering any constraints on  $\Gamma$  can be found as

$$\bar{\Gamma}_{uc}^{opt} = -W^{-T} \Gamma_A. \quad (3.18)$$

This is the perfect case when no quantization is considered. For the quantized case, the optimal solution can be given by using Theorem 1 as

$$\bar{\Gamma}^{opt} = W^{-\frac{T}{2}} Q_{NN}(-W^{-1/2} \Gamma_A) \quad (3.19)$$

where  $Q_{NN}(\cdot)$  is the nearest-neighbor quantizer that maps  $\mathbb{R}^{nN}$  (since there are  $N$  state vectors of size  $n$  in  $\Gamma$ ) to the finite set  $V$  defined as

$$V = \{v_1, v_2, \dots, v_K\} \quad (3.20)$$

where  $K$  is the number of elements in  $V$  and

$$v_i = W^{\frac{T}{2}} u_i. \quad (3.21)$$

After finding the optimal quantized states  $\bar{\Gamma}^{opt}$  corresponding to times  $k, k+1, \dots, k+N-1$ , the first quantized state  $\hat{x}_{k|k}^{1 \rightarrow 2}$  is sent to the fusion center.

$$\Gamma^{opt}(k) = [I_n \quad 0_n \quad \dots \quad 0_n] \bar{\Gamma}^{opt} \quad (3.22)$$

where  $I_n$  and  $0_n$  are the identity and zero matrices of size  $n \times n$ .

At time  $k+1$ , in order to be able to find  $\Gamma^{opt}(k+1)$  the same optimization is applied on the interval  $[k+1, k+N]$ , in other words, the horizon is shifted by one.

For the other time instants, the procedure is continued in a similar fashion.

### 3.2. The General Case

We have explained how the optimal quantization is performed in Section 3.1 when the quantized covariance matrices are known. When the covariance matrices  $\{P_{l|l}^{1 \rightarrow 2}\}_{l=k}^{k+N-1}$  sent from the local agent, are not known, which is the general and the realistic case, the cost function in (2.20) becomes too complicated in terms of all of the unknown optimization variables. Therefore, there is no simple solution to the optimization problem as in Section 3.1. On the other hand, in practice, only few different quantized values are communicated between the local center and the fusion center. In other words, the covariance quantization table in the local center is composed of only a small number of covariances. In this section, using this practical information, we propose the following simple solution for the general case when the

quantized covariance values are unknown and they are to be found by the optimization problem.

Suppose that the covariance quantization table in the local center is composed  $K$  different covariances where  $K$  is a small positive integer ( $\ll 100$ ). We assume that all covariances to be sent from the local agent in the optimization horizon are the same, i.e.,

$$P_{k|k}^{1 \rightarrow 2} = P_{k+1|k+1}^{1 \rightarrow 2} = \dots = P_{k+N-1|k+N-1}^{1 \rightarrow 2}. \quad (3.23)$$

The optimization problem (2.20) is then solved for each covariance in the covariance quantization table as explained in Section 3.1 and the covariance value which gives the minimum cost is selected as the optimal quantized covariance. Note that when we try to solve the optimization (2.20) after selecting one of the covariance values in the covariance quantization table, thanks to the assumption (3.23), all quantized covariance values to be sent to the fusion center (in the optimization horizon) are known. Therefore the problem turns into the case of known quantized covariances and hence the solution obtained in Section 3.1 is valid.

### 3.3. Implementation Issue: Size of the Quantization Table

In this section, we consider an implementation issue for the quantization algorithm proposed in the previous two sections. For this purpose, we consider a simple tracking problem where the state vector is composed to 4 variables, namely x-position, y-position, x-velocity and y-velocity. Suppose that we use  $B$  bits in order to quantize the state vector. If we use equal number of bits to quantize each variable in the state vector, then each variable will be quantized with  $B/4$  bits. Assuming minimum and maximum values for the variables, we form a quantization table of  $K_k = 2^{B/4} \times 2^{B/4} \times 2^{B/4} \times 2^{B/4} = 2^B$  values for the state vector for time  $k$ . Since a different state vector can be selected from the quantization table for each time instant  $k$  in the optimization horizon, the overall quantization table over the whole optimization horizon turns out to be  $K = K_k^N = 2^{BN}$ . Therefore, the quantization table in Theorem 1 will have length  $K = 2^{BN}$  which is impossible to process. This is

the main issue that would be encountered in implementing the quantization algorithm proposed in the previous sections.

Reducing the quantization table size by reducing the number of bits in order to solve this issue is not a reasonable approach since such an approach would need to use too few bits for each variable and the performance would be terrible. In this section, we propose an algorithm which reduces the size of the quantization table cleverly by removing unlikely rows from the quantization table by using the unquantized optimal state vector given in (3.18).

We propose the following reduction method. For each time instant  $k$

- Find the unquantized optimal state vector  $\bar{\Gamma}_{uc}^{opt}$  given in (3.18).
- Consider each element of  $\gamma_i^{opt}$  of  $\bar{\Gamma}_{uc}^{opt}$  where  $i = 1, \dots, 4N$ .
  - If  $\gamma_i^{opt}$  is a position variable, find the nearest three quantized position values  $\gamma_i^1, \gamma_i^2, \gamma_i^3$  to  $\gamma_i^{opt}$  from the position quantization table.
  - If  $\gamma_i^{opt}$  is a velocity variable, find the nearest three quantized velocity values  $\gamma_i^1, \gamma_i^2, \gamma_i^3$  to  $\gamma_i^{opt}$  from the velocity quantization table.
- Construct the reduced quantization from all possible combinations of the sets  $\{\gamma_1^j\}_{j=1}^3, \{\gamma_2^j\}_{j=1}^3, \dots, \{\gamma_{4N}^j\}_{j=1}^3$ .

In the reduction procedure described above, since only 3 quantized values are used for each element of the vector  $\Gamma$ , the size of the reduced quantization table becomes  $K = 3^{4N}$  which is much smaller than the value  $2^{BN}$  and quite manageable for small horizon lengths.

## CHAPTER 4

### SIMULATION RESULTS

In this section we compare the quantization method applied in the literature with the one proposed in this study by making a simulation study on three fusion architectures. The simulation results obtained for each architecture are examined in separate sections.

#### 4.1. Architecture 1

Architecture 1 represents the simple fusion architecture used in the earlier chapters which consists of a local agent  $F_1$  and a fusion center  $F_2$ . The local agent has its own sensor collecting measurements. The local agent operates its own Kalman filter and quantizes the track information with static or dynamic quantization methods. After the quantization process, the local agent sends its quantized track information to the fusion center. At the same time, the fusion center operates its own Kalman filter as well. In this architecture there is no sensor belonging to the fusion center. Hence the only data that the fusion center takes from outside is the quantized track information of the local agent. Architecture 1 is illustrated in Figure 4.1.



**Figure 4.1:** Information graph of architecture 1

#### 4.1.1. Target Model

True target states and measurements are generated according to the following linear Gaussian model.

$$x_k = Ax_{k-1} + Bw_k \quad (4.1)$$

$$y_k^1 = Cx_k + v_k^1 \quad (4.2)$$

where the state  $x_k$  is composed of x-y positions and x-y velocities of the target and  $y_k^1$  is the position measurements of the local center. The matrices  $A$ ,  $B$  and  $C$  are given as

$$A = \begin{bmatrix} I_2 & TI_2 \\ 0_2 & I_2 \end{bmatrix}, \quad B = \begin{bmatrix} \frac{T^2}{2}I_2 \\ TI_2 \end{bmatrix}, \quad C = [I_2 \quad 0_2] \quad (4.3)$$

where  $T = 1$ s. The initial state  $x_0 = [p_0^T, v_0^T]^T$  of the target is constructed randomly as follows. The initial position vector  $p_0 = [5000\text{m}, 5000\text{m}]^T$  is fixed. The components of the initial velocity  $v_0$  are random and independent. Each component of  $v_0$  is selected to be Gaussian distributed with zero mean and standard deviation 100m/s. The white process noise  $w_k$  is Gaussian with zero-mean and covariance  $Q = 2^2I_2$ . The white measurement noise  $v_k^1$  is Gaussian with zero-mean and covariance  $R = 20^2I_2$ . The target data is formed for 20 seconds which corresponds to 21 samples.

#### 4.1.2. Quantization Parameters

For the quantized state vector sent from the local agent to the fusion center,  $M$  bits are used for quantizing the state vector where  $M$  is going to be selected as 16, 32 and 64. Additional 4 bits are used for quantizing the covariance.

Since the state vector is 4-dimensional,  $M/4$  bits are reserved for each element of the state vector. A uniform grid of  $2^{M/4}$  values between  $[0, 10000\text{m}]$  is formed as the quantization table for the position values. A uniform grid of  $2^{M/4}$  values between  $[-250\text{m/s}, 250\text{m/s}]$  is formed as the quantization table for the velocity values. The quantized covariances  $P_i^{1 \rightarrow 2}$ ,  $i = 1, 2, \dots, K$  are selected as



$$P_i^{1 \rightarrow 2} = \begin{bmatrix} \sigma_{p,i}^2 I_2 & 0_2 \\ 0_2 & \sigma_{v,i}^2 I_2 \end{bmatrix}$$

where the position and velocity standard deviation values  $\sigma_{p,i}$  and  $\sigma_{v,i}$  are selected are given in Table 4.1.

**Table 4.1:** Covariance quantization table

$i$	$\sigma_{p,i}$ (m)	$\sigma_{v,i}$ (m/s)
1	2	0.4
2	4	0.8
3	6	1.2
4	8	1.6
5	10	2
6	12	2.4
7	14	2.8
8	16	3.2
9	20	20/3
10	24	24/3
11	28	28/3
12	32	32/3
13	36	36/3
14	40	40/3
15	44	44/3
16	48	48/3

For dynamic quantization the horizon length is selected to be  $N = 3$  and the constant  $\lambda$  determining the effect of covariances in the cost function (2.21) is selected as  $\lambda = 1000$ .

### 4.1.3. Methods

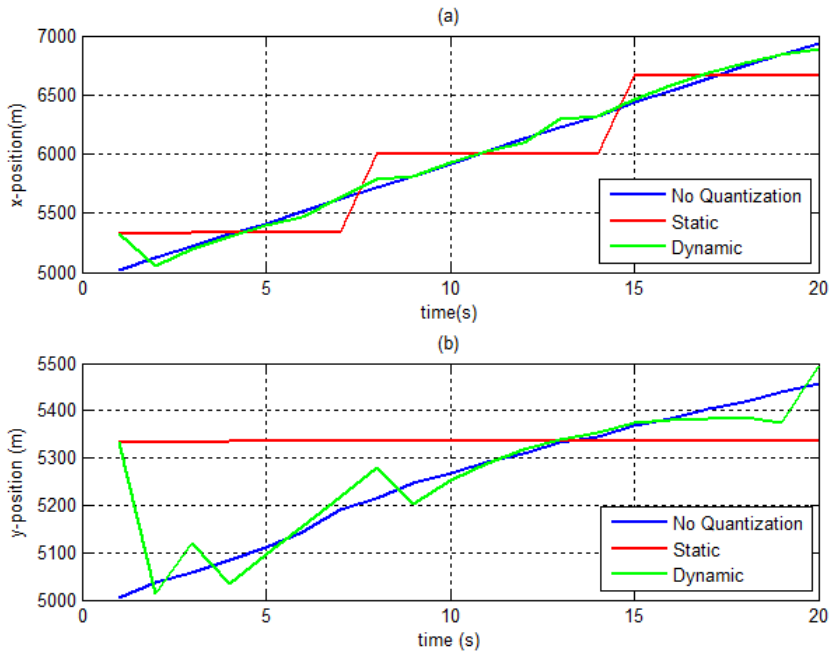
Three sets of results are obtained. In the first set of results, the optimal fusion result that would be obtained in the fusion center if no quantization was involved is calculated. This result is used as a baseline for comparing the results of the fusion centers using static and dynamic quantization methods.

The second set of results is the fusion results in the fusion center when static quantization is used between the local center and the fusion center.

The third set of results is the fusion results in the fusion center when dynamic quantization is used between the local center and the fusion center.

#### 4.1.4 Results

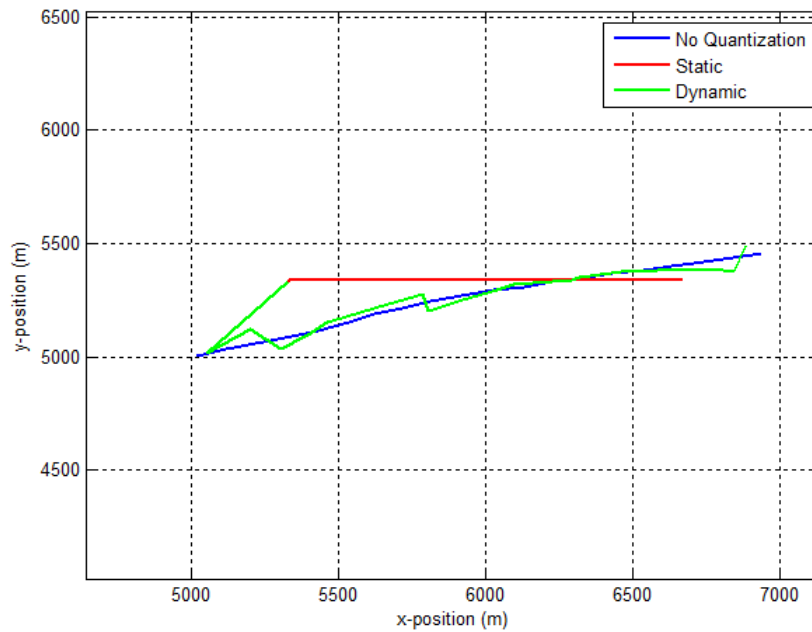
A total of 100 Monte Carlo simulations are made for  $M = 16, 32$  and  $64$  bits. In each run, a different realization for the true target state vectors and measurements is used. Figure 4.2 shows the position estimates at the fusion center for  $M = 16$  for a typical single run.



**Figure 4.2:** Position estimates in the fusion center with no quantization, static and dynamic quantization when  $M = 16$  bits: (a) x-positions (b) y-positions

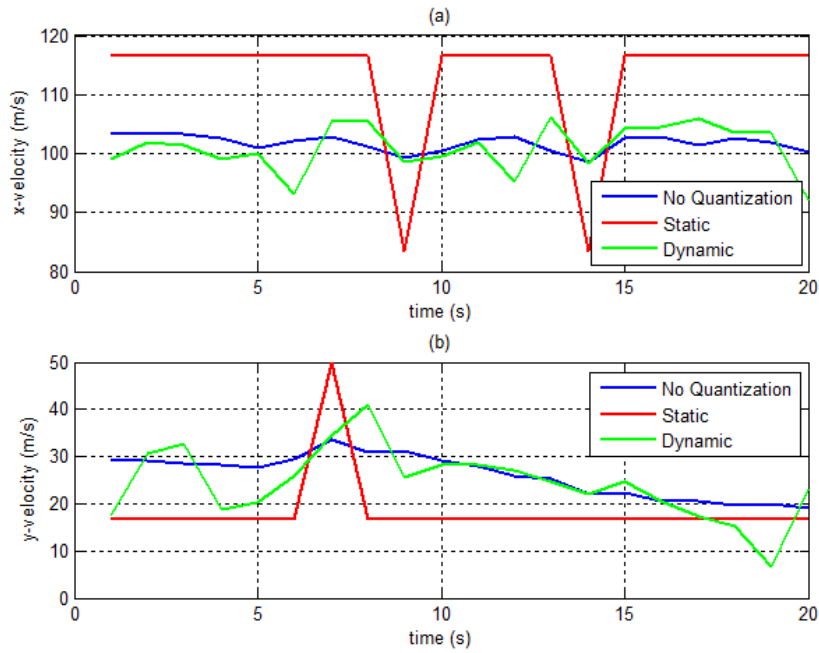
As shown in Figure 4.2-a, static quantization method is able to send only 3 different x-position values to the fusion center from quantization table. On the other hand, although dynamic quantization has utilized the same model and exactly the same quantization table with static quantization method, it has obtained much closer

results to the no quantization case than the static quantization by using low pass structure of Kalman filter in the fusion center. Likewise, there is only one value which static quantization sends to the fusion center from the quantization table in Figure 4.2-b. In this case, the fusion center results obtained by dynamic quantization are again much closer to the results of the no quantization case. Figure 4.3 shows the x-y position estimates in Figure 4.2 on a single graph.



**Figure 4.3:** Position estimates in the fusion center with no quantization, static and dynamic quantization when  $M = 16$  bits.

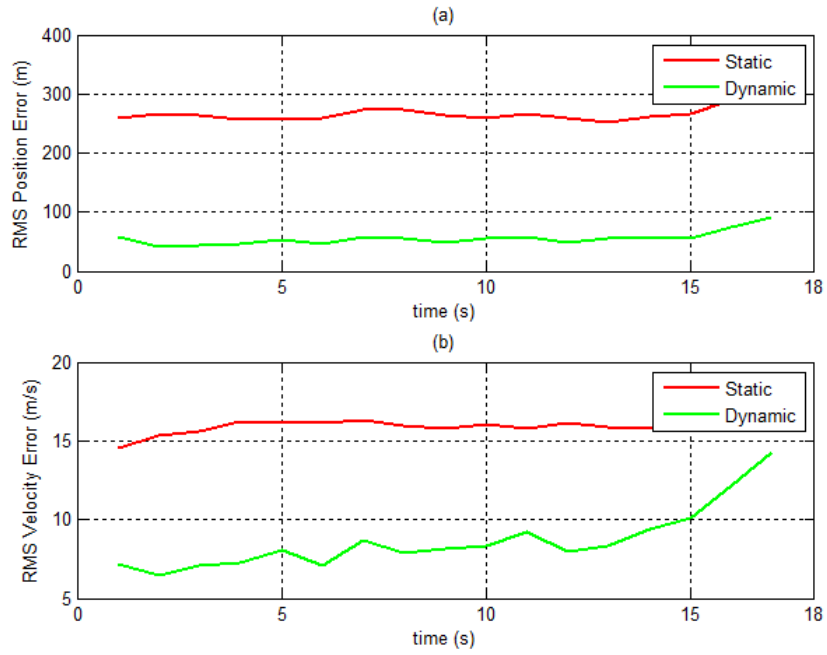
Figure 4.4 shows the velocity estimates at the fusion center for  $M = 16$  for a single run. As shown in the figure, the velocity estimates obtained with dynamic quantization are again much closer to the no quantization case.



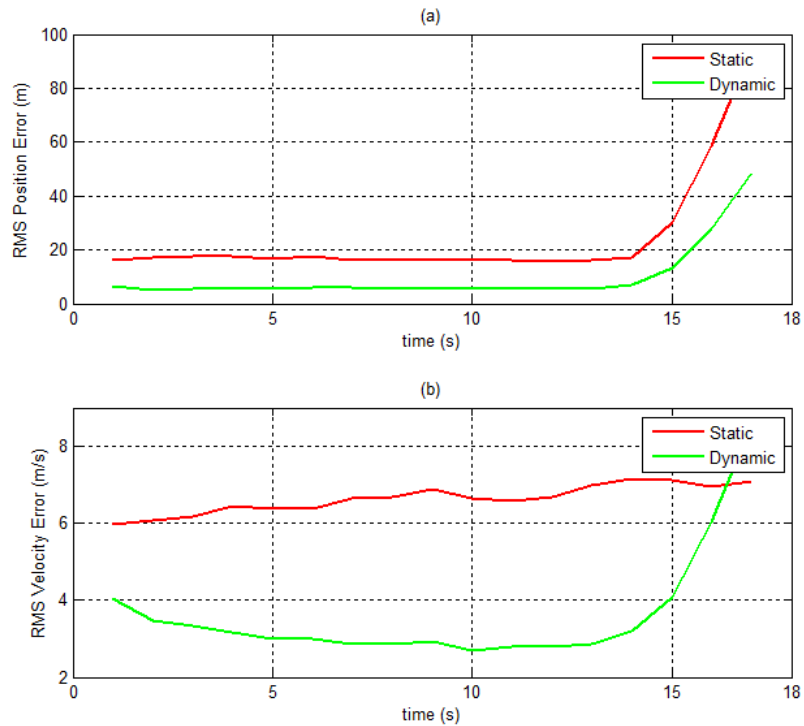
**Figure 4.4:** Velocity estimates in the fusion center with no quantization, static and dynamic quantization when  $M = 16$  bits: (a) x-velocity (b) y-velocity

Over the Monte Carlo runs RMS position and velocity errors are calculated between the results obtained with and without quantization. The RMS errors are shown in Figure 4.5. As expected the RMS errors of dynamic quantization is much lower than those of static quantization.

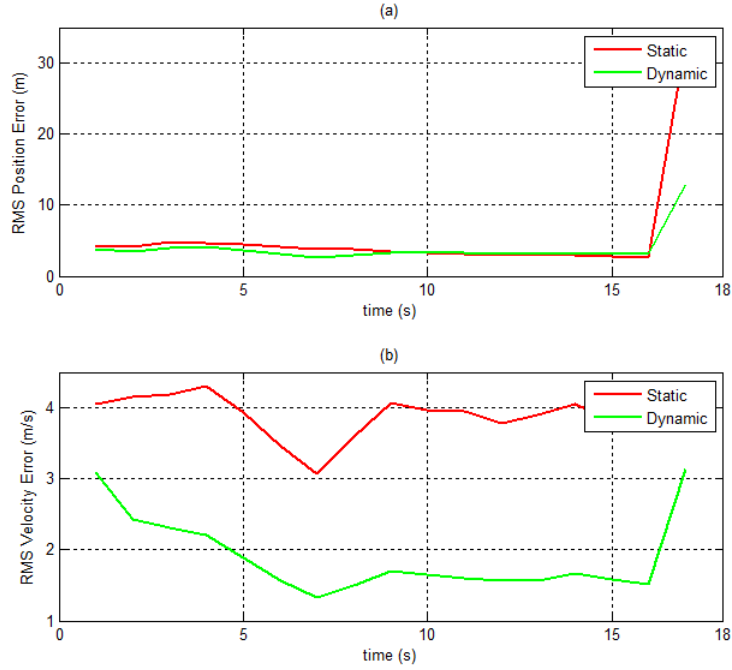
For  $M = 32$  and  $64$  bits, the simulations are repeated. In these cases, the single run results are not shown. Figure 4.6 and Figure 4.7 show the RMS position and velocity errors obtained by the algorithms for  $M = 32$  and  $M = 64$  respectively. As expected, errors for both quantization methods get smaller as the number of bits increases. The errors obtained with dynamic quantization are still much smaller than those obtained with static quantization.



**Figure 4.5:** RMS errors in the fusion center with static and dynamic quantization when  $M = 16$ : (a) RMS position errors (b) RMS velocity errors



**Figure 4.6:** RMS errors in the fusion center with static and dynamic quantization when  $M = 32$ : (a) RMS position errors (b) RMS velocity errors



**Figure 4.7:** RMS errors in the fusion center with static and dynamic quantization when  $M = 64$ : (a) RMS position errors (b) RMS velocity errors

In order to summarize, average RMS errors (over time) for  $M = 16, 32$  and  $64$  bits are given in Table 4.2. In the table, in addition to the average RMS position and velocity errors related to the state vector, average RMS position and velocity covariance errors are also provided. The covariance errors in each case are calculated using the Frobenius norm of the difference between the covariances obtained by methods with quantization and without quantization.

**Table 4.2:** RMS errors for Architecture 1

RMS Error	16 Bit		32 Bit		64 Bit	
	Static	Dynamic	Static	Dynamic	Static	Dynamic
Position (m)	266.80	56.70	31.78	14.96	8.74	4.51
Velocity (m/s)	15.87	8.89	6.65	3.91	3.89	1.98
Cov. Position (m)	3.99	3.22	3.95	3.22	3.81	3.23
Cov. Velocity (m/s)	3.96	2.59	3.96	2.59	3.96	2.60

As observed in the table, the difference between errors in the covariances obtained with dynamic quantization and static quantization is not as large as the difference between the position and velocity errors of the two methods. Nevertheless, the covariance errors of the dynamic quantization are still slightly smaller than static quantization.

## 4.2. Architecture 2

Architecture 2 consists of a local agent and a fusion center as Architecture 1. The only difference of Architecture 2 from Architecture 1 is that the fusion center in Architecture 2 has its own sensor. Architecture 2 is illustrated in Figure 4.8.



**Figure 4.8:** Information graph of Architecture 2

### 4.2.1. Target Model

True target states and measurements are generated according to the following linear Gaussian model.

$$x_k = Ax_{k-1} + Bw_k \quad (4.4)$$

$$y_k^1 = Cx_k + v_k^1 \quad (4.5)$$

$$y_k^2 = Cx_k + v_k^2 \quad (4.6)$$

The quantities  $x_k$ ,  $y_k^1$ ,  $w_k$ ,  $v_k^1$  and the matrices  $A$ ,  $B$ ,  $C$  are the same as those defined in Section 4.1.1. The quantities  $y_k^2$  represent the measurements of the fusion center. The white measurement noise  $v_k^2$  is Gaussian with zero-mean and covariance  $R = 20^2 I_2$ . The measurement noise  $v_k^2$  is independent of  $v_k^1$ .

### 4.2.2. Quantization Parameters

Quantization parameters are the same as those given in Section 4.1.2.

### 4.2.3. Methods

In this architecture, the only difference in the methods used is that the fusion center has now measurements  $y_k^2$  to process in addition to the quantized information coming from the local center. Due to the additional measurements, the estimates of the fusion center are no longer directly equal to the quantized values coming from the local center. Therefore, the Channel Filter equations to obtain  $\hat{x}_{k|k}^f$  and  $P_{k|k}^f$  from  $\hat{x}_{k|k}^{1 \rightarrow 2}$  and  $P_{k|k}^{1 \rightarrow 2}$  in the fusion center used in practice are different from (2.11)-(2.12) due to the presence of measurements  $y_k^2$ . The corresponding Channel Filter equations are given as follows.

$$\left(P_{k|k}^f\right)^{-1} \hat{x}_{k|k}^f = \left(P_{k|k}^2\right)^{-1} \hat{x}_{k|k}^2 + \left(P_{k|k}^{1 \rightarrow 2}\right)^{-1} \hat{x}_{k|k}^{1 \rightarrow 2} - \left(P_{k|k-1}^{1 \rightarrow 2}\right)^{-1} \hat{x}_{k|k-1}^{1 \rightarrow 2} \quad (4.7)$$

$$\left(P_{k|k}^f\right)^{-1} = \left(P_{k|k}^2\right)^{-1} + \left(P_{k|k}^{1 \rightarrow 2}\right)^{-1} - \left(P_{k|k-1}^{1 \rightarrow 2}\right)^{-1} \quad (4.8)$$

. The explanations for the terms in (4.7)-(4.8) are given below.

- $\hat{x}_{k|k}^2$  and  $P_{k|k}^2$  are the posterior state estimate and covariance in the fusion center obtained using only the local measurements  $y_k^2$ ;
- $\hat{x}_{k|k}^f$  and  $P_{k|k}^f$  are the updated state estimate and covariance in the fusion center;
- $\hat{x}_{k|k-1}^{1 \rightarrow 2}$  and  $P_{k|k-1}^{1 \rightarrow 2}$  are the predicted versions of the previously communicated information  $\hat{x}_{k-1|k-1}^{1 \rightarrow 2}$  and  $P_{k-1|k-1}^{1 \rightarrow 2}$  between the local center and the fusion center.

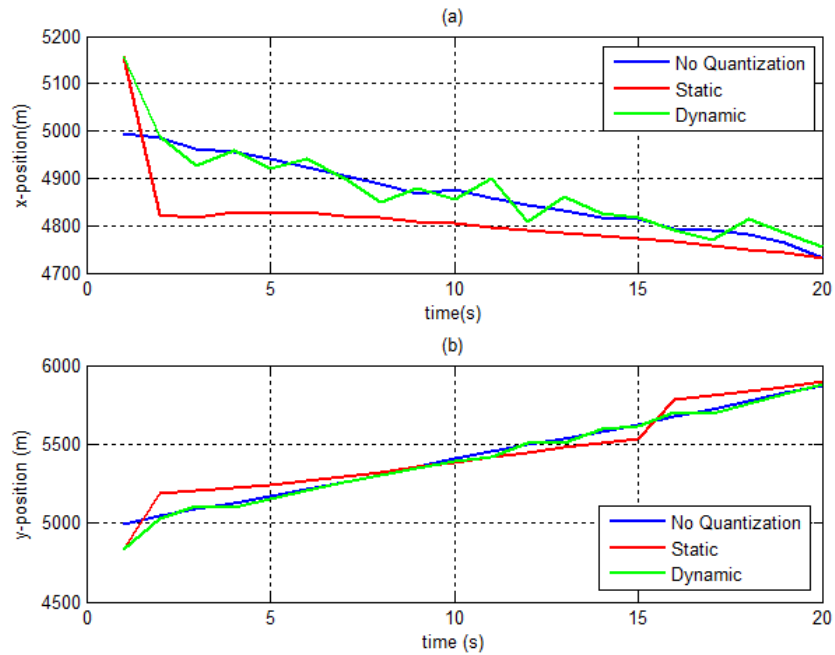
Note that, in this case, since the measurements  $y_k^2$  are not available in the local center, the local copy of the fusion center Kalman filter running at the local center still uses only the quantized information of the local center as measurements.



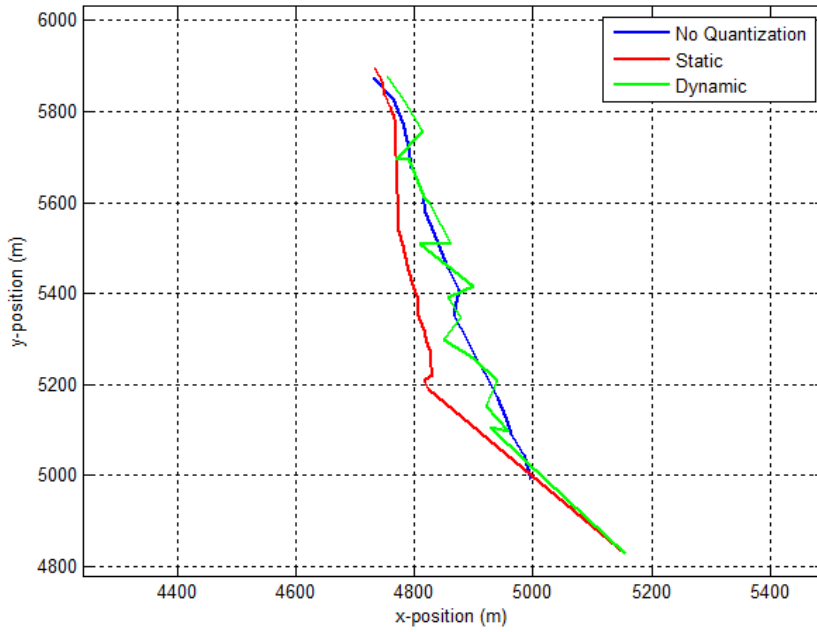
#### 4.2.4. Results

A total of 100 Monte Carlo simulations are made for  $M = 16, 32$  and  $64$  bits. In each run, a different realization for the true target state vectors and measurements is used.

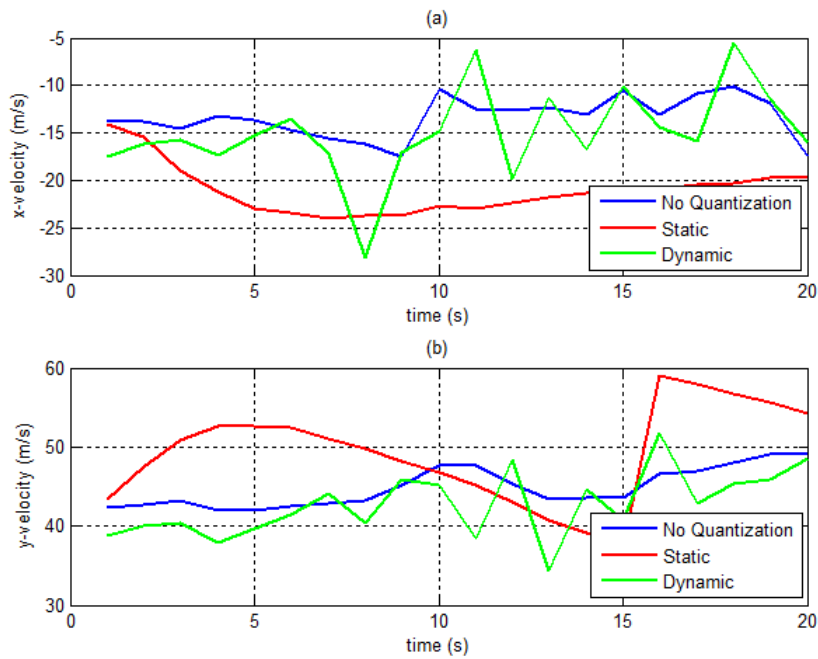
Figure 4.9 shows the position estimates obtained in the fusion center for the no quantization case and static and dynamic quantization based methods when  $M = 16$  bits for a typical single run. Figure 4.10 shows the results in Figure 4.9 on a single graph. Velocity estimates obtained in the fusion center for the different quantization schemes are illustrated in Figure 4.11. All three figures confirm the superiority of the dynamic quantization over static quantization approach.



**Figure 4.9:** Position estimates in the fusion center with no quantization, static and dynamic quantization when  $M = 16$  bits: (a) x-positions (b) y-positions



**Figure 4.10:** Position estimates in the fusion center with no quantization, static and dynamic quantization when  $M = 16$  bits.



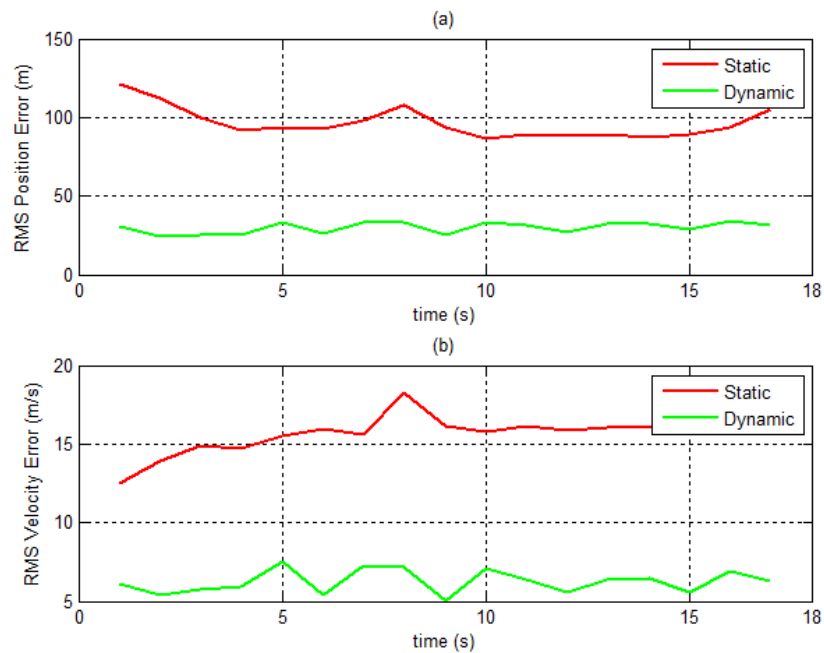
**Figure 4.11:** Velocity estimates in the fusion center with no quantization, static and dynamic quantization when  $M = 16$  bits: (a) x-velocity estimates (b) y-velocity estimates.

RMS position and velocity errors are calculated between the results obtained with and without quantization. The RMS errors are shown in Figure 4.12 which shows the better performance of dynamic quantization once again.

For  $M = 32$  and 64 bits, the simulations are repeated. In these cases, the single run results and RMS errors are not illustrated with figures. Instead, only the average RMS errors (over time) for  $M = 16, 32$  and 64 bits are presented in Table 4.3. The results observed in the table are similar to those obtained in Table 4.2.

### 4.3 Architecture 3

Architecture 3 consists of two local agents and a fusion center. All agents in the architecture have their own sensors collecting measurements independently. Architecture 3 is illustrated in Figure 4.13.



**Figure 4.12:** RMS errors in the fusion center with static and dynamic quantization when  $M = 16$  bits: (a) RMS position errors (b) RMS velocity errors.

**Table 4.3:** RMS errors for Architecture 2

RMS Error	16 Bit		32 Bit		64 Bit	
	static	dynamic	static	dynamic	static	dynamic
Position (m)	96.91	30.12	43.37	22.57	8.25	5.51
Velocity (m/s)	16.02	6.30	11.51	6.13	6.29	1.39
Cov. Position (m)	6.86	2.74	6.86	2.73	6.87	2.74
Cov. Velocity (m/s)	3.42	2.05	3.42	2.05	3.42	2.05

In Figure 4.13,  $F_1$  and  $F_2$  represent the local agents and  $S_1$  and  $S_2$  represent the sensors belonging to the corresponding local agents.  $F_3$  represents the fusion center and  $S_3$  represents the sensor belonging to the fusion center.

#### 4.3.1. Target Model

True target states and measurements are generated according to the following linear Gaussian model.

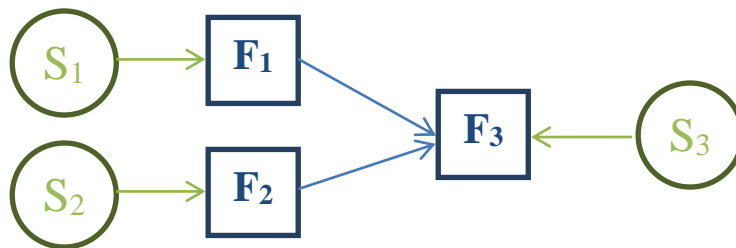
$$x_k = Ax_{k-1} + Bw_k \quad (4.9)$$

$$y_k^1 = Cx_k + v_k^1 \quad (4.10)$$

$$y_k^2 = Cx_k + v_k^2 \quad (4.11)$$

$$y_k^3 = Cx_k + v_k^3 \quad (4.12)$$

The quantities in (4.9)-(4.12) are defined similarly to those in Section 4.1.1 and 4.2.1.



**Figure 4.13:** Information graph of architecture 3

### 4.3.2. Quantization Parameters

Quantization parameters are the same as those given in Section 4.1.2.

### 4.3.3. Methods

Methods are the same as those given in Section 4.1.3. The difference is that the fusion center has now measurements  $y_k^3$  to process in addition to the quantized pieces of information coming from the first and the second local centers. Note that, in this case, each the local center keeps a local copy of the fusion center Kalman filter using only the quantized information of the corresponding local center as measurements.

### 4.3.4 Results

For Architecture 3, only the final RMS errors are shown in Table 4.4. The results are very similar to those obtained for the earlier architectures.

**Table 4.4:** RMS errors for Architecture 3

RMS Error	16 Bit		32 Bit		64 Bit	
	static	dynamic	static	dynamic	static	dynamic
Position (m)	119.62	30.99	9.33	5.40	8.08	5.22
Velocity (m/s)	13.8	6.71	4.51	1.29	4.91	1.45
Cov. Position (m)	6.04	2.96	6.04	2.95	6.04	2.95
Cov. Velocity (m/s)	3.46	2.33	3.46	2.33	3.46	2.33

#### 4.4 Effects of Horizon Length

In this section we will show the effects of the time horizon length on the performance of the quantization operation. Note that in all simulations above the time horizon was selected as  $N = 3$ . Now it is appropriate to change the time horizon and observe the quantization results. For this purpose we repeat the simulations of Architecture 1 by only changing the time horizon. RMS errors for  $M = 16, 32$  and  $64$  bits are calculated for static and dynamic quantization methods and they are shown in Table 4.5.

While the RMS covariance errors obtained by dynamic quantization methods are similar to the errors obtained by static quantization, in terms of RMS position and velocity errors, all dynamic quantization methods are significantly better than the static quantization irrespective of the horizon length used. If we compare the RMS errors of dynamic quantization methods with different horizon lengths among themselves, it is seen that reducing the horizon length does not reduce the performance of dynamic quantization. In fact, we observe interestingly that RMS position and velocity errors slightly decrease as the horizon length is reduced. The reason for this counter-intuitive reduction in the RMS errors for shorter horizons lies in the covariance selection process applied in our algorithm. It is seen in Table 4.5 that the covariance errors of the algorithms decrease as the horizon length is decreased. Hence, the algorithm can select better covariances as the time horizon gets shorter. The assumption (3.23) used in covariance selection constrains all of the quantized covariances over the horizon to be equal. Due to this restriction, as the horizon gets longer, worse and worse covariances are selected in the algorithm which leads to the increase in the RMS errors in the state variables. As the horizon becomes shorter, the effect of the restriction diminishes making the results better.

**Table 4.5:** RMS errors of quantization methods for different time horizons

RMS Error		Position (m)	Velocity (m/s)	Cov. Position (m)	Cov. Velocity (m/s)
16 Bit	Static	266.7994	15.8678	3.9922	3.9609
	Dynamic (N=3)	56.6983	8.8933	3.2209	2.5929
	Dynamic (N=2)	54.9329	8.4419	2.9896	2.5933
	Dynamic (N=1)	52.6651	7.4909	2.9121	2.5904
32 Bit	Static	31.7848	6.6476	3.9450	3.9602
	Dynamic (N=3)	14.9589	3.9120	3.2246	2.5928
	Dynamic (N=2)	14.9678	3.9043	2.9941	2.5939
	Dynamic (N=1)	14.7358	3.8469	2.8957	2.5921
64 Bit	Static	8.7383	3.8937	3.8140	3.9630
	Dynamic (N=3)	4.5133	1.9750	3.2302	2.5956
	Dynamic (N=2)	4.3491	1.9421	2.9765	2.5979
	Dynamic (N=1)	4.4926	1.9525	2.8865	2.5954

It must be said here that, if receding horizon idea could be applied without any restrictions as the one imposed in the assumption (3.23), increasing the horizon length would cause a decrease in the errors. However, when we impose the assumption (3.23), the errors increase more than they would decrease thanks to the increase in the horizon length which results in the counter-intuitive behavior observed in Table 4.5.

As a final remark, we can still conclude that the dynamic quantization method proposed in this study is more advantageous than the static quantization method used in practice even with a short time horizon.

## 4.5 Computation Time

In this section, we give the computation times of dynamic and static quantization approaches (per each time step). The computation times for different time horizon lengths, which were calculated in Matlab R2011b running on a PC with Intel Core 2 Duo P8800 2.67GHz CPU and 4GB RAM, are given in Table 4.6.

**Table 4.6:** Computation times for different time horizons

Computation Time (s)	16 Bit	32 Bit	64 Bit
Static	0.0135	0.0138	0.0157
Dynamic (N=3)	6.6548	7.3894	8.0594
Dynamic (N=2)	0.1258	0.1339	0.2577
Dynamic (N=1)	0.0174	0.0179	0.0715

As shown in Table 4.6, the computation times of static quantization are very low for all bit rates. As the time horizon gets shorter, the computation times of dynamic quantization decreases dramatically. When the time horizon length is unity, the computation times of dynamic quantization are very close to those of static quantization.



## CHAPTER 5

### CONCLUSION AND FUTURE WORK

This thesis study has proposed a new quantization algorithm for track fusion under communication constraints. The quantization algorithm used in practice is a simple nearest neighbor methodology which is named here as static quantization. On the other hand, the new quantization algorithm proposed here involves posing the quantization problem in an optimization framework. The objective function of the optimization measures the distance between the estimates/covariances of the local and remote Kalman filters over a time horizon. Thanks to the time horizon, the predicted future values of the track are brought into the picture. Since this approach considers the inherent dynamic characteristics of the tracks, the corresponding methodology has been named as dynamic quantization.

The optimization involved in the quantization operation has been solved using the receding horizon control methodology. The solution of the optimization was achieved in two stages. In the first stage, the assumption of known quantized covariances was made. In this case, it turned out that the cost function was quadratic in the unknown quantized state vectors. An existing solution in the literature has been used to solve the optimization problem for this case. In the second stage, the assumption of known quantized covariances was removed and the general problem is solved. For this purpose, the assumption that the quantized covariances are the same over the time horizon was made for ensuring computational feasibility.

The simulation studies have shown that the proposed quantization methodology is much better than the static quantization used in practice in terms of RMS errors in

state variables for all time horizon lengths considered. In terms of RMS covariance errors, the difference between the methods is less although dynamic quantization still is slightly better than static quantization. It has been seen that the assumption of “equal quantized covariances over the time horizon” used for solving the optimization problem results in counter-intuitive results for different time horizon lengths. The investigation of less restrictive assumptions under which the optimization problem can still be solved is left as a future work.

It has been seen that dynamic quantization takes much more computation time than static quantization for horizon lengths  $N = 2$  and  $3$ . However, for horizon length  $N = 1$  there is still no performance degradation and the computation time difference between static and dynamic quantization approaches is negligible. Therefore, horizon length can safely be decreased until the computation time for dynamic quantization is feasible for a specific application.

The simulation results show that the proposed dynamic method achieves much better results than static approach even with very low bit rates. As expected, the difference between static and dynamic quantization approaches decreases as the bit rate is increased. The extreme case of  $M = 16$  bits showed that while the static quantization approach can barely go below 100m position error standard deviation, the dynamic quantization approach can well obtain position errors around 50m standard deviation or less.

As future work, the applicability of the methodology for more complicated state estimators, such as interacting multiple model (IMM) filter, extended Kalman filter etc., can be investigated. Also tests in more complex fusion architectures can be useful as future work.

## REFERENCES

- [1] Gök, G. and Orguner, U., “İletişim Kısıtları Altında İz Birleştirme için Dinamik Nicemleme Kullanımı”, *IEEE 23<sup>th</sup> Sinyal İşleme ve İletişim Uygulamaları Kurultayı*, April 2015.
- [2] Challa, S., Morelande, M. R., Musicki, D. and Evans, R. J., “Introduction to Object Tracking”, Ch.1, *Fundamentals of Object Tracking*, Cambridge University Press, 2011.
- [3] Bar-Shalom, Y., Li, R., and Kirubarajan, T., “*Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*”, A Wiley Interscience Publication, 2001.
- [4] Kalman, R. “A New approach to Linear Filtering and Prediction Problems”, *Transactions of the ASME–Journal of Basic Engineering*, 82 (Series D), (1960), 35-45.
- [5] Kailath, T., Sayed, A., and Hassibi, B., “*Linear Estimation (Information and System Sciences Series)*”, Prentice-Hall, 2000.
- [6] Schmidh, S., “Application of State Space Methods to Navigation Problems”, *Advances in Control Systems*, (1966), 293-340.
- [7] Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. F., “A New approach for Filtering Nonlinear Systems”, In *Proceedings of American Control Conference*, Vol.3, 1995, 1628-1632.

- [8] Julier, S. J., and Uhlmann, J. K., “Unscented Filtering and Nonlinear Estimation”, *Proceedings of IEEE*, 92, 3 (Mar. 2004), 401- 422.
- [9] Arulampalam, M. S., Maskell, S., Gordon, N. and Clapp, T., “A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking”, *IEEE Transactions on Signal Processing*, Vol. 50, No. 2, Feb. 2002.
- [10] Gustafsson, F., “Particle Filter Theory and Practice with Positioning Applications Part 2: Tutorials”, *IEEE A&E Systems Magazine*, Vol. 25, No. 7, Jul. 2010.
- [11] Liggins, II, M. E., Chong, C., Kadar, I., Alford, M. G., Vannolica, V. and Thomopoulos, V., “Distributed Fusion Architectures and Algorithms for Target Tracking”, *Proceedings of the IEEE*, Vol.85, No.1, Jan. 1997.
- [12] Liggins, M. E. and Chang, K., “Distributed Fusion Architectures, Algorithms, and Performance within a Network-Centric Architecture”, Ch.17, *Handbook of Multisensor Data Fusion: Theory and Practice*, Taylor & Francis Group, Second Edition, 2009.
- [13] Bar-Shalom, Y., “Update with Out-of-Sequence Measurements in Tracking: Exact Solution”, *IEEE Transactions on Aerospace and Electronics Systems*, Vol. 38, No. 3, Jul. 2002.
- [14] Chang, K. C., Saha, R. K. and Bar-Shalom, Y., “On Optimal Track-to-Track Fusion”, *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 33, No. 4, Oct. 1997.
- [15] Blackman, S. and Popoli, R., *Design and Analysis of Modern Tracking Systems*, Norwood MA: Artech House, 1999.

- [16] Bar-Shalom, Y., "On the Track-to-Track Correlation Problem", *IEEE Transactions on Automatic Control*, Vol. AC-26, No. 2, Apr. 1981.
- [17] Chang, K. C., Saha, R. K., Bar-Shalom, Y., and Alford, M., "Performance Evaluation of Multisensor Track-to-Track Fusion", *IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Vol. 8, No. 11, Dec. 1996.
- [18] Chong, C. Y., "Hierarchical Estimation", *Proc. MIT/ONR Workshop on C3*, 1979.
- [19] Chong, C. Y., Chang, K. C., and Mori, S., "Distributed Tracking in Distributed Sensor Networks", *Proc. Amer. Contr. Conf.*, 1986.
- [20] Chong, C. Y., Mori, S., and Chang, K. C., "Distributed Multitarget Sensor Tracking", Chapter 8, *Multitarget-Multisensor Tracking: Applications and Advances*, Edited by Bar-Shalom, Y., Artech House.
- [21] Bar-Shalom, Y., and Campo, L., "The Effect of the Common Process Noise on the Two-Sensor Fused-Track Covariance", *SPAWAR*, Mar. 1986.
- [22] Saha, R. K., "Track-to-Track Fusion With Dissimilar Sensors", *IEEE Transactions on Aerospace and Electronics Systems*, Vol. 32, No. 3, Jul. 1996.
- [23] Saha, R. K., and Chang, K.C., "An Efficient Algorithm for Multisensor Track Fusion", *IEEE Transactions on Aerospace and Electronics Systems*, Vol. 34, No. 1, Jan. 1998.
- [24] Li, X. R., Zhu, Y., Wang, J., and Han, C. Z., "Unified Optimal Linear Estimation Fusion-PartI: Unified Models and Fusion Rules", *Proc. International*

*Conf. on Information Fusion*, pp. MoC2.10-MoC2.17, Paris, France, Jul. 2000. IEEE Transactions on Information Theory, Vol. 49, No. 9, Sept. 2003.

[25] Li, X. R., and Wang, J., ““Unified Optimal Linear Estimation Fusion-PartII: Discussions and Examples”, *Proc. International Conf. on Information Fusion*, pp. MoC2.18-MoC2.25, Paris, France, Jul. 2000.

[26] Li, X. R., and Zang, P., “Optimal Linear Estimation Fusion-PartIII: Cross-Correlation of Local Estimation Errors”, *Proc. International Conf. on Information Fusion*, pp. WeB1.11-WeB1.18, Montreal, QC, Canada, Aug. 2001.

[27] Li, X. R., and Zang, P., “Optimal Linear Estimation Fusion-PartIV: Optimality and Efficiency of Distributed Fusion”, *Proc. International Conf. on Information Fusion*, pp. WeB1.191-WeB1.26, Montreal, QC, Canada, Aug. 2001.

[28] Li, X. R., Zhang, K., Zhao, J., and Zhu, Y., “Optimal Linear Estimation Fusion-PartV: Relationships”, *Proc. International Conf. on Information Fusion*, pp. 497-504, Annapolis, MD, July 2002.

[29] Li, X. R., “Optimal Linear Estimation Fusion-PartVII: Dynamic Systems”, *Proc. International Conf. on Information Fusion*, pp. 445-462, Cairns, Australia, July 2003.

[30] Tian, X. and Bar-Shalom, Y., “Sequential Track-to-Track Fusion Algorithm: Exact Solution and Approximate Implementation”, *Proc. Of SPIE*, Vol. 6969, No. 696910-1, Mar. 2008.

[31] Goavers, F., and Koch, W., “An Exact Solution to Track-to-Track Fusion at Arbitrary Communication Rates”, *IEEE Transactions on Aerospace and Electronic Systems*, Jan. 2012.

- [32] Koch, W., “On Optimal Distributed Kalman Filtering and Retrodiction at Arbitrary Communication Rates for Maneuvering Targets”, *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2008)*, 2008.
- [33] Koch, W., “Exact Update Formulae for Distributed Kalman Filtering and Retrodiction at Arbitrary Communication Rates”, *Proceedings of 12<sup>th</sup> International Conference on Information Fusion*, 2009.
- [34] Goavers, F., and Koch, W., “Distributed Kalman Filter Fusion at Arbitrary Instants of Time”, *Proceedings of 13<sup>th</sup> International Conference on Information Fusion*, 2010.
- [35] Goavers, F., and Koch, W., “On the Globalized Likelihood Function for Exact Track-to-Track Fusion at Arbitrary Instants of Time”, *Proceedings of 14<sup>th</sup> International Conference on Information Fusion*, 2011.
- [36] Chang, K. C., Zhi, T., and Saha, R. K., “Performance Evaluation of Track Fusion with Information Matrix Filter”, *IEEE Transactions on Aerospace and Electronics Systems*, Vol. 38, No. 2, Apr. 2002.
- [37] Chen, H., Kirubarajan, T., and Bar-Shalom, Y., “Performance Limits of Track-to-Track Fusion versus Centralized Estimation Theory and Application”, *IEEE Transactions on Aerospace and Electronics Systems*, Vol. 39, No. 2, Apr. 2003.
- [38] Roecker, J. A., and McGillem, C. D., “Comparison of Two-Sensor Tracking Methods based on State Vector Fusion and Measurement Fusion”, *IEEE Transactions on Aerospace and Electronics Systems*, Vol. 24, 1988.
- [39] Julier, S., Uhlmann, J. K., “General Decentralized Data Fusion with Covariance Intersection (CI)”, Ch.12, *Handbook of Data Fusion*, CRC Press, 2001.

[40] Chang, K. C., Chong, C. and Mori, S., “On Scalable Distributed Sensor Fusion”, *Proceedings of 11<sup>th</sup> International Conference on Information Fusion*, Jul. 2008.

[41] Günay, M., “Representations of Covariance Matrices in Track Fusion Problems”, *A Thesis Submitted to the Graduate School of Natural and Applied Sciences of Middle East Technical University*, November 2007.

[42] Goodwin, G.C. and Quevedo, D.E., “Finite Alphabet Control and Estimation”, *International Journal of Control, Automation, and Systems*, Vol.1, No. 4, December 2003.



## APPENDIX A

### DERIVATIONS FOR THE TERMS $W$ , $\Gamma_A$ AND $\Lambda$

In this Appendix, we are going to present a derivation for the expressions for the variables  $W$ ,  $\Gamma_A$  and  $\Lambda$  in the cost function (3.7).

#### A.1 Expressing the Cost in Matrix Form

When the quantized covariance matrices are known the cost function for the optimization problem equivalently becomes

$$J\left(\Gamma|\{P_{l|l}^{1\rightarrow 2}\}_{l=k}^{k+N-1}\right) = \sum_{l=k}^{k+N-1} E\{\|\hat{x}_{l|l}^2 - \hat{x}_{l|l}^1\|_2^2 | y_{0:k}^1\} \quad (\text{A.1})$$

In the following, the cost function  $J\left(\Gamma|\{P_{l|l}^{1\rightarrow 2}\}_{l=k}^{k+N-1}\right)$  will be called as  $J(\cdot)$ . In order to be able to make our analysis in a structured manner, we are going to write the cost function in (A.1) as given below.

$$J(\cdot) = \sum_{l=k}^{k+N-1} E\{\| [I_n \quad -I_n] \xi_l \|^2 | y_{0:k}^1\} \quad (\text{A.2})$$

$$= \sum_{l=k}^{k+N-1} E\{\xi_l^T \Omega \xi_l | y_{0:k}^1\} \quad (\text{A.3})$$

$$= \sum_{l=k}^{k+N-1} E\{\|\xi_l\|_{\Omega}^2 | y_{0:k}^1\} \quad (\text{A.4})$$

$$= \sum_{l=0}^{N-1} E \left\{ \|\xi_{k+l}\|_{\Omega}^2 | y_{0:k}^1 \right\} \quad (\text{A.5})$$

where the augmented state vector  $\xi_k$  is defined as

$$\xi_k \triangleq \begin{bmatrix} \hat{x}_{k|k}^1 \\ \hat{x}_{k|k}^2 \end{bmatrix}. \quad (\text{A.6})$$

The matrix  $\Omega$  and the Mahalanobis norm  $\|\cdot\|_{\Omega}$  in (A.3)-(A.5) are defined as

$$\Omega \triangleq \begin{bmatrix} I_n & -I_n \\ -I_n & I_n \end{bmatrix} \quad (\text{A.7})$$

$$\|x\|_{\Omega} \triangleq x^T \Omega x. \quad (\text{A.8})$$

The recursions for the components of the augmented state, i.e.,  $\hat{x}_{k|k}^1$  and  $\hat{x}_{k|k}^2$ , can be written as

$$\hat{x}_{k|k}^1 = (I - K_k^1 C) A_1 \hat{x}_{k-1|k-1}^1 + K_k^1 y_k^1, \quad (\text{A.9})$$

$$\hat{x}_{k|k}^2 = (I - K_k^2) A_2 \hat{x}_{k-1|k-1}^2 + K_k^2 \hat{x}_{k|k}^{1 \rightarrow 2}, \quad (\text{A.10})$$

Using (A.9) and (A.10) we can write the recursion for the augmented state  $\xi_k$  as

$$\xi_k = A_k \xi_{k-1} + B_k^1 y_k^1 + B_k^2 \hat{x}_{k|k}^{1 \rightarrow 2} \quad (\text{A.11})$$

where

$$A_k \triangleq \begin{bmatrix} (I - K_k^1 C) A_1 & 0 \\ 0 & (I - K_k^2) A_2 \end{bmatrix}, \quad B_k^1 \triangleq \begin{bmatrix} K_k^1 \\ 0 \end{bmatrix}, \quad B_k^2 \triangleq \begin{bmatrix} 0 \\ K_k^2 \end{bmatrix}.$$

Since the cost function  $J(\cdot)$  has stochastic terms, namely  $y_{k+1}^1, y_{k+2}^1, \dots, y_{k+N-1}^1$ , our first aim is to get rid of expectations inside the cost function and to eliminate these random terms. Defining  $u_k \triangleq \hat{x}_{k|k}^{1 \rightarrow 2}$ . Then the equation (A.11) becomes

$$\xi_k = A_k \xi_{k-1} + B_k^1 y_k^1 + B_k^2 u_k \quad (\text{A.12})$$

The augmented state at time  $k + 1$  can be written as:

$$\xi_{k+1} = A_{k+1} \xi_k + B_{k+1}^1 y_{k+1}^1 + B_{k+1}^2 u_{k+1}. \quad (\text{A.13})$$

Substituting  $\xi_k$  in (A.12) into (A.13), the augmented state at time  $k + 1$  becomes:

$$\xi_{k+1} = A_{k+1} (A_k \xi_{k-1} + B_k^1 y_k^1 + B_k^2 u_k) + B_{k+1}^1 y_{k+1}^1 + B_{k+1}^2 u_{k+1} \quad (\text{A.14})$$

$$\begin{aligned} &= A_{k+1} A_k \xi_{k-1} + A_{k+1} (B_k^1 y_k^1 + B_k^2 u_k) + B_{k+1}^1 y_{k+1}^1 \\ &\quad + B_{k+1}^2 u_{k+1} \end{aligned} \quad (\text{A.15})$$

Similarly, the augmented state at time  $k + 2$  can be written as follows.

$$\xi_{k+2} = A_{k+2}\xi_{k+1} + B_{k+2}^1 y_{k+2}^1 + B_{k+2}^2 u_{k+2} \quad (\text{A.16})$$

When substituting  $\xi_{k+1}$  in (A.15) into (A.16), the augmented state becomes:

$$\begin{aligned} \xi_{k+2} &= A_{k+2}(A_{k+1}A_k\xi_{k-1} + A_{k+1}(B_k^1 y_k^1 + B_k^2 u_k) + B_{k+1}^1 y_{k+1}^1 \\ &\quad + B_{k+1}^2 u_{k+1}) + B_{k+2}^1 y_{k+2}^1 + B_{k+2}^2 u_{k+2} \end{aligned} \quad (\text{A.17})$$

$$\begin{aligned} &= A_{k+2}A_{k+1}A_k\xi_{k-1} + A_{k+2}A_{k+1}(B_k^1 y_k^1 + B_k^2 u_k) \\ &\quad + A_{k+2}(B_{k+1}^1 y_{k+1}^1 + B_{k+1}^2 u_{k+1}) + B_{k+2}^1 y_{k+2}^1 + B_{k+2}^2 u_{k+2}. \end{aligned} \quad (\text{A.18})$$

These equations can be generalized as follows.

$$\begin{aligned} \xi_{k+n} &= (A_{k+n} \dots A_{k+1}A_k)\xi_{k-1} + (A_{k+n} \dots A_{k+2}A_{k+1})(B_k^1 y_k^1 + B_k^2 u_k) \\ &\quad + \dots + (B_{k+n}^1 y_{k+n}^1 + B_{k+n}^2 u_{k+n}) \end{aligned} \quad (\text{A.19})$$

$$= A_{k+n:k}\xi_{k-1} + \sum_{i=0}^n A_{k+n:k+i+1}(B_{k+i}^1 y_{k+i}^1 + B_{k+i}^2 u_{k+i}) \quad (\text{A.20})$$

where  $A_{k+n:k} \triangleq A_{k+n} \dots A_{k+1}A_k$ . Hence we have

$$\xi_{k+l} = A_{k+l:k}\xi_{k-1} + \sum_{i=0}^l A_{k+l:k+i+1}(B_{k+i}^1 y_{k+i}^1 + B_{k+i}^2 u_{k+i}) \quad (\text{A.21})$$

We can write (A.21) in matrix form as follows.

$$\xi_{k+l} = A_{k+l:k}\xi_{k-1} + \bar{B}_l^1 Y + \bar{B}_l^2 \Gamma \quad (\text{A.24})$$

where

$$\bar{B}_l^1 \triangleq [A_{k+l:k+1}B_k^1 \quad A_{k+l:k+2}B_{k+1}^1 \quad \dots \quad A_{k+l}B_{k+l-1}^1 \quad B_{k+l}^1 \quad 0_m \quad \dots \quad 0_m]$$

$$\bar{B}_l^2 \triangleq [A_{k+l:k+1}B_k^2 \quad A_{k+l:k+2}B_{k+1}^2 \quad \dots \quad A_{k+l}B_{k+l-1}^2 \quad B_{k+l}^2 \quad 0_n \quad \dots \quad 0_n].$$

$$Y \triangleq \begin{bmatrix} y_k^1 \\ y_{k+1}^1 \\ \vdots \\ y_{k+l}^1 \\ \vdots \\ y_{k+N-1}^1 \end{bmatrix}, \quad \Gamma \triangleq \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+l} \\ \vdots \\ u_{k+N-1} \end{bmatrix}$$

$\bar{B}_l^1$  and  $\bar{B}_l^2$  have  $N$  partitions first  $l+1$  of which are nonzero.  $Y$  and  $\Gamma$  are the measurement matrix of the local agent and the quantized state matrix respectively.

Substituting  $y_{k+l}^1 = Cx_{k+l} + v_{k+l}^1$  into the measurement matrix  $Y$ , we get

$$Y = \begin{bmatrix} y_k^1 \\ y_{k+1}^1 \\ \vdots \\ y_{k+l}^1 \\ \vdots \\ y_{k+N-1}^1 \end{bmatrix} = \begin{bmatrix} y_k^1 \\ Cx_{k+1} + v_{k+1}^1 \\ \vdots \\ Cx_{k+l} + v_{k+l}^1 \\ \vdots \\ Cx_{k+N-1} + v_{k+N-1}^1 \end{bmatrix}. \quad (\text{A.25})$$

Note that the first partition of  $Y$ , i.e.,  $y_k^1$ , in (A.25) has not been changed since  $y_k^1$  is given in the conditioning of the expected value in the cost  $J(\cdot)$ . The expression (A.26) can be written as follows.

$$Y = \begin{bmatrix} y_k^1 \\ Cx_{k+1} + v_{k+1}^1 \\ \vdots \\ Cx_{k+l} + v_{k+l}^1 \\ \vdots \\ Cx_{k+N-1} + v_{k+N-1}^1 \end{bmatrix} = \begin{bmatrix} y_k^1 \\ Cx_{k+1} \\ \vdots \\ Cx_{k+l} \\ \vdots \\ Cx_{k+N-1} \end{bmatrix} + \begin{bmatrix} 0 \\ v_{k+1}^1 \\ \vdots \\ v_{k+l}^1 \\ \vdots \\ v_{k+N-1}^1 \end{bmatrix} = \begin{bmatrix} y_k^1 \\ \bar{C}\bar{X}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ \bar{V}_1 \end{bmatrix} \quad (\text{A.26})$$

where

$$\bar{X}_1 = \begin{bmatrix} x_{k+1} \\ x_{k+2} \\ \vdots \\ x_{k+l} \\ \vdots \\ x_{k+N-1} \end{bmatrix} \text{ and } \bar{V}_1 = \begin{bmatrix} v_{k+1}^1 \\ v_{k+2}^1 \\ \vdots \\ v_{k+l}^1 \\ \vdots \\ v_{k+N-1}^1 \end{bmatrix}$$

The vectors  $\bar{X}_1$  and  $\bar{V}_1$  are the state and measurement noise matrix of the local agent.

The matrix  $\bar{C}$  is defined as

$$\bar{C} = \text{blkdiag}(C)_{N-1 \times N-1}.$$

For the sake of simplicity, we now define

$$\hat{X} \triangleq \begin{bmatrix} y_k^1 \\ \bar{C}\bar{X}_1 \end{bmatrix}, \quad \hat{V} \triangleq \begin{bmatrix} 0 \\ \bar{V}_1 \end{bmatrix}.$$

which enables us to write (A.26) as shown below.

$$Y = \begin{bmatrix} y_k^1 \\ \bar{C}\bar{X}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ \bar{V}_1 \end{bmatrix} = \hat{X} + \hat{V} \quad (\text{A.27})$$

## A.2 Calculation of the Expected Values

We now calculate the expectation in the cost function given in (A.5).

$$\begin{aligned} \|\xi_{k+l}\|_{\Omega}^2 &\triangleq \xi_{k+l}^T \Omega \xi_{k+l} \\ &= (A_{k+l:k} \xi_{k-1} + \bar{B}_l^1 Y + \bar{B}_l^2 \Gamma)^T \Omega (A_{k+l:k} \xi_{k-1} + \bar{B}_l^1 Y + \bar{B}_l^2 \Gamma) \end{aligned} \quad (\text{A.28})$$

$$\begin{aligned} &= (\xi_{k-1}^T (A_{k+l:k})^T + Y^T (\bar{B}_l^1)^T + \Gamma^T (\bar{B}_l^2)^T) \Omega (A_{k+l:k} \xi_{k-1} \\ &\quad + \bar{B}_l^1 Y + \bar{B}_l^2 \Gamma) \end{aligned} \quad (\text{A.29})$$

$$\begin{aligned} &= \xi_{k-1}^T (A_{k+l:k})^T \Omega (A_{k+l:k}) \xi_{k-1} \\ &\quad + 2 \xi_{k-1}^T (A_{k+l:k})^T \Omega (\bar{B}_l^1 Y + \bar{B}_l^2 \Gamma) \\ &\quad + (\bar{B}_l^1 Y + \bar{B}_l^2 \Gamma)^T \Omega (\bar{B}_l^1 Y + \bar{B}_l^2 \Gamma). \end{aligned} \quad (\text{A.30})$$

The last expression in (A.30) can also be expanded into three terms as shown below.

$$\begin{aligned} \|\xi_{k+l}\|_Q^2 &= \xi_{k-1}^T (A_{k+l:k})^T \Omega (A_{k+l:k}) \xi_{k-1} \\ &\quad + 2 \xi_{k-1}^T (A_{k+l:k})^T \Omega (\bar{B}_l^1 Y + \bar{B}_l^2 \Gamma) \\ &\quad + Y^T (\bar{B}_l^1)^T \Omega \bar{B}_l^1 Y + 2 Y^T (\bar{B}_l^1)^T \Omega \bar{B}_l^2 \Gamma \\ &\quad + \Gamma^T (\bar{B}_l^2)^T \Omega \bar{B}_l^2 \Gamma. \end{aligned} \quad (\text{A.31})$$

Substituting  $\dot{X} + \dot{V}$  for the measurement matrix  $Y$ , we get

$$\begin{aligned} \|\xi_{k+l}\|_Q^2 &= \xi_{k-1}^T (A_{k+l:k})^T \Omega A_{k+l:k} \xi_{k-1} \\ &\quad + 2 \xi_{k-1}^T (A_{k+l:k})^T \Omega (\bar{B}_l^1 (\dot{X} + \dot{V}) + \bar{B}_l^2 \Gamma) \\ &\quad + (\dot{X} + \dot{V})^T (\bar{B}_l^1)^T \Omega \bar{B}_l^1 (\dot{X} + \dot{V}) \\ &\quad + 2 (\dot{X} + \dot{V})^T (\bar{B}_l^1)^T \Omega \bar{B}_l^2 \Gamma + \Gamma^T (\bar{B}_l^2)^T \Omega \bar{B}_l^2 \Gamma. \end{aligned} \quad (\text{A.32})$$

Taking the expectation of both sides of (A.32) condition on the measurements  $y_{0:k}^1$ , we obtain

$$\begin{aligned} E[\|\xi_{k+l}\|_{\Omega}^2 | y_{0:k}^1] &= E[\xi_{k-1}^T (A_{k+l:k})^T \Omega A_{k+l:k} \xi_{k-1} | y_{0:k}^1] \\ &\quad + E[2 \xi_{k-1}^T (A_{k+l:k})^T \Omega (\bar{B}_l^1 (\dot{X} + \dot{V}) + \bar{B}_l^2 \Gamma) | y_{0:k}^1] \\ &\quad + E[(\dot{X} + \dot{V})^T (\bar{B}_l^1)^T \Omega \bar{B}_l^1 (\dot{X} + \dot{V}) | y_{0:k}^1] \\ &\quad + E[2 (\dot{X} + \dot{V})^T (\bar{B}_l^1)^T \Omega \bar{B}_l^2 \Gamma | y_{0:k}^1] \\ &\quad + E[\Gamma^T (\bar{B}_l^2)^T \Omega \bar{B}_l^2 \Gamma | y_{0:k}^1]. \end{aligned} \quad (\text{A.33})$$

The only random terms in (A.33) are  $\dot{X}$  and  $\dot{V}$ . Hence the expectations of the other terms are equal to themselves, which gives

$$\begin{aligned}
E \left[ \left\| \xi_{k+l} \right\|_{\Omega}^2 \middle| y_{0:k}^1 \right] &= \xi_{k-1}^T (A_{k+l:k})^T \Omega A_{k+l:k} \xi_{k-1} \\
&+ 2 \xi_{k-1}^T (A_{k+l:k})^T \Omega (\bar{B}_l^1 E[\dot{X} + \dot{V} | y_{0:k}^1] + \bar{B}_l^2 \Gamma) \\
&+ E \left[ (\dot{X} + \dot{V})^T (\bar{B}_l^1)^T \Omega \bar{B}_l^1 (\dot{X} + \dot{V}) \middle| y_{0:k}^1 \right] \\
&+ E \left[ 2(\dot{X} + \dot{V})^T \middle| y_{0:k}^1 \right] (\bar{B}_l^1)^T \Omega \bar{B}_l^2 \Gamma + \Gamma^T (\bar{B}_l^2)^T \Omega \bar{B}_l^2 \Gamma. \quad (\text{A.34})
\end{aligned}$$

Calculating the first and the third expectations in (A.34) gives the following.

$$\begin{aligned}
E \left[ \left\| \xi_{k+l} \right\|_{\Omega}^2 \middle| y_{0:k}^1 \right] &= \xi_{k-1}^T (A_{k+l:k})^T \Omega A_{k+l:k} \xi_{k-1} \\
&+ 2 \xi_{k-1}^T (A_{k+l:k})^T \Omega (\bar{B}_l^1 \bar{X} + \bar{B}_l^2 \Gamma) \\
&+ E \left[ (\dot{X} + \dot{V})^T (\bar{B}_l^1)^T \Omega \bar{B}_l^1 (\dot{X} + \dot{V}) \middle| y_{0:k}^1 \right] \\
&+ 2 \bar{X}^T (\bar{B}_l^1)^T \Omega \bar{B}_l^2 \Gamma + \Gamma^T (\bar{B}_l^2)^T \Omega \bar{B}_l^2 \Gamma \quad (\text{A.35})
\end{aligned}$$

where

$$\bar{X} = E[\dot{X} | y_{0:k}^1] \quad (\text{A.36})$$

$$= E \left[ \begin{bmatrix} y_k^1 \\ \bar{C} \bar{X}_1 \end{bmatrix} \middle| y_{0:k}^1 \right] \quad (\text{A.37})$$

$$= E \left[ \begin{bmatrix} y_k^1 \\ C x_{k+1} \\ \vdots \\ C x_{k+l} \\ \vdots \\ C x_{k+N-1} \end{bmatrix} \middle| y_{0:k}^1 \right] \quad (\text{A.38})$$

$$= \begin{bmatrix} y_k^1 \\ CE[x_{k+1} | y_{0:k}^1] \\ \vdots \\ CE[x_{k+l} | y_{0:k}^1] \\ \vdots \\ CE[x_{k+N-1} | y_{0:k}^1] \end{bmatrix} \quad (\text{A.39})$$

$$\begin{aligned}
&= \begin{bmatrix} y_k^1 \\ C\hat{x}_{k+1|k}^1 \\ \vdots \\ C\hat{x}_{k+l|k}^1 \\ \vdots \\ C\hat{x}_{k+N-1|k}^1 \end{bmatrix} \\
&= \begin{bmatrix} y_k^1 \\ \hat{y}_{k+1|k}^1 \\ \vdots \\ \hat{y}_{k+l|k}^1 \\ \vdots \\ \hat{y}_{k+N-1|k}^1 \end{bmatrix}
\end{aligned} \tag{A.40}$$

The second expectation in (A.35) can be calculated as follows.

$$\begin{aligned}
E \left[ (\hat{X} + \hat{V})^T (\bar{B}_l^1)^T \Omega \bar{B}_l^1 (\hat{X} + \hat{V}) | y_{0:k}^1 \right] &= E \left[ (\hat{X})^T (\bar{B}_l^1)^T \Omega \bar{B}_l^1 (\hat{X}) | y_{0:k}^1 \right] \\
&\quad + E \left[ (\hat{X})^T (\bar{B}_l^1)^T \Omega \bar{B}_l^1 (\hat{V}) | y_{0:k}^1 \right] \\
&\quad + E \left[ (\hat{V})^T (\bar{B}_l^1)^T \Omega \bar{B}_l^1 (\hat{X}) | y_{0:k}^1 \right] \\
&\quad + E \left[ (\hat{V})^T (\bar{B}_l^1)^T \Omega \bar{B}_l^1 (\hat{V}) | y_{0:k}^1 \right].
\end{aligned} \tag{A.41}$$

Since the term  $(\hat{X} + \hat{V})^T (\bar{B}_l^1)^T \Omega \bar{B}_l^1 (\hat{X} + \hat{V})$  is a scalar, the trace of this term is equal to itself. Therefore the first term in (A.41) can be written as follows.

$$E \left[ (\hat{X})^T (\bar{B}_l^1)^T \Omega \bar{B}_l^1 (\hat{X}) | y_{0:k}^1 \right] = E \left[ \text{tr}((\hat{X})^T (\bar{B}_l^1)^T \Omega \bar{B}_l^1 (\hat{X})) | y_{0:k}^1 \right] \tag{A.42}$$

$$= E \left[ \text{tr} \left( (\bar{B}_l^1)^T \Omega \bar{B}_l^1 (\hat{X}) (\hat{X})^T \right) | y_{0:k}^1 \right] \tag{A.43}$$

$$= \text{tr} \left( (\bar{B}_l^1)^T \Omega \bar{B}_l^1 E \left[ (\hat{X}) (\hat{X})^T | y_{0:k}^1 \right] \right) \tag{A.44}$$

$$= \text{tr}((\bar{B}_l^1)^T \Omega \bar{B}_l^1 \bar{P}) \tag{A.45}$$

where

$$\bar{P} = E \left[ (\hat{X}) (\hat{X})^T | y_{0:k}^1 \right] \tag{A.46}$$

$$= E \left( \begin{bmatrix} y_k^1 \\ Cx_{k+1} \\ \vdots \\ Cx_{k+l} \\ \vdots \\ Cx_{k+N-1} \end{bmatrix} [(y_k^1)^T \quad x_{k+1}^T C^T \quad \cdots \quad x_{k+N-1}^T C^T] | y_{0:k}^1 \right) \quad (\text{A.47})$$

$$= E \left[ \begin{bmatrix} y_k^1 (y_k^1)^T & y_k^1 x_{k+1}^T C^T & \cdots & y_k^1 x_{k+N-1}^T C^T \\ Cx_{k+1} (y_k^1)^T & Cx_{k+1} x_{k+1}^T C^T & \cdots & Cx_{k+1} x_{k+N-1}^T C^T \\ \vdots & \vdots & \ddots & \vdots \\ Cx_{k+N-1} (y_k^1)^T & Cx_{k+N-1} x_{k+1}^T C^T & \cdots & Cx_{k+N-1} x_{k+N-1}^T C^T \end{bmatrix} | y_{0:k}^1 \right]. \quad (\text{A.48})$$

Considering the components of the matrix in (A.48) one by one, we get

- $E[y_k^1 (y_k^1)^T | y_{0:k}^1] = y_k^1 y_k^1{}^T$
- $E[Cx_{k+i} (y_k^1)^T | y_{0:k}^1] = CE[x_{k+i} | y_{0:k}^1] (y_k^1)^T$   
 $= C\hat{x}_{k+i|k}^1 (y_k^1)^T = \hat{y}_{k+i|k}^1 (y_k^1)^T$
- $E[Cx_{k+1} x_{k+i}^T C^T | y_{0:k}^1] = CE[x_{k+1} x_{k+i}^T | y_{0:k}^1] C^T$   
 $= CE[x_{k+1} (A^{n-1} x_{k+1} + \sum \beta w_k)^T | y_{0:k}^1] C^T$   
 $= CE[x_{k+1} x_{k+1}^T (A^{n-1})^T + x_{k+1} (\sum \beta w_k)^T | y_{0:k}^1] C^T$   
 $= C(P_{k+1|k}^1 + \hat{x}_{k+1|k}^1 (\hat{x}_{k+1|k}^1)^T) (A^{n-1})^T C^T.$

Note that the last equality can be written because  $E[x_{k+1} (\sum \beta w_k)^T | y_{0:k}^1]$  is '0' since  $x_{k+1}$  and the process noise terms in  $\sum \beta w_k$  are independent and the expectations of the process noise terms are zero. The general expression is given as

$$E[Cx_{k+i} x_{k+j}^T C^T | y_{0:k}^1] = CE[x_{k+i} x_{k+j}^T | y_{0:k}^1] C^T \quad (\text{A.49})$$

$$= C(P_{\min(k+i, k+j)|k}^1 + \hat{x}_{\min(k+i, k+j)|k}^1 (\hat{x}_{\min(k+i, k+j)|k}^1)^T) (A^{|i-j|})^T C^T. \quad (\text{A.50})$$

Continuing with the second term of (A.41) we write



$$\begin{aligned}
E \left[ (\hat{X})^T (\bar{B}_l^1)^T \Omega \bar{B}_l^1 (\hat{V}) | y_{0:k}^1 \right] &= E \left[ \text{tr}((\hat{X})^T (\bar{B}_l^1)^T \Omega \bar{B}_l^1 (\hat{V})) | y_{0:k}^1 \right] \\
&= E \left[ \text{tr}((\bar{B}_l^1)^T \Omega \bar{B}_l^1 (\hat{V}) (\hat{X})^T) | y_{0:k}^1 \right] \\
&= \text{tr} \left( (\bar{B}_l^1)^T \Omega \bar{B}_l^1 E \left[ (\hat{V}) (\hat{X})^T | y_{0:k}^1 \right] \right) \\
&= 0
\end{aligned}$$

where the last equality can be written since  $\hat{V}$  and  $\hat{X}$  matrices are independent and the expectations of the measurement noise terms are zero. Similarly, the third term of (A.41), i.e.,  $E \left[ (\hat{V})^T (\bar{B}_l^1)^T \Omega \bar{B}_l^1 (\hat{X}) | y_{0:k}^1 \right]$ , is zero.

The fourth term of (A.41) can be calculated as follows.

$$E \left[ (\hat{V})^T (\bar{B}_l^1)^T \Omega \bar{B}_l^1 (\hat{V}) | y_{0:k}^1 \right] = E \left[ \text{tr}((\hat{V})^T (\bar{B}_l^1)^T \Omega \bar{B}_l^1 (\hat{V})) | y_{0:k}^1 \right] \quad (\text{A.51})$$

$$= \text{tr} \left( (\bar{B}_l^1)^T \Omega \bar{B}_l^1 E \left[ (\hat{V}) (\hat{V})^T | y_{0:k}^1 \right] \right) \quad (\text{A.52})$$

$$= \text{tr}((\bar{B}_l^1)^T \Omega \bar{B}_l^1 \bar{R}) \quad (\text{A.53})$$

where

$$\begin{aligned}
\bar{R} &\triangleq E \left[ (\hat{V}) (\hat{V})^T | y_{0:k}^1 \right] \\
&= E \left[ \begin{array}{c|c} \begin{bmatrix} 0_{m,1} \\ v_{k+1} \\ \vdots \\ v_{k+l} \\ \vdots \\ v_{k+N-1} \end{bmatrix} & \begin{bmatrix} 0_{1,m} & v_{k+1}^T & v_{k+2}^T & \cdots & v_{k+N-1}^T \end{bmatrix} \\ \hline & y_{0:k}^1 \end{array} \right] \\
&= E \left( \begin{array}{cccc|c} 0_m & 0_m & \cdots & 0_m & \\ 0_m & v_{k+1} v_{k+1}^T & \cdots & v_{k+1} v_{k+N-1}^T & \\ \vdots & \vdots & \ddots & \vdots & \\ 0_m & v_{k+N-1} v_{k+1}^T & \cdots & v_{k+N-1} v_{k+N-1}^T & \\ \hline & & & & y_{0:k}^1 \end{array} \right) \\
&= \begin{bmatrix} 0_m & 0_m & \cdots & 0_m \\ 0_m & R & \cdots & 0_m \\ \vdots & \vdots & \ddots & \vdots \\ 0_m & 0_m & \cdots & R \end{bmatrix}.
\end{aligned}$$

As a result, the expectation in (A.41) can be written as shown below.

$$E \left[ (\dot{X} + \dot{V})^T (\bar{B}_l^1)^T \Omega \bar{B}_l^1 (\dot{X} + \dot{V}) \right] = \text{tr}((\bar{B}_l^1)^T \Omega \bar{B}_l^1 \bar{P}) + \text{tr}((\bar{B}_l^1)^T \Omega \bar{B}_l^1 \bar{R}) \quad (\text{A.54})$$

Finally, substituting the result (A.54) into the expectation in (A.36) we can write

$$\begin{aligned} E \left[ \|\xi_{k+l}\|_{\Omega}^2 \middle| y_{0:k}^1 \right] &= \xi_{k-1}^T (A_{k+l:k})^T \Omega A_{k+l:k} \xi_{k-1} \\ &\quad + 2\xi_{k-1}^T (A_{k+l:k})^T \Omega (\bar{B}_l^1 \bar{X} + \bar{B}_l^2 \Gamma) + \text{tr}((\bar{B}_l^1)^T \Omega \bar{B}_l^1 \bar{P}) \\ &\quad + \text{tr}((\bar{B}_l^1)^T \Omega \bar{B}_l^1 \bar{R}) + 2\bar{X}^T (\bar{B}_l^1)^T \Omega \bar{B}_l^2 \Gamma + \Gamma^T (\bar{B}_l^2)^T \Omega \bar{B}_l^2 \Gamma \end{aligned} \quad (\text{A.55})$$

### A.3 Final form of the Cost Function

We can now separate the terms which are independent of the quantized states as follows.

$$\begin{aligned} E \left[ \|\xi_{k+l}\|_{\Omega}^2 \middle| y_{0:k}^1 \right] &= 2\xi_{k-1}^T (A_{k+l:k})^T \Omega \bar{B}_l^2 \Gamma + 2\bar{X}^T (\bar{B}_l^1)^T \Omega \bar{B}_l^2 \Gamma \\ &\quad + \Gamma^T (\bar{B}_l^2)^T \Omega \bar{B}_l^2 \Gamma + J_l \end{aligned} \quad (\text{A.53})$$

where the term  $J_l$  which is independent of the quantized states is defined as follows.

$$\begin{aligned} J_l &\triangleq \xi_{k-1}^T (A_{k+l:k})^T \Omega A_{k+l:k} \xi_{k-1} + 2\xi_{k-1}^T (A_{k+l:k})^T \Omega \bar{B}_l^1 \bar{X} \\ &\quad + \text{tr}((\bar{B}_l^1)^T \Omega \bar{B}_l^1 \bar{P}) + \text{tr}((\bar{B}_l^1)^T \Omega \bar{B}_l^1 \bar{R}). \end{aligned} \quad (\text{A.54})$$

Note that although the term  $J_l$  is independent of the quantized states, it is still dependent on the quantized covariances. Hence for the general optimization problem where the quantized covariances are also unknown, the terms  $J_l$  will be necessary.

By using (A.53), the cost function  $J(\cdot)$  in (A.5) can be written as follows.

$$J(\cdot) = \sum_{l=0}^{N-1} E \left[ \|\xi_{k+l}\|_{\Omega}^2 \middle| y_{0:k}^1 \right] \quad (\text{A.55})$$

$$= \sum_{l=0}^{N-1} 2\xi_{k-1}^T (A_{k+l:k})^T \Omega \bar{B}_l^2 \Gamma + 2\bar{X}^T (\bar{B}_l^1)^T \Omega \bar{B}_l^2 \Gamma + \Gamma^T (\bar{B}_l^2)^T \Omega \bar{B}_l^2 \Gamma + J_l \quad (\text{A.56})$$

$$= \Gamma^T W \Gamma + 2\Gamma_A^T \Gamma + \Lambda. \quad (\text{A.57})$$

where

$$W \triangleq \sum_{l=0}^{N-1} (\bar{B}_l^2)^T \Omega \bar{B}_l^2 \quad (\text{A.58})$$

$$\Gamma_A^T \triangleq \sum_{l=0}^{N-1} (\xi_{k-1}^T (A_{k+l:k})^T + \bar{X}^T (\bar{B}_l^1)^T) \Omega \bar{B}_l^2 \quad (\text{A.59})$$

$$\Lambda \triangleq \sum_{l=0}^{N-1} J_l \quad (\text{A.60})$$

$$\triangleq \sum_{l=0}^{N-1} \left( \begin{array}{l} \xi_{k-1}^T (A_{k+l:k})^T \Omega A_{k+l:k} \xi_{k-1} + 2 \xi_{k-1}^T (A_{k+l:k})^T \Omega \bar{B}_l^1 \bar{X} \\ + \text{tr}((\bar{B}_l^1)^T \Omega \bar{B}_l^1 \bar{P}) + \text{tr}((\bar{B}_l^1)^T \Omega \bar{B}_l^1 \bar{R}) \end{array} \right) \quad (\text{A.61})$$