Transmission Rate Compression Based on Kalman Filter Using Spatio-temporal Correlation for Wireless Sensor Networks

Dem Fachbereich Physik und Elektrotechnik
Der Universität Bremen

Dissertation zur Erlangung des akademischen Grades eines
Doktor-Ingenieurin (Dr.-Ing.)

von

M.Sc. Yanqiu Huang

Referent:	Prof. Dr.-Ing. Alberto García-Ortiz
Korreferentin:	Prof. Dr. Anna Förster
Eingereicht am:	15.12.2016
Tag des Promotionskolloquiums:	24.01.2017

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my doctoral supervisor Prof. Dr.-Ing. Alberto García-Ortiz. I feel so lucky that I have such a wonderful supervisor at the beginning of my research career. His infinite patience, careful guidance and significant advice inspire me and improve me in my past four years' study. I have learned from him not only the technique stuff, but also the way to think, the kindness to student and the enthusiasm for research. These are the treasures for my future career.

I would also like to express my appreciation to Prof. Dr. Anna Förster. She accepts to review my dissertation in a very urgent situation and finishes the review very fast. Many thanks to Prof. Dr.-Ing. Axel Gräser, who spends time to discuss and understand my work, and Prof. Dr.-Ing. Steffen Paul, who helps me with my postdoc application. I also appreciate the talk with Prof. Dr. Rainer Laur, who is sagacious, admirable and full of experience.

Our research group is like a family to me. Everybody is very agreeable and helpful. Kerstin Janssen, a kind-hearted and beautiful women, has spared no effort with the administration work, the language problems and daily life supports to ease my burden of research irrelevant stuff. My colleagues Lennart Bamberg, Wolfgang Büter, Amir Najafi, Ardalan Najafi, Ayad Dalloo and Peter Lutzen, have granted me many helps, fun, happiness and encouragement. Sincere gratitude also to colleagues Christof Osewold, Daniel Gregorek, Parham Haririan, Jochen Rust, Janpeter Höffmann, Jonas Pistor, Maike Schröder, who have helped me to integrate into the institute, I really enjoy every moment with them. In addition, I really appreciate my friends who helped me a lot in the abroad daily life. And I gratefully acknowledge the financial support from China Scholarship Council.

Above all, I am deeply grateful for my parents. Thank you for giving me enough space to grow up, supporting me to insist on what I would like to do and tolerating the mistakes I made. To my fiancé Wanli Yu, I am so grateful for meeting you in my life and always having you by my side.

Abstract

Wireless sensor networks (WSNs) composed of spatially distributed autonomous sensor nodes have been applied to a wide variety of applications. Due to the limited energy budget of the sensor nodes and long-term operation requirement of the network, energy efficiency is a primary concern in almost any application. Radio communication, known as one of the most expensive processes, can be suppressed thanks to the temporal and spatial correlations. However, it is a challenge to compress the communication as much as possible, while reconstructing the system state with the highest quality.

This work proposes the PKF method to compress the transmission rate for cluster based WSNs, which combines a k-step ahead Kalman predictor with a Kalman filter (KF). It provides the optimal reconstruction solution based on the compressed information of a single node for a linear system. Instead of approximating the noisy raw data, PKF aims to reconstruct the internal state of the system. It achieves data filtering, state estimation, data compression and reconstruction within one KF framework and allows the reconstructed signal based on the compressed transmission to be even more precise than transmitting all of the raw measurements without processing.

The second contribution is the detailed analysis of PKF. It not only characterizes the effect of the system parameters on the performance of PKF but also supplies a common framework to analyze the underlying process of prediction-based schemes. The transmission rate and reconstruction quality are functions of the system parameters, which are calculated with the aid of (truncated) multivariate normal (MVN) distribution. The transmission of the node using PKF not only determines the current optimal estimate of the system state, but also indicates the range and the transmission probability of the k-step ahead prediction of the cluster head. Besides, one of the prominent results is an explicit expression for the covariance of the doubly truncated MVN distribution. This is the first work that calculates it using the Hessian matrix of the probability density function of a MVN distribution, which improves the traditional methods using moment-generating function and has generality. This contribution is important for WSNs, but also for other domains, e.g., statistics and economics.

The PKF method is extended to use spatial correlation in multi-nodes systems without any intra-communication or a coordinator based on the above analysis. Each leaf node executes a PKF independently. The reconstruction quality is further improved by the cluster head using the received information, which is equivalent to further reduce the transmission rate of the node under the guaranteed reconstruction quality. The optimal reconstruction solution, called Rand-ST, is obtained, when the cluster head uses the incomplete information by taking the transmission of each node as random. Rand-ST actually solves the KF fusion problem with colored and randomly transmitted observations, which is the first work addressing this problem to the best of our knowledge. It proves the KF with state augment method is more accurate than the measurement differencing approach in this scenario. The suboptimality of Rand-ST by neglecting the useful information is analyzed, when the transmission of each node is controlled by PKF. The heuristic EPKF methods are thereupon proposed to utilize the complete information, while solving the nonlinear problem through linear approximations. Compared with the available techniques, EPKF methods not only ensure an error bound of the reconstruction for each node, but also allow them to report the emergency event in time, which avoids the loss of penitential important information.

The proposed approaches are firstly evaluated using simulated systems to observe how far the reconstructions are from the real states. Then the real WSN datasets are used to compare the performance of the approaches with other techniques. Besides, the proposed approaches are implemented in the WSN Openmotes to study how much communication energy cost can be saved and how much lifetime can be improved.

Kurzfassung

Drahtlose Sensornetzwerke (WSNs), die aus räumlich verteilten autonomen Sensorknoten bestehen, werden bereits für eine Vielzahl von Anwendungen eingesetzt. Aufgrund des begrenzten Energiebudgets der Sensorknoten und der Anforderung einer langfristigen Betriebsdauer des Netzwerks ist Energieeffizienz bei WSNs von besonders hoher Bedeutung. Die Funkkommunikation ist für einen Großteil des Energieverbrauchs eines WSN-Knotens verantwortlich, welcher, unter Ausnutzung der zeitlichen und räumlichen Korrelationen der Datenströme reduziert, werden kann. Die besondere Herausforderung besteht dabei darin, die zu übertragenden Daten so weit wie möglich zu komprimieren, ohne die Systemperformance zu beeinträchtigen.

In dieser Arbeit wird die PKF-Methode zur Reduktion der erforderlichen Übertragungsrate für Cluster-basierte WSNs vorgestellt. Sie kombiniert einen Kalman-Prädiktor mit einem Kalman-Filter (KF). Die Methode liefert eine optimale Rekonstruktionslösung, basierend auf der komprimierten Information eines Knotens in einem linearen System. Der Ansatz der PKF-Methode ist es, den internen Zustand des Systems zu rekonstruieren, statt die verrauschten Rohdaten zu approximieren. Die Methode führt die Datenfilterung, Zustandsschätzung, Datenkompression und Rekonstruktion innerhalb eines KF-Frameworks aus und ermöglicht, dass das auf der Grundlage der komprimierten Übertragung rekonstruierte Signal genauer ist als bei der Übertragung aller nicht aufbereiteten Rohmessungen.

Ein weiterer Teil dieser Arbeit beinhaltet die detaillierte Analyse der PKF-Methode. Die Analyse charakterisiert nicht nur die Wirkung der Systemparameter auf die Leistungsfähigkeit der PKF, sondern sie liefert auch ein einheitliches Framework für die Analyse des zugrundeliegenden Prozesses der Prädiktor-basierten Ansätze. Die Übertragungsrate und die Rekonstruktionsqualität sind abhängig von den Systemparametern, die mit Hilfe der (beschränkten) mehrdimensionalen Normalverteilung (MVN) berechnet werden. Die Datenübertragung des Knotens unter Anwendung der PKF-Methode bestimmt nicht nur die aktuell beste Einschätzung der Systemperformance, sondern auch die Weite und die Übertragungswahrscheinlichkeit des dynamischen Prädiktors im Cluster-Head. Zudem ist ein bedeutendes Ergebnis dieser Arbeit ein expliziter Ausdruck für die Kovarianz der

zweifach beschränkten mehrdimensionalen Normalverteilung. Eine Literatur-Recherchen ergab, dass die vorliegende Arbeit die erste ist, welche die Hesse-Matrix der Wahrscheinlichkeitsdichtefunktion einer MVN-Verteilung für die Berechnung nutzt, die herkömmlichen Verfahren (welche die Momenterzeugende Funktion nutzen) verbessert und zudem allgemeingültig ist. Dieses hat für WSNs, aber auch für andere Bereiche (z. B. aus Statistik und Wirtschaft), eine große Bedeutung.

Weiterhin erweitert diese Arbeit das PKF-Verfahren, so dass die präsente räumliche Korrelation in einem Mehrknotensystem ausgenutzt wird, ohne dafür jegliche clusterinterne Kommunikation oder Koordination (basierend auf der zuvor beschriebenen Analyse) zu verwenden. Jeder Sensorknoten führt unabhängig eine PKF aus. Die Rekonstruktionsqualität wird durch den Cluster-Head, unter Verwendung der empfangenen Informationen, weiter verbessert, was einer weiteren Reduktion der Übertragungsrate des Knotens unter Einhaltung der garantierten Rekonstruktionsqualität entspricht. Die optimale Rekonstruktionslösung, genannt Rand-ST, wird erreicht, wenn der Cluster-Head die unvollständigen Informationen verwendet, indem er die Übertragung eines jeden Knotens als zufällig annimmt. Die Rand-ST löst eigentlich das KF-Fusionsproblem mit farbigen und zufällig gesendeten Daten-Sampeln. Eine Literatur-Recherche ergab, dass dies die erste Arbeit ist, welche diese Problematik untersucht. Die Ergebnisse zeigen auf, dass der KF in Kombination mit der State-Augment-Methode im untersuchten Szenario genauer ist als der Ansatz der Differenzen-Messung. Aufgrund der Vernachlässigung relevanter Informationen tritt eine Suboptimalität bei der Rand-ST-Lösung auf. Diese wird unter der Annahme, dass die Übertragung über alle Sensorknoten mittels einer PKF gesteuert wird, analysiert. Anhand dieser Analyse zeigt die vorliegende Arbeit die Notwendigkeit der Verwendung der heuristischen EPKF-Methoden. Die EPKF-Methoden ermöglichen es, den kompletten Informationsgehalt auszuschöpfen und gleichzeitig das Problem der Nicht-Linearität durch eine Approximation ersten Grades zu lösen. Verglichen mit bisherigen Verfahren stellen die EPKF-Methoden nicht nur eine obere Fehlergrenze für die Daten-Rekonstruktion in jedem Knoten sicher, sondern ermöglichen zudem eine frühzeitige Detektion systemkritischer Ereignisse. Dadurch wird der Verlust besonders relevanter Informationen vermieden.

Die in dieser Arbeit vorgestellten Verfahren werden zunächst anhand einer Simulations-Plattform evaluiert, um zu quantifizieren wie weit die Rekonstruktionen von den ursprünglichen Werten abweichen. Anschließend werden reale WSN-Datenströme verwendet um die vorgestellten Verfahren mit den bisherigen zu vergleichen. Zudem werden die Verfahren in WSN *Openmotes* implementiert, um die Reduktion des Energieverbrauchs und die daraus folgende Erhöhung der Akkulaufzeit zu untersuchen.

Contents

1 Introduction

Wireless sensor networks (WSNs) consist of spatially distributed and mutually communicated sensor nodes to monitor physical or environmental phenomena [1]. Each node is able to collect information from the surrounding environment with a sensing unit, elaborate this information locally with a processing unit, and communicate with other nodes with a communication unit [2]. The WSN has been considered as one of the most important technologies for the 21st century [3] and has gained much attention from the research and industrial communities in the past decades. This key technology enables a wide range of new applications and services including monitoring of physical environments [4] [5], enhanced industrial control [6] [7], remote health care [8] [9], logistic [10] [11] and so on.

The sensor nodes are usually required to be operational for long periods, ranging from several days in the case of long-term health monitoring, months for supply chain management, and years or even decades for applications such as weather monitoring. However, they are typically battery-powered and it is hard or even impossible to change or recharge batteries due to the large quantities or the harsh physical environments. This would lead to the fragmentation of the network and loss of potentially crucial information. Thus, in almost any application of WSNs, energy efficiency is a primary concern.

This dissertation aims to reduce the energy consumption of sensor nodes by compressing their transmission rates, while providing sufficient information to understand and interpret the monitored systems.

1.1 Motivation

The energy consumption of the sensor nodes typically involves sensing, processing and communication [12]. As widely recognized by the research community, one of the most energy intensive processes of a sensor node is the wireless communication [13]. In a classical architecture for instance, a single bit transmission can consume over 1000 times more energy than a single 32-bit computation [13]. In addition to the energy consumption of data packets transmission, significant energy is also required by overhead activities,

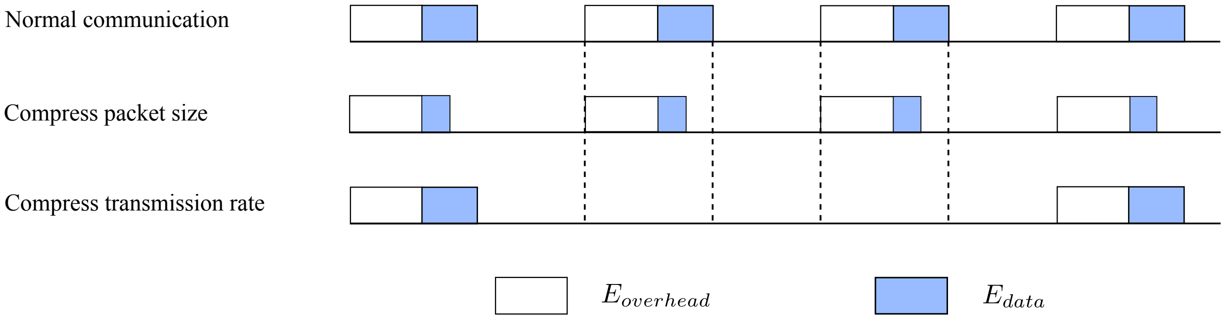


Figure 1.1: Schematic comparison between data packet compression and transmission rate compression.

such as radio start-up, channel accessing, control packets, turnaround, idle listening, overhearing, and collision as analyzed in [14]. Thus, most of the research focuses on developing energy efficient schemes for reducing the communication cost.

Data compression is very attractive due to the inherent existence of spatial and temporal correlation in the physical phenomena [2]. Spatially adjacent sensor nodes have correlated observations and the consecutive measurements of a sensor node are temporal correlated. Exploiting this characteristic can efficiently compress the redundant information. The related algorithms aim to either compress the packet size or the transmission rate as illustrated in Fig. 1.1.

The approaches for packet size compression typically refer to dictionary-based compression [15] [16] or predictive coding [17]. They usually suffer from the growing dictionary or the latency problems depending on the specific techniques. Even if these techniques are able to compress the data size with a high compression ratio, they are incapable of reducing overhead of each transaction which can dominate the energy consumption in some cases [14]. In contrast, the schemes for transmission rate compression [18][19] can decrease the total communication energy cost during the transaction (see Fig. 1.1). Therefore, compressing the transmission rate using spatio-temporal correlation is preferred in this work.

The reduction of the transmission rate leads to a decrease of the reconstruction quality for the monitored system. The problem is how to compress the transmission rate of the sensor nodes as much as possible, while reconstructing the system state with the highest accuracy.

1.2 Main Contribution

This work addresses the above mentioned problem and proposes a communication ratio compression scheme utilizing spatio-temporal correlation for cluster-based WSNs. The main contribution are as follows:

- It provides the optimal reconstruction solution based on the compressed information of a single node for a linear system. The proposed approach, termed as PKF, combines a k-step ahead Kalman predictor with a Kalman filter (KF) to suppress the communication between the leaf node and the cluster head, while reconstructing the system state in the best manner. It achieves data filtering, state estimation, data compression and reconstruction within one KF framework and allows the reconstructed signal based on the compressed transmission to be even more precise than transmitting all of the raw measurements without processing.

- It provides an in-depth mathematical analysis of PKF, which is helpful to understand the underlying process of the scheme and to find the effect of the system parameters on its performance. The transmission rate and reconstruction quality using PKF are calculated with the aid of multivariate normal (MVN) distribution. The transmission of the node not only tells the current optimal estimate of the system state, but also indicates the range and the transmission probability of the k-step ahead prediction of the cluster head. Besides, one of the prominent results is an explicit expression for the covariance of the doubly truncated MVN. We believe this is the first work that calculates it using the Hessian matrix of the probability density function (PDF) of an MVN distribution, which improves the traditional methods using moment generating function and has generality. This contribution is important for WSNs, but also for other domains, e.g., statistics and economics.

- It extends PKF to use spatial correlation in multi-nodes systems without intra-communication based on the above analysis. The optimal reconstruction solution is obtained, called Rand-ST, when the cluster head uses the incomplete information by taking the transmission of each node as random. Rand-ST actually solves the KF fusion problem with colored and randomly transmitted observations, which is the first work that addresses this problem to the best of our knowledge. It proves the KF with state augment method is more accurate than the measurement differencing approach in this scenario. The suboptimality of Rand-ST by neglecting the useful information is analyzed, when the transmission of each node is controlled by PKF.

The heuristic methods are proposed based on Rand-ST, called EPKF-simp, EPKF-norm and EPKF-mix, to utilize the complete information, while solving the nonlinear problem through linear approximations. The reconstruction quality can be improved by using EPKF methods, which is equivalent to further reduce the transmission rate under the guaranteed quality.

- It implements the proposed approaches in the WSN Openmotes. The transmission rate reduction using PKF and the reconstruction quality improvement by further using EPKF are obtained in an arbitrary formed network. The computation energy consumption of PKF and the communication energy consumption are compared by visualizing the current profile on an oscilloscope. Considering the overall per-day current consumption of the leaf node and using the obtained transmission rate, the lifetime improvements are obtained.

1.3 Publications

The related publications of this work include [20, 14, 21, 22, 23, 24, 25] as shown below:

Journal Articles

- Yanqiu Huang, Wanli Yu, Christof Osewold, and Alberto Garcia-Ortiz. Analysis of PKF: A communication cost reduction scheme for wireless sensor networks. *IEEE Transactions on Wireless Communications*, 15(2):843–856, Feb 2016.

- Yanqiu Huang, Wanli Yu, and Alberto Garcia-ortiz. Accurate energy-aware workload distribution for wireless sensor networks using a detailed communication energy cost model. *Journal of Low Power Electronics*, 10(2):183–193(11), June 2014.

- Yanqiu Huang, Wanli Yu, and Alberto Garcia-Ortiz. EPKF: transmission rate compression based on Kalman filter using spatio-temporal correlation for WSNs. *Submitted to IEEE Transactions on Wireless Communications.*

- Wanli Yu, Yanqiu Huang, and Alberto Garcia-Ortiz. An On-line Optimal Distributed Workload Scheduling Algorithm for Wireless Sensor Networks. *Submitted to IEEE Sensors Journal.*

Conference Proceedings

- Yanqiu Huang, Wanli Yu, and Alberto Garcia-Ortiz. PKF-ST: A communication cost reduction scheme using spatial and temporal correlation for wireless sensor networks. In *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks (EWSN)*, pages 47–52, 2016.

- Wanli Yu, Yanqiu Huang, and Alberto Garcia-Ortiz. Modeling optimal dynamic scheduling for energy-aware workload distribution in wireless sensor networks. In *2016 International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 116–118, May 2016.

- Wolfgang Buter, Yanqiu Huang, Daniel Gregorek, and Alberto Garcia-Ortiz. A decentralised, autonomous, and congestion-aware thermal monitoring infrastructure for photonic network-on-chip. In *Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), 2015 10th International Symposium on*, pages 1–8, June 2015.

- Wanli Yu, Yanqiu Huang, and Alberto Garcia-Ortiz. An altruistic compression-scheduling scheme for cluster-based wireless sensor networks. In *Sensing, Communication, and Networking (SECON), 2015 12th Annual IEEE International Conference on*, pages 73–81, Seattle, USA, June 2015.

- Yanqiu Huang, Wanli Yu, and Alberto Garcia-Ortiz. PKF: A communication cost reduction schema based on kalman filter and data prediction for wireless sensor networks. In *Proceedings of the 26th IEEE International system-on-chip conference*, pages 73–78. CAS, Sep. 2013.

1.4 Dissertation Structure

The dissertation is organized in the classical form of three main parts: an introduction where the state of the art and related background are stated, a central core where the proposed methods are developed, and a final part with the validation of the approaches and the conclusion.

I. **Introduction**: Chapter 2 and Chapter 3 introduce the state of the art and the background.

   The dissertation starts with a detailed discussion of existing data compression techniques in Chapter 2. It concludes the advantages and disadvantages of each approach and motivates the use of KF for data compression. Chapter 3 introduces the state-space model of a system and how to estimate the system state using KF. It deeply studies the optimality of KF from Bayesian estimation and presents the variants of KF with correlated noise and colored noise. This chapter provides the solid theoretical foundations for our proposed approaches in the following chapters.

II. **Core:** the proposed approaches are presented in Chapters 4 and 5.

    Chapter 4 proposes our PKF approach using temporal correlation that combines a k-step ahed KF predictor and a KF to compress the transmission rate for cluster-based WSNs. For understanding the underlying process of PKF and finding the effect of the system parameters on its performance, an in-depth mathematical analysis is studied in this chapter. Based on this analysis, Chapter 5 extends PKF to further exploit spatial correlation. The nonlinear reconstruction problem is solved from the linear approximations using different methods.

III. **Conclusion**: the validation of the methods and the final conclusion are described in the last two chapters.

    The performance of the proposed approaches PKF and EPKF are estimated using real WSN signals. To measure the energy consumption and lifetime improvement by using the proposed approaches, the algorithms are implemented in Openmotes. Finally, we conclude our work and present the future research directions in Chapter 7.

2 Review and Comparison of Data Compression Techniques

2.1 Introduction

The WSN nodes are typically powered by batteries, which are with limited energy budget. It is hard or impossible to recharge or replace the battery due to the large quantities or the harsh environments. Besides, the WSN applications often require the network last for long time. Therefore, how to achieve the energy efficiency is alway concerned by the research or industrial communities. Since the communication process is much more costly in terms of energy use than the data computation, most of the research focuses on developing energy efficient schemes for reducing the communication cost. In general, the existing techniques are mainly devoted to either regulating the communication across the whole network (e.g., the design of routing and clustering protocols [26], as well as scheduling strategies [14]) or reducing the amount of transmission information for each node by data processing (e.g., data aggregation [27] and data compression [2]). Data compression is very attractive due to the inherent existence of spatial and temporal correlation in the physical phenomena. It can be combined with the network-based strategies to improve the lifetime [28], [29].

The data compression approaches are classified into two categories: *data packet size compression* approach and *transmission rate compression* approach in Section 2.2. The detailed descriptions of related approaches are presented in Section 2.3 and 2.4, respectively. Section 2.5 critically evaluates these approaches based on energy conservation and reconstruction quality. The gaps in previous research are outlined, which motivates our approach in Chapters 4 and 5. A summary of the chapter is presented in Section 2.6.

2.2 Taxonomy of Data Compression Approaches

In this work, we classify the data compression approaches used in WSNs into two categories: *data packet size compression* and *transmission rate compression*.

- *data packet size compression*, which refers to approaches that compress the volume of the data packet at each transmitted time to reduce the communication energy cost. The related work can be broadly classified into four main classes [2, 30, 31]: dictionary based compression, distributed source coding, transform based compression and compressed sensing (also known as compressive sensing, compressive sampling, or sparse sampling).

- *transmission rate compression*, which refers to approaches that compress the transmission frequency to achieve the energy reduction. This category mainly includes: time series forecasting, stochastic based approach and compressed sensing [2, 30].

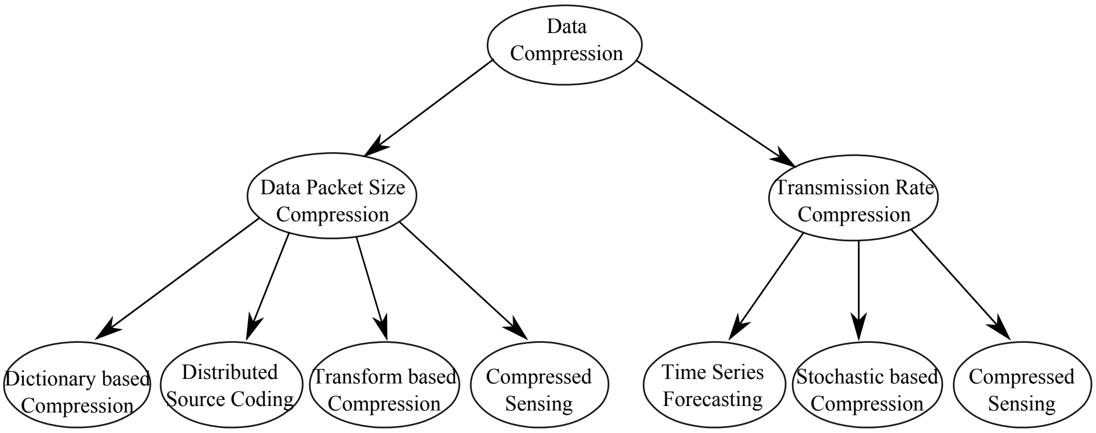

Data Compression
Data Packet Size Compression
Transmission Rate Compression
Dictionary based Compression
Distributed Source Coding
Transform based Compression
Compressed Sensing
Time Series Forecasting
Stochastic based Compression
Compressed Sensing

Figure 2.1: Taxonomy of data compression approaches used in WSNs.

The detailed description of each approach in these two categories are introduced in Sections 2.3 and 2.4.

2.3 Data Packet Size Compression

This section presents an overview of algorithms in the *data packet size compression* category. The critical analysis and comparison of these algorithms will be discussed in Section 2.5. According to Section 2.2, we mainly focus on the algorithms of the dictionary based, distributed source coding, transform based and compressed sensing.

Dictionary Based Compression

Dictionary based compression aims to build a list of commonly occurring patterns, named dictionary, and encode these patterns by transmitting their index in the list. The decoder maintains the same predefined dictionary to recover the information. Although dictionary based algorithms can be used to compress all kinds of data, traditional algorithms are not suitable for WSNs due to the large requirements of processing and memory [32].

S-LZW for sensor node, is developed in [16] by balancing the dictionary size, the size of the data to be compressed and the protocol to follow when the dictionary fails. When it is applied to several datasets of real WSN applications, the energy consumption can be reduced up to a factor of 4.5X. Authors in [33] propose a simple lossless entropy compression (LEC) scheme. The LEC algorithm is similar to the baseline JPEG algorithm for compressing the DC coefficients. Compared with S-LZW, LEC requires lower computation power and uses smaller dictionary. The size of the dictionary is determined by the resolution of the analog-to-digital converter. An adaptive lossless data compression (ALDC) algorithm has been designed in [34]. It firstly partitions the data sequence that needs to be transmitted into blocks, then compresses the block of data using two adaptive lossless entropy compression (ALEC) code options: 2-Huffman Table and 3-Huffman Table ALEC. Since the compression is allowed to dynamically adjust to a changing source, ALDC outperforms LEC and S-LZW. The extension of LEC, GA-LEC and FA-LEC [35], are proposed to achieve the adaptive compression. These two schemes implement the adaptation based on the concept of appropriately rotating the prefix-free table. Shorter prefix-free codes for a larger percentage of samples are used in both GA-LEC and FA-LEC.

Distributed Source Coding

Distributed source coding approaches are very popular for data compression in WSNs. They typically compress the data inside the network based on the Slepian and Wolf theorem [36], which involves coding of two or more dependent sources with separate encoders and a joint decoder. Fig. 2.2 shows one example with two correlated data streams X and Y. If the encoder and the decoder processes of two sources are executed independently, the coding rates, R_1 and R_2, have to be larger than or equal to the entropies of two sources, $H(X)$ and $H(Y)$, respectively, to achieve lossless compression. Although joint encoding can reduce the coding rates from $H(X) + H(Y)$ to $H(X, Y)$, it requires intra-communication between two sources. By using the Slepian and Wolf theorem, two sources can be independently encoded, while the coding rates can be reduced using a joint decoder as depicted in Fig. 2.2. The theoretical bound for lossless coding rates of two sources subject to $R_1 \geq H(X|Y)$, $R_2 \geq H(Y|X)$, $R_1 + R_2 \geq H(X, Y)$, according to the Slepian and Wolf theorem.

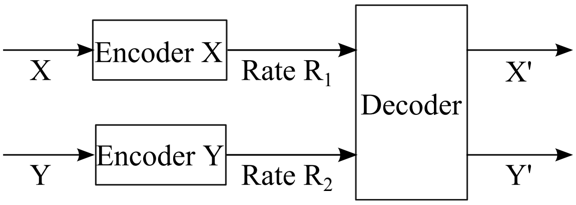


Figure 2.2: Spelain and Wolf theorem: independent encoding and joint decoding of two correlated data streams X and Y.

The work of [37] proposes a compression method by exploiting existing correlations in sensor data based on distributed source coding principles. The decoder collects the correlations among the sensor nodes and broadcasts to them. Each node encodes the observations according to the received corrections. A clustered Slepian and Wolf coding (CSWC) is designed in [38], combing with inter-cluster explicit entropy coding to compress the data based on the spatial correlations. Similar approaches based on the distributed source coding can be found in the survey papers [2, 30].

Transform Coding

The transform coding is widely used in image or video compression algorithms. Recently, these approaches are adopted in wireless multimedia sensor networks (WMSNs). Wavelets transform and cosine transform are two common approaches. In [39], this work firstly collects N-samples signal in transform coding and then approximates the data by K-sparse representation. The signal is represented in a basis expansion to be sparse using the transform theory, e.g., wavelet transform, Fourier transform, etc. The K largest coefficients and the corresponding locations are encoded and transmitted. A modified version of distributed wavelet transform is proposed in [40] to address the energy reduction problem of WSN. A scheme named Set Partitioning in Hierarchical Trees [41] achieves high compression ratio by setting a partition algorithm in wavelet transform. The power consumption of the cosine transform approaches is usually larger than the wavelet transform methods [42]. To reduce the complexity, an integer cosine transform is used in [43]. In addition, [44] proposes an adaptive data compression approach based on Fuzzy transform to minimize the memory space and communication cost.

Compressed Sensing

Compressed sensing (CS) has attracted the attention from various scientific research communities. It promises a reconstruction of a sparse signal by using a sampling rate significantly below the Nyquist rate [45]. Given a proper transformation basis $\Psi = [\psi_1, \psi_2, \cdots, \psi_N]$, the signal X can be transformed to a K-sparse representation S, i.e., $X = \Psi S$. The theory of CS demonstrates that the signal X can be compressed as $Y = \Phi X$, with a $M \times N$ $(M \leq N)$ sized measurement matrix $\Phi = [\phi_1, \phi_2, \cdots, \phi_N]$ whose row vectors are largely incoherent with Ψ. The recovery of X can be achieved by the ℓ_1 minimization:

$$\hat{S} = \arg\min \|S\|_{\ell_1} \quad \text{subject to } Y = \Phi X = \Phi\Psi S \tag{2.1}$$

Current researches mainly apply CS in WMSNs to achieve *data packet size compression*. A CS based video encoder is designed in [45] to compress the raw samples that the camera captured by using the temporal correlations between consecutive video frames. A number of works apply CS into ECG monitoring like [46, 47]. In such scenarios, the original ECG signal is usually firstly represented by a linear transformation, then a sparse representation is calculated to get the compressed signal which will be transmitted.

2.4 Transmission Rate Compression

This section presents an overview of algorithms in the *transmission rate compression* category. Generally, the *transmission rate compression* category can be further classified into: compressed sensing, time series forecasting and stochastic based compression as shown in Fig. 2.1. Both temporal and spatial correlation can be exploited.

Time Series Forecasting

Exploiting the time series models, such as Moving Average (MA), Auto-Regressive (AR) and Auto-Regressive Moving Average (ARMA) models, for transmission rate compression is simpler and has a good data quality in many practical cases [31]. For example, in [28] a low-order AR model is built at each node to predict local readings. Nodes transmit these local models to a sink node, by which the sink node predicts their values without directly communicating with the nodes. When needed, nodes send information about outliers and model updates to the sink. Unlike [28], the method presented in [29] models the physical phenomenon as an AR model plus a linear trend during a time interval of a few hours rather than during the full history. It detects the variations in the data distribution to guarantee the accuracy of the system model. When the model is not accurate enough, a model update phase is triggered. Besides single model schemes, an adaptive multi-model selection mechanism is presented in [48], where all nodes save a set of models. At a given instant only one of them is used for data prediction. If the error between sensed value and prediction is higher than the allowed threshold, the current model is switched to the one satisfying the requested accuracy and minimizing the cost of the update. A similar method called DBP, derivative based prediction, uses a simple linear model to predict the trends of the data measured by sensor nodes [49]. This work is based on the assumption that the trends of sensed data in short and medium time intervals could be approximated by using a linear model.

Spatial correlation can be exploited to further decrease the communication cost. Some of the techniques require the intra-communication among nodes. For example, the node intercepts the information from its neighbors to compress its own data in [24]. Similarly, the node receives the model parameters from its neighbors to decide whether to transmit its own parameters in [50]. To avoid this overhead, clustering the nodes and selecting a part of them to be active in a period is one of the most popular approaches. An energy-efficient data collection framework, EEDC, is proposed in [18]. Each node stores the latest sampling values until its buffer is full and calculates the line segments approximating the original time series. The transmission rate is reduced by only transmitting the end points of every line segment. To further reduce communication cost, the cluster head selects an appropriate number of nodes to be active. A similar approach is [51], where a sensing framework using virtual sensors is proposed. It uses an autocorrelation based transversal filter to predict data using temporal correlation and selects the active nodes in the coordinator to minimize the energy consumption of the network and balance the energy expenditure of nodes.

Stochastic Based Compression

The stochastic based compression techniques vary according to the way that the model is built, e.g., probabilistic models and state space models. The probabilistic models are constructed by exploiting a characterization of the phenomenon in terms of a random process time series. In other words, the physical phenomenon is considered as a random process by means of a probability density function (PDF). For instance, in [52] a specific model based on the PDF of time-varying multivariate Gaussian distribution is established at the sink node with historical data. When the user queries e.g., if an attribute is located in a given range, the cluster head uses this model to compute the probability rather than communicating with the sensor node. This is a “pull-based” approach where the user initiates the transaction. By contrast, a “push-based” approach is presented in [19], which acquires data at a steady rate and proactively reports anomalies to the user. It uses a pair of replicated probabilistic models synchronously running in both the leaf node and the cluster head. With this model, the cluster head predicts the approximated data and the leaf node follows this prediction to guarantee the prediction quality by transmitting the inaccurate data. An extension of [19] is given in [53], where a dynamic probabilistic model is exploited to enable real-time applications.

A state space representation of the phenomenon can be derived. It provides the dynamics as a set of coupled first-order differential equations in a set of internal variables known as state variables, together with a set of algebraic equations that combine the state variables into physical output variables [54]. With the help of filtering and prediction techniques, the communication can be suppressed. In [55], the SIP method is proposed to estimate the system state and compress the transmission between the leaf node and the cluster head. It consists of three steps: data filtering, state estimation and, data prediction and reconstruction. Each node firstly uses a filter, e.g., Exponentially Weighted Moving Average (EWMA), LMS, NLMS or KF, to remove the measurement noise in the collected raw data. Then the node provides an estimate of the system state using either Piece-wise Linear Approximation (PLA) or Piece-wise Constant Approximation (PCA) from the smoothed data. The head predicts the system state based on its last prediction with PLA or PCA. The leaf node follows the prediction of the cluster head and compares it with its own estimation using the new collection. When the error between the prediction and the local estimation exceeds a given threshold, the leaf node sends the current state vector to the cluster head.

A sophisticated approach using dual KFs (DKF) is proposed in [56], where the system model is constructed in accordance with a KF. DKF uses a pair of KFs in both leaf node and cluster head to synchronously predict the raw data. When the data contains noise, each node firstly uses an additional KF with a controllable process covariance to remove the noise and provide the smoothed data. This data is treated as the measurement for the second KF. When the prediction error using the second KF compared with the smoothed data is bigger than a threshold, the smoothed data is transmitted to the cluster head.

The authors in [57] provide a method named CoGKDA, which combines the Grey model and KF together. The leaf node collects the raw data z_k at time k and predicts x_{k+1} using the latest stored l samples in the actual data queue (ADQ). Besides, it follows the prediction of the cluster head y_{k+1} using the latest stored l samples in the predicted value queue (PVQ). If $y_{k+1} - z_k < \epsilon$ and $y_{k+1} - x_{k+1} < \theta$, the transmission can be suppressed; otherwise, the current collected value z_k is transmitted.

Compressed Sensing

Besides the use of CS [58] to compress the data size as introduced in Section 2.3, it can also be used for transmission rate compression using both temporal and spatial correlation [59, 60, 61, 62].

Distributed CS is applied in the network by [59]. Each node executes CS coding to reduce the sampling rate and thus to decrease the number of transmitted packets, while the reconstruction progress is executed in the sink node which does not have energy limitation. In [60] and [61], temporal and spatial correlations are utilized in the CS decoder to achieve more sampling reduction in the network. CS is also used for localization in WSNs [62], comparing with traditional localization approaches that require a large number of sensor nodes to transmit the received signal strength (RSS), using CS enables few RSS samples since the authors claimed that the RSS vector can be sparsely represented. More relevant researches can be found in a survey paper [63].

2.5 Critical Analysis

We compare the above classified methods from two metrics: energy conservation and reconstruction quality. For the first comparison, the communication cost models are reviewed. There are many studies available [14, 64, 65, 66, 67]. Basically, the total communication energy consumption of a node consists of data packet transmission and overhead cost: $E_{cmn} = E_{overhead} + E_{data}$. The overhead activities involve radio startup, channel accessing, control packets, turnaround, idle listening, overhearing, collision, etc. Before transmitting the data packet, the sensor node needs to turn on the radio and tries to access the wireless channel. Some control packets may also be needed. After that, the actual transaction commences and once finished, the radio is shut down. During this period, the node may turn on its receiver prior to the actual reception because of the unawareness of the destination active state (it is the so called idle listening) and may receive some packets that are not intended for it (namely overhearing). Due to collision, the packets may not be transmitted or received successfully which causes retransmission and extra energy cost.

Based on the above model, we can conclude that the *data packet size compression* approaches focus on reducing the energy cost of data packet transmission E_{data}. In contrast, the *transmission rate compression* aims to reduce the overall communication cost E_{cmn}. As analyzed in [14], the overhead can dominate the total energy consumption of the sensor nodes in some cases. Although many approaches are able to compress the data size with a high compression ratio, e.g., 45-75% by LEC [33], up to 93% by CSWC [38], they are incapable of reducing the overhead consumption in the communication. The transmission rate can be compressed from 50-99% ranging from techniques and data types as summarized in [68]. In this case, it is more efficient in reducing the communication cost.

Among transmission rate compression techniques, we compare their reconstruction quality. Many techniques, based on time series modeling, probability modeling or even compressive sensing, supply only the approximated data of the measurements. However, the raw measurements are inevitably corrupted by noise in practical WSN scenarios [1, 69]. It makes the reconstructions using these schemes unable to reflect the true state of the monitored environment. In this sense, the approaches based on the filtering techniques could produce more accurate reconstructions by removing the noise.

Considering the state-space model provides a much *richer* description of the dynamic phenomenon, the objective by using a WSN in the end becomes to reconstruct the state information from the data supplied by the sensor nodes. From this point of view, the node performing local state estimation and transmitting the estimated state when needed, may provide more information and have better reconstruction in the cluster head. As the real world systems are frequently able to be represented in terms of very simple models of first- or second-order [70], transmitting a low-order state, typically a small proportion in the data packet (taking the packet header into account), may not consume notable energy cost. For a linear system, the best candidate for noise reduction and state estimation is the KF, since it promises the optimal state estimate in the sense of minimum mean square error (MMSE) [71]. It has been widely used in WSNs, such as target tracking [72], outlier detection [73], [74]. KF-based data fusion is one of the most significant approaches to overcome sensor failures and spatial coverage problems [75], [76]. In order to fully utilize the KF IP, we restrict ourselves to the techniques employing KF for transmission rate compression in this work.

The existing methods using the KF for transmission rate compression still need to be improved. They only exploit the partial functionality of KF in noise filtering but not the essence in state estimation. For example, CoGKDA [57] uses the filtered value by KF as a reference to compare with the prediction of the cluster head. When the prediction error or the cumulative error exceeds the bounds, the raw data in the leaf node is sent. However, once there is a missing point in the cluster head, i.e., the data is intermittently transmitted, the reconstruction error starts to cumulate even with the update of a new observation, since the past information is not contained in the current measurement. Instead of transmitting the raw data, the leaf node should transmit the current state estimate. It can calibrate the estimation of the cluster head and reset the cumulative error. One of these methods is SIP [55]. It takes the KF as a candidate for noise reduction in the leaf node and approximates the system state using PLA or PLC methods from the smoothed data. However, it scarifies the computation cost by separating data filtering, state estimation and prediction into different frameworks, while providing only the approximations of the system state. DKF [56] removes the noise in the raw data by setting a controllable covariance of the process noise to a KF. The second KF treats the output of the first KF as the measurement for further prediction. In this case, the optimal system model for the second KF should be the augmented model, rather than the model with the same state transition matrix as the smoothing KF claimed by the authors, due to the colored measurement noise.

The above analysis motivates our proposed approach, PKF, in Chapter 4 that uses a KF for transmission rate compression. It takes the full advantage of the KF for data filtering and state estimation, and aims to optimally reconstruct the state information for a linear system characterized by a state-space model. To exploit the spatial correlation without intra-communication and coordinator, the extension of PKF is further proposed in Chapter 5.

2.6 Summary

This chapter has reviewed the literature relevant to data compression techniques used in WSNs, summarizes their limitations by a critical analysis and motivates the use of the KF in our proposed approaches. According to the compressed objects of the techniques, *data size* or *transmission rate*, we firstly classify the approaches into two categories: data packet size compression and transmission rate compression. The detailed descriptions of the related approaches in each category are then presented.

Compared with the data packet size compression, compressing the transmission rate can save the overall energy communication cost in one transaction, including the overhead and the cost for real data packet transmission. Due to the fact that the observations collected by the sensor node are accompanied by the ubiquitous noise, reconstructing the raw data with approximations provides inaccurate information for the monitored system. It indicates that the local preprocessing in the node is needed. As the dynamic phenomenon can be well described by a state-space model, the objective by using a WSN is to reconstruct the state information by using the data of the sensor nodes. The node performing local state estimation based on the obtained measurement and transmitting the estimated state when needed, may provide better reconstruction in the cluster head, since the estimation of the cluster head can be calibrated and the cumulative error can be reset. For a linear system, which is the main focus of this work, the best candidate for noise reduction and state estimation is the KF. However, the existing methods using KF for transmission rate compression exploit only its partial functionality in noise filtering, but not the essence in state estimation. This motivates us to take the full advantage of the KF in our proposed approaches, combining data filtering, state estimation with data prediction, to compress the transmission rate while reconstructing the state information.

3 Kalman Filter and Optimality Study

3.1 Introduction

The state vector contains all information about the system at a given time instant. It can not be directly determined by the input and output of the system in most practical scenarios, because of the unknown disturbances, the partially observation and so on. Instead, the internal state can only be estimated from a model and the available measurements by using the state estimation methods. Kalman filter, as one of the estimation methods, produces the optimal state estimates of the linear dynamic systems with Gaussian noise. It is a recursive algorithm and combines the prediction from the previous time step with the current measurement to produce an improved estimate of the current state [77].

This chapter firstly introduces the definition of a system. The two typical modeling methods, difference equation and state space model, are compared in Section 3.2. The general process of KF to estimate the inner state of the linear dynamic systems is introduced in Section 3.3. Then we try to understand the optimality of KF from Bayesian estimation in Section 3.4 including conditional expectation and maximum *a posteriori* (MAP) estimation. The optimality study of KF is helpful to analyze our proposed approach in Chapter 4. Further on, the variants of KF for systems with correlated noise and systems with colored measurement noise are presented in Section 3.5. These variants are needed in Chapter 4 and Chapter 5.

3.2 State-space Model

"*A system is considered to be an object in which different variables interact at all kinds of time and space scales and that produces observable signals*" [78]. There are five sets of variables in a system, known as the input u, the system disturbance w, the state x, the measurement disturbance v, and the output z. The input u represents the external forces that are acting upon the system, which is measurable and can be manipulated directly by the user. The disturbance w is indicated as system noise, which originates from the environment and directly affects the behavior of the system. It cannot be manipulated and is considered as possibly structured uncertainty in the input u or in the relationship between u and x [78]. The system state x stores all the effects of the past inputs u and disturbances w to the system. When the state depends only on the current input and disturbance, it is a static system; otherwise, the system exhibits dynamic behavior. The number of the system states, n, is equal to the number of independent energy storage elements (such as mass, spring, capacitor, inductance [79]) in the system [54]. The real-world systems are frequently able to be represented in terms of very simple models of first- or second-order [70]. The output disturbance v represents the uncertainty introduced in the measurement process, which cannot be manipulated. The output z is the observable behavior of the dynamic phenomenon that are of interest to the user. These variables could be continuous or discrete functions of time. We are interested in the discrete-time signals here. The continuous-time signals, such as electrical voltages produced by sound or image recording instruments, can be converted to discrete-time signals by sampling and quantization [80].

There are typically two methods to model a discrete-time dynamic system. One is to directly relate the input u, the disturbance w and v to the output z in one difference equation, such as:

$$z_k = g_k(z_{k-1}, \cdots z_{k-n}, u_k, \cdots, u_{k-m}, w_k, \cdots, w_{k-n}, v_k, \cdots, v_{k-n}) \tag{3.1}$$

where $g_k(\cdot)$ is an arbitrary and vector-valued function. This method only considers the input-output characteristic. It can not provide any knowledge of the interior structure and state information of the system.

An alternative solution is the so-called state-space model. Instead of viewing a system simply as a relation between inputs and outputs, state space models consider this transformation as taking place via the transformation of the internal state of the system [81]. By defining an $n \times 1$ vector x_k to indicate the internal state, the above n-order difference equation Eq. (3.1) can be described as n first order difference equations:

$$x_k = f_k(x_{k-1}, u_{k-1}, w_{k-1}) \tag{3.2}$$

where $f_k(\cdot)$ is a vector function with n components. The output of the system can be calculated from the internal state x_k, the input u_k and the disturbance v_k:

$$z_k = h_k(x_k, u_k, v_k) \tag{3.3}$$

where $h_k(\cdot)$ is a vector function with p components.

State space models are more akin to the classical mathematical models used in physics, chemistry, and economics [81]. They offer a standardized way for defining the inner states for both linear and nonlinear systems and are more adapted to computations with n first order difference equations. When $f_k(\cdot)$ and $h_k(\cdot)$ are linear functions of x, u, w and v, the system is a linear discrete dynamic system. It is the main focus of this work. In this case, the process model of Eq. (3.2) written in the state-space form is:

$$x_k = A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + w_{k-1} \tag{3.4}$$

where A_k is the transition matrix which relates the system state at time k to the state at time $k+1$; B_k is the control-input matrix manipulating the effect of the control input on the system state; u_k is the known input vector (steering angle, throttle setting, braking force); w_k accounts for the inexactitudes of the model and is also known as the process noise. The observation z_k is mapped from x_k by the observation matrix H_k and corrupted with a measurement noise:

$$z_k = H_k x_k + v_k \tag{3.5}$$

The diagram of the state-space system model is shown in Fig. 3.1.

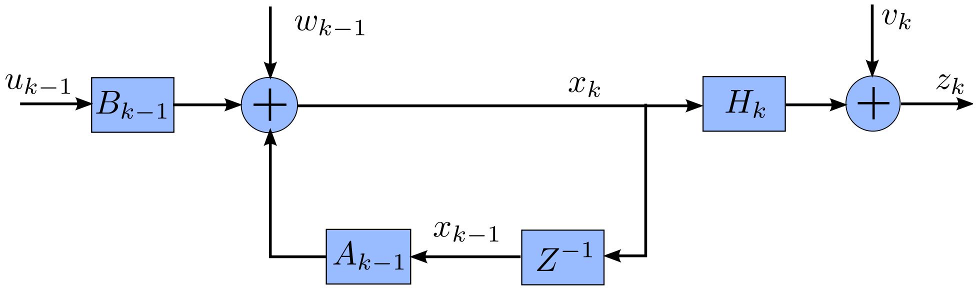


Figure 3.1: The diagram of the state-space model for a linear discrete dynamic system.

3.3 Kalman Filter

The internal state of a linear dynamic system can be estimated from the noisy observations by a KF. It combines the estimate from the previous time step with the current measurement to produce an improved estimate of the current state [77]. It is a recursive algorithm that produces the minimum mean square error of the estimation for a system with Gaussian noise.

In the standard KF, the process noise $w_k \sim \mathcal{N}(0, Q_k)$ and the measurement noise $v_k \sim \mathcal{N}(0, R_k)$ are assumed to be Gaussian white noise with zero mean and known covariance, namely,

$$E[w_k] = 0 \qquad E[v_k] = 0 \tag{3.6}$$

$$E[w_k w_j^T] = Q_k \delta_{kj} \qquad E[v_k v_j^T] = R_k \delta_{kj} \tag{3.7}$$

where $E[\cdot]$ denotes expectation and δ_{kj} denotes the Kronecker delta function with $\delta_{kj} = 1$ if $k = j$; otherwise, $\delta_{kj} = 0$. Q_k and R_k are covariance matrices of the process and measurement noise, respectively. These two noise are mutual uncorrelated and also uncorrelated with the state, namely

$$E[w_k v_j^T] = 0 \quad E[x_k w_j^T] = 0 \quad E[x_k v_j^T] = 0 \tag{3.8}$$

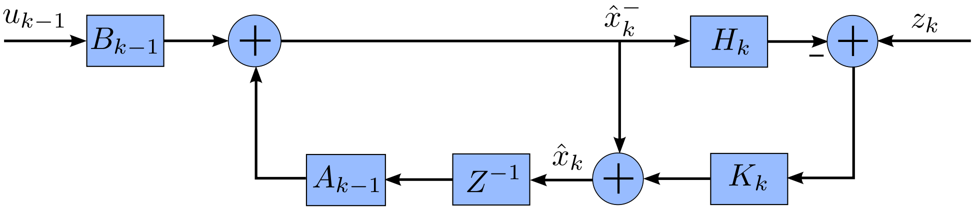


Figure 3.2: The diagram of the Kalman filter for discrete dynamical system.

The process of the KF involves two steps: *prediction* and *update*. The diagram is shown in Fig. 3.2. In the *prediction* phase, the state estimate of the previous time step $\hat{x}_{k-1}$ is used to generate an *a priori* estimate of the current state $\hat{x}_k^-$.

$$\hat{x}_k^- = A_{k-1}\hat{x}_{k-1} + B_{k-1}u_{k-1} \tag{3.9}$$

Let $\hat{e}_k^- = \hat{x}_k^- - x_k$ denote the error between this *a priori* estimate and the true state. The uncertainty of this prediction is measured by the covariance of the error. It is calculated by:

$$P_k^- = E\big[\hat{e}_k^- \, \hat{e}_k^{-T}\big] = A_{k-1}P_{k-1}A_{k-1}^T + Q_{k-1} \tag{3.10}$$

where P_{k-1} is the *a posteriori* covariance of the last time step. It will be discussed in more detail later in this section. In the *update* phase, the current measurement z_k is incorporated into the *a priori* prediction to produce an improved *a posteriori* state estimate $\hat{x}_k$. For convenience, we call it *optimal value* in the following. The basic idea behind this phase is to use a weighted average, with more weight K_k being given to the *a priori* estimate with higher certainty. The weight, also known as the Kalman gain, satisfies:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \tag{3.11}$$

The updated estimate of the system lies between the predicted and measured state, and is given by:

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k \hat{x}_k^-) \tag{3.12}$$

Let $\hat{e}_k = \hat{x}_k - x_k$ denote the error of this optimal estimate. Its covariance indicates the uncertainty of the final estimate, which is:

$$P_k = E\left[\hat{e}_k\, \hat{e}_k^T\right] = (I - K_k H_k) P_k^- \tag{3.13}$$

In the time invariant systems, the KF typically enters a steady state after several steps, where the Kalman gain and the covariance converge to constant values: $K_{k\to\infty} = K$, $P_{k\to\infty}^- = P^-$ and $P_{k\to\infty} = P$. Then only Eqs. (3.9) and (3.12) are needed to predict the future state.

3.4 Understand the Optimality of KF from Bayesian Estimation

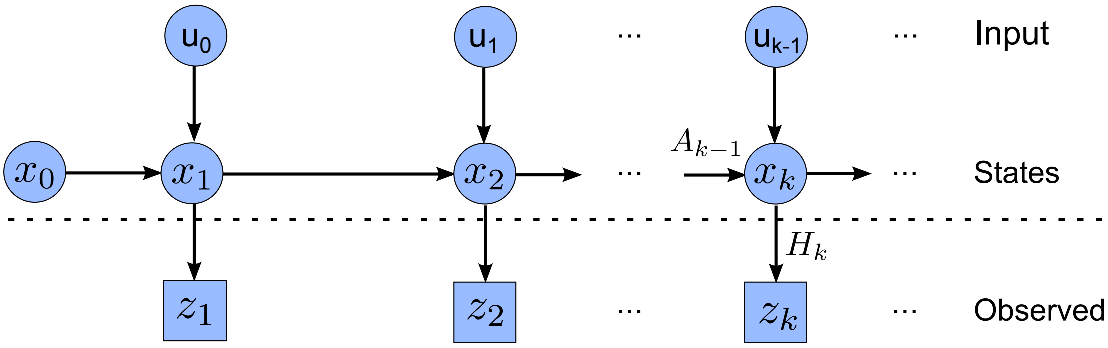


Figure 3.3: Bayesian framework of a hidden Markov model.

In the recursive Bayesian estimation [82], the true state $(x_0, \cdots, x_k)$ is assumed to be an unobserved Markov process, and the measurements $(z_1, \cdots, z_k)$ are the observed state of a hidden Markov model (HMM) as shown in Fig. 3.3. The probability of the current state and the measurement only depend on the state at last time step, i.e. $p(x_k \mid$

$x_{k-1}, x_{k-2}, \cdots, x_0, u_{k-1}, \cdots, u_0) = p(x_k \mid x_{k-1}, u_{k-1})$ and $p(z_k \mid x_k, x_{k-1}, \cdots, x_0) = p(z_k \mid x_k)$ because of the Markov assumption. Bayes estimator minimizes the posterior expected value of a loss function and maximizes the posterior probability density function (PDF) for state x_k, given the observation set $\boldsymbol{Z_k} = [z_k, \cdots, z_1]$ and the control input $\boldsymbol{U_k} = [u_{k-1}, \cdots, u_0]$. We obtain the equivalent estimators from conditional expectation and maximum a posteriori (MAP) estimation in the following to illustrate the optimality of KF.

3.4.1 Conditional Expectation

Given two random variables X and Y, the conditional expected value of Y given $X = a$, $E[Y|X = a]$, is a number that depends on a, i.e., it is a function of a. Thus, the conditional expected value of Y given X, denoted as $E[Y|X]$, is a random variable, which is a function of X. It has been proved that $E(Y|X)$ is closest to Y of all functions of X, in the sense of minimum mean square error (MMSE) [83]. Thus, we aim to obtain the conditional expectation of x_k based on $\boldsymbol{Z_k}$ and $\boldsymbol{U_k}$, i.e., $E[x_k \mid (\boldsymbol{Z_k}, \boldsymbol{U_k})]$, in the following.

The following theorem is the basis for our derivation, which can be derived from the Bayes' rule as illustrated in [84, 85]. If two random vector X_1 and X_2 have joint Gaussian distribution, such as:

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right)$$

then the distribution of X_1 conditional on $X_2 = a$ is multivariate normal $(X_1|X_2 = a) \sim \mathcal{N}(\bar{\mu}, \bar{\Sigma})$, where

$$\begin{aligned} \bar{\mu} &= \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(a - \mu_2) \\ \bar{\Sigma} &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \end{aligned} \tag{3.14}$$

because of the Bayes' rule [82]:

$$p(a|b) = \frac{p(a, b)}{p(b)} \tag{3.15}$$

The best prediction of x_k based on $\boldsymbol{Z}_{k-1}$ and $\boldsymbol{U_k}$ is the conditional expectation $E\big[x_k \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k}\big]$, which can be calculated by:

$$\begin{aligned} \hat{x}_k^- &= E\big[x_k \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k}\big] \\ &= E\big[A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + w_{k-1} \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k}\big] \\ &= A_{k-1}\hat{x}_{k-1} + B_{k-1}u_{k-1} \end{aligned} \tag{3.16}$$

This is the *a priori* prediction of Kalman filter with the prediction covariance:

$$P_k^- = E[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T] = A_{k-1}P_{k-1}A_k^T + Q_k \tag{3.17}$$

Thus, the distribution of x_k conditional on $(\boldsymbol{Z}_{k-1}, \boldsymbol{U_k})$ has normal distribution, namely,

$$x_k \mid (\boldsymbol{Z}_{k-1}, \boldsymbol{U_k}) \sim \mathcal{N}(\hat{x}_k^-, P_k^-) \tag{3.18}$$

The random variable z_k conditional on $(\boldsymbol{Z}_{k-1}, \boldsymbol{U_k})$ has the mean

$$\begin{aligned} E[z_k \mid (\boldsymbol{Z}_{k-1}, \boldsymbol{U_k})] &= E[H_k x_k + v_k \mid (\boldsymbol{Z}_{k-1}\boldsymbol{U_k})] \\ &= H_k E[x_k \mid (\boldsymbol{Z}_{k-1}\boldsymbol{U_k})] = H_k\hat{x}_k^- \end{aligned}$$

and covariance

$$\begin{aligned} E[(z_k - H_k\hat{x}_k^-)(z_k - H_k\hat{x}_k^-)^T] &= E\Big[\big(H_k(x_k - \hat{x}_k^-) + v_k\big)\big(H_k(x_k - \hat{x}_k^-) + v_k\big)^T\Big] \\ &= H_k P_k^- {H_k}^T + R_k \end{aligned}$$

Thus,

$$z_k \mid (\boldsymbol{Z}_{k-1}, \boldsymbol{U_k}) \sim \mathcal{N}(H_k\hat{x}_k^-,\ H_k P_k^- H_k^T + R_k) \tag{3.19}$$

Observing Eqs. (3.18) and (3.19), the two random variables x_k and z_k conditional on $\boldsymbol{Z}_{k-1}$ and $\boldsymbol{U_k}$ have jointly Gaussian distribution. The cross correlation between the two variables $P_{xz} = E[(x_k - \hat{x}_k^-)(z_k - H_k\hat{x}_k^-)] = P_k^- H_k^T$. Thus, the joint distribution of x_k and z_k conditional on $\boldsymbol{Z}_{k-1}$ can be expressed as

$$\begin{bmatrix} x_k \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k} \\ z_k \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \hat{x}_k^- \\ H_k\hat{x}_k^- \end{bmatrix},\ \begin{bmatrix} P_k^- & P_k^- H_k^T \\ H_k P_k^- & H_k P_k^- H_k^T + R_k \end{bmatrix} \right) \tag{3.20}$$

Then the distribution of $(x_k \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k})\Big|(z_k \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k})$ is the conditional distribution of x_k conditional on $\boldsymbol{Z_k}$ and $\boldsymbol{U_k}$, i.e.,

$$x_k \mid (\boldsymbol{Z_k}, \boldsymbol{U_k}) = (x_k \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k})\Big|(z_k \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k}) \tag{3.21}$$

because of the Bayes' rule:

$$p(a|b,c) = \frac{p(a,b|c)}{p(b|c)} \tag{3.22}$$

Using Eq. (3.14), we can obtain the mean:

$$\begin{aligned}\hat{x}_k &= \hat{x}_k^- + P_k^- {H_k}^T(H_k P_k^- {H_k}^T + R_k)^{-1}(z_k - H_k \hat{x}_k^-) \\ &= \hat{x}_k^- + K_k(z_k - H_k \hat{x}_k^-)\end{aligned} \tag{3.23}$$

and the covariance:

$$\begin{aligned}P_k &= P_k^- - P_k^- {H_k}^T(H_k P_k^- {H_k}^T + R_k)^{-1} H_k P_k^- \\ &= (I - K_k H_k)P_k^-\end{aligned} \tag{3.24}$$

where

$$K_k = P_k^- H_k'(H_k P_k^- H_k' + R_k)^{-1} \tag{3.25}$$

The derived equations using conditional expectation, i.e., Eqs. (3.16), (3.17) and (3.23) to (3.25), are exactly the same as the five Kalman filter equations.

3.4.2 Maximum a posteriori Estimation

This section derives the KF equations from MAP estimation [71]. It aims to find the mode of posterior probability within a Bayesian framework.

From Bayes rule we have:

$$\begin{aligned}p(x_k \mid \boldsymbol{Z_k}, \boldsymbol{U_k}) &= \frac{p(x_k, \boldsymbol{Z_k}, \boldsymbol{U_k})}{p(\boldsymbol{Z_k}, \boldsymbol{U_k})} \\ &= \frac{p(x_k, z_k, \boldsymbol{Z}_{k-1}, \boldsymbol{U_k})}{p(z_k, \boldsymbol{Z}_{k-1}, \boldsymbol{U_k})}\end{aligned} \tag{3.26}$$

where the joint PDF in the numerator can be further expressed by

$$\begin{aligned}p(x_k, z_k, \boldsymbol{Z}_{k-1}, \boldsymbol{U_k}) &= p(z_k|x_k, \boldsymbol{Z}_{k-1}, \boldsymbol{U_k})p(x_k, \boldsymbol{Z}_{k-1}, \boldsymbol{U_k}) \\ &= p(z_k|x_k, \boldsymbol{Z}_{k-1}, \boldsymbol{U_k})p(x_k|\boldsymbol{Z}_{k-1}, \boldsymbol{U_k})p(\boldsymbol{Z}_{k-1}, \boldsymbol{U_k}) \\ &= p(z_k|x_k)p(x_k|\boldsymbol{Z}_{k-1}, \boldsymbol{U_k})p(\boldsymbol{Z}_{k-1}, \boldsymbol{U_k})\end{aligned} \tag{3.27}$$

The third equality is based on the fact that z_k only depends on the current state x_k, and v_k is independent of $\boldsymbol{Z}_{k-1}$ and $\boldsymbol{U_k}$. Substituting Eq. (3.27) into Eq. (3.26), we can obtain

$$p(x_k \mid \boldsymbol{Z_k}, \boldsymbol{U_k}) = \frac{p(z_k|x_k)p(x_k|\boldsymbol{Z}_{k-1}, \boldsymbol{U_k})p(\boldsymbol{Z}_{k-1}, \boldsymbol{U_k})}{p(z_k, \boldsymbol{Z}_{k-1}, \boldsymbol{U_k})}$$

$$
\begin{aligned}
&= \frac{p(z_k \mid x_k)p(x_k \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k})p(\boldsymbol{Z}_{k-1}, \boldsymbol{U_k})}{p(z_k \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k})p(\boldsymbol{Z}_{k-1}, \boldsymbol{U_k})} \\
&= \frac{p(z_k \mid x_k)p(x_k \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k})}{p(z_k \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k})}
\end{aligned}
\tag{3.28}
$$

where the denominator $p(z_k \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k})$ is the normalizing constant, denoted as a in the following. Under the Gaussian assumption of process noise and measurement noise, the mean and covariance of $p(z_k|x_k)$ are:

$$
\begin{aligned}
&E[z_k|x_k] = E[H_k x_k + v_k|x_k] = H_k x_k \\
&E\Big[\big(z_k - E[z_k \mid x_k]\big)\big(z_k - E[z_k \mid x_k]\big)^T\Big] = E[v_k v_k^T] = R_k
\end{aligned}
$$

Thus,

$$
p(z_k \mid x_k) = \frac{1}{\sqrt{(2\pi)^p |R_k|}}\, exp\Big(-\frac{1}{2}(z_k - H_k x_k)^T {R_k}^{-1}(z_k - H_k x_k)\Big) \tag{3.29}
$$

As obtained from Eqs. (3.16) and (3.17), the mean and covariance of $x_k \mid (\boldsymbol{Z}_{k-1}, \boldsymbol{U_k})$ are $\hat{x}_k^-$ and P_k^-, respectively. Then,

$$
p(x_k \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k}) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_k|}}\, exp\Big(-\frac{1}{2}(x_k - \hat{x}_k^-)^T P_k^{-(-1)}(x_k - \hat{x}_k^-)\Big) \tag{3.30}
$$

By substituting Eqs. (3.29) and (3.30) to Eq. (3.28), the posterior PDF $p(x_k \mid \boldsymbol{Z_k}, \boldsymbol{U_k})$ satisfies:

$$
p(x_k \mid \boldsymbol{Z_k}, \boldsymbol{U_k}) = \frac{exp\Big(-\frac{1}{2}(z_k - H_k x_k)^T {R_k}^{-1}(z_k - H_k x_k) - \frac{1}{2}(x_k - \hat{x}_k^-)^T \Sigma_k^{-1}(x_k - \hat{x}_k^-)\Big)}{a\sqrt{(2\pi)^{m+n}|R_k||\Sigma_k|}} \tag{3.31}
$$

The update step of Kalman filter is to maximize this posterior PDF. Let $\hat{x}_k^{MAP}$ denote the MAP estimate of the state, it then follows:

$$
\left.\frac{\partial \log p(x_k \mid \boldsymbol{Z_k}, \boldsymbol{U_k})}{\partial x_k}\right|_{x_k = \hat{x}_k^{MAP}} = 0 \tag{3.32}
$$

Combining Eq. (3.31) and Eq. (3.32), we can derive that:

$$
\hat{x}_k^{MAP} = \big(H_k^T {R_k}^{-1} H_k + P_k^{-(-1)}\big)^{-1}\big(\Sigma_k^{-1}\hat{x}_k^- + H_k^T {R_k}^{-1} z_k\big) \tag{3.33}
$$

Thanks to the lemma of inverse matrix in [86] that

$$(P^{-1} + B^T R^{-1} B)^{-1} = P - PB^T(BPB^T + R)^{-1}BP$$
$$(P^{-1} + B^T R^{-1} B)^{-1} B^T R^{-1} = PB^T(BPB^T + R)^{-1}$$

we can simplify Eq. (3.33) as:

$$\hat{x}_k^{MAP} = \hat{x}_k^- + K_k(z_k - H_k\hat{x}_k^-) \tag{3.34}$$

where K_k is the Kalman gain and satisfies

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \tag{3.35}$$

The covariance of the MAP estimate follows:

$$P_k = E[(x_k - \hat{x}_k^{MAP})(x_k - \hat{x}_k^{MAP})^T] = (I - K_k H_k)P_k^- \tag{3.36}$$

Thus, the equations derived from MAP estimation are consistent with KF.

3.5 Variants of Kalman Filter

In the standard KF derived in previous sections, the process noise and the measurement noise are assumed to be white and uncorrelated with each other. However, in some applications, they may have mutual correlations and have color. This section presents the variants of KF coping with these problems.

3.5.1 Kalman Filter with Correlated Noise

When w_k and v_k are correlated, we present the derivation of KF using conditional distribution of MVN in this section.

The system model and the measurement model still satisfy Eq. (3.4) and Eq. (3.5). The only difference is that w_k and v_k are correlated, which is defined as[1]:

$$E[w_k v_j^T] = M_k \delta_{kj} \tag{3.37}$$

[1] The definition is consistent with Matlab system identification toolbox. There are also other definitions of the correlation, e.g. $E[w_k v_j^T] = M_k \delta_{(k-1)j}$ in [87]. The obtained equations are slightly different.

We firstly calculate the distribution of x_k conditional on $\boldsymbol{Z}_{k-1}$ and $\boldsymbol{U_k}$, i.e., $x_k \mid (\boldsymbol{Z}_{k-1}, \boldsymbol{U_k})$. It has the mean value:

$$\begin{aligned} E[x_k \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k}] &= E[A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + w_{k-1} \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k}] \\ &= E[A_{k-1}x_{k-1} \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k}] + B_{k-1}u_{k-1} + E[w_{k-1} \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k}] \\ &= A_{k-1}\hat{x}_{k-1} + B_{k-1}u_{k-1} + E[w_{k-1} \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k}] \end{aligned} \tag{3.38}$$

In the standard KF, we have obtained $E[w_{k-1} \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k}] = 0$ because w_k and v_k are uncorrelated. But when they are correlated, we use the conditional distribution of two joint Gaussian vectors to calculate it, namely

$$w_{k-1} \mid (\boldsymbol{Z}_{k-1}, \boldsymbol{U_k}) = (w_{k-1} \mid Z_{k-2}, \boldsymbol{U_k}) \Big| (z_{k-1} \mid Z_{k-2}, \boldsymbol{U_k}) \tag{3.39}$$

The conditional distribution of the random vector $z_k \mid (\boldsymbol{Z}_{k-1}, \boldsymbol{U_k})$ is different from Eq. (3.19). It has a new covariance

$$\begin{aligned} \Sigma_{zz} &= E[(z_k - H_k\hat{x}_k^-)(z_k - H_k\hat{x}_k^-)^T] \\ &= E[(H_kx_k - H_k\hat{x}_k^-)(H_kx_k - H_k\hat{x}_k^-)^T] + E[v_k(H_kx_k - H_k\hat{x}_k^-)^T] \\ &+ E[(H_kx_k - H_k\hat{x}_k^-)v_k^T] + E[v_kv_k^T] \\ &= H_kP_k^-H_k^T + R_k \end{aligned} \tag{3.40}$$

Thus, when the process and measurement noise are correlated,

$$z_k \mid (\boldsymbol{Z}_{k-1}, \boldsymbol{U_k}) \sim \mathcal{N}(H_k\hat{x}_k^-,\ H_kP_k^-H_k^T + M_k^TH_k^T + H_kM_k + R_k) \tag{3.41}$$

The vector $w_{k-1} \mid (Z_{k-2}, \boldsymbol{U_k}) \sim \mathcal{N}(0,\ Q_{k-1})$, since w_{k-1} is uncorrelated with v_{k-2}. The cross covariance between $w_{k-1} \mid (Z_{k-2}, \boldsymbol{U_k})$ and $z_{k-1} \mid (Z_{k-2}, \boldsymbol{U_k})$ is:

$$\Sigma_{xz} = E[w_{k-1}(z_{k-1} - H_{k-1}\hat{x}_{k-1}^-)^T] = E[w_{k-1}v_{k-1}^T] = M_{k-1} \tag{3.42}$$

Thus, according to Eq. (3.14) and Eq. (3.38), the *a priori* estimate of the state for the noise correlated system is

$$\hat{x}_k^- = E[x_k \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k}] = A_{k-1}\hat{x}_{k-1} + B_{k-1}u_{k-1} + G_{k-1}(z_{k-1} - H_{k-1}\hat{x}_{k-1}^-) \tag{3.43}$$

with the *a priori* estimate covariance

$$P_k^- = A_{k-1}P_{k-1}A_{k-1}^T + Q_{k-1} - M_{k-1}K_{k-1}^T A_{k-1}^T - A_{k-1}K_{k-1}M_{k-1}^T - G_{k-1}M_{k-1}^T \tag{3.44}$$

where

$$G_{k-1} = M_{k-1}(H_{k-1}P_{k-1}^- H_{k-1}^T + R_{k-1})^{-1} \tag{3.45}$$

We then calculate the distribution of $x_k \mid (\boldsymbol{Z_k}, \boldsymbol{U_k}) = (x_k \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k})\Big|(z_k \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k})$. The cross covariance between $(x_k \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k})$ and $(z_k \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U_k})$ is:

$$\begin{aligned} P_{xz} &= E[(x_k - \hat{x}_k^-)(z_k - H_k\hat{x}_k^-)^T] \\ &= E[(x_k - \hat{x}_k^-)(H_k x_k - H_k\hat{x}_k^-)^T] + E[(A_{k-1}x_{k-1} + w_{k-1} - \hat{x}_k^-)v_k^T] \\ &= P_k^- H_k^T \end{aligned} \tag{3.46}$$

Thus, the update equations of KF according to Eq. (3.14) become:

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k\hat{x}_k^-) \tag{3.47}$$

with the *a posteriori* estimation covariance

$$P_k = (I - K_kH_k)P_k^- \tag{3.48}$$

where

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \tag{3.49}$$

In summary, when the process noise and system noise are correlated, the best estimate can be obtained using KF with the equations Eqs. (3.43), (3.44) and (3.47) to (3.49), which are derived from the conditional distribution of joint Gaussian.

3.5.2 Kalman Filter with Colored Measurement Noise

When the measurement noise has color, there are a couple of ways to obtain the optimal estimate [88]. The approaches can be roughly categorized into two types. One is measurement differencing [89] [90]. The idea is to obtain a new process model with the difference of two time step measurements as an auxiliary signal to remove the correlation. The other method is called state augment [91] [92]. It augments the system state vector to include the colored measurement noise. We present two common methods in this section.

The process and observation models of the system with colored measurement noise remain Eqs. (3.4) and (3.5), while the measurement noise satisfies:

$$\begin{aligned}
&v_k = \psi_{k-1}v_{k-1} + \varepsilon_{k-1}\\
&\varepsilon_k \sim \mathcal{N}(0, R_k)\\
&E[w_k\varepsilon_j^T] = 0\\
&E\{v_kv_{k-1}^T\} = E[(\psi_{k-1}v_{k-1} + \varepsilon_{k-1})v_{k-1}^T] = \psi_{k-1}E[v_{k-1}v_{k-1}^T]
\end{aligned} \tag{3.50}$$

Measurement differencing

The first method [89] is to define an auxiliary signal y_k using measurement differencing as follows:

$$\begin{aligned}
y_k &= z_{k+1} - \psi_kz_k\\
&= (H_{k+1}x_{k+1} + v_{k+1}) - \psi_k(H_kx_k + v_k)\\
&= (H_{k+1}A_k - \psi_kH_k)x_k + H_{k+1}w_k + v_{k+1} - \psi_kv_k\\
&= (H_{k+1}A_k - \psi_kH_k)x_k + H_{k+1}w_k + \varepsilon_k\\
&= H_k^*x_k + v_k^*
\end{aligned} \tag{3.51}$$

where $H_k^* = H_{k+1}A_{k+1} - \psi_kH_k$ and $v_k^* = H_{k+1}w_k + \varepsilon_k$. Then, the new but equivalent system can therefore be written as:

$$\begin{aligned}
x_k &= A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + w_{k-1}\\
y_k &= H_k^*x_k + v_k^*\\
{R_k}^* &= E[v_k^*v_k^{*T}] = H_{k+1}Q_kH_{k+1}^T + R_k\\
M_k &= E[w_kv_k^{*T}] = E\big[w_k(H_{k+1}w_k + \varepsilon_k)^T\big] = Q_kH_{k+1}^T
\end{aligned} \tag{3.52}$$

where ${R_k}^*$ is the covariance of the new measurement noise and M_k is the cross covariance between the process and measurement noise.

Using the KF equations Eqs. (3.43) and (3.47) for correlated noise with these new models, we can obtain the estimate of the *a priori* and the *a posteriori* state estimates:

$$\begin{aligned}
\hat{x}_k^- &= E[x_k|y_1, \cdots y_{k-1}]\\
\hat{x}_k &= E[x_k|y_1, \cdots y_k] = \hat{x}_k^- + K_k(y_k - H_k^*\hat{x}_k^-)
\end{aligned} \tag{3.53}$$

Since y_k is based on z_{k+1}, this means that $\hat{x}_k^-$ is the best estimate and $\hat{x}_k$ is the single state smoothing solution.

State augment

Another method [91] is to augment the original system model as follows:

$$\begin{bmatrix} x_k \\ v_k \end{bmatrix} = \begin{bmatrix} A_{k-1} & 0 \\ 0 & \psi_{k-1} \end{bmatrix} \begin{bmatrix} x_{k-1} \\ v_{k-1} \end{bmatrix} + \begin{bmatrix} B_{k-1} \\ 0 \end{bmatrix} u_{k-1} + \begin{bmatrix} w_{k-1} \\ \varepsilon_{k-1} \end{bmatrix} \tag{3.54}$$

Then the measurement model becomes:

$$z_k = \begin{bmatrix} H_k & I \end{bmatrix} \begin{bmatrix} x_k \\ v_k \end{bmatrix} + 0 \tag{3.55}$$

The covariance of the process noise is:

$$E\left[\begin{pmatrix} w_k \\ \varepsilon_k \end{pmatrix} \begin{pmatrix} w_k^T & \varepsilon_k^T \end{pmatrix}\right] = \begin{bmatrix} Q_k & 0 \\ 0 & R_k \end{bmatrix} \tag{3.56}$$

and there is no measurement noise. The optimal estimate of this system can be obtained using the standard KF equations Eqs. (3.9) to (3.13).

3.6 Summary

This chapter has introduced the two steps of KF for estimating the system state including prediction and update phases. The diagram is summarized in Fig. 3.2 and there are five key equations Eqs. (3.9) to (3.13) for the uncorrelated Gaussian noise systems. The optimality of KF is deeply examined. The KF equations can be obtained from conditional expectation and MAP estimation, which demonstrates that KF produces the optimal estimate for the system with white noise.

When the process noise and the measurement noise are correlated, the best estimate can be obtained using Eqs. (3.43), (3.44) and (3.47) to (3.49), which are derived from the conditional distribution of joint Gaussian. These equations are useful in practical. For example, when we try to find the system parameters using the Matlab system identification toolbox, the process noise and the measurement noise are assumed to be the same source, which are correlated.

Moreover, for the system with colored measurement noise, which means the noise is time correlated, we have presented two common methods to deal with this problem including time differencing and state augment. The first one obtains a new measurement model by using the difference of two time step measurements to reduce the correlation. It needs to further use the solution for dealing with correlated noise, since the new measurement noise is correlated with the process noise. The second method augments the original system model to include the process of the colored measurement noise. There is no measurement noise in the new observation model, which is equivalent to say that the measurement noise is white with a mean of zero and a covariance of zero. The optimal estimate can be obtained using the stand KF equations.

4 PKF: Transmission Rate Compression Based on KF Using Temporal Correlation

4.1 Introduction

As concluded in Chapter 2, radio communication is one of the most energy-intensive processes in WSNs and an efficient way to reduce it is to suppress the transmission rate of sensor nodes. This chapter presents our compression scheme, called PKF (predictor combined with KF), for cluster based WSNs. Different from previous methods, PKF aims to optimally reconstruct the system state from the compressed information as illustrated in Section 4.2.1. To estimate the reconstruction of PKF and find the effect of the system parameters, Section 4.2.2 employs artificial systems, where the real states are known. The reconstructed signal of PKF is compared with the raw data, the KF-optimal and the real state. The transmission rate and the corresponding reconstruction quality are numerically examined under different system parameters.

In order to understand the underlying process of PKF and estimate the effect of the system parameters, we formulate the trade-off between energy efficiency and reconstruction quality through an in-depth mathematical analysis in Section 4.3. The distribution of the prediction error is studied in Section 4.3.1. Based on this analysis, Section 4.3.2 models the process of PKF as a Markov chain and obtains the transmission rate as a function of the system parameters. The reconstruction accuracy of PKF is estimated by the covariance of the reconstruction error. It is formulated in Section 4.3.3 with the help of truncated MVN (MVN) distribution and we further reduce the computation complexity by an approximated uniform distribution. The mathematical analysis is validated in Section 4.3.4 by comparing the theoretical results with the simulated results in Section 4.2.2. This study is important for understanding PKF but also vital for extending it to further exploit spatial correlation in Chapter 5.

4.2 Functionality of the PKF Approach

4.2.1 Compression Strategy and Reconstruction Solution

In this section, we describe our PKF scheme. As shown in Fig. 4.1, we assume that sensor nodes have already formed sets of clusters, according to a certain clustering algorithm, such as LEACH [64], Directed Diffusion [93] or CAG [94]. The leaf nodes are in charge of collecting information about surrounding environment and transmitting the data packets to the cluster heads. The cluster heads then forward the information to the sink node. The goal of a WSN is to understand the monitoring system from the observations of the sensor nodes. The objective of PKF is to suppress the communication between a leaf node and a cluster head, while optimally reconstructing the system state with a guaranteed quality using the compressed information.

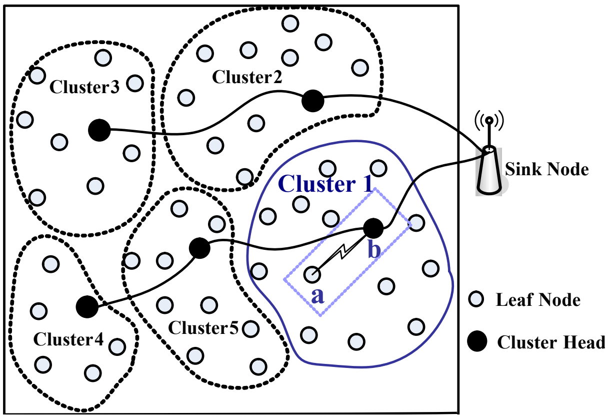


Figure 4.1: A cluster-based WSN.

To achieve communication compression, each node can intermittently transmit either the raw data or the preprocessed data based on some suppression schemes. Since the observations of the leaf nodes are usually corrupted by noise, the latter method would produce more accurate reconstructions. There are different methods to remove the noise of the raw data, such as EWMA filter, LMS filter, NLMS filter, KF and so on. Among them, KF can not only remove the noise but also provide the optimal estimate of the system state as introduced in Chapter 3. This coincides exactly with the objective of the WSNs. Thus, each leaf node is required to firstly execute a KF to filter noise and produce the optimal estimate of the system state. When a transmission is needed, the current optimal state is transmitted instead of the raw data, which calibrates the estimation of the cluster head to the optimal one and resets the cumulative error. Now we examine the suppression strategy and the optimal reconstruction scheme from Bayesian estimation.

Let $\hat{\boldsymbol{X}}_{\boldsymbol{k}} = [\hat{x}_1, \cdots \hat{x}_k]$ collect all the local estimates of a leaf node from time 1 to time k. A suppression strategy specifies which estimates in $\hat{\boldsymbol{X}}_{\boldsymbol{k}}$ would be transmitted to the cluster head in the k time instants duration. Under different strategies, $\hat{\boldsymbol{X}}_{\boldsymbol{k}}$ changes. While in any case, it is a subset of $\hat{\boldsymbol{X}}_{\boldsymbol{k}}$ and we define it as $\hat{\boldsymbol{X}}_{\boldsymbol{k\_s}}$. The last element of $\hat{\boldsymbol{X}}_{\boldsymbol{k\_s}}$ is the latest transmitted data $\hat{x}_j$. For example, considering the transmitted data from time 1 to time 5 are $\hat{x}_1$ and $\hat{x}_3$, then $\hat{\boldsymbol{X}}_{\boldsymbol{5\_s}} = [\hat{x}_1,\ \hat{x}_3] \subset \hat{\boldsymbol{X}}_{\boldsymbol{5}} = [\hat{x}_1,\ \hat{x}_2,\ \hat{x}_3,\ \hat{x}_4,\ \hat{x}_5]$. Then the best estimator of x_k based on $\hat{\boldsymbol{X}}_{\boldsymbol{k\_s}}$, denoted as $\tilde{x}_k$, is the conditional expectation $E[x_k|\hat{\boldsymbol{X}}_{\boldsymbol{k\_s}}]$. Since $\hat{\boldsymbol{X}}_{\boldsymbol{k\_s}} \subset \hat{\boldsymbol{X}}_{\boldsymbol{k}}$, it can be simplified using the tower property [95] of the conditional expectation:

$$\tilde{x}_k = E[x_k|\hat{\boldsymbol{X}}_{\boldsymbol{k\_s}}] = E[E(x_k|\hat{\boldsymbol{X}}_{\boldsymbol{k}})|\hat{\boldsymbol{X}}_{\boldsymbol{k\_s}}] = E[\hat{x}_k|\hat{\boldsymbol{X}}_{\boldsymbol{k\_s}}] \tag{4.1}$$

The equation implies that the best estimator of x_k based on the received data sequence $\hat{\boldsymbol{X}}_{\boldsymbol{k\_s}}$ is equivalent to the best estimator of $\hat{x}_k$ using $\hat{\boldsymbol{X}}_{\boldsymbol{k\_s}}$. Since the state estimate of KF contains all the past information, i.e., $\hat{x}_k = E[x_k \mid \boldsymbol{Z_k}, \boldsymbol{U_k}]$, as derived in Chapter 3, we can derive that $E[\hat{x}_k|\hat{\boldsymbol{X}}_{\boldsymbol{k\_s}}]$ is only related with the last received data. Assuming the last received data is $\hat{x}_k$, then $\tilde{x}_k = E[\hat{x}_k|\hat{\boldsymbol{X}}_{\boldsymbol{k\_s}}] = E[\hat{x}_k \mid \hat{x}_k] = \hat{x}_k$. If the last received data is $\hat{x}_{k-1}$, then $E[\hat{x}_k|\hat{\boldsymbol{X}}_{\boldsymbol{k\_s}}] = E[\hat{x}_k \mid \hat{x}_{k-1}] = A_{k-1}\hat{x}_{k-1} + B_{k-1}u_{k-1} = \hat{x}_k^-$, which is the 1-step ahead KF predictor. For a general case, assuming $\hat{x}_j$ $(j \leq k)$ is the last element in $\hat{\boldsymbol{X}}_{\boldsymbol{k\_s}}$, then the best estimator is the k-step ahead KF predictor, namely $\tilde{x}_k = E[\hat{x}_k|\hat{\boldsymbol{X}}_{\boldsymbol{k\_s}}] = E[\hat{x}_k|\hat{x}_j] = \big(\prod_{i=j}^{k-1} A_i\big)\hat{x}_j + \sum_{i=j}^{k-1}\big(\prod_{m=k-1}^{i+1} A_m\big)B_i u_i$. In the time invariant system model, $A_k = A$, $E[\hat{x}_k|\hat{x}_j] = A^{k-j}\hat{x}_j + \sum_{i=j}^{k-1}\big(\prod_{m=k-1}^{i+1} A\big)B_i u_i$. Thus, the best estimator based on the received data sequence written in a recursive form is a linear predictor:

$$\tilde{x}_k = A_{k-1}\tilde{x}_{k-1} + B_{k-1}u_{k-1} \tag{4.2}$$

It uses the received optimal value to replace the current prediction and further predict the states of the next several time instants, which is equivalent to the k-step ahead KF predictor. Once the cluster head receives $\hat{x}_k$, all the past information of the observation is also obtained, which is more accurate than transmitting the raw data z_k that only contains the current information.

Depending on different suppression strategies, the received data sequence $\hat{\boldsymbol{X}}_{\boldsymbol{k\_s}}$ changes and the reconstruction quality varies. To guarantee the reconstruction accuracy, the predicted observation, $\tilde{z}_k = H_k\tilde{x}_k$, should be as close to the ideal observation $H_k x_k$ as possible. This implies that a threshold to restrict the prediction error is needed. Since x_k is unknown to each node, we can use $\hat{x}_k$ instead to calculate the error $\epsilon_k = H_k(\tilde{x}_k - \hat{x}_k)$

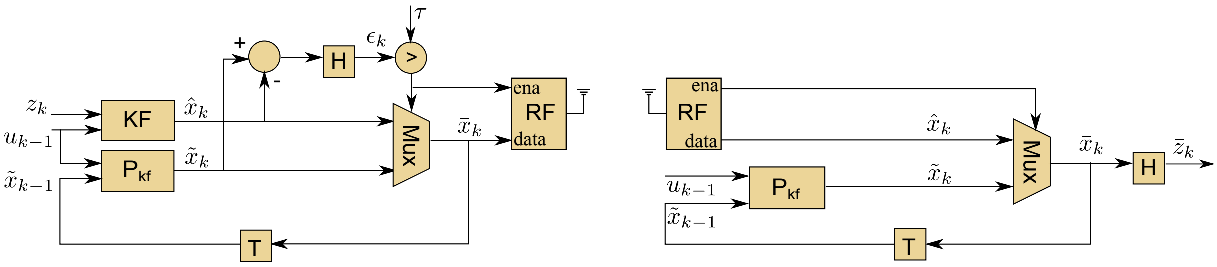
because of Eq. (4.1). Then each node needs to follow the prediction of the cluster head and keeps the prediction error ϵ_k within the threshold interval. When the absolute value of the error is larger than the threshold, the current optimal value is transmitted. Note that, in this case, $\hat{\boldsymbol{X}}^{\boldsymbol{i}}_{\boldsymbol{k\_s}}$ consists of two components. One is the transmitted data sequence. Each element is a Gaussian random variable. The other is the accuracy indication of the prediction, which can be treated as a Boolean indicator. It equals 0, when the cluster head receives the optimal value and indicates the prediction is inaccurate; otherwise, it equals 1 indicating that the prediction of the untransmitted state is accurate enough. The reconstruction problem in Eq. (4.1) is actually nonlinear. Even so, the linear predictor is still surprisingly the optimal estimator.

Taking cluster 1 in Fig. 4.1 as an example, we present the whole process of PKF in the following. The leaf node a firstly performs a KF to reduce the measurement noise and obtain the optimal estimate of the state, $\hat{x}_k$, using Eqs. (3.9) to (3.13). The cluster head b uses Eq. (4.2) to obtain the best estimation of the current state based on the received information. To guarantee the reconstruction quality, the node a follows the estimation of the cluster head b by synchronously executing the same predictor and calculates the prediction error ϵ_k. When the absolute value of ϵ_k exceeds a given threshold τ, the optimal value $\hat{x}_k$, is transmitted to head b. The current prediction $\tilde{x}_k$, of both node a and head b are replaced by $\hat{x}_k$. Thus, the final reconstructed value of the cluster head b is $\bar{z}_k = H_k\bar{x}_k$, where

$$\bar{x}_k = \begin{cases} \tilde{x}_k, & \text{if } ||\epsilon_k|| \leq \tau \\ \hat{x}_k, & \text{otherwise} \end{cases} \tag{4.3}$$

Note that PKF allows multiple sensor types to be encoded in a single state vector. When the node has many sensors to measure different kinds of data, the observation z_k is a vector and correspondingly ϵ_k is a vector. We compare the norm of ϵ_k with the constant threshold τ, which depends on the requirement of a specific application. The diagram of PKF is shown in Fig. 4.2, where $\mathbf{P_{kf}}$ denotes the k-step ahead Kalman predictor. Accordingly, Algorithm 1 and Algorithm 2 depict the process of PKF performed by the leaf node and the cluster head, respectively, where Algorithm 2 is executed by the cluster head for each leaf node.

In addition, PKF is suitable for both time variant and invariant systems in accordance with the KF. Without loss of generality, the system parameters in the algorithms, e.g. A_k, B_k and H_k, are denoted to be time variant, which does not affect the underline process of PKF. For time invariant systems, these parameters can be found offline by analyzing the historical data as done in Chapter 6, while for time variant systems, they can be



The diagram of PKF-en in slave node

The diagram of PKF-de in the cluster head

Figure 4.2: The block diagram of PKF.

Algorithm 1 PKF algorithm performed by the leaf node

```
Initialize $\hat{x}_0$, $\hat{P}_0$ and $\tilde{x}_0$
for each $z_k$ do
    Calculate $\hat{x}_k$ using Eqs. (3.9) to (3.13)
    Follow the prediction of the cluster head $\tilde{x}_k$ using Eq. (4.2)
    Calculate the prediction error $\epsilon_k = H_k(\tilde{x}_k - \hat{x}_k)$
    if $||\epsilon_k|| > \tau$ then
        Send the current optimal value $\hat{x}_k$
        Replace the current prediction by the optimal value $\tilde{x}_k = \hat{x}_k$
    end if
end for
```

Algorithm 2 PKF algorithm performed by the cluster head

```
Initialize $\tilde{x}_0$
for each prediction time do
    if the update $\hat{x}_k$ is available then
        $\tilde{x}_k = \hat{x}_k$
    else
        Predict $\tilde{x}_k$ using Eq. (4.2)
    end if
    Obtain the current reconstructed state $\bar{x}_k = \tilde{x}_k$
    Obtain the current reconstructed observation $\bar{z}_k = H_k\bar{x}_k$
end for
```

updated using offline or online methods. For example, we can obtain the evolution of these parameters as functions of time offline using historical data and store these functions in leaf node and cluster head to update their models during the running time as illustrated in Section 6.2. The model update process can also be done online using an approach similar to [29]. When the transmission rate is higher than an expected value, the update phase is triggered in the cluster head. The analyzed parameters are transmitted to the leaf node to synchronize the model. The precise schemes to find the optimal online modeling method are beyond the scope of this work.

In summary, PKF achieves data filtering, state estimation, data compression and reconstruction within one KF framework. It provides the optimal reconstructions based on the compressed information, which have the restricted error bound compared with the ideal observation.

4.2.2 The Reconstruction of PKF Using Simulated Systems

In this section, we use artificial systems to estimate the reconstructions of PKF. Since the real states are known in this case, the comparisons can be done precisely. We firstly generate an arbitrary system and set up a threshold to examine the reconstructed signal of PKF. Then various thresholds and different covariances of measurement noise are used to find the effect on the performance of PKF.

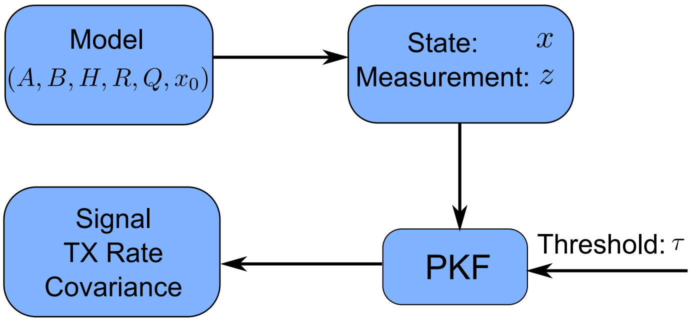


Figure 4.3: The procedure of the first experiment.

The procedure of the first experiment is depicted in Fig. 4.3. We firstly generate a system using a model including the state and measurement. Then assign a threshold to execute PKF. A section of the reconstructed signals is presented, and the transmission rate and the covariance of the reconstruction errors are compared. The parameters of the system and the threshold are listed in Table 4.1. There are total 2^{16} data points in our simulation and the pseudo-random values of the noise are drawn from the standard normal distribution.

Table 4.1: System parameters and the threshold for estimating the reconstruction of PKF.

System and user parameters						
A	B	H	Q	x_0	R	τ
0.9	0	1	0.01	0	0.25	0.1

Fig. 4.4 depicts a section of the reconstructed signal of PKF in comparison with the raw data, the optimal estimate of KF (KF-optimal) and the real state. There are 21

out of 100 transmitted points during this time. Obviously, the KF successfully removes the noise of the raw data and the estimates are closer to the real state. PKF follows the estimate of KF pretty well and replaces the inaccurate prediction as shown in Fig. 4.4. The reconstructed signal is even closer to the real state compared with transmitting all of the raw data.

In the following, we use the numbers of transmission rate and covariance of reconstruction errors to measure the performance of PKF, instead of presenting the signals. The obtained transmission rate using PKF is 19.75% under this setup. The covariance of the reconstruction errors compared with the real state is 0.03, which is around 7 times smaller than the measurement noise R. Fig. 4.5 compares the distributions of the measurement noise, and the errors between KF-optimal and the real state. Most errors of KF-optimal are located in the range of $[-0.5 \quad 0.5]$, while the measurement noise is relatively larger and located in the range $[-1.5 \quad 1.5]$. The reconstruction errors of PKF compared with the KF-optimal are bounded in the interval of $[-0.1 \quad 0.1]$ as depicted in the Fig. 4.6b. The variance is much smaller than the variance of raw data w.r.t. KF-optimal as shown in Fig. 4.6a. Compared with the real state, the covariance of reconstruction errors of PKF is much smaller than the variance of the measurement noise as shown in Fig. 4.7.

In the second experiment, the system states and measurements are kept unchanged, while the threshold is tuned from 0 to 0.22 to find the effect on the performance of PKF. As shown in Fig. 4.8a and Fig. 4.8b (see the magenta lines), as the threshold increases, the transmission rate dramatically decreases and the covariance of reconstruction errors w.r.t. the KF-optimal increases. Combining the results of transmission rate and reconstruction quality, we obtain the trade-off between them as shown in Fig. 4.9a. There is no doubt that for a high reconstruction quality (small covariance of reconstruction errors), more transmission rate is required. The same trend holds for the trade-off between the transmission rate and the covariance of the reconstruction error w.r.t. the real state as shown in Fig. 4.9b, where the covaraince of reconstruction errors of PKF is compared with the covariance of measurement noise of raw data R. Surprisingly, the reconstruction of PKF with very few transmission could be even more accurate than always transmitting the raw data in this system.

One advantage of PKF is the use of KF that provides the optimal estimation of the state from the noisy measurements. In order to find how the measurement noise affects the gain of PKF, we performed an additional experiment. Here, we keep the system states unchanged and adjust the covariance R to obtain two more different measurement sets. The setup is equivalent to have three nodes to measuring the same system, but with

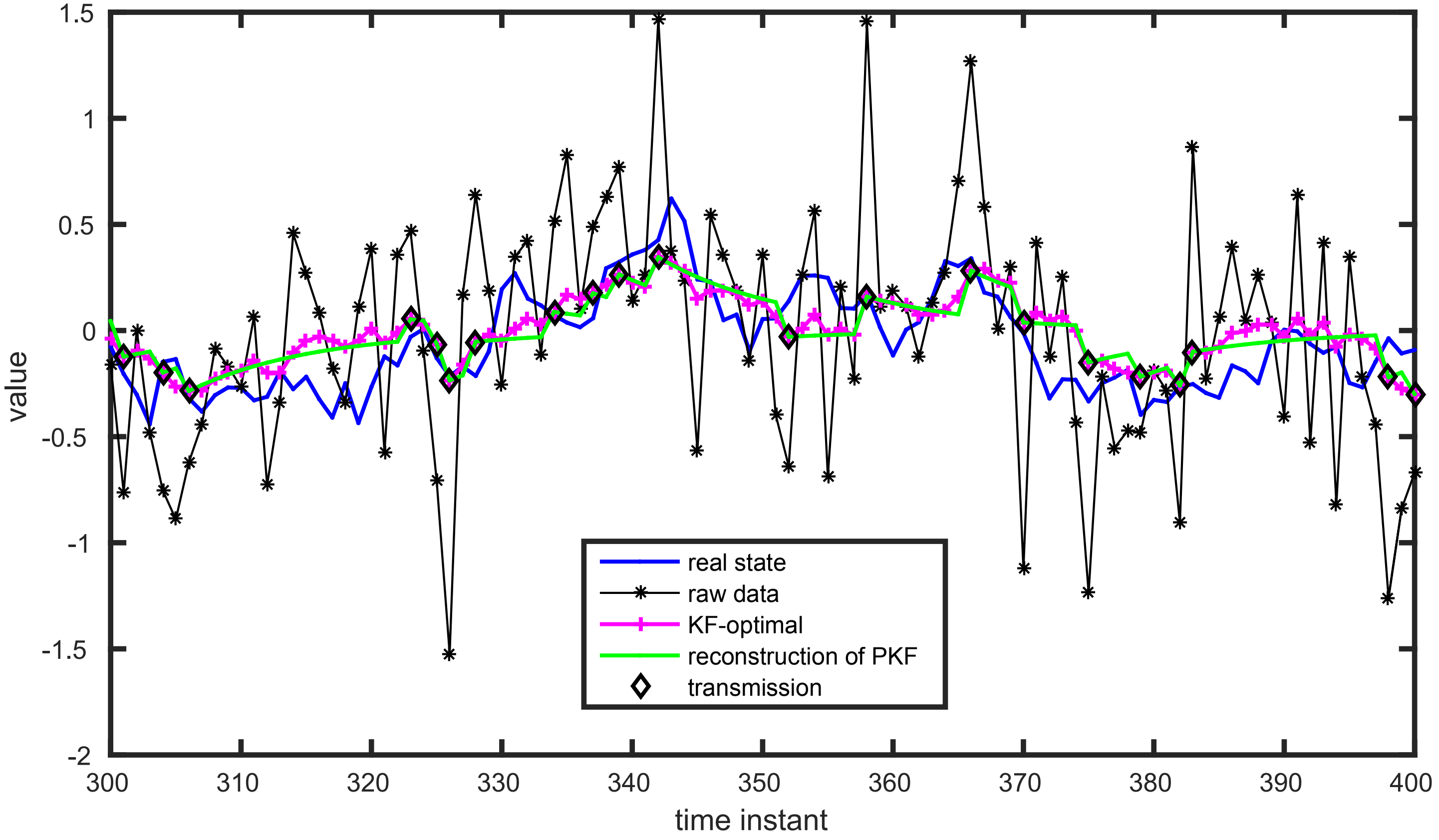
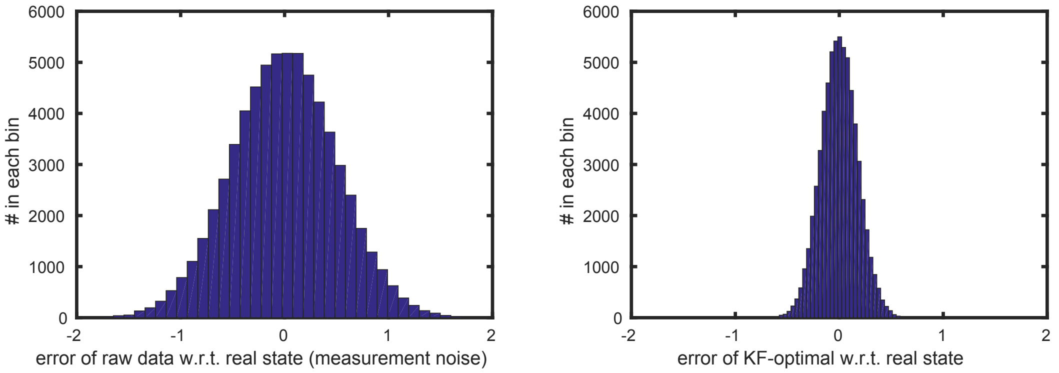


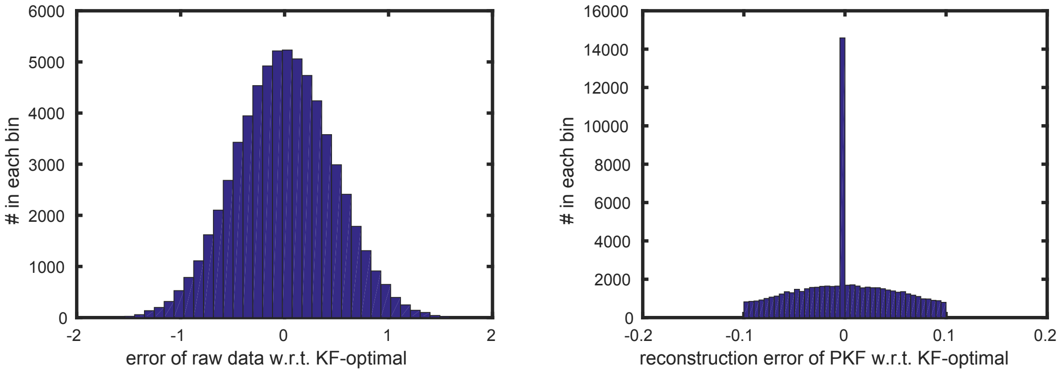
Figure 4.4: A section of reconstructed signals of PKF compared with the raw data, the KF-optimal and the real state in an artificial system, when the transmission rate of the node is 19.75%.



(a)

(b)

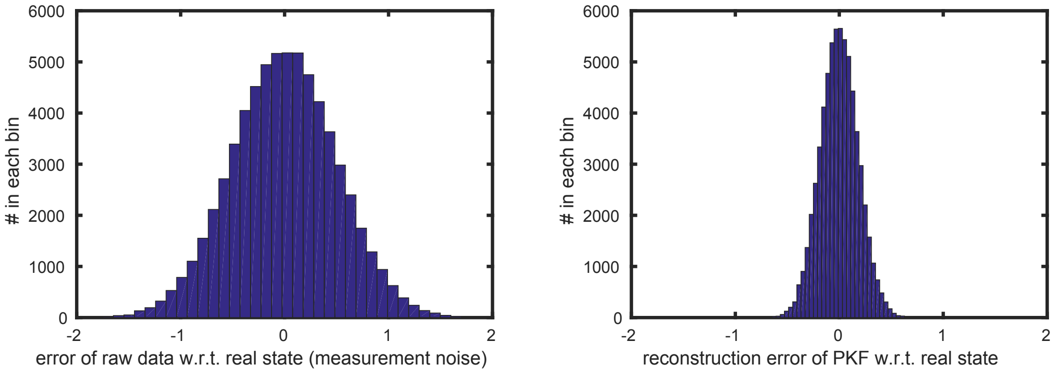
Figure 4.5: The error distribution of (a) the raw data and (b) the KF-optimal compared with real state.



(a)

(b)

Figure 4.6: The error of (a) the raw data and (b) the PKF reconstruction compared with KF-optimal.



(a)

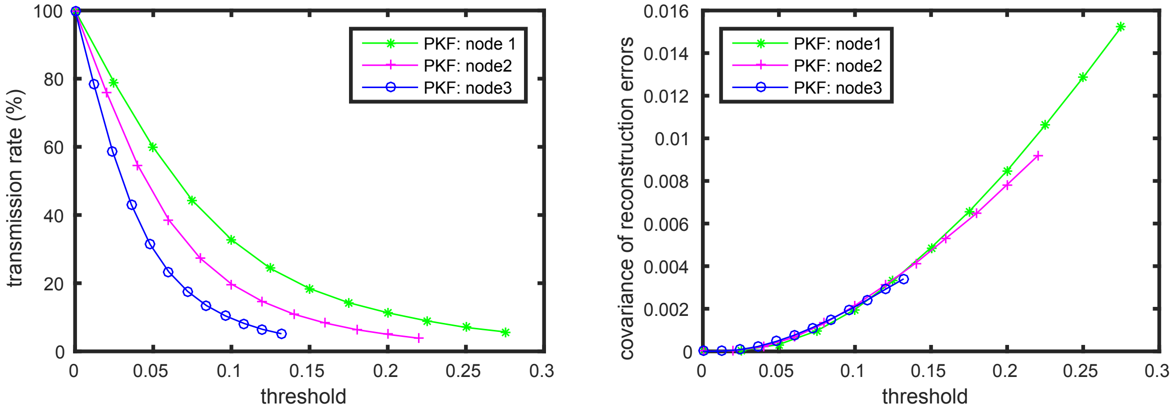
(b)

Figure 4.7: The error of (a) the raw data and (b) the PKF reconstruction compared with real state.

different measurement noises. The covariances of the measurement noise are listed in Table 4.2.

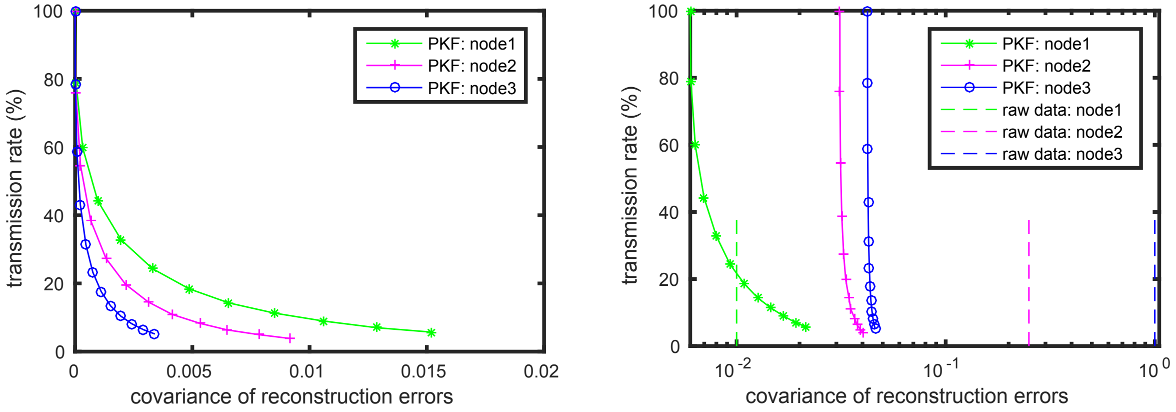
Table 4.2: The covariance of measurement noise of three nodes.

Node ID	1	2	3
R	0.01	0.25	1



(a) (b)

Figure 4.8: (a) Threshold vs. transmission rate; (b) Threshold vs. covariance of reconstruction errors w.r.t. KF-optimal, in node 1, node 2 and node 3, respectively.



(a) (b)

Figure 4.9: Trade-off between transmission rate and covariance of reconstruction errors (a) w.r.t. KF-optimal; (b) w.r.t. real state, in node 1, node 2 and node 3, respectively.

The basic relationships among the threshold, the transmission rate and the covariance of reconstruction errors, for node 2 still hold for node 1 and node 3 as shown in

Fig. 4.8a and Fig. 4.8b. As the threshold increases, the node requires less transmission rate, while producing higher reconstruction errors. The trade-offs between them are depicted in Fig. 4.9a. Under the same transmission rate, the higher measurement noise the node has, the smaller covariance of reconstruction errors (w.r.t. the KF-optimal) PKF produces. This is because the deviation between the *a priori* prediction and the KF-optimal decreases, as the measurement noise increases for a given process model. The KF trusts more on the *a priori* prediction than the noisy measurements. It indicates that the advantage by using PKF versus no compression increases as the measurement noise increases. The more obvious evidence is reported in Fig. 4.9b. It compares the covariance of reconstruction errors of PKF with the covariance of measurement noise of raw data as the transmission rate decreases. When $R = 0.01$ in node 1, PKF needs around 21.69% transmission to achieve the same reconstruction quality as transmitting all of the raw data. As R increases to 0.25 in node 2, the reconstruction quality of PKF using 21.69% transmission can be increased by around 6 times compared with transmitting all of the raw data. The improvement is even larger when R increases to 1 in node 3. In other words, the superiority of the reconstruction quality of PKF compared with the raw data becomes larger, as the measurement noise increases.

4.3 Mathematical Analysis of PKF

For a given node, the value of the threshold τ affects not only the communication energy cost but also the reconstruction accuracy. As the threshold increases, the node needs to transmit fewer packets, whereas the reconstruction quality decreases. For different nodes with different measurement noises, the communication rate and the reconstruction quality of PKF vary under a given threshold. The bigger the measurement noise is, the higher superiority PKF achieves. In the following, we aim to mathematically quantify the relationship among those quantities to evaluate the performance of PKF. For the sake of conciseness, we use a time invariant system here.

4.3.1 Distribution of the Prediction Error

In this section, we analyze the distribution of the single step ahead prediction error and find the joint distribution of the sequence of errors produced by different step ahead predictions, based on the theory from Chapter 3. It is the basis for the analysis in Section 4.3.2 and Section 4.3.3.

Given an initial value, $\tilde{x}_0 = \hat{x}_0$, we can obtain the k-step ahead prediction, $\tilde{x}_k = A^k\tilde{x}_0$ using Eq. (4.2). The k-step ahead prediction error is equivalent to the prediction error at time instant k, $\epsilon_k = H(\tilde{x}_k - \hat{x}_k)$. However, considering the inaccurate prediction at a time instant could be replaced by the optimal value, these two errors could be different. For example, assuming we have a sequence of reconstructions $\tilde{x}_1, \hat{x}_2, \tilde{x}_3, \cdots$, the prediction errors at time instant 1 is also the 1-step ahead prediction error evolving from $\hat{x}_0$, while the prediction error at time instant 3 becomes the 1-step ahead prediction error coming from $\hat{x}_2$. We firstly illustrate that the errors from k-step ahead prediction have the same distribution, no matter of the starting time instant. In this example, we are going to prove that ϵ_1 and ϵ_3 have the same distribution under the assumption that KF has converged. Because of the Gaussian assumption of KF, the prediction error at each time instant is a random variable that satisfies a normal distribution. Thus we can calculate the mean and the covariance of ϵ_1 and ϵ_3 to validate our assumption. By using Eqs. (3.4), (3.5), (3.9) and (3.12), we can obtain that:

$$\begin{aligned}
\epsilon_1 &= H(\tilde{x}_1 - \hat{x}_1)\\
&= H(A\hat{x}_0 - (I - K_1H)A\hat{x}_0 - K_1HAx_0 - K_1Hw_0 - K_1v_1)\\
&= H\big(K_1HA(\hat{x}_0 - x_0) - K_1Hw_0 - K_1v_1\big)\\
\epsilon_3 &= H(\tilde{x}_3 - \hat{x}_3)\\
&= H(A\hat{x}_2 - (I - K_3H)A\hat{x}_2 - K_3HAx_2 - K_3Hw_2 - K_3v_3)\\
&= H\big(K_3HA(\hat{x}_2 - x_2) - K_3Hw_2 - K_3v_3\big)
\end{aligned} \tag{4.4}$$

The mean of them can be calculated by:

$$\begin{aligned}
E[\epsilon_1] &= E[H\big(K_1HA(\hat{x}_0 - x_0) - K_1Hw_0 - K_1v_1\big)]\\
&= HK_1HA(E[\hat{x}_0] - E[x_0])\\
&= HK_1HA(\hat{x}_0 - \hat{x}_0) = 0\\
E[\epsilon_3] &= E[H\big(K_3HA(\hat{x}_2 - x_2) - K_3Hw_2 - K_3v_3\big)]\\
&= HK_3HA(E[\hat{x}_2] - E[x_2])\\
&= HK_3HA(\hat{x}_2 - \hat{x}_2) = 0
\end{aligned}$$

since the mean values of v_k and w_k are zero and the mean of x_k is $\hat{x}_k$ as introduced in

Chapter 3. The covariance of them are:

$$\begin{aligned}
E[\epsilon_1\epsilon_1^T] &= HK_1H(AP_0A^T + Q_0)H^TK_1^TH^T + HK_1R_1K_1^TH^T \\
&= HK_1(HP_1^-H^T + R_1)K_1^TH^T = HP_1^-H^TK_1^TH^T \\
E[\epsilon_3\epsilon_3^T] &= HK_3H(AP_2A^T + Q_2)H^TK_3^TH^T + HK_3R_3K_3^TH^T \\
&= HK_3(HP_3^-H^T + R_3)K_3^TH^T = HP_3^-H^TK_3^TH^T
\end{aligned} \tag{4.5}$$

When KF has converged, $P_k^- = P$ and $K_k = K$. Thus, we can obtain:

$$E[\epsilon_1\epsilon_1^T] = E[\epsilon_3\epsilon_3^T] = HP^-H'K'H'$$

Then the 1-step ahead prediction error evolving from $\hat{x}_0$, ϵ_1 and the 1-step ahead prediction error coming from $\hat{x}_2$, ϵ_3 have the same normal distribution. Thus, we can conclude that the errors of k-step ahead prediction, no matter starting from which time point, have the same distribution. We use the notation ε_k to denote the k-step ahead prediction error. In this example, both ϵ_1 and ϵ_3 are the 1-step ahead prediction error ε_1.

Without loss of generality, we can use ϵ_k to find the distribution of ε_k assuming that the head continues predicting without replacement. As k increases, it is too complex to obtain an explicit equation for ε_k using the iteration method used in Eq. (4.4). In the following, we aim to describe it in a recursive way.

Let $e_k = \tilde{x}_k - \hat{x}_k$ denote the prediction error of the state w.r.t. the KF-optimal, $\hat{e}_k = \hat{x}_k - x_k$ is the *a posteriori* estimate error of KF, and $\tilde{e}_k = \tilde{x}_k - x_k$ represents the prediction error with respect to the true state. The diagram of the notations is shown in Fig. 4.10. Then, $e_k = \tilde{e}_k - \hat{e}_k$ and the k-step ahead prediction error can be recalculated as:

$$\varepsilon_k = \epsilon_k = He_k = H[-I \quad I][\hat{e}_k \quad \tilde{e}_k]^T = H[-I \quad I]\, ee_k \tag{4.6}$$

where $ee_k = [\hat{e}_k \quad \tilde{e}_k]^T$. Initially, we have $\tilde{x}_0 = \hat{x}_0$, thus $\tilde{e}_0 = \hat{e}_0$ and $ee_0 = [\hat{e}_0 \quad \hat{e}_0]^T$. From Eqs. (3.4), (3.5), (3.9), (3.12) and (4.2), the errors, $\hat{e}_k$ and $\tilde{e}_k$, can be formulated in a recursive way:

$$\begin{aligned}
\hat{e}_k &= (I - KH)A\hat{e}_{k-1} + (KH - I)w_{k-1} + Kv_k \\
\tilde{e}_k &= A\tilde{e}_{k-1} - w_{k-1}
\end{aligned} \tag{4.7}$$

Thus, the error vector and the k-step ahead error satisfy:

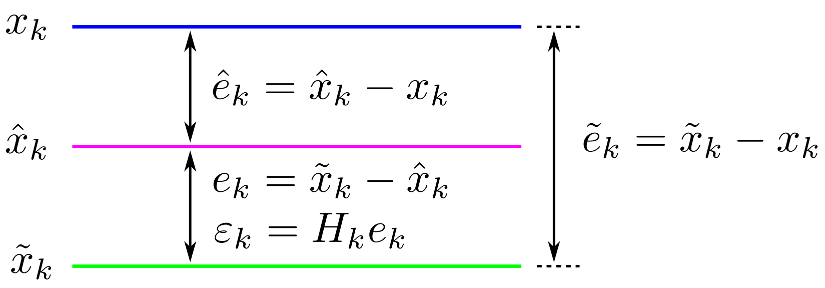

x_k
$\hat{x}_k$
$\tilde{x}_k$
$\hat{e}_k = \hat{x}_k - x_k$
$e_k = \tilde{x}_k - \hat{x}_k$
$\varepsilon_k = H_k e_k$
$\tilde{e}_k = \tilde{x}_k - x_k$

Figure 4.10: The schematic diagram of the error definitions.

$$\begin{aligned} ee_k &= \beta ee_{k-1} + C_w w_{k-1} + C_v v_k \\ \varepsilon_k &= \epsilon_k = He_k = H[-I \quad I]ee_k = H[-I \quad I](\beta ee_{k-1} + C_w w_k + C_v v_k) \end{aligned} \tag{4.8}$$

where $\beta = \begin{bmatrix} (I-KH)A & 0 \\ 0 & A \end{bmatrix}$, $C_w = \begin{bmatrix} KH - I & -I \end{bmatrix}^T$ and $C_v = \begin{bmatrix} K & 0 \end{bmatrix}^T$.

The mean of ε_k using Eqs. (3.4) and (4.2) can be calculated by:

$$\begin{aligned} E[\varepsilon_k] = E[\epsilon_k] &= E[H(\tilde{x}_k - x_k)] + E[H(x_k - \hat{x}_k)] \\ &= E[H(\tilde{x}_k - x_k)] \\ &= H(A^k \hat{x}_0 - A^k \hat{x}_0) = 0 \end{aligned} \tag{4.9}$$

The covariance of ε_k, denoted as σ_k^2, where σ_k is the standard deviation, can be obtained by calculating the covariance of ee_k. From Eqs. (3.10) and (3.13), the covariance of $\hat{e}_k$ is $P_k = P$. Let us denote the covariance of $\tilde{e}_k$ and the covariance between these two errors as $\tilde{P}_k$ and $\ddot{P}_k$, respectively. From Eq. (4.7), we can obtain that:

$$\begin{aligned} \tilde{P}_k &= E[\tilde{e}_k \, \tilde{e}_k^T] = A\tilde{P}_{k-1}A^T + Q \\ &= A^k P (A^k)^T + \sum_{i=0}^{k-1} A^i Q (A^i)^T \\ \ddot{P}_k &= E[\hat{e}_k \, \tilde{e}_k^T] = (I - KH)(A\ddot{P}_{k-1}A^T + Q) \\ &= P_k = P \end{aligned} \tag{4.10}$$

Then the covariance of ee_k is:

$$P_{ee_k} = E\left[ee_k \, ee_k^T\right] = E\left[\begin{bmatrix} \hat{e}_k & \tilde{e}_k \end{bmatrix}^T \begin{bmatrix} \hat{e}_k^T & \tilde{e}_k^T \end{bmatrix} \right]$$

$$= \begin{bmatrix} P_k & \ddot{P}_k \\ \ddot{P}_k & \tilde{P}_k \end{bmatrix} = \begin{bmatrix} P & P \\ P & \tilde{P}_k \end{bmatrix} \tag{4.11}$$

Thus, the covariance of e_k is $\tilde{P}_k - P$, which indicates that there is no correlation between e_k and $\hat{e}_k$. It is consistent with Eq. (4.1). The best prediction of the real state is the best prediction of the KF estimate. Combining Eq. (4.8), the covariance of ε_k is:

$$\begin{aligned} \sigma_k^2 &= H \begin{bmatrix} -1 & 1 \end{bmatrix} P_{ee_k} \begin{bmatrix} -1 & 1 \end{bmatrix}^T H^T \\ &= H(\tilde{P}_k - P)H^T \\ &= H\theta_k H^T \end{aligned} \tag{4.12}$$

where $\theta_k = \tilde{P}_k - P = A^k P (A^k)^T - P + \sum_{i=0}^{k-1} A^i Q (A^i)^T$. When $k = 1$, $\theta_1 = APA^T + Q - P = P^- - (I - KH)P^- = KHP^-$, then $\sigma_1^2 = HKHP^- H^T$, which is consistent with Eq. (4.5).

Therefore, the k-step ahead prediction error, ε_k, has a normal distribution, i.e., $\varepsilon_k \sim \mathcal{N}(0,\, \sigma_k^2)$. The probability density function is:

$$\varphi_{\varepsilon_k}(\chi) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{\chi^2}{2\sigma_k^2}}$$

To illustrate our above analysis, we obtain the prediction errors in our original system generated in Section 4.2.2. In this example, the covariance matrix of the *a posteriori* estimation of KF is $P = 0.15$. As shown in Fig. 4.11, the covariance of the prediction error increases[1] from 0 to 0.0222 when k increases from 1 to 100 and converges to 0.0222. The experimental obtained covariance fit quite well with the theoretical values calculated by Eq. (4.12). Moreover, the histogram of the k-step ($k = 1, 2, 20, 24$) ahead prediction error in Fig. 4.12) from the experiment satisfies a normal distribution, which is consistent with the analysis. The curve becomes shorter and wider as the standard deviation σ_k increases and remains similar when σ_k converges. Each normal curve satisfies the empirical rule [96] that around 68% of errors are within one σ_k away from the mean; around 95% of the errors lie within two σ_k; and about 99.7% are within three σ_k.

Then we analyze the distribution of the vector of k steps ahead prediction errors, $\boldsymbol{v\varepsilon_k} = [\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_k]^T \in \mathbb{R}^k$. From Eqs. (4.6) and (4.8), we can see that ε_k are linear functions of the errors from different step ahead, i.e., ε_{k-1}, ε_{k-2}, $\cdots$, ε_1. Then every linear combination of the components in $\boldsymbol{v\varepsilon_k}$ is still normally distributed. It indicates that the vector of k steps ahead prediction errors $\boldsymbol{v\varepsilon_k}$ satisfies a MVN distribution [97]. The mean

[1] Note that in some examples, σ_k may decrease then converge depending on the value of P and Q.

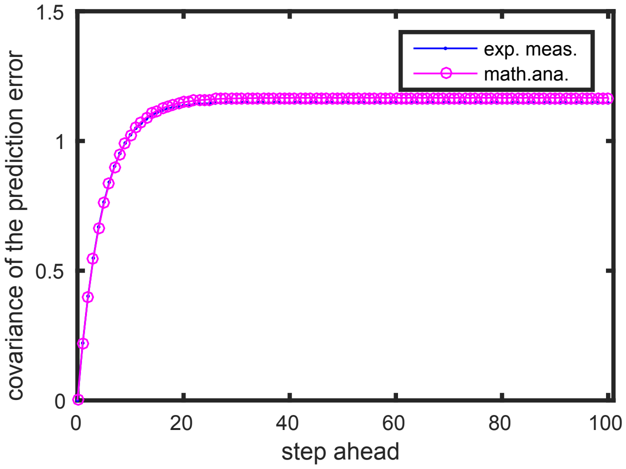


Figure 4.11: The comparison of the experimental measured and mathematical analysis of the covariance of different step ahead prediction errors

of $\boldsymbol{v\varepsilon_k}$ is a zero vector. The covariance matrix is denoted as $\boldsymbol{S_k}$. The diagonal elements of $\boldsymbol{S_k}$ are the covariance of each random variable, σ_1^2, $\sigma_2^2, \cdots, \sigma_k^2$. The $(i,\ j)$ entry of $\boldsymbol{S_k}$ is the covariance between ε_i and ε_j. When $i > j$, it can be calculated using Eqs. (4.6) and (4.8) to (4.12) as:

$$\begin{aligned} E\Big[\varepsilon_i\varepsilon_j^T\Big] &= E\Big[\epsilon_i\epsilon_j^T\Big] = E\Big[H[-I \quad I]\big(\beta^{i-j}ee_j + f(w,v,i,j)\big)ee_j^T[-I \quad I]^T H^T\Big] \\ &= E\Big[H[-I \quad I]\beta^{i-j}P_{ee_j}[-I \quad I]^T H^T\Big] \\ &= HA^{i-j}\theta_j H^T \end{aligned}$$

where $f(w,v,i,j) = \sum_{m=j+1}^{i}\beta^{i-m}\big(C_w w_m + C_v v_m\big)$. It is independent of ee_j. The $(j,\ i)$ entry of $\boldsymbol{S_k}$ is the transpose of the $(i,\ j)$ entry.

Thus, the covariance matrix $\boldsymbol{S_k}$ is:

$$\boldsymbol{S_k} = \begin{pmatrix} H\theta_1 H^T & \cdots & H\theta_1(A^{k-1})^T H^T \\ \vdots & \ddots & \vdots \\ HA^{k-1}\theta_1 H^T & \cdots & H\theta_k H^T \end{pmatrix} \tag{4.13}$$

Given a fixed i in Eq. (4.13), as j increases, the (i,j) entry of $\boldsymbol{S_k}$ gets smaller and smaller (because the eigenvalue of A is smaller than 1) and the jth diagonal element of $\boldsymbol{S_k}$ increases and converges to a constant as analyzed before. It means the correlation between the prediction errors from different steps ahead becomes lower, as the distance between them gets larger. The correlation can be characterized by Pearson's correlation

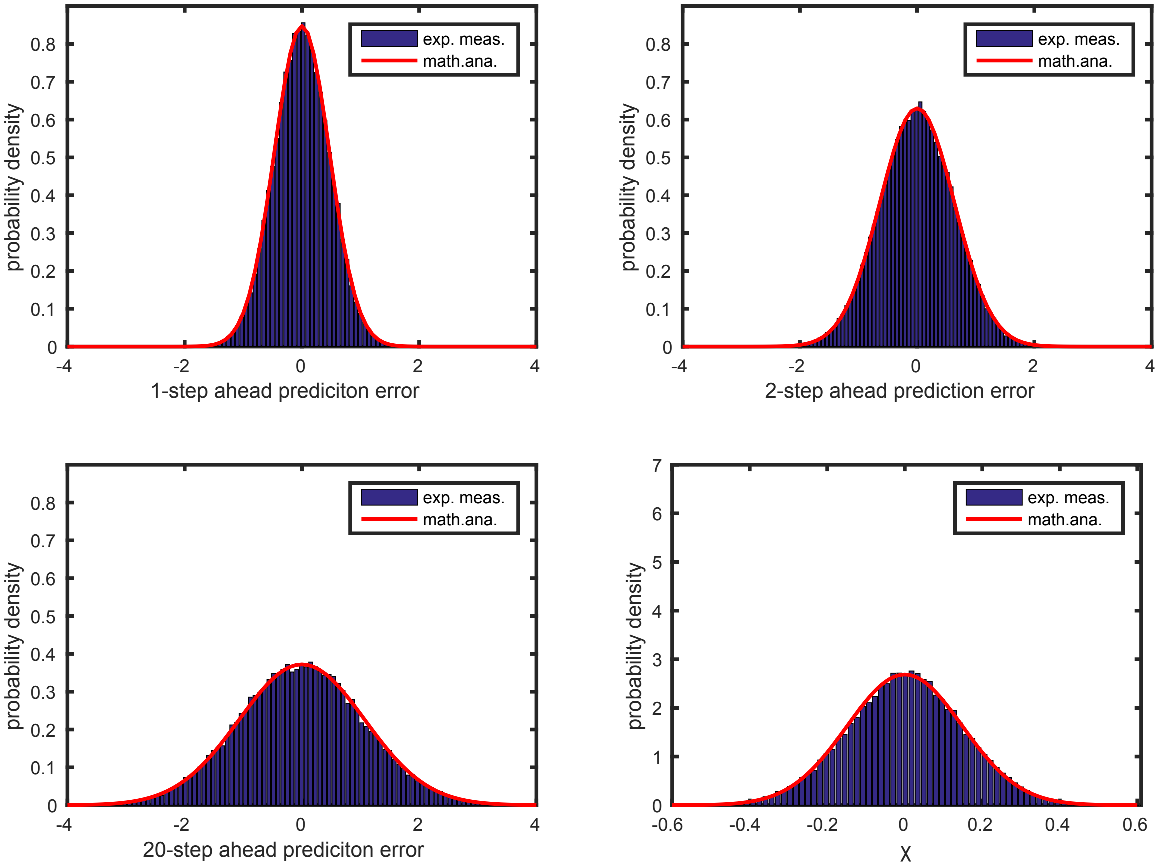


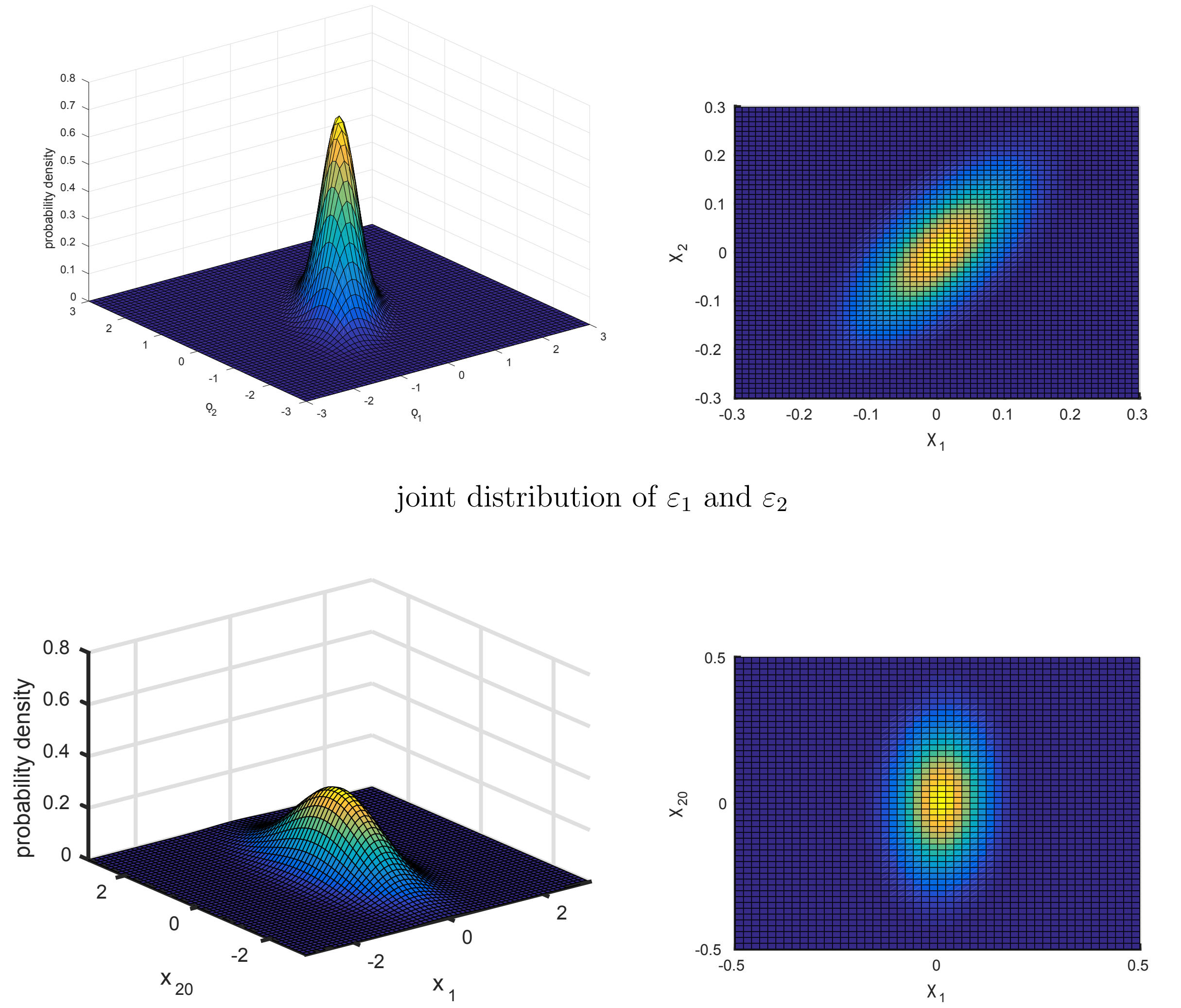
Figure 4.12: The probability density distribution of 1, 2, 20, 24-step ahead prediction error

coefficient [98], which is calculated by:

$$\rho_{i,j} = \boldsymbol{S_k}(i,j)/\sqrt{\boldsymbol{S_k}(i,i)\boldsymbol{S_k}(j,j)}$$

As an illustration, we depict the joint distribution of ε_1 and ε_2, and ε_1 and ε_{20} in Fig. 4.13. As expected, each pair of them has a bivariate normal distribution. The correlation between them can be viewed from the equidensity contours. The 1-step ahead prediction error ε_1 has a higher correlation with ε_2 ($\rho_{1,2} = 0.6686$) than ε_{20} ($\rho_{1,20} = 0.0632$). It is consistent with the covariance matrix Eq. (4.13), where the theoretical correlation coefficients between ε_1 and ε_2, and ε_1 and ε_{20} are 0.6690 and 0.0593, respectively.

In summary, a single step ahead prediction error, ε_k, has a normal distribution with zero mean and covariance σ_k^2, i.e., $\varepsilon_k \sim \mathcal{N}(0,\, \sigma_k^2)$. The covariance converges to a constant as k increases. The vector of k steps ahead prediction errors from 1-step ahead to k-step



joint distribution of ε_1 and ε_2

joint distribution of ε_1 and ε_{20}

Figure 4.13: Joint normal distribution of different step ahead prediction errors

ahead, $\boldsymbol{v\varepsilon_k} = [\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_k]^T$, satisfies a MVN distribution, i.e, $\boldsymbol{v\varepsilon_k} \sim \mathcal{N}(0, \boldsymbol{S}_k)$. The further two prediction errors are, the lower correlation they have. The probability density function (PDF) of $\boldsymbol{v\varepsilon_k}$ is:

$$\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi}) = \varphi_{\boldsymbol{v\varepsilon_k}}(\chi_1, \cdots, \chi_k) = \frac{1}{\sqrt{(2\pi)^k |\boldsymbol{S_k}|}} \exp\Big(-\frac{1}{2} \boldsymbol{\chi}^T \boldsymbol{S_k}^{-1} \boldsymbol{\chi} \Big)$$

where $\boldsymbol{\chi}$ is a real k-dimensional column vector.

4.3.2 Threshold vs. Transmission Rate

The process of PKF can be modeled as a Markov chain with the help of the previous analysis of prediction error distribution. We then obtain the transmission rate as a function of the threshold using the steady state equation of the chain in this section.

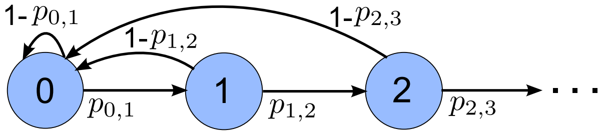


Figure 4.14: Description of PKF using success-runs Markov chain.

The sequence of reconstructions of PKF, $\bar{x}_1, \bar{x}_2, \bar{x}_3, \cdots$, is a random process [99]. Each random variable $\bar{x}_k$ could be either the prediction or the optimal value, which is independent of others. Supposing the outcome of the reconstruction at time k is $\hat{x}_k$, it does not affect the likelihood of getting $\hat{x}_{k+1}$ or $\tilde{x}_{k+1}$ at time $k+1$. Each random variable $\bar{x}_k$ results in either *success* or *failure*: if the outcome is the prediction, it corresponds to a *success*; otherwise, it is a *failure*. From Eq. (4.3) we know that the reconstruction is the prediction, when the prediction error at this time instant lies within the threshold interval $[-\tau \quad \tau]$. Assuming the prediction error at time instant k, ϵ_k, is the i-step ahead prediction error, ε_i, then the probability of the kth trial to have a *success* is actually $\Pr(-\tau \leq \varepsilon_i \leq \tau) = \Pr(-\tau \leq \varepsilon_i \leq \tau)$, the probability of the i-step ahead prediction error falling in the interval $[-\tau \quad \tau]$. The *failure* probability of each trial is correspondingly $1 - \Pr(\tau \leq \varepsilon_i \leq \tau)$.

Let Υ_n denote the number of most recent consecutive *successes* that have been observed at the nth trial [100]. If the nth trial is a *failure*, then $\Upsilon_n = 0$; if trial numbers n, $n-1$, $n-2, \cdots, n-m+1$ are all *successes* but trial number $n-m$ is a *failure*, then $\Upsilon_n = m$. The collection of $\{\Upsilon_1,\ \Upsilon_2,\ \Upsilon_3, \cdots\}$ is thereby a stochastic process, each of which is a random variable. Assuming $\Upsilon_n = k$ at the nth trial, then Υ_{n+1} will equal either $k+1$ or 0 at the next trial regardless of the values $\Upsilon_1, \cdots, \Upsilon_{n-1}$. It means the random process satisfies the Markov property [101] and can be modeled as a discrete-time Markov chain [102].

The state space [103] of the Markov chain should be $\mathbb{N}$, which is the set of all possible values of Υ_n. The transition probability of going from state k at time n to the next state $k+1$ at time $n+1$ should be $\Pr(\Upsilon_{n+1} = k+1 | \Upsilon_n = k, \cdots, \Upsilon_1 = 1) = \Pr(\Upsilon_{n+1} = k+1 | \Upsilon_n = k)$. The number of consecutive *successes* is k indicating that all the k steps ahead prediction errors are within the threshold intervals, i.e., $|\varepsilon_1| \leq \tau, |\varepsilon_2| \leq \tau, \cdots, |\varepsilon_k| \leq \tau$.

For conciseness, we define $\boldsymbol{R_k(\tau)} = \Big\{\chi_i = \boldsymbol{\chi}(i) \in \mathbb{R}^{\boldsymbol{k}} : |\chi_i| \leq \tau,\ i = 1 \cdots k\Big\}$ as the region of k-dimensional space that each variable, ε_i, lies within the interval of the threshold $[-\tau \quad \tau]$. Then the transition probability from state k to state $k+1$ is independent of the time n and can be represented as:

$$\begin{aligned} p_{k,k+1} &= \Pr(\Upsilon_{n+1} = k+1 | \Upsilon_n = k) \\ &= \Pr(\Upsilon_n = k+1 | \Upsilon_{n-1} = k) \\ &= \Pr\Big(|\varepsilon_{k+1}| \leq \tau \Big| |\varepsilon_k| \leq \tau, \cdots, |\varepsilon_1| \leq \tau\Big) \\ &= \Pr\Big(|\varepsilon_{k+1}| \leq \tau \Big| \boldsymbol{v\varepsilon_k} \in \boldsymbol{R_k(\tau)}\Big) \end{aligned}$$

Using the Bayes' theorem, it can be calculated by

$$\begin{aligned} p_{k,k+1} &= \frac{\Pr\Big(\boldsymbol{v\varepsilon_{k+1}} \in \boldsymbol{R_{k+1}(\tau)}\Big)}{\Pr\Big(\boldsymbol{v\varepsilon_k} \in \boldsymbol{R_k(\tau)}\Big)} \\ &= \Phi_{\boldsymbol{v\varepsilon_{k+1}}}(\tau) / \Phi_{\boldsymbol{v\varepsilon_k}}(\tau) \end{aligned} \tag{4.14}$$

where

$$\Phi_{\boldsymbol{v\varepsilon_k}}(\tau) = \Pr\Big(\boldsymbol{v\varepsilon_k} \in \boldsymbol{R_k(\tau)}\Big) = \int_{\boldsymbol{R_k(\tau)}} \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})\, d\boldsymbol{\chi} \tag{4.15}$$

It is the probability of $\boldsymbol{v\varepsilon_k}$ locating inside the region $\boldsymbol{R_k(\tau)}$. When $k = 0$, $\Phi_0(\tau) = 1$, since all errors in state 0 are 0. The transition probability of going from state k to the state 0 has the probability $p_{k,0} = 1 - p_{k,k+1}$. Thus, the transition matrix of the chain is:

$$\boldsymbol{P} = \begin{pmatrix} 1 - p_{0,1} & p_{0,1} & \cdots & 0 & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\ 1 - p_{k,k+1} & 0 & \cdots & p_{k,k+1} & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \end{pmatrix}$$

It is a time-homogeneous Markov chain [100]. The distribution over states can be written as a stochastic row vector $\boldsymbol{\pi} = \big[p_0,\, p_1, \cdots\big]^T$ with non-negative entries that add up to one [104]. The probability of the random variable Υ_n in the state k is $\Pr(\Upsilon_n = k) = \boldsymbol{\pi}(k) = p_k$. The distribution in state 0, p_0, is actually the transmission rate of the leaf node. According to the steady state equation, $\boldsymbol{P\pi} = \boldsymbol{\pi}$, and Eq. (4.14), the distribution over the state k

when $k > 0$ is:

$$p_k = p_{k-1}\, p_{k-1,k} = p_0 \prod_{i=1}^{k} p_{i-1,i} = p_0 \Phi_{\boldsymbol{v\varepsilon_k}}(\tau) \tag{4.16}$$

Since the total probability in all states equals one, i.e., $\sum_{k=0}^{\infty} p_k = p_0\Big(1 + \sum_{k=1}^{\infty} \Phi_{\boldsymbol{v\varepsilon_k}}(\tau)\Big) = 1$, the transmission probability p_0 is:

$$p_0 = \frac{1}{1 + \sum_{k=1}^{\infty} \Phi_{\boldsymbol{v\varepsilon_k}}(\tau)} \tag{4.17}$$

It is a function of the threshold τ.

4.3.3 Threshold vs. Reconstruction Accuracy

We calculate the covariance of the reconstruction errors of PKF w.r.t. KF in the following. Compared with the real state, the covariance only needs to be added by the *a posteriori* covariance of KF as analyzed in Section 4.3.1. The total error covariance compared with the KF, denoted as $\bar{\bar{\sigma}}^2$, is the summation of the covariance of reconstruction errors generated in each state. According to the law of total probability [105], it can be calculated by:

$$\bar{\bar{\sigma}}^2 = \sum_{k=1}^{\infty} p_k \bar{\sigma}_k^2 \tag{4.18}$$

where $\bar{\sigma}_k^2$ is the covaraince of the reconstruction error produced by state k. The goal of obtaining $\bar{\bar{\sigma}}^2$ gets reduced to calculate $\bar{\sigma}_k^2$. It is addressed in the following paragraphs.

As analyzed in Section 4.3.1, the joint distribution of k steps ahead prediction errors $\boldsymbol{v\varepsilon_k}$ has a MVN distribution. Then $\boldsymbol{v\varepsilon_k}$ conditional on $\boldsymbol{v\varepsilon_k} \in \boldsymbol{R_k(\tau)}$, have a truncated MVN distribution [106]. Its probability density function, f, for $\tau \le \chi_1 \le \tau, \cdots, \tau \le \chi_k \le \tau$, is given by:

$$f(\boldsymbol{\chi}, \tau) = \frac{\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\Phi_{\boldsymbol{v\varepsilon_k}}(\tau)} \tag{4.19}$$

It is indeed the density of the joint distribution of k states reconstruction errors, $\boldsymbol{v\varepsilon_k}|\boldsymbol{v\varepsilon_k} \in \boldsymbol{R_k(\tau)}$. The kth random variable, $\varepsilon_k\big|\boldsymbol{v\varepsilon_k} \in \boldsymbol{R_k(\tau)}$, is the reconstruction error generated in state k. Its covariance, $\bar{\sigma}_k^2$, can be calculated by:

$$\bar{\sigma}_k^2 = \frac{\int_{\boldsymbol{R_k(\tau)}} \chi_k^2 \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})\, d\boldsymbol{\chi}}{\Phi_{\boldsymbol{v\varepsilon_k}}(\tau)}$$

$$= \frac{\Psi_{\boldsymbol{v\varepsilon_k}}(\tau)}{\Phi_{\boldsymbol{v\varepsilon_k}}(\tau)} \tag{4.20}$$

where

$$\Psi_{\boldsymbol{v\varepsilon_k}}(\tau) = \int_{\boldsymbol{R_k(\tau)}} \chi_k^2 \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})\, d\boldsymbol{\chi} \tag{4.21}$$

It is the unnormalized covariance of the reconstruction error in state k. Then the final covariance of the reconstruction errors in Eq. (4.18), combining Eqs. (4.16) and (4.20), can be recalculated as:

$$\begin{aligned}\bar{\bar{\sigma}}^2 &= \sum_{k=1}^{\infty} p_k \bar{\sigma}_k^2 = \sum_{k=1}^{\infty} p_0 \Phi_{\boldsymbol{v\varepsilon_k}}(\tau) \Psi_{\boldsymbol{v\varepsilon_k}}(\tau) / \Phi_{\boldsymbol{v\varepsilon_k}}(\tau) \\ &= p_0 \sum_{k=1}^{\infty} \Psi_{\boldsymbol{v\varepsilon_k}}(\tau) \end{aligned} \tag{4.22}$$

As k increases, it is hard to obtain $\Psi_{\boldsymbol{v\varepsilon_k}}(\tau)$, because once the error in the former states are truncated, the distribution of the error in the current state is not normal anymore. Although in [107], they use moment generation to calculate this truncated covariance, the formula is hard to use. To the best of our knowledge, this is the first work that shows how to use the Hessian matrix [108] of the PDF of MVN to obtain the doubly truncated covariance and the code can be realized in Matlab.

The idea is to truncate all the errors at the same time. Then the problem to obtain $\Psi_{\boldsymbol{v\varepsilon_k}}(\tau)$ can be solved by calculating $\bar{\boldsymbol{S}}_{\boldsymbol{k}}$, which is the covariance matrix of the doubly truncated MVN distribution truncated over the region $\boldsymbol{R_k(\tau)}$ and satisfies:

$$\bar{\boldsymbol{S}}_{\boldsymbol{k}} = \frac{\int_{\boldsymbol{R_k(\tau)}} \boldsymbol{\chi}\boldsymbol{\chi}^T \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})\, d\boldsymbol{\chi}}{\Phi_{\boldsymbol{v\varepsilon_k}}(\tau)} = \frac{\tilde{\boldsymbol{S}}_{\boldsymbol{k}}}{\Phi_{\boldsymbol{v\varepsilon_k}}(\tau)}$$

where

$$\tilde{\boldsymbol{S}}_{\boldsymbol{k}} = \int_{\boldsymbol{R_k(\tau)}} \boldsymbol{\chi}\boldsymbol{\chi}^T \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})\, d\boldsymbol{\chi} \tag{4.23}$$

It is the numerator of $\bar{\boldsymbol{S}}_{\boldsymbol{k}}$ and denotes the unnormalized covariance of the doubly truncated MVN distribution. Then $\Psi_{\boldsymbol{v\varepsilon_k}}(\tau)$ is the (k, k) entry of the $\tilde{\boldsymbol{S}}_{\boldsymbol{k}}$, namely,

$$\Psi_{\boldsymbol{v\varepsilon_k}}(\tau) = \tilde{\boldsymbol{S}}_{\boldsymbol{k}}(k, k) \tag{4.24}$$

According to [108], the Hessian matrix of $\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})$ is given by:

$$\begin{aligned}
H_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi}) &= \begin{pmatrix} \frac{\partial^2\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial^2\chi_1}, & \frac{\partial^2\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial\chi_1\partial\chi_2}, & \cdots & \frac{\partial^2\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial\chi_1\partial\chi_k} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial\chi_k\partial\chi_1}, & \frac{\partial^2\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial\chi_k\partial\chi_2}, & \cdots & \frac{\partial^2\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial^2\chi_k} \end{pmatrix} \\
&= \frac{\partial^2\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial^2\boldsymbol{\chi}} = \frac{\partial\nabla\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial\boldsymbol{\chi}} \\
&= \frac{\partial\big(-\boldsymbol{\chi}^T\boldsymbol{S_k}^{-1}\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})\big)}{\partial\boldsymbol{\chi}} \\
&= (\boldsymbol{S_k}^{-1}\boldsymbol{\chi}\boldsymbol{\chi}^T\boldsymbol{S_k}^{-1} - \boldsymbol{S_k}^{-1})\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi}) \qquad (4.25)
\end{aligned}$$

where $\nabla\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})$ is the gradient of $\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})$ and satisfies

$$\begin{aligned}
\nabla\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi}) &= \Big[\tfrac{\partial\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial\chi_1}, \ \tfrac{\partial\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial\chi_2}, \ \cdots, \ \tfrac{\partial\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial\chi_k}\Big] \\
&= -\boldsymbol{\chi}^T\boldsymbol{S_k}^{-1}\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi}) \qquad (4.26)
\end{aligned}$$

It is easy to obtain $\boldsymbol{\chi}\boldsymbol{\chi}^T\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi}) = \boldsymbol{S_k}H_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})\boldsymbol{S_k} + \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})\boldsymbol{S_k}$ by multiplying both sides of Eq. (4.25) by $\boldsymbol{S_k}$ twice. Then integrating the result over the region $\boldsymbol{R_k(\tau)}$, we obtain:

$$\begin{aligned}
\tilde{\boldsymbol{S}}_{\boldsymbol{k}} = \int_{\boldsymbol{R_k(\tau)}} \boldsymbol{\chi}\boldsymbol{\chi}^T\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})\,d\boldsymbol{\chi} &= \int_{\boldsymbol{R_k(\tau)}} \boldsymbol{S_k}H_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})\boldsymbol{S_k}\,d\boldsymbol{\chi} + \int_{\boldsymbol{R_k(\tau)}} \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})\boldsymbol{S_k}\,d\boldsymbol{\chi} \\
&= \boldsymbol{S_k}\boldsymbol{\xi_k}\boldsymbol{S_k} + \boldsymbol{S_k}\Phi_{\boldsymbol{v\varepsilon_k}}(\tau) \qquad (4.27)
\end{aligned}$$

where $\boldsymbol{\xi_k} = \int_{\boldsymbol{R_k(\tau)}} H_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})\,d\boldsymbol{\chi}$. It is a $k \times k$ matrix.

In Eq. (4.27), only $\boldsymbol{\xi_k}$ is unknown. We calculate it in two steps: first the diagonal elements and then the off-diagonal elements. The *ith* diagonal element of $\boldsymbol{\xi_k}$ is $\boldsymbol{\xi_k}(i,i)$ and holds:

$$\begin{aligned}
\boldsymbol{\xi_k}(i,i) &= \int_{\boldsymbol{R_k(\tau)}} \frac{\partial^2\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial^2\chi_i}\,d\boldsymbol{\chi} \\
&= \int_{\boldsymbol{R_{k-1}(\tau)}} \frac{\partial\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi}_{-i},\chi_i)}{\partial\chi_i}\Big|_{\chi_i=-\tau}^{\tau} d\boldsymbol{\chi}_{-i} \\
&= 2\int_{\boldsymbol{R_{k-1}(\tau)}} \frac{\partial\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi}_{-i},\chi_i=\tau)}{\partial\tau} d\boldsymbol{\chi}_{-i} \qquad (4.28)
\end{aligned}$$

where $\boldsymbol{\chi}_{-i}$ denotes the elimination of χ_i from $\boldsymbol{\chi}$.

To obtain $\frac{\partial\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi}_{-i},\chi_i=\tau)}{\partial\tau}$, we first calculate $\frac{\partial\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial\chi_i}$. It can be obtained from the gradient of $\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})$. For computation convenience, the gradient equation Eq. (4.26) is rearranged as:

$$\begin{bmatrix} \frac{\partial \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial \boldsymbol{\chi}_{-i}} & \frac{\partial \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial \chi_i} \end{bmatrix} = -[\boldsymbol{\chi}_{-i} \quad \chi_i]\boldsymbol{\Sigma_k}^{-1}\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi}) \tag{4.29}$$

where $\boldsymbol{\chi}$ and $\boldsymbol{S_k}$ are rearranged into $(\boldsymbol{\chi}_{-i}, \chi_i)$ and $\boldsymbol{\Sigma_k}$, respectively. The rearranged covariance matrix $\boldsymbol{\Sigma_k}$ follows:

$$\boldsymbol{\Sigma_k} = \begin{bmatrix} \boldsymbol{\Sigma_{11}} & \boldsymbol{\Sigma_{12}} \\ \boldsymbol{\Sigma_{21}} & \Sigma_{22} \end{bmatrix}$$

where $\boldsymbol{\Sigma_{11}}$ is constructed by dropping the row and column of the *ith* element in $\boldsymbol{S_k}$; $\boldsymbol{\Sigma_{12}} = \boldsymbol{S_k}(j,i)$, $j \neq i$. It is the column of the *ith* element except its row in $\boldsymbol{S_k}$; $\boldsymbol{\Sigma_{21}} = \boldsymbol{\Sigma_{12}}^T = \boldsymbol{S_k}(i,j)$, $j \neq i$; $\Sigma_{22} = \boldsymbol{S_k}(i,i)$. It is the row and the column of the *ith* element in $\boldsymbol{S_k}$. For example when $k = 4$ and $i = 3$, the rearranged $\boldsymbol{\Sigma_k}$ is shown in Fig. 4.15 with

$$\boldsymbol{\Sigma_{11}} = \begin{bmatrix} \boldsymbol{S_4}(1,1) & \boldsymbol{S_4}(1,2) & \boldsymbol{S_4}(1,4) \\ \boldsymbol{S_4}(2,1) & \boldsymbol{S_4}(2,2) & \boldsymbol{S_4}(2,4) \\ \boldsymbol{S_4}(4,1) & \boldsymbol{S_4}(4,2) & \boldsymbol{S_4}(4,4) \end{bmatrix} \quad \boldsymbol{\Sigma_{22}} = \begin{bmatrix} \boldsymbol{S_4}(3,3) \end{bmatrix}$$

$$\boldsymbol{\Sigma_{21}} = \begin{bmatrix} \boldsymbol{S_4}(3,1) & \boldsymbol{S_4}(3,2) & \boldsymbol{S_4}(3,4) \end{bmatrix} \quad \boldsymbol{\Sigma_{12}} = \boldsymbol{\Sigma_{21}}^T$$

Multiplying both sides of the rearranged gradient equation Eq. (4.29) by $\boldsymbol{\Sigma_k}$ results in:

$$\begin{bmatrix} \frac{\partial \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial \boldsymbol{\chi}_{-i}} & \frac{\partial \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial \chi_i} \end{bmatrix} \boldsymbol{\Sigma_k} = -[\boldsymbol{\chi}_{-i} \quad \chi_i]\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})$$

Then, extracting the second column of the equation,

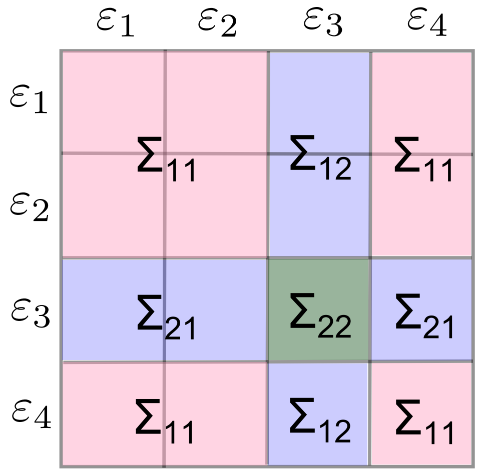


Figure 4.15: One example of the rearranged $\boldsymbol{\Sigma_4}$ when $i = 3$ and $k = 4$.

$$\frac{\partial \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial \boldsymbol{\chi}_{-i}}\boldsymbol{\Sigma_{12}} + \frac{\partial \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial \chi_i}\Sigma_{22} = -\chi_i \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})$$

the formula for $\frac{\partial \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial \chi_i}$ is:

$$\frac{\partial \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial \chi_i} = -\Big(\chi_i \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi}) + \frac{\partial \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial \boldsymbol{\chi}_{-i}}\boldsymbol{\Sigma_{12}}\Big)\Sigma_{22}^{-1}$$

Plugging it into Eq. (4.28), we can obtain that

$$\begin{aligned}
\boldsymbol{\xi_k}(i,i) &= 2\int_{\boldsymbol{R_{k-1}(\tau)}} \frac{\partial \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi}_{-i}, \chi_i = \tau)}{\partial \tau} d\boldsymbol{\chi}_{-i} \\
&= -2\int_{\boldsymbol{R_{k-1}(\tau)}} \tau \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi}_{-i}, \chi_i = \tau)\Sigma_{22}^{-1} d\boldsymbol{\chi}_{-i} \\
&\quad - 2\int_{\boldsymbol{R_{k-1}(\tau)}} \frac{\partial \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi}_{-i}, \chi_i = \tau)}{\partial \boldsymbol{\chi}_{-i}}\boldsymbol{\Sigma_{12}}\Sigma_{22}^{-1}\, d\boldsymbol{\chi}_{-i} \\
&= -2\tau F_i^{+}(\tau)\Sigma_{22}^{-1} - 2\Big(F_{i,j}^{++}(\tau) - F_{i,j}^{+-}(\tau)\Big)_{j\neq i}\boldsymbol{\Sigma_{12}}\Sigma_{22}^{-1} \qquad (4.30)
\end{aligned}$$

where

$$F_i^{+}(\tau) = \int_{\boldsymbol{R_{k-1}(\tau)}} \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi}_{-i}, \chi_i = \tau)\, d\boldsymbol{\chi}_{-i} \qquad (4.31)$$

$$F_{i,j}^{++}(\tau) = \int_{\boldsymbol{R_{k-2}(\tau)}} \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi}_{-i,j}, \chi_i = \tau, \chi_j = \tau)\, d\boldsymbol{\chi}_{-i,j} \qquad (4.32)$$

$$F_{i,j}^{+-}(\tau) = \int_{\boldsymbol{R_{k-2}(\tau)}} \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi}_{-i,j}, \chi_i = \tau, \chi_j = -\tau)\, d\boldsymbol{\chi}_{-i,j} \qquad (4.33)$$

where $\boldsymbol{\chi}_{-i,j}$ denotes the elimination of χ_i and χ_j from $\boldsymbol{\chi}$. The results of these three equations can be obtained by the conditional MVN distribution. According to Bayes's theorem, $\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi}_{-i}, \chi_i = \tau) = \varphi_{\boldsymbol{v\varepsilon_{k-i}}}(\boldsymbol{\chi}_{-i}|\chi_i = \tau)\varphi_{\varepsilon_i}(\chi_i = \tau)$. As we known from Chapter 3, the distribution of $\boldsymbol{v\varepsilon_{k-i}}$ conditional on $\varepsilon_i = \tau$ satisfies a MVN with mean value $\tilde{\boldsymbol{\mu}}$ and covariance matrix $\tilde{\boldsymbol{\Sigma}}$, i.e., $(\boldsymbol{v\varepsilon_{k-i}}|\varepsilon_i = \tau) \sim N(\tilde{\boldsymbol{\mu}},\ \tilde{\boldsymbol{\Sigma}})$, where $\tilde{\boldsymbol{\mu}} = \boldsymbol{\Sigma_{12}}\Sigma_{22}^{-1}\tau$ is a constant, and $\tilde{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma_{11}} - \boldsymbol{\Sigma_{12}}\Sigma_{22}^{-1}\boldsymbol{\Sigma_{21}}$. The probability of $\boldsymbol{v\varepsilon_{k-i}}$ located in the region $\boldsymbol{R_{k-1}(\tau)}$ conditional $\varepsilon_i = \tau$ is denoted as $\Phi_{\boldsymbol{v\varepsilon_{k-i}}}(\tau)$. Thus, Eq. (4.31) can be calculated by:

$$F_i^{+}(\tau) = \Phi_{\boldsymbol{v\varepsilon_{k-i}}}(\tau)\varphi_{\varepsilon_i}(\tau) \qquad (4.34)$$

This expression can be easily obtained by Matlab using standard functions. When $1 < i < k$, the complexity of this equation can be further reduced as illustrated in Section 8.1. It can be reduced from a $(k-1)$ dimensional integral to the multiplication of one $(i-1)$ and one $(k-i)$ dimensional integral, namely,

$$F_i^+(\tau) = \Phi_{\boldsymbol{v\varepsilon_{i-1}}}(\tau)\Phi_{\boldsymbol{v\varepsilon_{k-i}}}(\tau)\varphi_{\varepsilon_i}(\tau) \tag{4.35}$$

Similarly, the formula for $F_{i,j}^{++}(\tau)$ and $F_{i,j}^{+-}(\tau)$ can be simplified with the conditional distribution. The vector of the random variables is separated as $\boldsymbol{v\varepsilon_{k}}_{-i,j}$ and $[\varepsilon_i,\ \varepsilon_j]$. Then $(\boldsymbol{v\varepsilon_{k}}_{-i,j} \mid [\varepsilon_i,\ \varepsilon_j] = [\tau,\ \tau]^T)$ satisfies $N(\tilde{\boldsymbol{\mu}}_{\mathbf{3}}, \tilde{\boldsymbol{\Sigma}}_{\mathbf{3}})$. The mean $\tilde{\boldsymbol{\mu}}_{\mathbf{3}}$ and the covariance $\tilde{\boldsymbol{\Sigma}}_{\mathbf{3}}$ can still be calculated using Eq. (3.14) with the rearranged matrix. An example of the rearranged matrix is shown in Fig. 4.16. Then Eqs. (4.32) and (4.33) can be calculated by:

$$F_{i,j}^{++}(\tau) = \Phi_{\boldsymbol{v\varepsilon_{k}}_{-i,j}}(\tau)\varphi_{\varepsilon_i,\varepsilon_j}(\tau,\ \tau) \tag{4.36}$$

$$F_{i,j}^{+-}(\tau) = \Phi_{\boldsymbol{v\varepsilon_{k}}_{-i,j}}(\tau)\varphi_{\varepsilon_i,\varepsilon_j}(\tau, -\tau) \tag{4.37}$$

Depending on the position of i and j, the complexity of these two formulas can be reduced similarly to $F_i^+(\tau)$. For the sake of conciseness, the detailed discussion is not presented here. Plugging Eqs. (4.36), (4.37) and (4.34) into Eq. (4.30), we can obtain $\boldsymbol{\xi_k}(i,i)$.

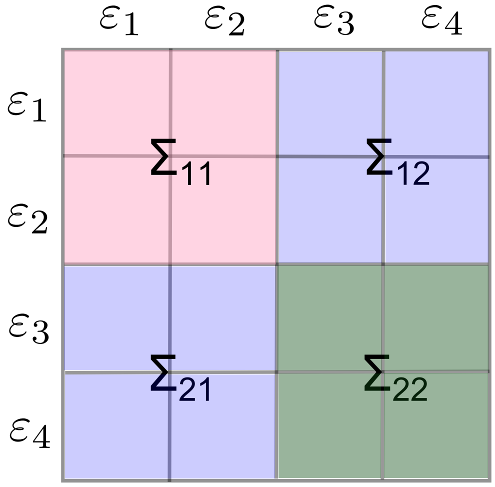


Figure 4.16: The rearranged $\boldsymbol{\Sigma_4}$ when $i = 3$ and $j = 4$.

Next, we calculate the (i,j) entry of $\boldsymbol{\xi_k}$. It can be obtained by:

$$\boldsymbol{\xi_k}(i,j) = \int_{\boldsymbol{R_k(\tau)}} \frac{\partial^2 \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})}{\partial\chi_i\partial\chi_j}\, d\boldsymbol{\chi}$$

$$= \int_{\boldsymbol{R_{k-1}(\tau)}} \frac{\partial \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi}_{-i}, \chi_i = \tau)}{\partial \chi_j} d\boldsymbol{\chi}_{-i} - \int_{\boldsymbol{R_{k-1}(\tau)}} \frac{\partial \varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi}_{-i}, \chi_i = -\tau)}{\partial \chi_j} d\boldsymbol{\chi}_{-i}$$

$$= 2\Big(F_{i,j}^{++}(\tau) - F_{i,j}^{+-}(\tau)\Big) \tag{4.38}$$

where $F_{i,j}^{++}(\tau)$ and $F_{i,j}^{+-}(\tau)$ can be calculated using Eqs. (4.36) and (4.37). After calculating $\boldsymbol{\xi_k}$ using Eqs. (4.30) and (4.38), we can obtain $\tilde{\boldsymbol{S}}_{\boldsymbol{k}}$ and $\Psi_{\boldsymbol{v\varepsilon_k}}(\tau)$ using Eqs. (4.24) and (4.27).

The complexity for computing $\bar{\bar{\sigma}}^2$ can be reduced, since in practical scenarios, the threshold is much smaller than the standard deviation of the prediction errors in each state[2], i.e., $\tau \ll \sigma_k$. The distribution of the error in state k can be approximated as a uniform distribution: $f(\tau) = \frac{1}{2\tau}$. Then the covariance, $\bar{\sigma}_k^2$ in Eq. (4.20), can be approximated as:

$$\bar{\sigma}_k^2 \approx \int_{-\tau}^{\tau} \chi_k^2 \, \frac{1}{2\tau} \, d\chi_k = \frac{\tau^2}{3} \tag{4.39}$$

The total covariance of reconstruction errors of N states in Eq. (4.22) is close to:

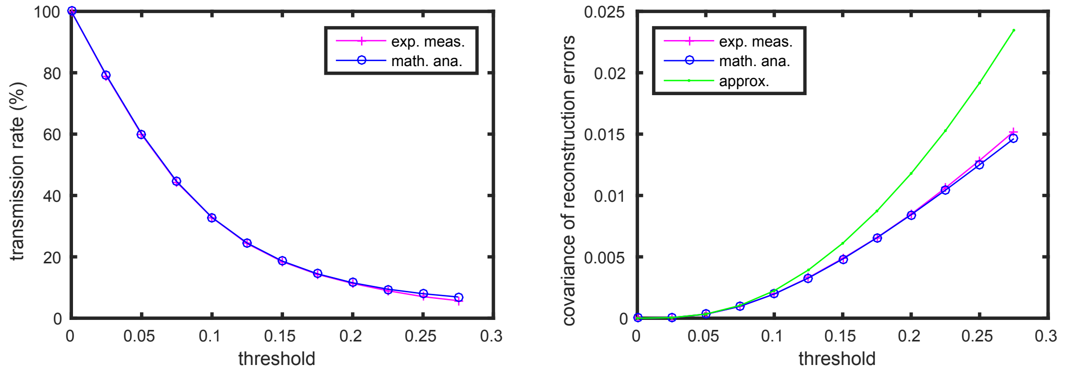
$$\bar{\bar{\sigma}}^2 \approx \frac{\tau^2}{3} \sum_{k=1}^{\infty} p_k = \frac{\tau^2}{3}(1 - p_0) \tag{4.40}$$

As the threshold τ increases, the error increases due to the approximation.

4.3.4 Validation of the Mathematical Analysis

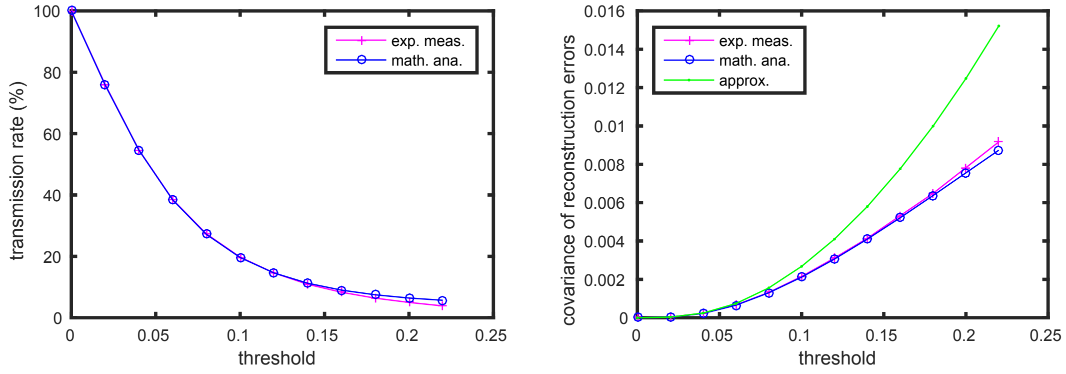
In this section, we validate our mathematical analysis by comparing the theoretical results with the simulated results in Section 4.2.2. As shown in Figs. 4.17 to 4.19, the mathematical analysis using Eqs. (4.17) and (4.22) precisely follow the experimental measured transmission rate and the covariance for these three nodes. The trade-offs between transmission rate and covariance of reconstruction errors, w.r.t. KF-optimal and real state, are depicted in Figs. 4.20 to 4.22. As the threshold increases, each node requires less transmission. Under the same threshold, the bigger the measurement noise of the leaf node is, the less transmission rate is required. This is due to that the reconstruction quality of the node with smaller measurement noise is more accurate. As expected, the approximated covariance of reconstruction errors using Eq. (4.40) becomes inaccurate as the threshold increases. Taking node 1 for example, when $\tau = 1.1\sigma_1$, the inaccuracy is 3.19% compared to the experimental measurements.

[2] Before truncated, the prediction error in state k is actually $\chi_k | \boldsymbol{\chi_{k-1}} \in \boldsymbol{R_{k-1}(\tau)}$. It can be approximated as the k-step ahead prediction error $\chi_k \sim (0, \sigma_k^2)$ when τ is small.



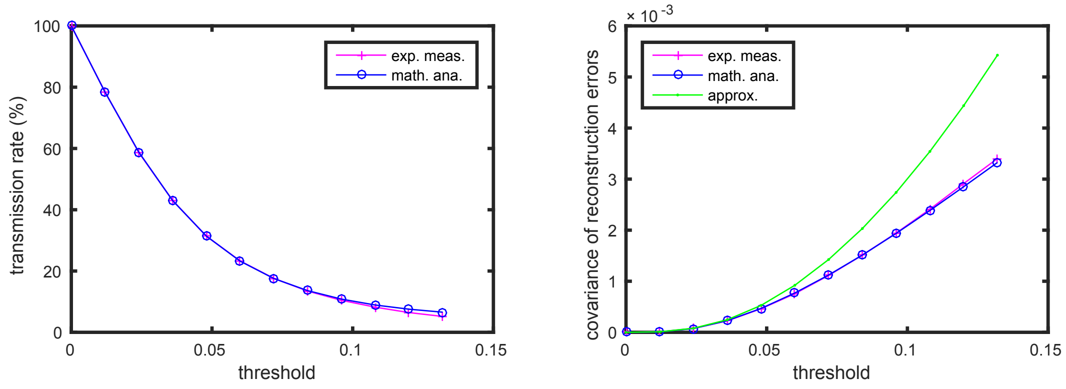
(a) (b)

Figure 4.17: Comparisons of (a) transmission rates (b) covariance of reconstruction errors w.r.t. KF optimal between experimental measurements and mathematical analysis in node 1;



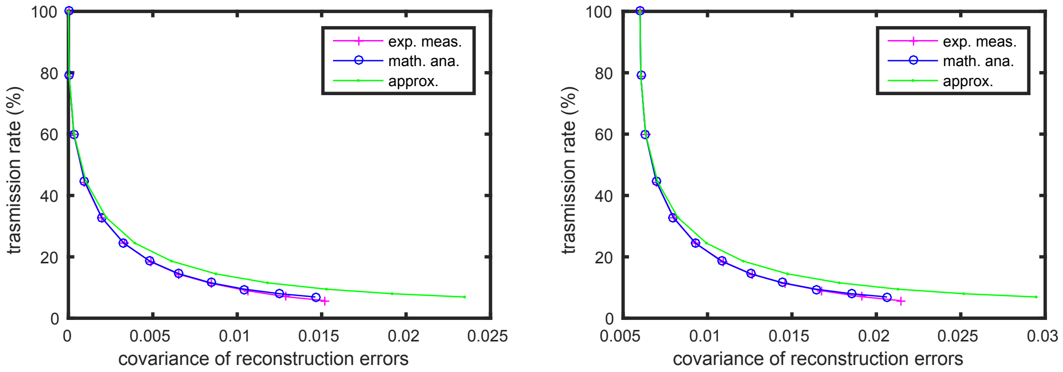
(a) (b)

Figure 4.18: Comparisons of (a) transmission rates (b) covariance of reconstruction errors w.r.t. KF optimal between experimental measurements and mathematical analysis in node 2;



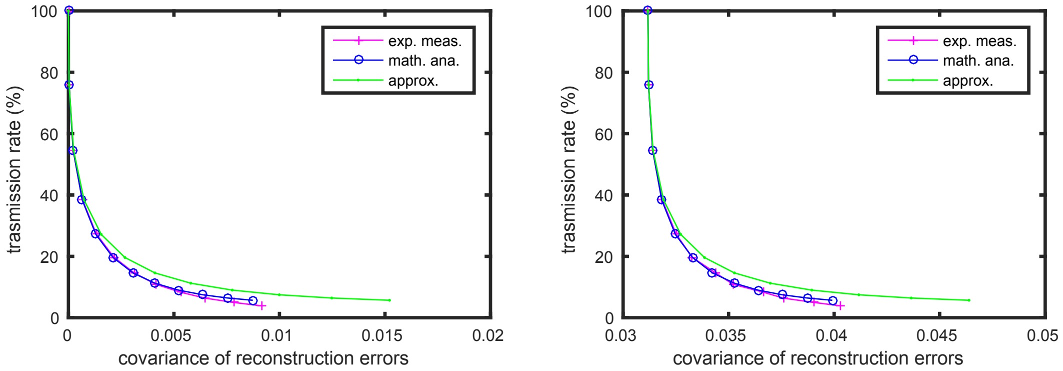
(a) (b)

Figure 4.19: Comparisons of (a) transmission rates (b) covariance of reconstruction errors w.r.t. KF optimal between experimental measurements and mathematical analysis in node 3.



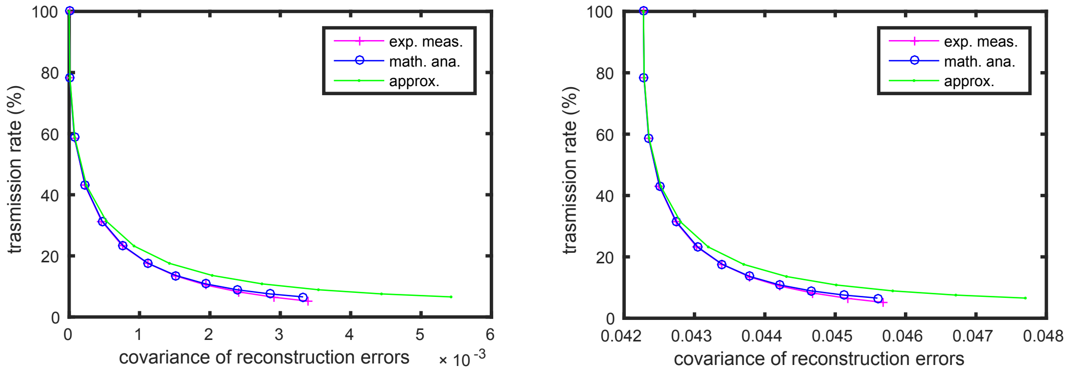
(a) (b)

Figure 4.20: Comparisons of trade-off between transmission rate and covariance of reconstruction errors w.r.t. (a) KF-optimal; (b) real state between experimental measurements and mathematical analysis in node 1;



(a) (b)

Figure 4.21: Comparisons of trade-off between transmission rate and covariance of reconstruction errors w.r.t. (a) KF-optimal; (b) real state between experimental measurements and mathematical analysis in node 2;



(a) (b)

Figure 4.22: Comparisons of trade-off between transmission rate and covariance of reconstruction errors w.r.t. (a) KF-optimal; (b) real state between experimental measurements and mathematical analysis in node 3.

Due to the limitation of Matlab functions, the maximum state, N, is set to 25 in Eq. (4.22). As the threshold increases, there exists more than 25 states. This makes that the analysis of the transmission rate is slightly larger and the covariance of the reconstruction errors is slightly smaller than the experimental measurements. However, it does not significantly affect the accuracy of the analysis, since the probability in state k is smaller and smaller as k increases. For example, when $R = 0.25$ and $\tau = 0.1$ (the original system), there are total 45 states from the simulation result. The probabilities in the first 25 states are depicted in Fig. 4.23a. It decays and gradually converges to zero. The larger probabilities are in the first several states. Fig. 4.23b shows the covariance of reconstruction errors in each state. It has some fluctuations, but is relatively stable. The distribution of the reconstruction errors w.r.t. the KF-optimal in state 1, 2 and 5 are depicted in Figs. 4.24a, 4.25a and 4.26a. The errors are truncated at the threshold interval form -0.1 to 0.1. The distribution of the errors from all state as depicted in Fig. 4.6b is the combination of the error from each state. The reconstruction errors w.r.t. the real state in state 1, 5 and 10 have the distribution shown in Figs. 4.24b, 4.25b and 4.26b. They do not have a normal distribution any more. Fig. 4.7b is actually the combination of many non-Gaussian noise from different states.

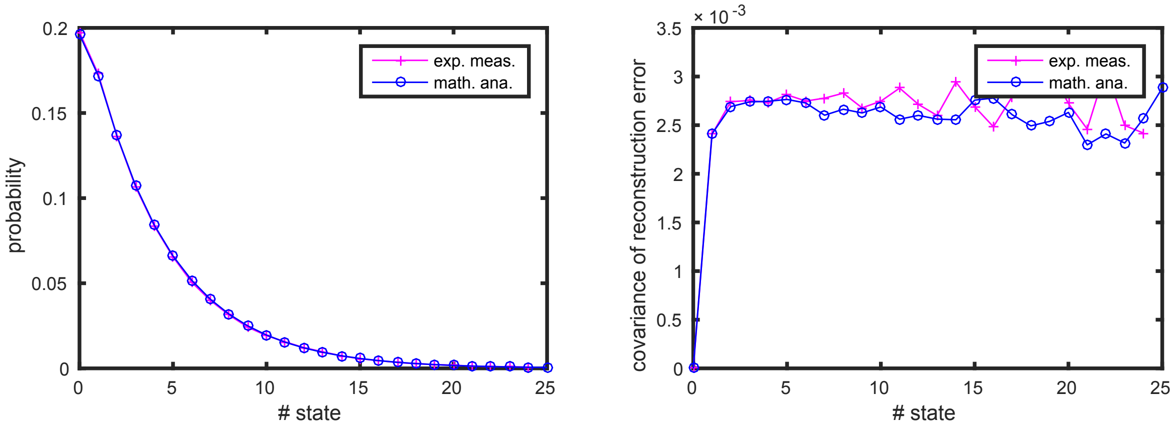
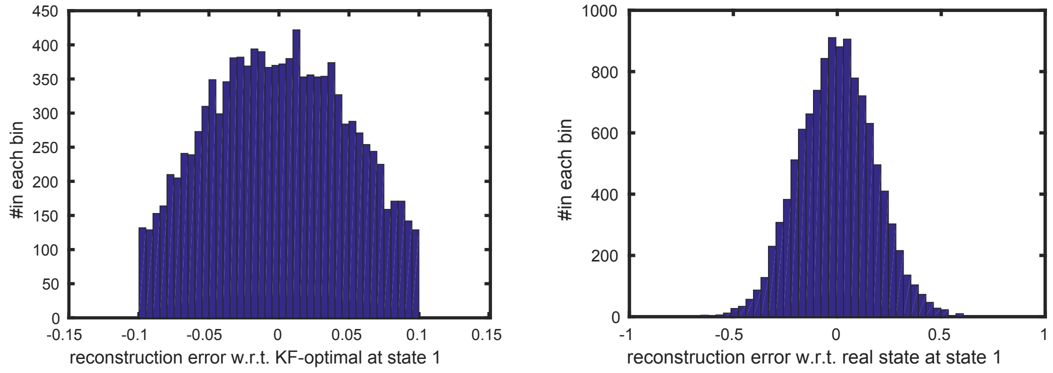


Figure 4.23: The probability and the covariance of each state in node 2 when $\tau = 0.1$.

4.4 Summary

In this chapter, we present our PKF scheme. It combines a KF and a k-step ahead KF-predictor to suppress the communication between the leaf node and the cluster head, while reconstructing the system state with the compressed information in the best way.



(a) (b)

Figure 4.24: The distribution of the reconstruction errors (a) w.r.t. the KF-optimal (b) w.r.t. the real state at state 1, in node 2 when $\tau = 0.1$;

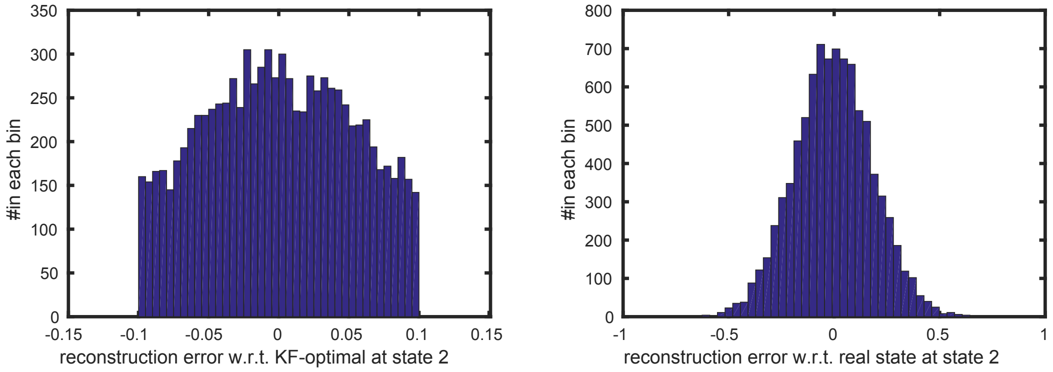


Figure 4.25: The distribution of the reconstruction errors (a) w.r.t. the KF-optimal (b) w.r.t. the real state at state 2, in node 2 when $\tau = 0.1$;

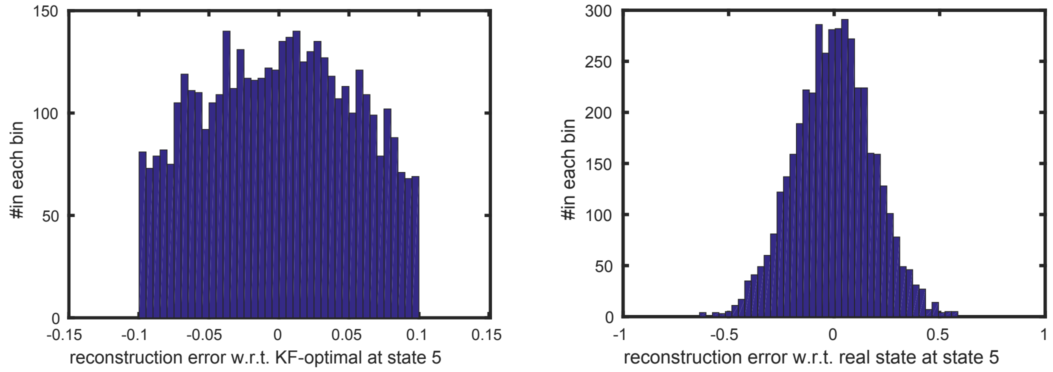


Figure 4.26: The distribution of the reconstruction errors (a) w.r.t. the KF-optimal (b) w.r.t. the real state at state 5, in node 2 when $\tau = 0.1$;

PKF allows multiple sensor types to be encoded in a single state vector and achieves data filtering, state estimation, data compression and reconstruction within one KF framework. The reconstructed signal of PKF with 19.75% transmission is much closer to the real state compared with transmitting all of the raw data in an arbitrary simulated system. In this case, the covariance of the reconstruction error is only 0.03, while the covaraince of the measurement noise is 0.25. As the measurement noise of the system increases, the superiority of the reconstruction quality of PKF compared with the raw data becomes even larger.

In order to quantify the gain of PKF in different scenarios, an in depth mathematical analysis is carried out. It is important to understand the underlying process of PKF but also vital for extending it to exploit spatial correlation. A single step ahead prediction error is proved to have a normal distribution with zero mean. The covariance is calculated from the system parameters and converges to a constant as k increases. The vector of k steps ahead prediction errors from 1-step ahead to k-step ahead satisfies a MVN distribution with zero mean vector and a covariance matrix which is from the system parameters. The further between two prediction errors, the lower correlation they have. Based on the analysis of error distribution, PKF is modeled as a Markov chain. The transmission rate is obtained with the help of the CDF of the MVN. The covariance of the reconstruction errors is obtained by calculating the covariance of the doubly truncated MVN distribution. We use the Hessian matrix of the PDF of a MVN distribution for this calculation, which improves the traditional methods using moments and has generality. The result is important for WSNs, but also for other domains, e.g., statistic, economics, etc. The calculation complexity is further reduced by an approximated computation, where the distribution of the error in each state is approximated as a uniform distribution when the threshold is much less than the covariance. Using the same simulated systems, the theoretical results of the mathematical analysis including transmission rate and reconstruction accuracy follow the simulated results precisely. As expected, the approximated covariance becomes less accurate as the threshold increases.

5 Extension of PKF Using Spatial Correlation

5.1 Introduction

In Chapter 4, we have presented our PKF approach. The cluster head optimally reconstructs the state from the transmitted data of an individual node using temporal correlation. However, when there are many sensor nodes, the spatial correlation could be further exploited to improve the reconstruction quality. This chapter extends PKF to exploit spatial correlation.

We firstly find the suppression strategy and formulate the reconstruction problem for a multi-nodes system using Bayesian estimation in Section 5.2. The problem is non-linear and requires intensive computation in the cluster head. A linear reconstruction solution, termed as Rand-ST, is proposed using the incomplete information in Section 5.3.1. The feasibility of Rand-ST is estimated using the simulated system in Section 5.3.2 and analyzed from the mathematical point of view in Section 5.3.3.

In order to utilize the complete information while still solving the problem through linear approximations based on the above proposed approach Rand-ST, Section 5.4.1 proposes three different heuristic methods, EPKF-simp, EPKF-norm and EPKF-mix depending on different scenarios. The reconstructions of them are estimated and compared using simulated systems in Section 5.4.2.

5.2 Compression Strategy and Nonlinear Reconstruction Problem for Multi-nodes Systems

In Section 4.2.1, we have illustrated that the k-step ahead KF-predictor provides the best reconstruction using the information of a single node and the transmission of the local estimates are suppressed when the prediction is accurate enough. This section aims to find the compression strategy and the reconstruction solution when there are many nodes in the system.

When the system is monitored by m ($m \in \mathbb{Z}^{+}, m > 1$) nodes, we add the superscript i to the symbols related with the node ID to differentiate each node. Then z_k^i denotes the observation of node i at time k; the corresponding observation matrix is H_k^i; the measurement noise is v_k^i that satisfies a Gaussian distribution with zero mean and covariance R_k^i, $v_k^i \sim N(0, R_k^i)$; the measurement sequence of node i till time k is $\boldsymbol{Z_k^i} = [z_1^i, z_2^i, \cdots, z_k^i]$.

The direct use of PKF needs an ideal component that knows the measurement sequence of each node. It executes KF to remove the measurement noise of each node and provides the optimal estimate at time k, $\hat{x}_k = E[x_k|\boldsymbol{Z}_k^1, \cdots, \boldsymbol{Z_k^m}]$. The cluster head then uses the linear predictor Eq. (4.2) to estimate x_k based on a subset of $\boldsymbol{\hat{X}_k} = [\hat{x}_1, \cdots, \hat{x}_k]$. The ideal component follows the prediction of the cluster head using the same predictor Eq. (4.2) to guarantee the prediction quality and transmits the optimal value when the prediction is inaccurate. However, this is impractical in reality, since each node needs to firstly transmit their observations to a center to obtain $\hat{x}_k$, which significantly reduces the energy savings. Instead, each leaf node could process its own data independently. It removes the measurement noise and provides the local estimate $\hat{x}_k^i = E[x_k|\boldsymbol{Z_k^i}]$ based on its own observations. The collection of the local estimates of node i till time k is $\boldsymbol{\hat{X}_k^i} = [\hat{x}_1^i, \cdots, \hat{x}_k^i]$. Under a suppression strategy, a subset $\boldsymbol{\hat{X}_{k\_s}^i} \subset \boldsymbol{\hat{X}_k^i}$ would be transmitted to the cluster head. Then the best reconstruction in the head is $E[x_k|\boldsymbol{\hat{X}_{k\_s}^1, \cdots, \hat{X}_{k\_s}^m}]$. To guarantee the prediction accuracy, each leaf node should run the same predictor. However, it is impossible without intra-communication because of the absence of the neighbors' estimates.

To avoid extra communication, each leaf node could execute PKF independently. A subset of the local estimates of each node $\boldsymbol{\hat{X}_{k\_s}^i}$ is transmitted to the cluster head under the control of PKF, which is the compression strategy for each node. Then the best reconstruction of x_k in the cluster head based on the received local estimates from all nodes is:

$$\bar{\bar{x}}_k = E[x_k|\boldsymbol{\hat{X}_{k\_s}^1, \cdots, \hat{X}_{k\_s}^m}] \tag{5.1}$$

and the corresponding estimated observation for node i at time k is:

$$\bar{\bar{z}}_k^i = H_k^i \bar{\bar{x}}_k \tag{5.2}$$

Note that $\boldsymbol{\hat{X}_{k\_s}^i}$ consists of two components. One is the transmitted data sequence. Each element is a Gaussian random variable. The other is the accuracy indication of the prediction, which can be treated as a Boolean indicator. It equals 0, when the cluster head receives the optimal value and indicates the prediction is inaccurate; otherwise, it equals 1. Thus, the reconstruction problem in Eq. (5.1) is nonlinear. It requires intensive computation in the cluster head. In the following, we aim to find the linear reconstruction solution.

5.3 Rand-ST: Linear Reconstruction Solution Using Incomplete Information

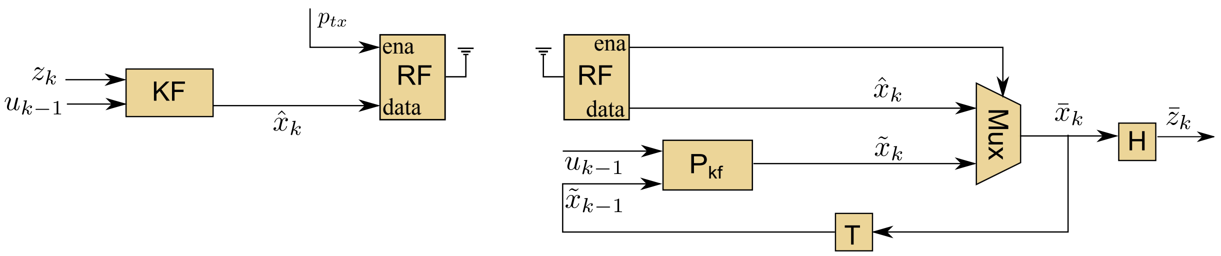
From the above analysis, when the Boolean indicator is neglected in the received data, the nonlinear problem Eq. (5.1) can be converted to a linear estimation. We aim to find the optimal reconstruction solution in this case and examine the feasibility from the simulation results and the mathematical analysis in this section.

5.3.1 Optimal Reconstruction Method Under the Random Transmission Scheme

Without using the Boolean indicator, the compression strategy is actually switched from transmitting at the time points that the predictions are inaccurate to the random transmission. Each local KF estimate has a probability p_{tx} to be transmitted. The transmitted data sequence of node i till time k is $\hat{\boldsymbol{X}}^{\boldsymbol{i}}_{\boldsymbol{k\_s}}$. The best estimate of the cluster head based on the received information from a single node is $E[x_k|\hat{\boldsymbol{X}}^{\boldsymbol{i}}_{\boldsymbol{k}}]$. It is exactly the same as the linear predictor Eq. (4.2) in PKF, while the received data sequence of the cluster head is different because of the unguaranteed transmission. The cluster head predicts the future states using Eq. (4.2) and replaces the current prediction when it receives the data from the leaf node. This approach is called Rand-idp and the diagram is depicted in Fig. 5.1.

Since the transmitted data of each node is a Gaussian random variable in this case, KF provides the optimal reconstruction solution based on the transmitted data sequence of m nodes in Eq. (5.1). Although [109] and [110] have addressed the problems using KF with intermittent observations for the systems with one and multi-nodes, the problem addressed here is more complex and we can not directly apply the solutions here. Since the observations for the KF are the transmitted local estimates, we need to find a new observation model to map $\hat{x}^i_k$ from the system state.

Combing Eqs. (3.4), (3.5), (3.9) and (3.12), the local estimate of node i at time k, $\hat{x}^i_k$,



The diagram of Rand-idp in slave node

The diagram of Rand-idp in the cluster head

Figure 5.1: The diagram of Rand-idp: the leaf node runs a KF and randomly transmits the KF estimate with a probability p_{tx}; the cluster head predicts the state using the predictor and replaces the current prediction when it receives data from the leaf node.

can be described as a function of the system state:

$$\hat{x}^i_k = K^i_k H^i_k A_{k-1} x_{k-1} + (I - K^i_k H^i_k) A_{k-1} \hat{x}^i_{k-1} + K^i_k H^i_k w_{k-1} + K^i_k v^i_k + B_{k-1} u_{k-1} \qquad (5.3)$$

It has colored measurement noise:

$$\Delta x^i_k = \hat{x}^i_k - x_k = (I - K^i_k H^i_k) A_{k-1} \Delta x^i_{k-1} + (K^i_k H^i_k - I) w_{k-1} + K^i_k v^i_k \qquad (5.4)$$

This can be solved using the modified KF with either measurement differencing [89] or state augment methods [91] as introduced in Section 3.5.2. The new system obtained by using measurement differencing is derived in Appendix A, where the system model remains the original model Eq. (3.4) and the observation model with the auxiliary signal $\bar{\hat{x}}^i_k$ to remove the correlation becomes:

$$\begin{aligned} \bar{\hat{x}}^i_{k-1} &= \hat{x}^i_k - (I - K^i_k H^i_k) A_{k-1} \hat{x}^i_{k-1} \\ &= K^i_k H^i_k A_{k-1} x_{k-1} + K^i_k H^i_k w_{k-1} + K^i_k v^i_k + B_{k-1} u_{k-1} \qquad (5.5) \end{aligned}$$

The cluster head can use a KF with the original system model Eq. (3.4) and this new observation model Eq. (5.5) for reconstruction based on the randomly and intermittently received local estimates. This solution is called Rand-ST-dec. More specifically, in Rand-ST-dec, each node randomly transmits the local KF estimates with a probability p^i_{tx}; the cluster head decolors the received data and executes a KF with the original process model and new observation model to improve the local estimation by exploiting spatial correlation. The diagram is shown in Fig. 5.2, where m_o is the original process model Eq. (3.4) and observation model Eq. (3.5), m_d is the original process model Eq. (3.4)

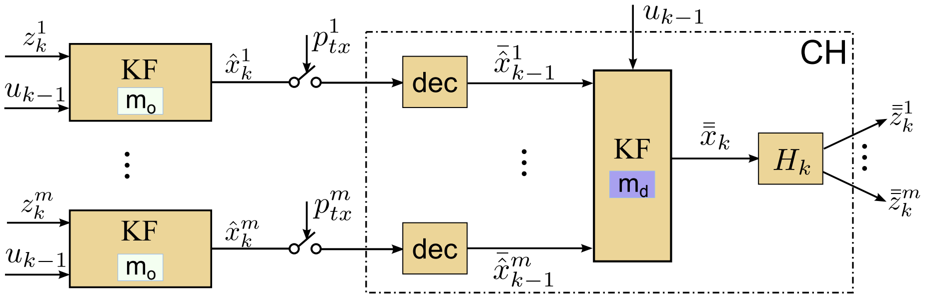


Figure 5.2: The diagram of Rand-ST-dec: each leaf node runs a KF with m_o (the original system model Eq. (3.4) and observation model Eq. (3.5)) and randomly transmits the local KF estimates with a probability p_{tx}^i; the cluster head decolors the received data and executes a KF with m_d (the original process model Eq. (3.4) and new observation model Eq. (5.5)) to improve the local estimation by exploiting spatial correlation.

and new observation model Eq. (5.5). Comparing Eqs. (5.3) and (5.5), $\hat{x}_k^i = E[x_k|\boldsymbol{Z_k^i}]$ contains all the past information of the measurements, while $\bar{\hat{x}}_{k-1}^i$ only has the information of the raw data at time k. When each node continuously transmits, there is no difference between using $\bar{\hat{x}}_k^i$ and $\hat{x}_k^i$. However, when each node intermittently transmits, a part of information is lost after decolor and using $\hat{x}_k^i$ can produce more accurate reconstructions. Moreover, to obtain $\bar{\hat{x}}_k^i$, the local estimates of two consecutive estimates are needed, which increases the transmission in reality.

To avoid the loss of useful information and extra transmission, the state augment method can be used in the cluster head to include the colored measurement noise into the state vector. Then the state is expanded as $X_k = [x_k, \hat{x}_k^1, \cdots, \hat{x}_k^m]^T$. Combining Eqs. (3.4) and (5.3), the system model is correspondingly expanded as:

$$X_k = F_{k-1}X_{k-1} + G_{k-1}U_{k-1} + W_{k-1} \tag{5.6}$$

where F_k is the expanded transition matrix with the corresponding transition coefficients from the past X_{k-1} to the current state X_k, G_{k-1} is the expanded control input matrix, U_{k-1} the expanded control input, W_{k-1} is the combined system noise matrix. They satisfy:

$$F_k = \begin{bmatrix} A_{k-1} & 0 & 0 & \cdots & 0 \\ K_k^1 H_k^1 A_{k-1} & (I - K_k^1 H_k^1)A_{k-1} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ K_k^m H_k^m A_{k-1} & 0 & 0 & \cdots & (I - K_k^m H_k^m)A_{k-1} \end{bmatrix}$$

$$G_k = [B_{k-1}, \cdots, B_{k-1}]^T$$

$$
\begin{aligned}
U_{k-1} &= [u_{k-1}, \cdots, u_{k-1}] \\
W_{k-1} &= \Big[w_{k-1}, \quad K_k^1 H_k^1 w_{k-1} + K_k^1 v_k^1, \quad \cdots, \quad K_k^m H_k^m w_{k-1} + K_k^m v_k^m\Big]^T
\end{aligned}
$$

Since $\hat{x}_k^i$ is the measurement, the expanded observation model satisfies:

$$Y_k = D_k X_k \tag{5.7}$$

where Y_k collects the transmitted local estimates of the leaf nodes at time k, which is $[\hat{x}_k^1, \cdots, \hat{x}_k^m]^T$ or a subset of it; D_k is the observation matrix, which varies from time to time depending on the transmission state of each node. For example when $m = 2$, there are two nodes in the system. The augmented state is $X_k = [x_k,\ \hat{x}_k^1,\ \hat{x}_k^2]^T$. Assuming at time k, only node 1 transmits its local estimate $\hat{x}_k^1$, then $Y_k = \hat{x}_k^1$ and $D_k = [0, I, 0]$. The KF in the head updates all states using the received data by changing the corresponding observation matrix and provides the best estimation for the expanded state vector $\tilde{X}_k = E[X_k|Y_1, \cdots, Y_k] = E[X_k|\hat{\boldsymbol{X}}_{\boldsymbol{ks}}^{\boldsymbol{1}}, \cdots, \hat{\boldsymbol{X}}_{\boldsymbol{ks}}^{\boldsymbol{m}}]$. When no data is received, the best estimate is $\tilde{X}_k^- = E[X_k|Y_1, \cdots, Y_{k-1}] = F_{k-1}\hat{X}_{k-1} + G_{k-1}U_{k-1}$, namely, the *a priori* prediction of KF for the expanded state vector. The final estimation of the real state is the first element in the expanded state vector:

$$\bar{\bar{x}}_k = S_l \tilde{X}_k \tag{5.8}$$

where $S_l = \begin{bmatrix} I & \mathbf{0} \end{bmatrix}$ and I is the identity matrix to select the estimation of the original state and $\mathbf{0}$ is an all zero matrix with the same number of columns as $[\hat{x}_k^1, \cdots, \hat{x}_k^m]$. The reconstructed observation for each node can be calculated using Eq. (5.2).

The solution that the cluster head uses KF with the state augment method to reconstruct the state based on the randomly and intermittently transmitted local estimates is called Rand-ST. The diagram is shown in Fig. 5.3. Each leaf node runs a KF with the original system model m_o and randomly transmits the local estimates $\hat{x}_k^i$ with a probability p_{tx}^i; the cluster head executes a KF with m_c (the expanded process model Eq. (5.6) and the expanded observation model Eq. (5.7)) to improve the local estimation by exploiting spatial correlation.

5.3.2 The Reconstruction of Rand-ST Using Simulated Systems

In this section, we keep using the simulated systems in Section 4.2.2 to examine the reconstruction of Rand-ST. There are three nodes in the system and they have different measurement noise as introduced in Section 4.2.2. We compare the trad-off between

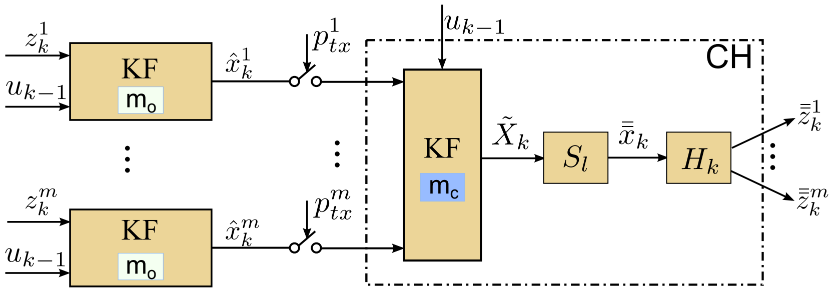
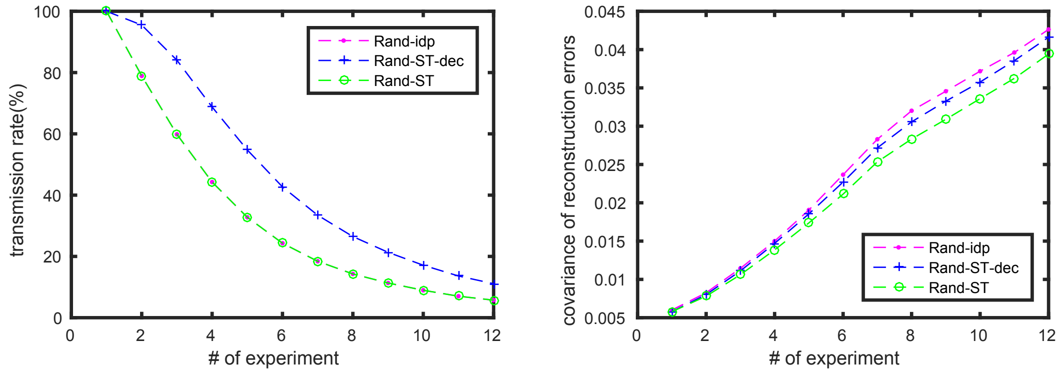


Figure 5.3: The diagram of Rand-ST: each leaf node runs a KF with the original system model m_o (the original system model Eq. (3.4) and observation model Eq. (3.5)) and randomly transmits the local estimates $\hat{x}_k^i$ with a probability p_{tx}^i; the cluster head executes a KF with m_c (the expanded process model Eq. (5.6) and the expanded observation model Eq. (5.7)) to improve the local estimation by exploiting spatial correlation. .

transmission rate and reconstruction quality of Rand-idp, Rand-ST-dec and Rand-ST, when each node randomly transmits the local estimates. Then the performance of Rand-ST is further examined, when the transmission of each node is controlled by PKF.

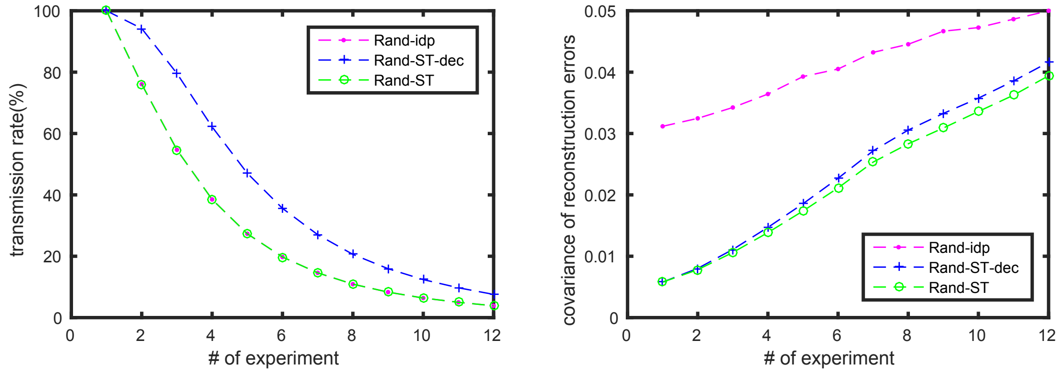
Each node randomly transmits the local estimates with various transmission probability from 100% to less than 10% as shown in Fig. 5.4a, Fig. 5.5a and Fig. 5.6a. The transmission rates of Rand-idp and Rand-ST are the same, since the cluster head uses the transmitted local estimates in Rand-ST for reconstruction. While to remove the colored noise in the local estimates, Rand-ST-dec requires more transmission. The two consecutive estimates are needed to obtain the new observation, which means if the estimates at time k, $\hat{x}_k^i$ is transmitted, the leaf node also needs to transmit $\hat{x}_{k-1}^i$ to obtain $\bar{\hat{x}}_{k-1}^i$, if it has not been transmitted.

The corresponding covariances of reconstruction errors w.r.t. the real state are depicted in in Fig. 5.4b, Fig. 5.5b and Fig. 5.6b. The reconstruction quality is further improved by using spatial correlation. Both Rand-ST-dec and Rand-ST produce smaller covariances of reconstruction error compared with Rand-idp. The best reconstruction quality can be achieved in the cluster head, when each node continuously transmits the local estimates. In this case, Rand-ST-dec produces the same covariance of reconstruction errors 0.0058 as Rand-ST. While when each node intermittently transmits, Rand-ST-dec has larger covariance than Rand-ST-dec due to the loss of useful information, which is consistent with our former analysis. When the measurement noise is smaller, the local estimation is closer to the best estimation using spatial correlation. E.g., the covariance of the local estimation error in node 1 is 0.006 which is very close to 0.0058, while the covariances of node 2 and node 3 are around 5.4 and 7.3 times as large as the covariance of the best estimation.



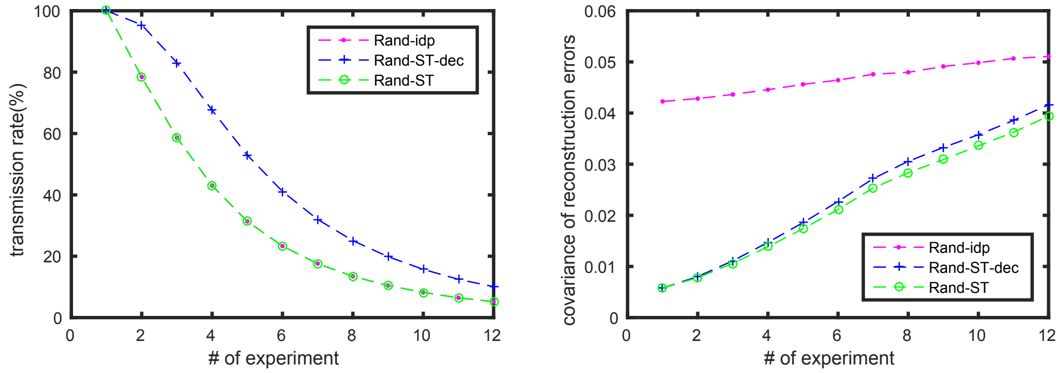
(a) (b)

Figure 5.4: Comparison of transmission rate and covariance of reconstruction errors w.r.t. real state among Rand-idp, Rand-ST-dec and Rand-ST in node 1.



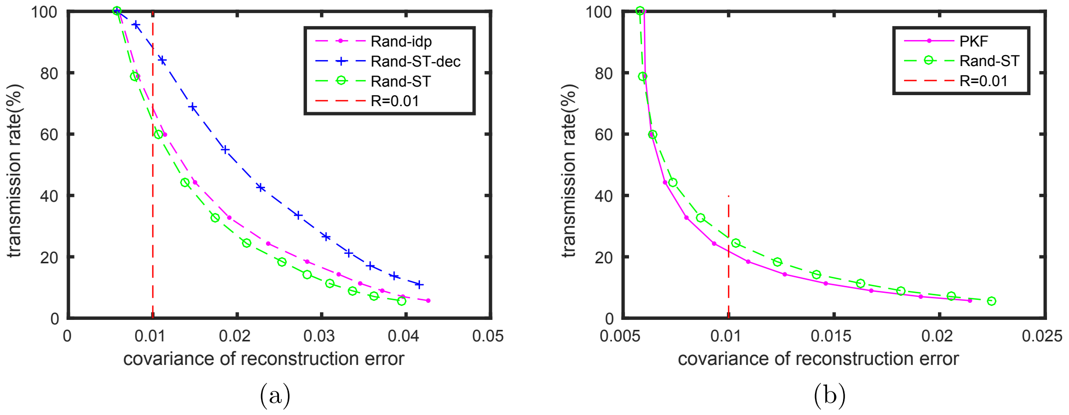
(a) (b)

Figure 5.5: Comparison of transmission rate and covariance of reconstruction errors w.r.t. real state among Rand-idp, Rand-ST-dec and Rand-ST in node 2.



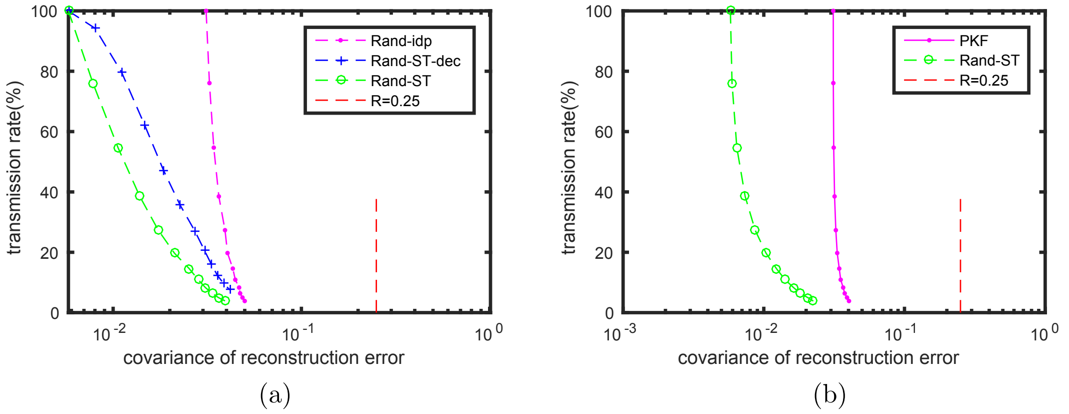
(a) (b)

Figure 5.6: Comparison of transmission rate and covariance of reconstruction errors w.r.t. real state among Rand-idp, Rand-ST-dec and Rand-ST in node 3.



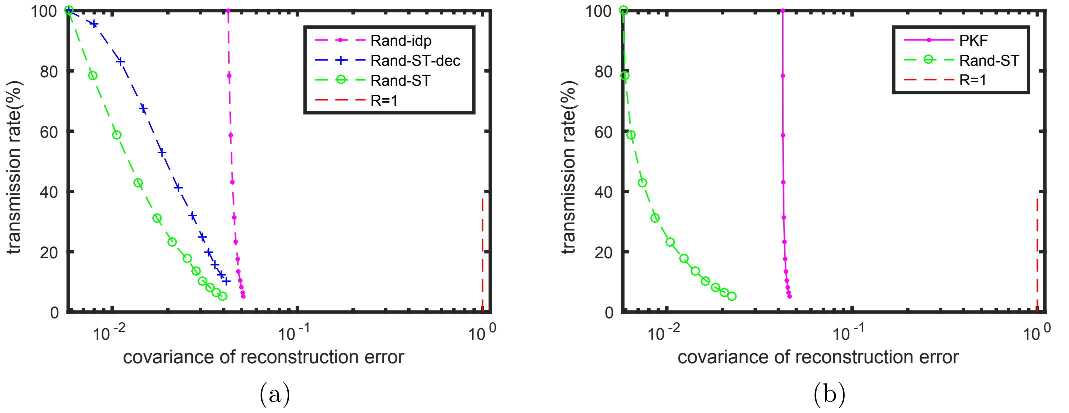
(a) (b)

Figure 5.7: Comparison of the trade-off between transmission rate and covariance of reconstruction errors w.r.t. real state in node 1. (a) among Rand-idp, Rand-ST-dec, Rand-ST; (b) between Rand-ST and PKF.



(a) (b)

Figure 5.8: Comparison of the trade-off between transmission rate and covariance of reconstruction errors w.r.t. real state in node 1. (a) among Rand-idp, Rand-ST-dec, Rand-ST; (b) between Rand-ST and PKF.



(a) (b)

Figure 5.9: Comparison of the trade-off between transmission rate and covariance of reconstruction errors w.r.t. real state in node 1. (a) among Rand-idp, Rand-ST-dec, Rand-ST; (b) between Rand-ST and PKF.

The bigger the measurement noise is, the larger improvement can be achieved by using spatial correlation. There is no big improvement in node 1 by further using neighbors' information, while the improvement becomes more obvious for node 2 and node 3 as the measurement noise increases.

The trade-off between the transmission rate and the reconstruction quality for these three nodes are shown in Fig. 5.7a, Fig. 5.8a and Fig. 5.9a. Due to the extra transmission and small improvement on reconstruction quality, Rand-ST-dec even underperforms Rand-idp in node 1. To achieve the same quality as the raw data, Rand-idp needs 68.46% transmission, while Rand-ST-dec requires about 20% more. Rand-ST reduces the transmission rate to 64% by using spatial correlation. In node 2 and node 3, the improvement on reconstruction quality becomes more obvious. Under the same transmission rate, Rand-ST-dec produces less error than Rand-idp. Rand-ST only requires 10% and 5% transmission rate to achieve the same reconstruction quality as Rand-idp with 100% transmission for node 2 and node 3, respectively.

Thus, we conclude that when each node randomly transmits the local estimates, Rand-ST is the best reconstruction solution that can always improve the reconstruction quality by further exploiting spatial correlation and the improvement compared with only using temporal correlation increases as the measurement noise increases.

Now we estimate the performance of Rand-ST when each node transmits the local estimates using PKF. By assigning appropriate thresholds, the transmission rates of each node are the same as random transmission. As shown in Fig. 5.7b, Fig. 5.8b and Fig. 5.9b, using the transmitted data controlled by PKF, Rand-ST can further improve the reconstruction quality for node 2 and node 3 under a given transmission rate, while it decreases the reconstruction quality for node 1 when the transmission rate is lower than 68.67%. It implies that Rand-ST is suboptimal when the transmission is controlled by PKF. We aim to analyze the reason in the next section.

5.3.3 Suboptimality Study of Rand-ST Under the PKF Controlled Compression Scheme

From the simulation results, we have found Rand-ST is the optimal reconstruction solution when the local estimates are transmitted randomly, while it is inefficient when the transmission is controlled by PKF. We have analyzed the reconstruction quality and the transmission rate of PKF in Section 4.3. Here we analyze the trade-off of Rand-idp to understand the suboptimality of Rand-ST method in PKF controlled transmission scheme.

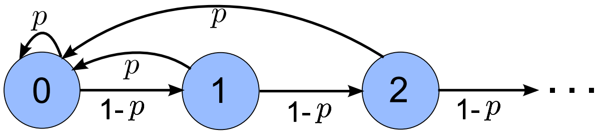


Figure 5.10: State graph of the success-runs chain for random transmission.

The reconstruction of the cluster head at time k, denoted as $\bar{x}_k$, using Rand-idp is a random variable. It could be either the prediction $\tilde{x}_k$ or the optimal value $\hat{x}_k$, which is independent of others. Suppose the outcome of the reconstruction at time k is $\hat{x}_k$, it does not affect the likelihood of getting $\hat{x}_{k+1}$ or $\tilde{x}_{k+1}$ at time $k+1$. The random variable $\bar{x}_k$ results in either *success* or *failure*: if the outcome is the prediction $\tilde{x}_k$, it corresponds to a *success*; otherwise, it is a *failure*. Different from PKF that the success probability depends on the prediction accuracy, in this case the probability of success is the same p every time the experiment is conducted. In other words, each variable $\bar{x}_k$ is associated with a Bernoulli trial [111]. Thus, the sequence of independent random variables $\bar{x}_1, \bar{x}_2, \bar{x}_3, \cdots$ is a Bernoulli process[112].

Let Γ_n denote the number of most recent consecutive successes that have been observed at the nth trial [100]. If the nth trial is a failure, then $\Gamma_n = 0$; if trial numbers $n, n-1, n-2, \cdots, n-m+1$ are all successes but trial number $n-m$ is a failure, then $\Gamma_n = m$. The collection of $\{\Gamma_1,\ \Gamma_2,\ \Gamma_3, \cdots\}$ is thereby a stochastic process, each of which is a random variable. Assuming $\Gamma_n = k$ at the nth trial, then Γ_{n+1} will equal either $k+1$ or 0 at the next trial regardless of the values $\Gamma_1, \cdots, \Gamma_{n-1}$. It means the random process satisfies the Markov property [101] and can be modeled as a discrete-time Markov chain [102].

The state space [103] of the Markov chain should be $\mathbb{N}$, which is the set of possible values of Γ_n. The transition probability of going from state k at time n to the next state $k+1$ at time $n+1$ should be $\Pr(\Gamma_{n+1} = k+1|\Gamma_n = k, \cdots, \Gamma_1 = 1) = \Pr(\Gamma_{n+1} = k+1|\Gamma_n = k) = 1-p$. Since the number of consecutive successes is independent of the time n, we can discard the time and obtain the transition probability from state k to state $k+1$ is:

$$\begin{aligned} p_{k,k+1} &= \Pr(Y_{n+1} = k+1|Y_n = k) \\ &= \Pr(Y_n = k+1|Y_{n-1} = k) \\ &= 1-p \end{aligned}$$

The transition probability from state k to the state 0 has the probability $p_{k,0} = p$. Thus, the transition matrix of the chain is:

$$\boldsymbol{P} = \begin{pmatrix} p & 1-p & \cdots & 0 & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\ p & 0 & \cdots & 1-p & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \end{pmatrix}$$

It is a time-homogeneous Markov chain [100]. The distribution over states can be written as a stochastic column vector $\boldsymbol{\pi} = \left[p_0,\ p_1, \cdots\right]^T$ with non-negative entries that add up to one [104]. The probability of the random variable Γ_n in the state k is $\Pr(\Gamma_n = k) = \boldsymbol{\pi}(k) = p_k$. The distribution in state 0, p_0, is actually the transmission rate of the leaf node. According to the steady state equation, $\boldsymbol{P}\boldsymbol{\pi} = \boldsymbol{\pi}$, and Eq. (4.14), $p_0 = p$ and the distribution over the state k when $k > 0$ is

$$p_k = p_{k-1}(1-p) = p(1-p)^k \tag{5.9}$$

The reconstruction quality of Rand-idp is measured by the covariance of the reconstruction errors w.r.t. the optimal estimate of the local KF. The total covariance is the summation of the covariance of reconstruction errors generated in each state. Compared with the KF-optimal, the error in each state is the k-step ahead prediction error ε_k, which has the normal distribution $\varepsilon_k \sim \mathcal{N}(0, \sigma_k^2)$ as analyzed in Section 4.3.1. Thus, according to the law of total probability, the final reconstruction covariance of the errors w.r.t. the optimal estimate of the local KF is:

$$\sigma_{rand}^2 = \sum_{k=0}^{\infty} p_k \sigma_k^2 \tag{5.10}$$

Note that, in the simulated systems that the real state is known, the reconstruction quality can be measured by the covariance of the reconstruction error w.r.t. the real state. The reconstruction error at state k compared with the real state is $H_k\tilde{e}_k$ (see Eq. (4.6)). It has the normal distribution with the covariance $\tilde{\sigma}_k^2 = H_k \tilde{P}_k H_k^T$ from Eq. (4.10). Then the total covariance of the reconstruction error w.r.t. the real state of Rand-idp is:

$$\tilde{\sigma}_{rand}^2 = \sum_{k=0}^{\infty} p_k \tilde{\sigma}_k^2 \tag{5.11}$$

Taking node 2 for example, when the transmission probability equals 19.75%, there are total 39 states in this experiment. The probability and the covariance of reconstruction

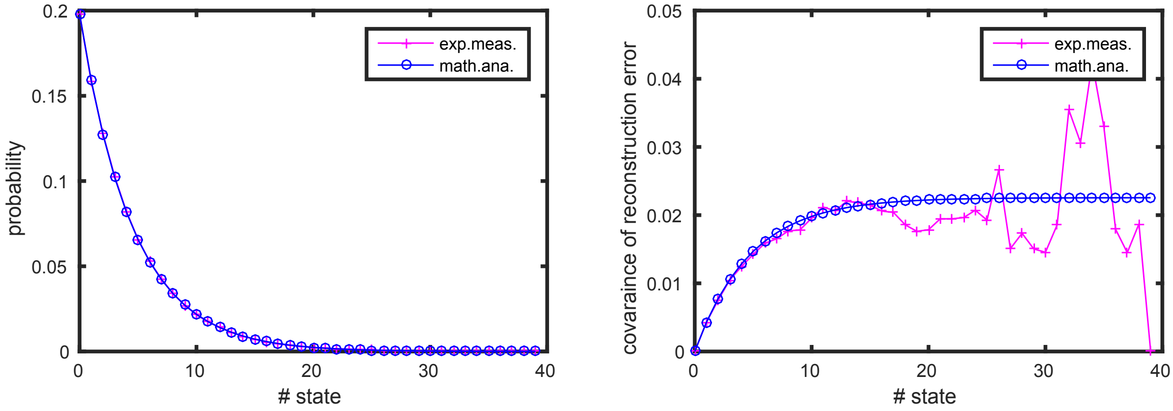
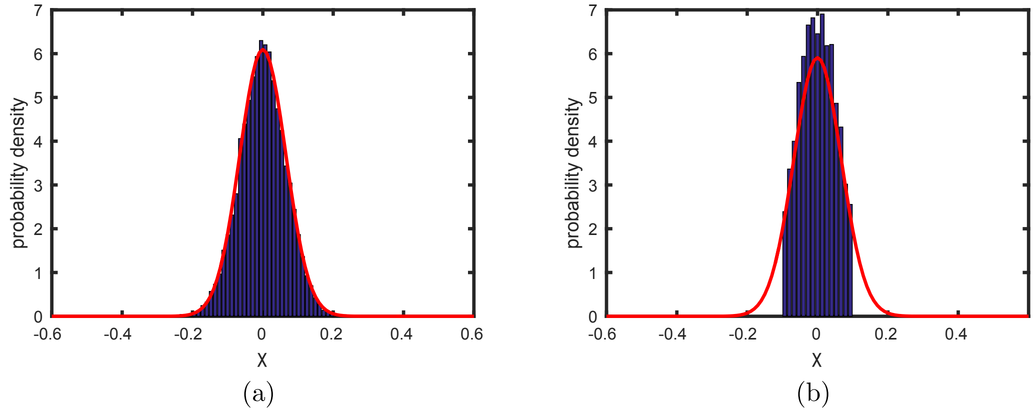


Figure 5.11: The probability and the covariance of each state in node 2 when it randomly transmits the local estimates with transmission probability 19.75%.

errors w.r.t. the optimal estimate of the local KF of each state are shown in Fig. 5.11a and Fig. 5.11b. Most of the samples are in the first 15 states. The analyzed probability fits quite well with the experiment measurement. There are some deviations between the measured and analyzed covariance when the state gets bigger, since the number of samples becomes smaller. Compared with Fig. 4.23b, the covariance of Rand-idp at each state is 10 times larger than that of PKF. The probability density distribution of the reconstruction errors in state 1, 2 and 5 w.r.t. the optimal estimate of the local KF are shown in Fig. 5.12a, Fig. 5.13a and Fig. 5.14a. They have the normal distribution. Comparing Fig. 5.12a with Fig. 4.12a, and Fig. 5.13a with Fig. 4.12b, they are exactly the same. Thus, we can conclude that when the node randomly transmits the local estimates, the reconstruction error w.r.t. the KF-optimal in state k is the k-step ahead prediction errors ε_k.

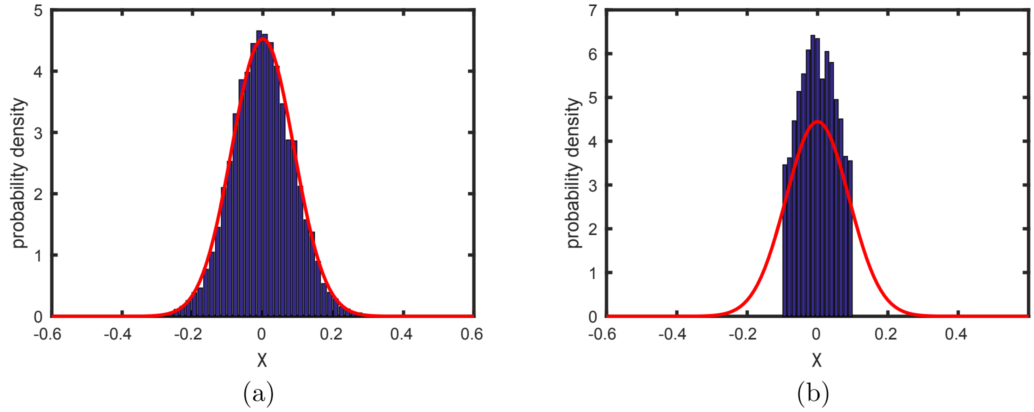
Now we examine how far is the probability density distribution of the reconstructed error of PKF at each state from the analyzed normal distribution by considering a random transmission. When each node transmits the local estimates under the control of PKF, we have depicted the error distribution at state 1, 2 and 5 in Fig. 4.24a, Fig. 4.25a and Fig. 4.26a. The probability density distribution of them are far away from the analyzed normal distribution as shown in Figs. 5.12b, 5.13b and 5.14b. This deviation degenerates the performance of Rand-ST when each node transmits under the control of PKF.

The analyzed trad-off between transmission rate and covariance of the reconstruction errors w.r.t. the KF-optimal using Eq. (5.10) follows pretty well with the experiment measurements of Rand-idp for these three nodes as shown in Fig. 5.15a, Fig. 5.16a and



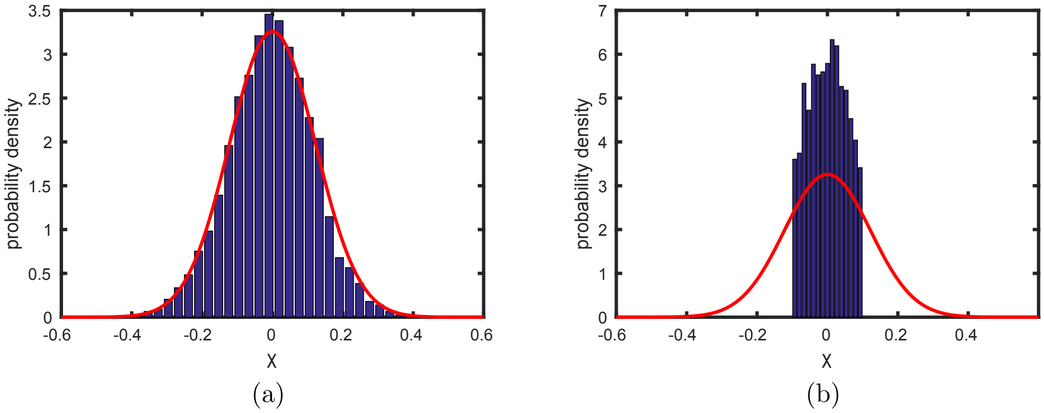
(a) (b)

Figure 5.12: Probability density distribution of reconstruction errors w.r.t. KF-optimal when $P_{tx} = 19.75\%$ in node 2 at state 1: (a) Rand-idp vs. analysis of Rand-idp; (b) PKF vs. analysis of Rand-idp.



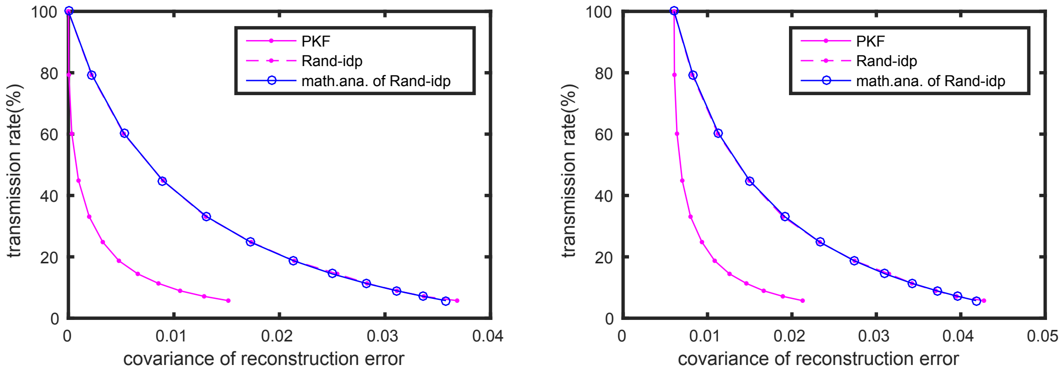
(a) (b)

Figure 5.13: Probability density distribution of reconstruction errors w.r.t. KF-optimal when $P_{tx} = 19.75\%$ in node 2 at state 2: (a) Rand-idp vs. analysis of Rand-idp; (b) PKF vs. analysis of Rand-idp.



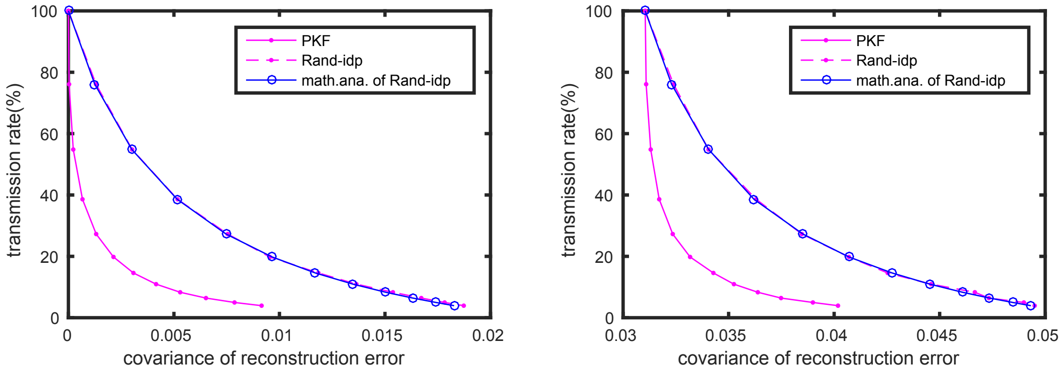
(a) (b)

Figure 5.14: Probability density distribution of reconstruction errors w.r.t. KF-optimal when $P_{tx} = 19.75\%$ in node 2 at state 5: (a) Rand-idp vs. analysis of Rand-idp; (b) PKF vs. analysis of Rand-idp.



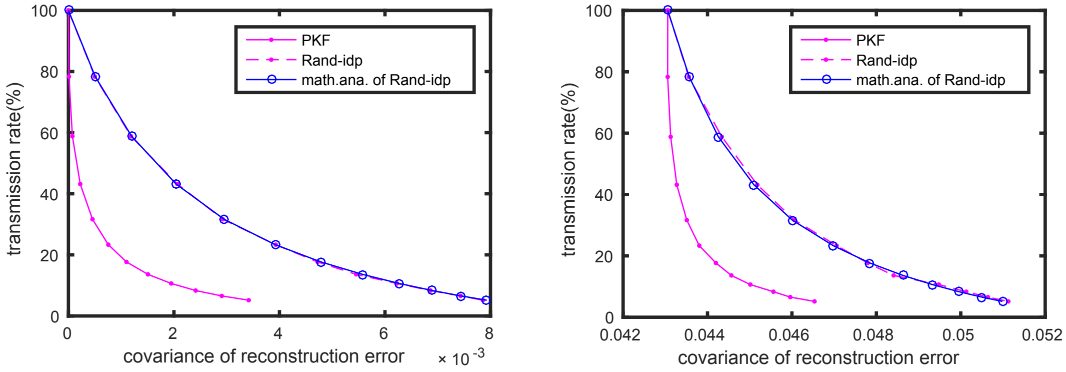
(a) (b)

Figure 5.15: Comparison of the trad-off between transmission rate and covariance of the reconstruction error among PKF, Rand-idp and the analysis of Rand-idp in node 1 (a) w.r.t. KF-optimal; (b) w.r.t. real state.



(a) (b)

Figure 5.16: Comparison of the trad-off between transmission rate and covariance of the reconstruction error among PKF, Rand-idp and the analysis of Rand-idp in node 2 (a) w.r.t. KF-optimal; (b) w.r.t. real state.



(a) (b)

Figure 5.17: Comparison of the trad-off between transmission rate and covariance of the reconstruction error among PKF, Rand-idp and the analysis of Rand-idp in node 3 (a) w.r.t. KF-optimal; (b) w.r.t. real state.

Fig. 5.17a. Compared with PKF, the covariance produced by Rand-idp is bigger under the same transmission rate. It increases from 2.05 times to 41.39 times as big as the covariance of reconstruction errors produced by PKF for node 2, as the transmission rate increases from 3.93% to 100%. The same trend holds for the trade-off between transmission rate and covariance of the reconstruction errors w.r.t. the real state using Eq. (5.11) as shown in Fig. 5.15b, Fig. 5.16b and Fig. 5.17b.

In summary, the above analysis and simulation results indicate that Rand-ST provides the optimal reconstruction by using spatial correlation, when each node randomly transmits the local estimates. The gain compared with only using temporal correlation increases as the measurement noise increases. However, when the transmission is controlled by PKF, the reconstruction error is much overestimated (e.g. 41 times in an example) by Rand-ST by treating the data sequence as randomly transmitted. In the cases that the local estimates are very accurate, Rand-ST using spatial correlation could produce even worse reconstruction than PKF with only temporal correlation.

5.4 EPKF: Linear Reconstruction Solutions Using Complete Information

We have presented a linear reconstruction solution for the nonlinear problem Eq. (5.1) in Section 5.3.1, which is proved to be impractical by neglecting the useful information in the received data sequence. In this section, we aim to utilize the complete information while still solving the problem through linear approximations based on the above proposed approach Rand-ST. There are three different heuristic methods depending on different scenarios as introduced in Section 5.4.1. The reconstructions of them are estimated and compared using simulated systems in Section 5.4.2.

5.4.1 Heuristic Reconstruction Methods

As known from Chapter 4, the untransmitted data of each leaf node indicates that the prediction in the cluster head is close to the local KF estimates with a deviation $||\epsilon^i_k|| \leq \tau$, i.e., $\hat{x}^i_k = E[x_k|\hat{\boldsymbol{X}}^{\boldsymbol{i}}_{\boldsymbol{k}}] = E[x_k|\hat{\boldsymbol{X}}^{\boldsymbol{i}}_{\boldsymbol{k\_s}}] + \epsilon^i_k = \tilde{x}^i_k + \epsilon^i_k$, where $\bar{x}^i_k$ is the reconstruction of PKF of node i at time k. Thus, we can approximate the unknown local estimates of each leaf node using this information. One possible solution is to take the deviation as zero. The alternative way is to approximate the deviation as normal distributed noise. In both cases, the nonlinear problem can be solved by the KF with the state augment method.

We firstly present the simplest method EPKF-simp. It directly takes the reconstructions of PKF as the optimal estimates of the local KF without any noise. This is motivated from Fig. 4.5b, Fig. 4.6b and Fig. 4.7b. The distribution of the reconstruction error of PKF compared with the real state in Fig. 4.7b is nearly the same as Fig. 4.5b, because the reconstruction error of PKF compared with KF in Fig. 4.6b is too small. The untransmitted estimate of node i at time k can be approximated as the reconstruction of PKF, i.e., $\hat{x}_k^i \approx \bar{x}_k^i$. Then the collection of the local estimates of node i till time k satisfies $\hat{\boldsymbol{X}}_{\boldsymbol{k}}^{\boldsymbol{i}} \approx \bar{\boldsymbol{X}}_{\boldsymbol{k}}^{\boldsymbol{i}}$, where $\bar{\boldsymbol{X}}_{\boldsymbol{k}}^{\boldsymbol{i}} = [\bar{x}_1^i, \cdots, \bar{x}_k^i]$ is the reconstruction sequence of the cluster head using PKF for node i till time k. The problem in Eq. (5.1) can thereby be approximated as:

$$\bar{\bar{x}}_k \approx E[x_k|\bar{\boldsymbol{X}}_{\boldsymbol{k}}^{\boldsymbol{m}}, \cdots, \bar{\boldsymbol{X}}_{\boldsymbol{k}}^{\boldsymbol{1}}] \tag{5.12}$$

This is equivalent to assume that each node continuously transmits the local estimates. The optimal reconstruction solution can be provided by the KF with the state augment method as introduced in Section 5.3.1. The KF in the cluster head uses the expanded process model Eq. (5.6) and the expanded process model Eq. (5.7) to improve the reconstruction quality of each node. Different from Rand-ST, the time variant observation matrix D_k is always $D = [\mathbf{0},\ I]$, where $\mathbf{0}$ is an all zero vector and I is an identity matrix with the same columns as $[\hat{x}_k^1, \cdots, \hat{x}_k^m]$. The cluster head executes KF to firstly calculate the *a priori* prediction $\tilde{X}_k^- = E[X_k|\bar{\boldsymbol{X}}_{\boldsymbol{k-1}}^{\boldsymbol{1}}, \cdots, \bar{\boldsymbol{X}}_{\boldsymbol{k-1}}^{\boldsymbol{m}}]$ for the expanded state vector. Then it takes the reconstructions $[\bar{x}_k^1, \cdots, \bar{x}_k^m]$ for each node using PKF at time k as the measurements for the KF to update the prediction and produces the *a posteriori* estimate $\tilde{X}_k = E[X_k|\bar{\boldsymbol{X}}_{\boldsymbol{k}}^{\boldsymbol{1}}, \cdots, \bar{\boldsymbol{X}}_{\boldsymbol{k}}^{\boldsymbol{m}}]$.

The computation complexity in the head can be further reduced. When the measurement has no noise, the *a posteriori* estimate of the state is actually the measurement. Thus, there is no need to update the whole expanded state vector. Only the original state needs to be updated and the other states can be filled with the reconstructions of PKF. Let $\tilde{X}_k^- = [\bar{\bar{x}}_k^-, \bar{x}_k^{1-}, \cdots, \bar{x}_k^{m-}]^T$ denote the *a priori* prediction of the augmented states using Eq. (3.9), the update equation for the whole expanded state should be $\tilde{X}_k = \tilde{X}_k^- + K_k\big([\bar{x}_k^1, \cdots, \bar{x}_k^m]^T - [\bar{x}_k^{1-}, \cdots, \bar{x}_k^{m-}]^T\big) = [\bar{\bar{x}}_k,\ \bar{x}_k^1, \cdots, \bar{x}_k^m]$. To update the original state, it can be simplified as

$$\begin{aligned}\bar{\bar{x}}_k &= \bar{\bar{x}}_k^- + C_k^1(\bar{x}_k^1 - \bar{x}_k^{1-}) + \cdots + C_k^m(\bar{x}_k^m - \bar{x}_k^{m-}) \\ &= \Big[I,\ \ -C_k^1,\ \ \cdots,\ \ -C_k^m\Big]\,\tilde{X}_k^- + \Big[C_k^1,\ \ \cdots,\ \ C_k^m\Big]\,\Big[\bar{x}_k^1,\ \ \cdots,\ \ \bar{x}_k^m\Big]^T\end{aligned}$$

$$
\begin{aligned}
&= \begin{bmatrix} I, & -C_k^1, & \cdots, -C_k^m \end{bmatrix} F_{k-1}\tilde{X}_{k-1} + \sum_{i=1}^{m} C_k^i \bar{x}_k^i \\
&= \begin{bmatrix} \phi_k^0, & \phi_k^1, & \cdots, & \phi_k^m \end{bmatrix} \tilde{X}_{k-1} + \sum_{i=1}^{m} C_k^i \bar{x}_k^i \\
&= \phi_k^0 \bar{\bar{x}}_{k-1} + \sum_{i=1}^{m} \phi_k^i \bar{x}_{k-1}^i + \sum_{i=1}^{m} C_k^i \bar{x}_k^i
\end{aligned} \tag{5.13}
$$

where ϕ_k^i is the $(i+1)$th entry in the vector $\begin{bmatrix} I, & -C_k^1, & \cdots, -C_k^m \end{bmatrix} F_k$.

The coefficients C_k^i can be extracted from the calculation of the KF gain K_k using Eqs. (3.10), (3.11) and (3.13). The a posteriori covariance of $\hat{x}_k^i$ in the augmented states should be zero, since the local KF promises the optimal estimation. Assuming ρ_k is the a posteriori estimation covariance of the original state at time k, then the covariance of the augmented states P_k follows:

$$
P_k = \begin{bmatrix} \rho_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}
$$

Then the *a priori* estimation covariance using Eq. (3.10) with the model m_c is:

$$
P_k^- = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}
$$

where $p_{11} = A_{k-1}\rho_k A_{k-1}^T + Q_{k-1}$ is the *a priori* estimation covariance of the original states; $p_{12} = [K_k^1 H_k^1 p_{11}, \cdots, K_k^m H_k^m p_{11}]^T$; $p_{12} = p_{21}^T$; The ith diagonal elements of p_{22} is $p_{22}(i,i) = K_k^i H_k^i p_{11} H_k^{iT} K_k^{iT} + K_k^i R_k^i K_k^{iT}$ and the (i,j) entry is $p_{22}(i,j) = K_k^i H_k^i p_{11} H_k^{jT} K_k^{jT}$. Because of the special shape of the observation matrix $D = [\mathbf{0} \quad I]$ and the zero measurement noise, the Kalman gain calculated by Eq. (3.11) is actually:

$$
K_k = \begin{bmatrix} p_{12} & p_{22} \end{bmatrix}^T p_{22}^{-1}
$$

The required coefficients for updating the original state is thereby:

$$
\begin{bmatrix} C_k^1 & \cdots & C_k^m \end{bmatrix} = p_{12} p_{22}^{-1} \tag{5.14}
$$

The diagram of EPKF-simp is shown in Fig. 5.18. Each leaf node runs a PKF-en independently to transmit the local estimates $\hat{x}_k^i$ when the prediction of the cluster head is inaccurate. The cluster head executes a PKF-de for each node to reconstruct the state

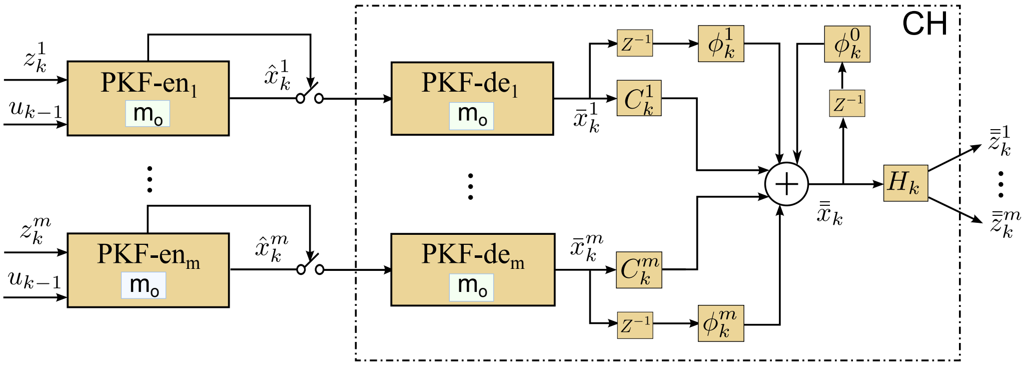


Figure 5.18: The diagram of EPKF-simp: each leaf node runs a PKF-en independently to transmit the local estimates $\hat{x}_k^i$ when the prediction of the cluster head is inaccurate; the cluster head executes a PKF-de for each node to reconstruct the state based on the transmitted data of a single node. It further takes the reconstructions for each node $\bar{x}_k^i$ as the nodes' local estimates and uses the linear combination of them to further improve the estimation for each node.

based on the transmitted data of a single node. It further takes the reconstructions for each node $\bar{x}_k^i$ as the nodes' local estimates and uses the linear combination of them with the coefficients C_k^i to further improve the estimation for each node. The approaches based on the combined system model are not suitable for the fast change systems. The system matrix F_k requires the local KF gains of each node K_k^i at each time step. It costs synchronization overhead. However, for the time invariant system and the system undergoing slow time variation, the implementation complexity can be much reduced. The KF gain and the estimation covariance of the local KF converge after several steps for the time invariant system. The combined system matrix F and the converged coefficients C^i are thereby constants, which can be easily calculated offline. The head stores these parameters. It only needs to receive $\hat{x}_k^i$ and compute Eq. (5.13) online. For the system with slow time variation, an additional synchronous process is needed to update the stored parameters at the beginning of the new system.

Now, let us illustrate the philosophy of the EPKF-norm method with an example. It approximates the reconstruction error of PKF w.r.t. the local estimates of KF as normal distribution. Naively, we can approximate all the reconstruction errors together as a normal distribution. When $\tau = 0.1$ and $\tau = 0.2$ in node 2, the entire errors approximations are shown in Fig. 5.20a and Fig. 5.20b. The mean values of them are zero. The covarinces are $\bar{\bar{\sigma}}^2$, namely the covariance of reconstructon errors of PKF w.r.t. the local estimates of KF. They are 0.0022 and 0.0092, when τ equals 0.1 and 0.2, respectively. However, as analyzed in Chapter 4, the reconstruction errors are composed of the errors generated at

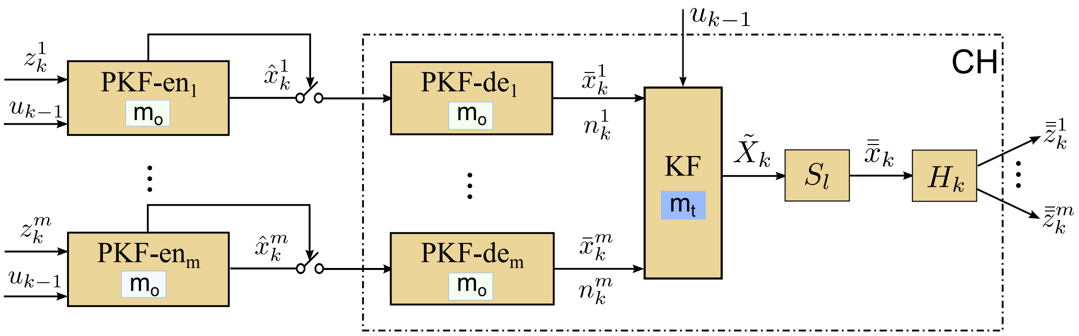


Figure 5.19: The diagram of EPKF-norm: each leaf node runs a PKF-en independently to transmit the local estimates $\hat{x}_k^i$ when the prediction of the cluster head is inaccurate. The cluster head executes a PKF-de for each node to reconstruct the state based on the transmitted data of a single node. It distinguishes the error distribution with an indicator n_k^i to denote which state is the current reconstruction from. Then it takes the reconstructions $\bar{x}_k^i$ as the measurements with the corresponding normal distributed noise and remove the noise by the KF with m_t (the expanded process model Eq. (5.6) and the expanded observation model with the time variant normal distributed noise Eq. (5.15)

different states with individual covariance and the probability in each state decreases as the state number increases. In any cases, the error produced at state 0 is zero, because the reconstructions are the received local estimates. The probability at this state is the largest. The covariance generated at state 1 and state 2 are $\bar{\sigma}_1^2 = 0.0024$ and $\bar{\sigma}_2^2 = 0.0027$ when $\tau = 0.1$; $\bar{\sigma}_1^2 = 0.0043$ and $\bar{\sigma}_2^2 = 0.0072$, when $\tau = 0.2$. The probability density distribution of them are shown in Figs. 5.21a to 5.21b. If we naively mix them together, the most often appeared errors are incorrectly approximated. A more sophisticated method should distinguish the errors produced in each state and approximate them separately. For example, the individually approximated normal distributions of the errors at the first two states are shown in Figs. 5.21a to 5.21b. Different from Rand-ST that the covariance of the reconstruction error at state k is the covariance of the k-step ahead prediction error σ_k^2, the approximated normal distribution in EPKF-norm has the truncated covariance $\bar{\sigma}_k^2$, namely the covariance of reconstruction errors of PKF generated in state k. Then the untransmitted local estimates are approximated as the reconstruction of PKF with a normal distributed noise, which depends on how many time steps ahead is the reconstruction of PKF from. If the reconstruction of PKF at time k in node i is the j-step ahead prediction, then the covariance of the error is $\bar{\sigma}_j^2$. After noise approximation, the cluster head can utilize a KF with the state augment method to improve the reconstruction of each node. The process model remains Eq. (5.6), while the observation model Eq. (5.2) becomes

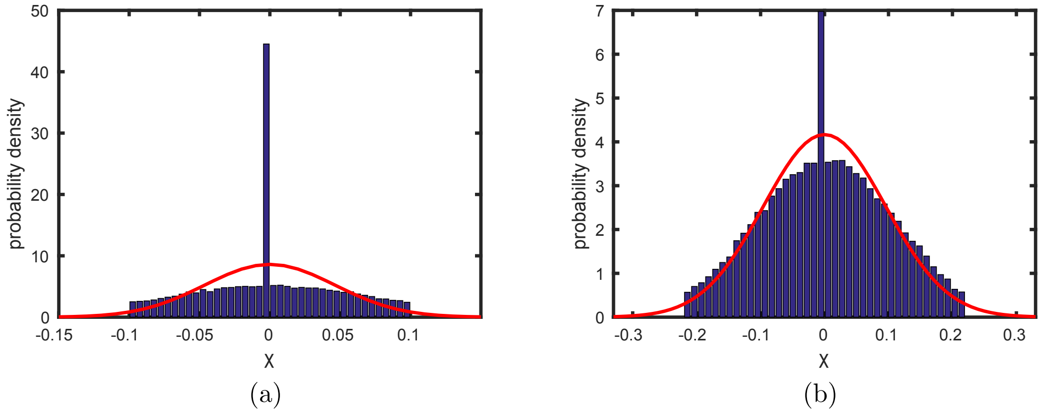
$$Y_k = DX_k + V_k \tag{5.15}$$

where $D = [\mathbf{0}, I]$. Each leaf node continuously transmits the local estimates, which is the same as EPKF-simp. The difference is that the measurement has a time variant normal distributed noise and the covariance of the noise depends on the number of step ahead that the reconstruction comes from in PKF.

The diagram of EPKF-norm is shown in Fig. 5.19, where m_t is the expanded process model Eq. (5.6) and the expanded observation model with the time variant normal distributed noise Eq. (5.15). Each leaf node runs a PKF-en to transmit the local estimates $\hat{x}_k^i$ when the prediction of the cluster head is inaccurate. The cluster head executes a PKF-de for each node to reconstruct the state based on the transmitted data of a single node. It distinguishes the error distribution with an indicator n_k^i to denote which state is the current reconstruction from. Then it takes the reconstructions $\bar{x}_k^i$ as the measurements with the normal distributed noise and remove the noise by the KF with m_t.

Comparing the approximated distributions of node 2 when $\tau = 0.1$ and $\tau = 0.2$ in Fig. 5.22, we can find that the error is distributed more close to the normal distribution than zero valued noise when the threshold gets larger. The same trend holds also for node 1 and node 3 but with different threshold intervals. Through our exhaustive experiments, we have found that when the transmission rate of each node is below about 15%, the error distributions at the first several states are closer to normal distribution. When the transmission rate is higher, there is no need to approximate the errors at each state separately, which reduces the implementation complexity. Inspired from the above results, we propose another method, called EPKF-mix. It switches the two approximation methods of the error distribution depending on the transmission rate of each node. When the transmission rate is higher than 15%, the reconstruction errors of PKF w.r.t. the estimates of the local KF is approximated as zero; otherwise, the errors produced at each state is approximated as a normal distribution with the truncated covariance $\bar{\sigma}_k^2$. After noise approximation, the cluster head utilizes a KF with the state augment method to further improve the reconstruction for each node. Compared with EPKF-norm, the process model remains Eq. (5.6), while the noise vector V_k in observation model Eq. (5.15) are filled with zero if the measurement noise of some nodes are approximated as zero. The diagram of EPKF-mix is nearly the same as EPKF-norm in Fig. 5.19 and we do not present again. The only difference is that for the node whose error is approximated as zero, the indicator n_k^i can be deleted.

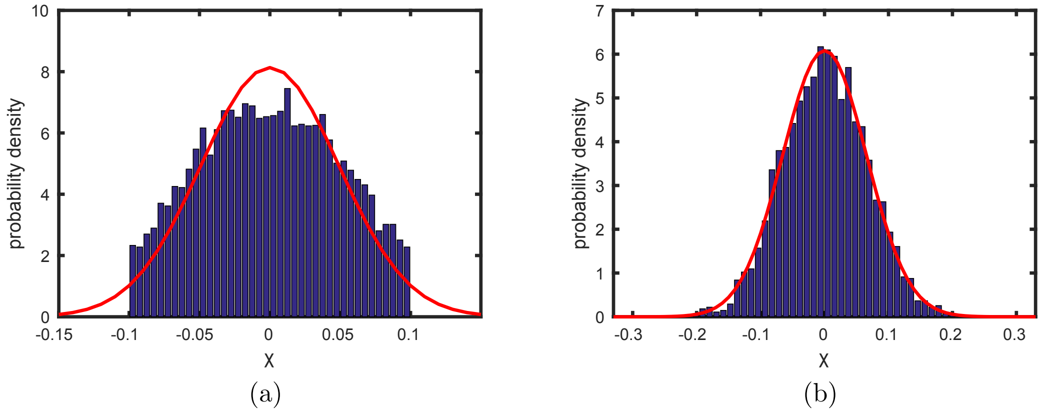
Now we compare the implementation complexity among these three heuristic methods.



(a)

(b)

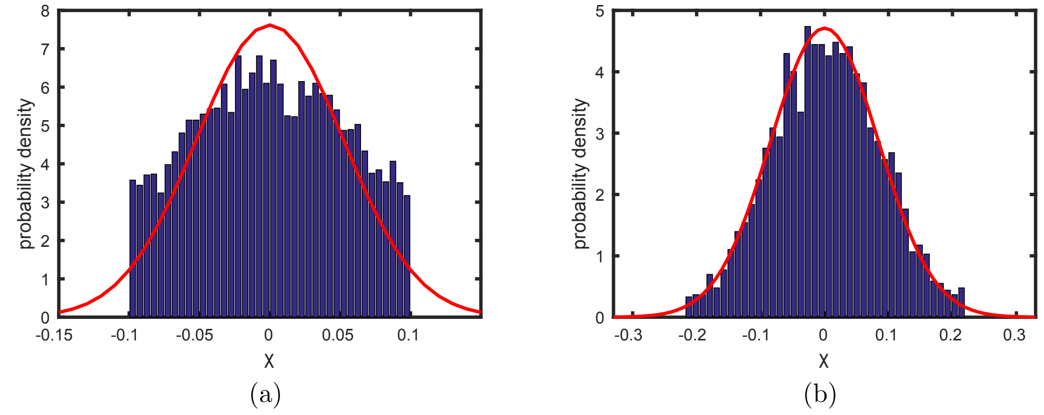
Figure 5.20: Approximating the entire reconstruction errors of PKF w.r.t local KF estimates as a normal distribution with zero mean and $\bar{\bar{\sigma}}^2$ covariance when (a) $\tau = 0.1$ (19.75%); (b) $\tau = 0.2$ (5.18%).



(a)

(b)

Figure 5.21: Approximating reconstruction errors of PKF w.r.t local KF estimates at state 1 as a normal distribution with zero mean and $\bar{\sigma}_1^2$ covariance when (a) $\tau = 0.1$ (19.75%); (b) $\tau = 0.2$ (5.18%).



(a)

(b)

Figure 5.22: Approximating reconstruction errors of PKF w.r.t local KF estimates at state 2 as a normal distribution with zero mean and $\bar{\sigma}_2^2$ covariance when (a) $\tau = 0.1$ (19.75%); (b) $\tau = 0.2$ (5.18%).

EPKF-simp neglects the reconstruction errors of PKF w.r.t. the local estimates. It directly takes the reconstructions of PKF as the estimates of the local KF without any noise. In this case, the cluster head only needs to calculate a linear combination of the reconstructions of each node, which has the simplest implementation complexity. An alternative method is EPKF-norm. It approximates the deviation between the reconstructions of PKF and the local estimates of KF as normal distributed noise and uses a KF with state augment method to remove the noise. The covariance of the noise varies depending on which step ahead the reconstruction comes from. Thus, it requires a indicator in the cluster head to mark how many steps of the current reconstruction is from the last transmission and dynamically changes the covariance of the measurement noise for the KF. The complexity is increased compared with EPKF-simp. Due to the fact that the error at each state is distributed closer to a normal distribution when the transmission rate gets lower, the EPKF-mix is proposed to adjust the two approximation methods. When the transmission rate is higher than 15%, the errors are approximated as zero; otherwise, the reconstruction error generated at each step is approximated individually as a normal distribution with zero mean and truncated covariance $\bar{\sigma}_k^2$. The cluster head then utilizes a KF with the state augment method to further improve the reconstruction for each node. Compared with EPKF-norm, the cluster head may not need the indicator n_k^i and change the covariance of the measurement noise any more for the node whose error is approximated as zero. Thus, among these three methods, it has the intermediate implementation complexity.

In summary, we have proposed three heuristic methods for the cluster head to utilize the complete information of the received data to solve the reconstruction problem Eq. (5.1) in this section. Each leaf node transmits the estimates of the local KF when the prediction is inaccurate under the control of PKF. The cluster head approximates the unreceived estimates using the reconstruction of PKF. Among them EPKF-simp has the lowest implementation complexity by always taking the reconstructions of PKF as the estimates of the local KF without any noise. EPKF-norm is the most complex method by approximating the reconstruction error at each state individually as a normal distribution. EPKF-mix adjusts the two approximation methods depending on the transmission rate of each node, which has the intermediate implementation complexity.

5.4.2 The Reconstruction of EPKF Using Simulated Systems

This section aims to estimate the reconstruction quality of the above proposed EPKF methods. The simulated system with three nodes in Section 4.2.2 is still used here to present a fair comparison. We firstly estimate the improvements of EPKF methods compared with PKF approach and examine how far the reconstructed signal of EPKF is from the real state. Then the trade-off between transmission rate and reconstruction quality is studied to compare with the results in Section 5.3.2. At last, we aim to find a trade-off between the implementation complexity and the reconstruction quality among the proposed approaches.

Table 5.1: The comparison of the covariance of reconstruction errors between using PKF and EPKF methods when $\tau_1 = 0.275$, $\tau_2 = 0.1$ and $\tau_3 = 0.132$.

	Method	node 1	node 2	node3
TX rate (%)	Any	5.72	19.75	5.18
Covariance of reconstruction errors w.r.t. real state	PKF	0.0214	0.0333	0.0460
	EPKF-simp	0.0204	0.0204	0.0204
	EPKF-norm	0.0194	0.0194	0.0194
	EPKF-mix	0.0193	0.0193	0.0193

In order to examine the improvement of EPKF compared with PKF, the threshold for node 2 is still assigned to $\tau_2 = 0.1$ as in our first experiment in Section 4.2.2. The transmission rate of node 2 using PKF is 19.75%. The thresholds for node 1 and node 3, τ_1 and τ_3 are arbitrary selected and are further adjusted to find the effect on the performance of different EPKF methods. The results when $\tau_1 = 0.275$ and $\tau_3 = 0.132$ are listed in Table 5.1. Under the control of PKF, the transmission rates of node 1 and node 3 are 5.72% and 5.18%, respectively. These rates hold for any method, since each node executes PKF independently to transmit its own estimate and the cluster head exploits the data from all nodes to further implement EPKF methods. The covariances of reconstruction errors of PKF w.r.t. the estimates of the local KF for each node are 0.0152, 0.0022 and 0.0034, respectively. Comparing the covariance of reconstruction errors of EPKF methods with that of PKF, the reconstruction quality for each node is further improved by exploiting the spatial correlation in the head. The improvement increases as the measurement noise increases. There is not too much improvement in node 1, no matter using which EPKF technique. Because of its lowest measurement noise, the temporal reconstruction of PKF is already the most accurate one among these three nodes. While the improvement for node 2 and node 3 are very obvious. For example, using the simplest method EPKF-simp that takes the reconstruction errors of each node as zero, the reconstruction quality are improved by 38.74% and 55.65% for node 2 and node 3, respectively. By approximating the reconstruction error of each node at each step as an individual normal distributed noise, the improvements of EPKF-norm are even further: 41.74% for node 2 and 57.83% for node 3. Different from EPKF-norm, EPKF-mix approximates the reconstruction error

Table 5.2: The comparison of the covariance of reconstruction errors between using PKF and EPKF methods when $\tau_1 = 0.125$, $\tau_2 = 0.1$ and $\tau_3 = 0.06$.

	Method	node 1	node 2	node3
TX rate (%)	Any	24.31	19.75	23.30
Covariance of reconstruction errors w.r.t. real state	PKF	0.0093	0.0333	0.0431
	EPKF-simp	0.0090	0.0090	0.0090
	EPKF-norm	0.0090	0.0090	0.0090
	EPKF-mix	0.0090	0.0090	0.0090

of PKF for node 2 as zero valued noise, since its transmission rate are higher than 15%. The covariance of the reconstruction errors w.r.t the real state is slightly reduced by EPKF-mix compared with EPKF-norm; and the improvement w.r.t. PKF is increased to 9.81%, 42.04% and 58.04% for each node.

To examine the spatial effect, we keep the threshold for node 2 as $\tau_2 = 0.1$ and decrease the thresholds for node 1 and node 3. The transmission rates of them are increased to 24.31% and 23.30% as listed in Table 5.2 when $\tau_1 = 0.125$ and $\tau_3 = 0.06$ as an example. All of the EPKF methods produce the same covariance of reconstruction errors w.r.t. the real state in this case. Since the transmission rate of each node is over 15%, there is no doubt that EPKF-mix should produce the same results as EPKF-simp. The covariances of reconstruction errors of PKF w.r.t. the estimates of the local KF for each node are 0.0033, 0.0022 and 0.0008, respectively. Compared with the *a posteriori* covariance of the local KFs, 0.0060, 0.0304 and 0.0426, they are very small. The two methods for approximating the very small noise do not affect a lot on the reconstruction quality. Thus, EPKF-norm produce the same covariance of reconstruction errors as EPKF-simp as well. Considering the implementation complexity of these three methods, EPKF-simp is the best solution in this case. Compared with PKF, the reconstruction quality is improved to 72.97% by using spatial correlation for node 2, because the temporal reconstructions of its neighbors with higher transmission rate are more accurate than before. The reconstructed signal of EPKF-simp is depicted in Fig. 5.23. Compared with the reconstruction of PKF, it is closer to the real state. It is even more accurate than the estimates of its local KF, because the KF in the cluster head trusts more on the less noisy data of node 1. When there are more than one node in the system, the best estimate of the system state should be the global KF-optimal, produced by a KF with all of the raw data from each node. It can be treated as the reference to measure the reconstruction quality of each methods, when the real state is absent. In Fig. 5.24a and Fig. 5.24b, we report the error distribution of PKF and EPKF-simp w.r.t. global KF-optimal in this case. Most of the reconstruction errors of

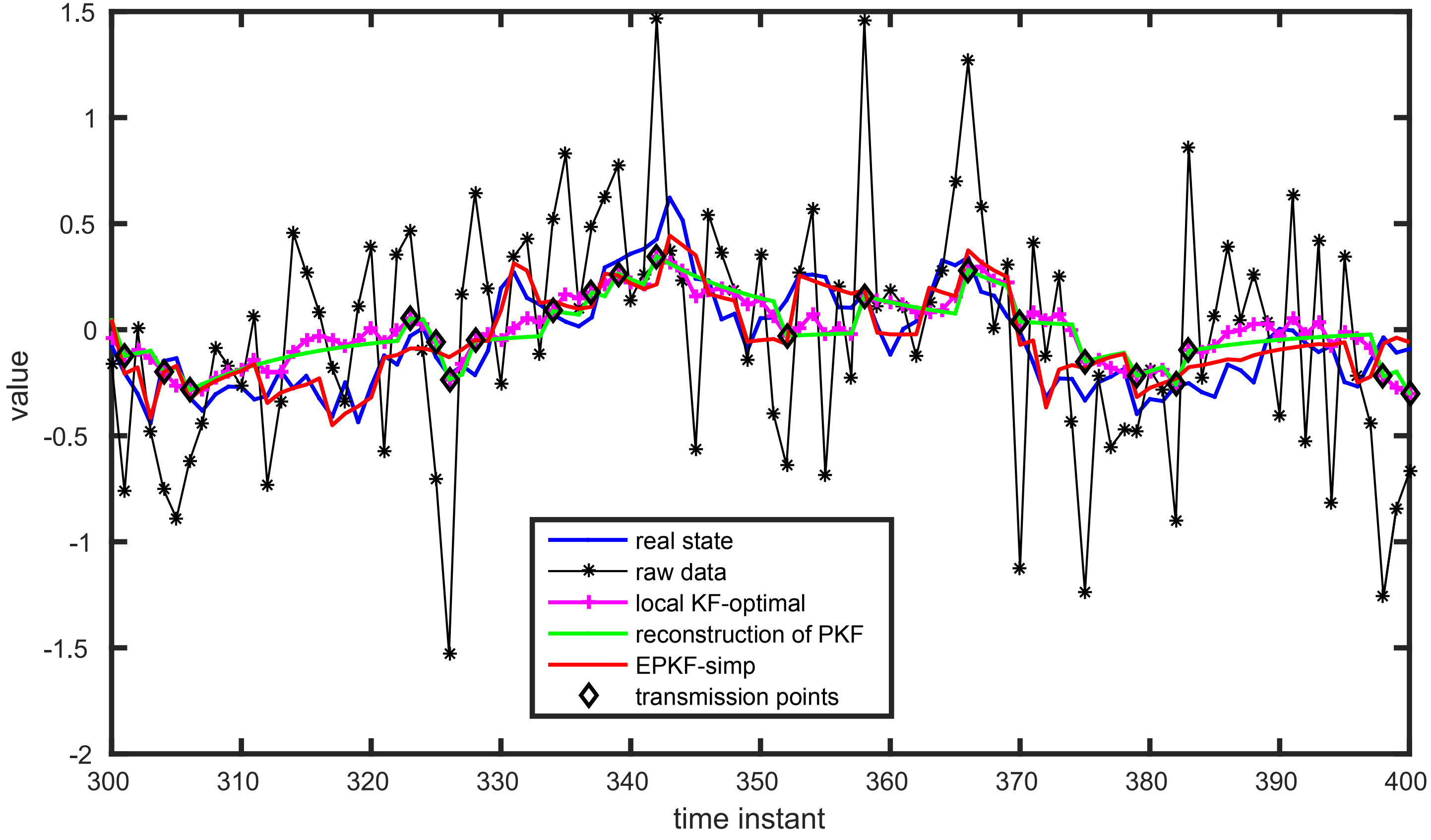
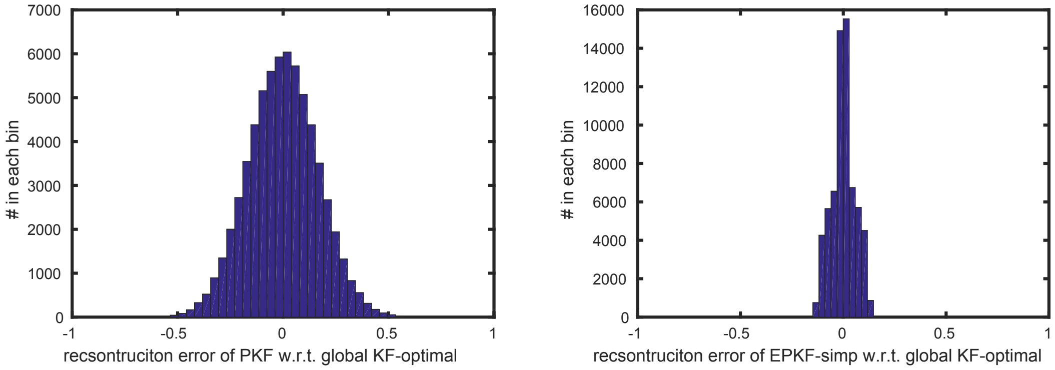


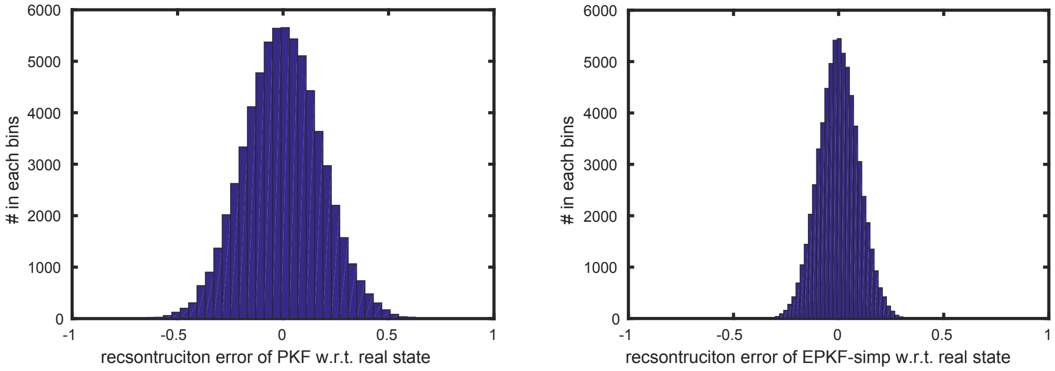
Figure 5.23: A section of reconstructed signal of EPKF compared with the raw data, the KF-optimal, the real state and the reconstructions of PKF in an artificial system with three nodes, when the transmission rates of each node are 24.31%, 19.75%, and 23.30%, respectively.

PKF are located in the interval $[-0.5 \quad 0.5]$, while in EPKF-simp they are restricted in the interval $[-0.15 \quad 0.15]$. The covariance of the reconstruction errors of EPKF-simp is 0.0032, which is around 8 times smaller than that of PKF. The distribution of the reconstruction errors w.r.t. the real state are depicted in Fig. 5.24a and Fig. 5.24b. Similarly, the reconstruction errors of EPKF-simp are distributed more concentrated in a smaller range.



(a) (b)

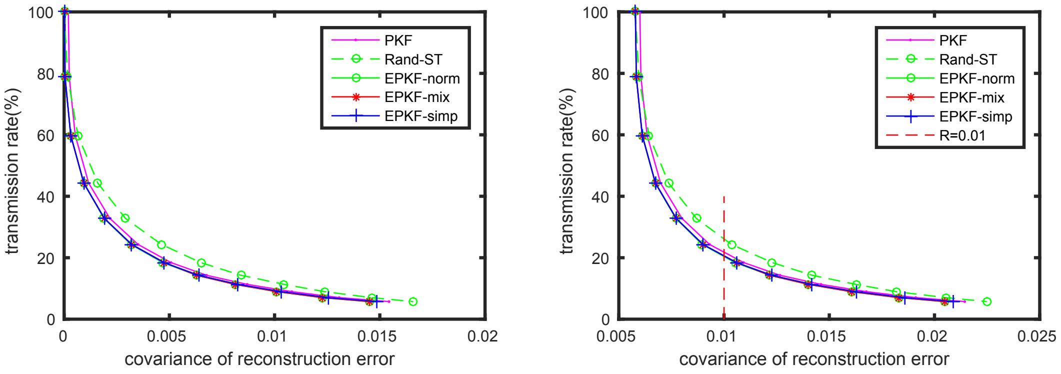
Figure 5.24: Distribution of the reconstruction error w.r.t. global KF-optimal in node 2, when the transmission rates of each node are 24.31%, 19.75%, and 23.30%, receptively: (a) PKF; (b) EPKF-simp.



(a) (b)

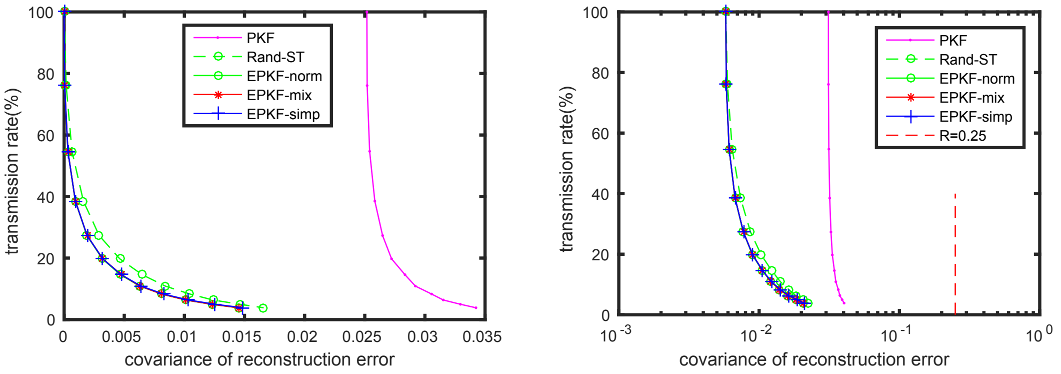
Figure 5.25: Distribution of the reconstruction error w.r.t. real state in node 2, when the transmission rates of each node are 24.31%, 19.75%, and 23.30%, receptively: (a) PKF; (b) EPKF-simp.

Now we observe the trade-off between transmission rate and reconstruction quality. The reconstruction qualities of PKF, Rand-ST and three EPKF methods are compared w.r.t. both global KF-optimal and the real state under the same transmission rate. The threshold of each node increases progressively as done in Section 4.2.2: node 1 from 0



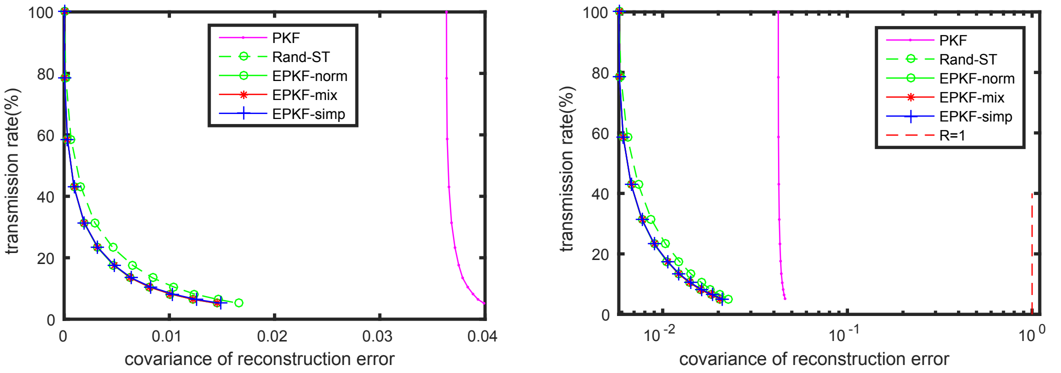
(a) (b)

Figure 5.26: Comparison of trade-off between transmission rate and covariance of reconstruction errors among PKF, Rand-ST and three EPKF methods in node 1 (a) w.r.t. global KF-optimal; (b) w.r.t. real state.



(a) (b)

Figure 5.27: Comparison of trade-off between transmission rate and covariance of reconstruction errors among PKF, Rand-ST and three EPKF methods in node 2 (a) w.r.t. global KF-optimal; (b) w.r.t. real state.



(a) (b)

Figure 5.28: Comparison of trade-off between transmission rate and covariance of reconstruction errors among PKF, Rand-ST and three EPKF methods in node 3 (a) w.r.t. global KF-optimal; (b) w.r.t. real state.

to 0.275, node 2 from 0 to 0.22 and node 3 from 0 to 0.132. The transmission rate of each node under every threshold has been reported in Figs. 4.17a, 4.18a and 4.19a in Section 4.2.2. As the transmission rate decreases, the covariance of reconstruction error increases as reported in Figs. 5.26a, 5.26b, 5.27a, 5.27b, 5.28a and 5.28b. When each node continuously transmits the local estimates, the reconstruction in the cluster head by using spatial correlation is the global KF-optimal. There is no difference at this point among Rand-ST and three EPKF methods, and the improvement for each node compared with the reconstruction of PKF increases as the measurement noise increases: 3.33 % for node 1, 81.41% for node 2 and 86.29% for node 3. However, as the transmission rate decreases, Rand-ST fails in node 1 as mentioned before. Whereas, the EPKF methods can always improve the reconstruction quality of PKF by further exploiting spatial correlation for these three nodes, although there is no big improvement in node 1. It is equivalent to further reduce the transmission of each node with the guaranteed quality. For example, the transmission rates of each node are reduced by 40 $\sim$ 95% under the same reconstruction quality as PKF when each node continuously transmits. Both Rand-ST and EPKF-norm approximate the errors as normal distribution. The former takes the covariance of k-step ahead prediction errors σ_k^2 as the covariance, while the later uses the truncated covariance of PKF $\bar{\sigma}_k^2$, which could be tens of times smaller than σ_k^2. Among the three EPKF methods, there is nearly no difference when the transmission rates of each node are small. Since the reconstruction error in this scenario is very small compared to the *a posteriori* covariance of the local KF, the two approximation methods have no obvious impact on the reconstruction quality. As the transmission rates get lower, EPKF-mix and EPKF-norm produce the same results, while the reconstruction quality of EPKF-simp is slightly lower by approximating the error as zero. For example, at the last point in Fig. 5.26a when the transmission rate of these three nodes are 5.72%, 3.85% and 5.18%, respectively, the covariance produced by using EPKF-mix and EPKF-norm is 0.0205, while it is 0.0209 by using EPKF-simp. Nevertheless, this small deviation can be neglected compared with its lowest implementation complexity.

5.5 Summary

We have extended PKF to further exploit spatial correlation for the multi-nodes system in this chapter. Each leaf node executes a PKF independently. A subset of the local KF-estimates is transmitted to the cluster head under the control of PKF. The cluster head collects the data from each node and reconstructs the system state by using spatial correlation. The reconstruction problem is formulated using Bayesian estimation, which is nonlinear and requires intensive computation. By using the incomplete information, the problem can be converted to a linear estimation problem. In this case, the compression strategy is switched from transmitting at the time points that the predictions are inaccurate to the random transmission. The KF can produce the optimal reconstruction. Since the transmitted local estimates of each node have colored noise, the modified KF with measurement differecing and state augment method are used, which corresponds to our Rand-ST-dec and Rand-ST methods. Through the analysis and the simulation results, Rand-ST is proved to be more accurate than Rand-ST-dec. It provides the optimal reconstructions when each node randomly transmits the local estimates. The transmission rate can be further reduced by 95% compared with only using temporal correctional for a node because of the improved reconstruction quality. However, under the PKF controlled transmission, the covariance of reconstruction errors produced by Rand-ST is 12% larger than PKF with only spatial correlation. The suboptimality is analyzed through a detailed analysis. The reconstruction error of PKF is much overestimated in Rand-ST due to the neglect of the useful information, e.g., 2.05 times to 41.39 times for a node as the transmission rate increases from 3.93% to 100%.

In order to utilize the complete information while solving the problem through linear approximations, we have proposed three heuristic methods based on Rand-ST. Each leaf node transmits the estimates of the local KF when the prediction is inaccurate under the control of PKF. The cluster head approximates the unreceived estimates using the reconstruction of PKF. The simplest method is EPKF-simp, which neglects the reconstruction errors of PKF w.r.t. the local estimates. It directly takes the reconstructions of PKF as the estimates of the local KF without any noise. In this case, the cluster head only needs to calculate a linear combination of the reconstructions of each node, which has the simplest implementation complexity. However, as the transmission rate decreases, the error is distributed more close to a normal distribution than zero valued noise. An alternative method EPKF-norm is proposed. It approximates the reconstruction errors of PKF w.r.t. the local estimates of KF at each state individually as a normal distribution and uses a KF with state augment method in the cluster head to remove the noise. It requires high implementation complexity. The third method EPKF-mix is proposed to adjust the two approximation methods. When the transmission rate is higher than 15%, the errors are approximated as zero; otherwise, the reconstruction error generated at each step is approximated individually as a normal distribution with zero mean and truncated covariance $\bar{\sigma}_k^2$. The cluster head then utilizes a KF with the state augment method to further improve the reconstruction for each node. Among these three methods, it has the intermediate implementation complexity.

The simulation results illustrate that EPKF methods can further improve the reconstruction quality by using spatial correlation. The improvement increases as the measurement noise increases. For example, the improvements w.r.t. to PKF are 3.33 %, 81.41%, and 86.29% as the covariance of the measurement noise increases from 0.01 to 1 in three nodes. It is equivalent to further reduce the transmission of each node with the guaranteed quality. In this case, the transmission rates of each node are reduced by 40 $\sim$ 95%. For a given node, the gain by using EPKF methods increases as the reconstruction quality of its neighbors increase. Taking node 2 for example, the improvement of reconstruction quality is increased from 41.74% to 72.97%, when the transmission rates of its neighbors increase from 5.72% to 24.31%. Comparing these three methods, when all the nodes have high transmission rate, the reconstruction qualities of them are the same. For example, when the transmission rates of three nodes are 24.31%, 19.75% and 23.30%, respectively, the covariances of reconstruction errors produced by three EPKF methods are the same 0.009. Considering the implementation complexity, EPKF-simp is the best candidate. If all nodes have very small transmission rate, EPKF-mix and EPKF-norm are the same. They have better reconstruction quality than EPKF-simp. For instance, when the transmission rates of three nodes are 5.72%, 3.85% and 5.18%, respectively, the covariance of reconstruction errors produced EPKF-mix is 0.0205, which is the same as EPKF-norm; and EPKF-simp has the covaraince 0.0209. Another case is that some nodes have much higher transmission rate than others, EPKF-mix produces the least reconstruction error. For example, when the transmission rate of three nodes are 5.72%, 19.75% and 5.18%, the covariances of reconstruction errors produced by EPKF-simp, EPKF-norm and EPKF-mix are 0.0204, 0.0194 and 0.0193, respectively.

6 Experimental Results

6.1 Introduction

In Chapter 4 and Chapter 5, we have provided the first look at the reconstructions of PKF and EPKF from the arbitrary simulated systems. This section aims to evaluate the performance of our techniques using the real world data and real hardware implementation. The experimental evaluation is structured into two phases. We firstly use two kinds of real temperature datasets taken from typical WSN scenarios with a single node to measure the performance of PKF in Section 6.2. In order to demonstrate the superiority of PKF, we compare it with DFK [56], SIP [55], PAQ [28], PLAMLiS [18] and CS [58]. Moreover, to illustrate that PKF can work in time variant systems, a simple example is also presented. In Section 6.3, the robustness of our mathematical analysis of PKF is evaluated using these datasets, by comparing with the experimental measured results. We further estimate the performance of EPKF using the datasets with different sizes of the cluster in Section 6.4. Besides the comparison among the PKF-based approaches, EPKF-simp, Rand-ST, EPKF-norm and EPKF-mix, we compare them with other techniques using both temporal and spatial correlations: EEDC [18] and CS [58].

In the second phase, to measure the energy consumption and lifetime improvement using our proposed approaches, the algorithms are implemented in the WSN motes, Openmote [113], running on Contiki OS [114] in Section 6.5. An arbitrary network is formed with four leaf nodes and one master node to measure the transmission rate of each node and the improvements on reconstruction quality by further implementing EPKF in the cluster head. Then the current profile of the leaf node during each process is visualized on an oscilloscope by measuring the voltage drop over a fixed resistor. The computation energy consumption of PKF and the communication energy consumption are compared to examine how much energy can be saved using PKF. At last, the lifetime improvement using PKF is studied by considering the overall per-day current consumption of the leaf node and using the obtained transmission rate of the leaf node.

6.2 Estimation of PKF Using WSN Datasets

The quality of the reconstructed signal of PKF is evaluated by the covariance of reconstruction errors with respect to the KF-optimal values, since the real state is absent when using the real datasets and KF provides the optimal estimate of the system state. It is compared with DKF [56], SIP [55], PAQ [28], PLAMLiS [18] and CS [58] approaches. Both DKF [56] and SIP [55] exploit a KF for noise reduction from the raw data and further reduce the transmission rate using different predictors based on the preprocessed data. Whereas, PAQ [28], PLAMLiS [18] and CS [58] are the popular techniques without KF to achieve this aim. The detailed introduction of the techniques can be found in Chapter 2. To illustrate the ability of PKF working in time variant systems, we present an example using offline stored models.

For these evaluations, we use two temperature datasets: dataset 1, 1024 values in singlehop-indoor-moteid1 and dataset 2, 3600 values in singlehop-indoor-moteid2 presented in [115]. The data is collected at intervals of 5 seconds from a simple single-hop WSN deployment using TelosB mote. In order to find the effect of the underling system models on the quality of the PKF approach, we establish two different models for each dataset without control inputs: PKF-constant and PKF-linear. In the constant model, there is only one variable in the state space, whereas in the linear model, the temperature value is considered to vary with a velocity $\dot{v}_k$. The system parameters including A, H, R, Q are obtained using Matlab system identification toolbox [116, 117] to fit the first 1024 data points of each dataset.

In order to provide fair comparisons, the required parameters of each approach are initialized to be consistent with PKF. The system parameters of the DKF [56] approach are optimized by Matlab as well. According to the process of DKF, each leaf node performs two KFs, where the first KF is to reduce noise. Here, we let it have the same parameters as PKF-linear. Since the output of the first KF is treated as the measurements input for the second KF and the state transition matrix is required unchanged, we further optimize the new parameters H, R, Q for the second KF. In SIP [55], the same KF filter as PKF-linear is selected to remove the noise, which is claimed to have the best reconstruction quality using their approach. The leaf node uses piece-wise linear to estimate the state, which consists of the current smoothed measurement and the deviation from the last smoothed data. The cluster head also uses PLA to predict the current state with the last received state. Two-order AR model of the raw data is obtained for PAQ [28]. The cluster head stores these two coefficients and the past two data points for the current prediction. For PLAMLiS [18], we assume the size of buffer is enough to store all values of dataset2. The leaf node calculates the line segment to approximate the raw data given an error bound and transmits the end points of each line segment to the cluster head. In order to implement CS [58], we exploit the discrete cosine transform (DCT) as the representation basis, Ψ, for these two datasets, which can sparsify the original signals sufficiently. An independent and identically distributed (i.i.d.) Gaussian matrix is used for random projection, Φ, which is incoherent with Ψ. The signal is reconstructed by ℓ_1 minimization method [118].

The trade-offs between transmission rate and reconstruction quality among these approaches are depicted in Fig. 6.1 and Fig. 6.2. As the number of transmission decreases, the quality of the reconstruction degrades. The transmission rates of these approaches gradually converge to a similar value but with different speed[1].

PKF-linear outperforms PKF-constant. It indicates that the performance of PKF relies on the accuracy of the system model, which is a common issue for the model-based techniques. The more accurate the model is, the better performance PKF achieves.

Compared with the approaches that use the same linear KF for noise reduction, namely, DKF [56] and SIP [55], PKF-linear requires the fewest transmission rate under the same covariance of reconstruction errors. It is interesting to compare the reconstruction quality with the covariance of measurement noise before KF filtering. As analyzed by the system identification toolbox, the covariance of the measurement noise for dataset 1 and datase 2 are 0.33 and 0.41, respectively. Without data degradation, PKF-linear only needs 11.11% and 11.91% transmission, which in turn saves the transmission of DKF by 23.11% and 13.93%, respectively. The second KF in DKF [56] uses the output of the first KF as the measurements. Then the optimal reconstruction method should be that the cluster head uses a KF with the combined system model m_c, since the estimate of the first KF contains colored noise as analyzed in Eqs. (5.3) and (5.4). When it receives the update, it calculates the KF gain, conducts the *a posteriori* estimate and obtains the *a posteriori* estimate covariance, which actually replaces the prediction by the update, since the first KF promises the optimal estimate for a linear system. In other words, the optimal reconstructions of KF in the head with the state augment method should be the same as PKF when there is only one node. However, the state transition matrix is required to be unchanged in DKF [56], which means the reconstructions produced by the second KF is suboptimal. Thus, the reconstruction quality of DKF is worse than PKF. SIP requires around 25.52% and 19.74% transmission to achieve the same reconstruction

[1] Here we only present the performance of CS with similar reconstruction quality as other techniques. The transmission rate of CS converges to 10% when the covariance of reconstruction errors is larger than $0.001°\text{C}^2$.

quality as the raw data, which are around 129.70% and 65.74% more transmission than PKF-linear. After noise reduction using the same liner KF as PKF-linear, the system state is estimated from another approximation method PLA in the leaf node instead of directly using the optimal estimate of the KF. The cluster head then uses PLA to predict the approximation of the state and the leaf node follows the prediction of the cluster head to guarantee the reconstruction quality. These approximations degrades the reconstruction quality.

Compared with the techniques without KF, the advantages of PKF are even more significant because of the KF. Without quality degradation, PKF-linear decreases the transmission rate of PAQ [28] by 37.98% and 29.08% for dataset 1 and dataset 2, respectively. To achieve the same quality as the raw measurements, PLAMLiS [18] and CS [58] require 2.14 and 6.60 times as much transmission as PKF for dataset 1; and 2.06 and 6.56 times for dataset 2, respectively. These approaches only provide the approximations of the raw data. As the measurement noise increases, the superiority of PKF could become more obvious.

We further illustrate how PKF can be used in time variant systems using real temperature values. Since the above used datasets are collected over several hours, they are too short to cause a change of the system model. Thus, another dataset from a typical WSN testbed is selected as shown in Fig. 6.3a. It is collected by Tinynode at intervals of 30 seconds on 4th. Nov. 2006 from sensorscope-meteo44 in LUCE WSN testbed [119].

Assuming the system is invariant over the whole day, we can find the system model m_{iv} with the aid of Matlab system identification toolbox. Under this model, the tradeoff between transmission rate and reconstruction quality using PKF is shown in Fig. 6.3b. As depicted in Fig. 6.3a, the measurements have larger noise during the day time, from 9 am to 17 pm. We can divide the data of the whole day into three parts: from 0 am to 9 am, 9 am to 17 pm and 17 pm to 0 am as shown in Fig. 6.3a. Each part has different system parameters and we can obtain the corresponding system model, m_1, m_2 and m_3. Both the leaf node and cluster head store these models and update them in time. The performance of PKF using these time variant models, is shown in Fig. 6.3b. Compared with using the time invariant model, the reconstruction quality of PKF is slightly improved, which is consistent with the above result that the more accurate the model is, the better performance PKF achieves.

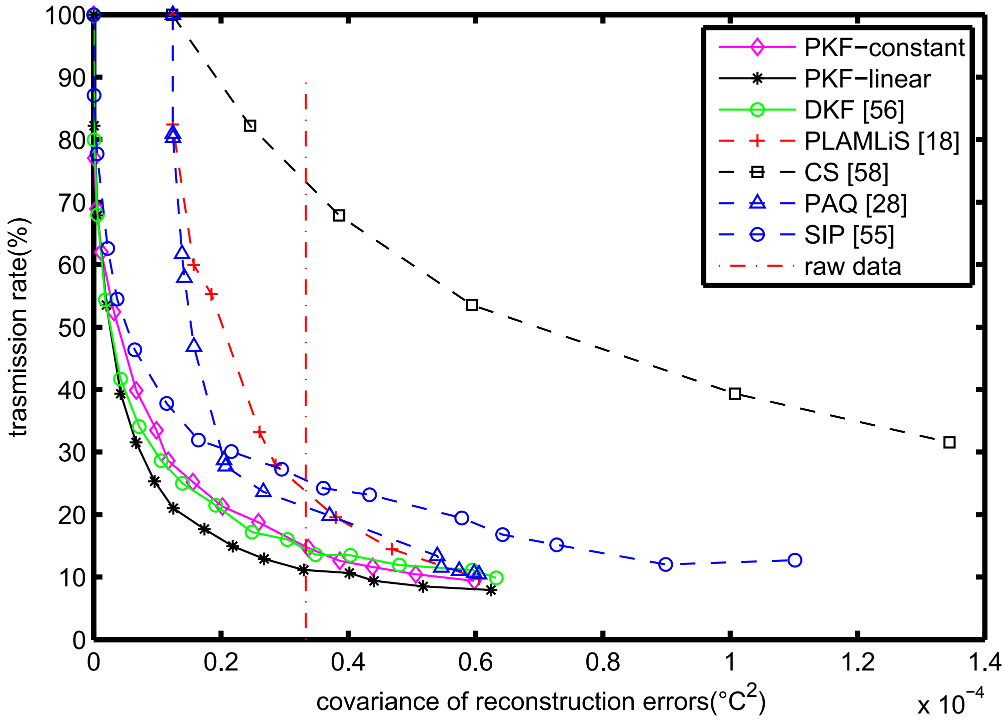


Figure 6.1: Performance comparisons of PKF, DKF [56], PAQ [28], PLAMLiS [18], CS [58] and SIP [55] using dataset 1.

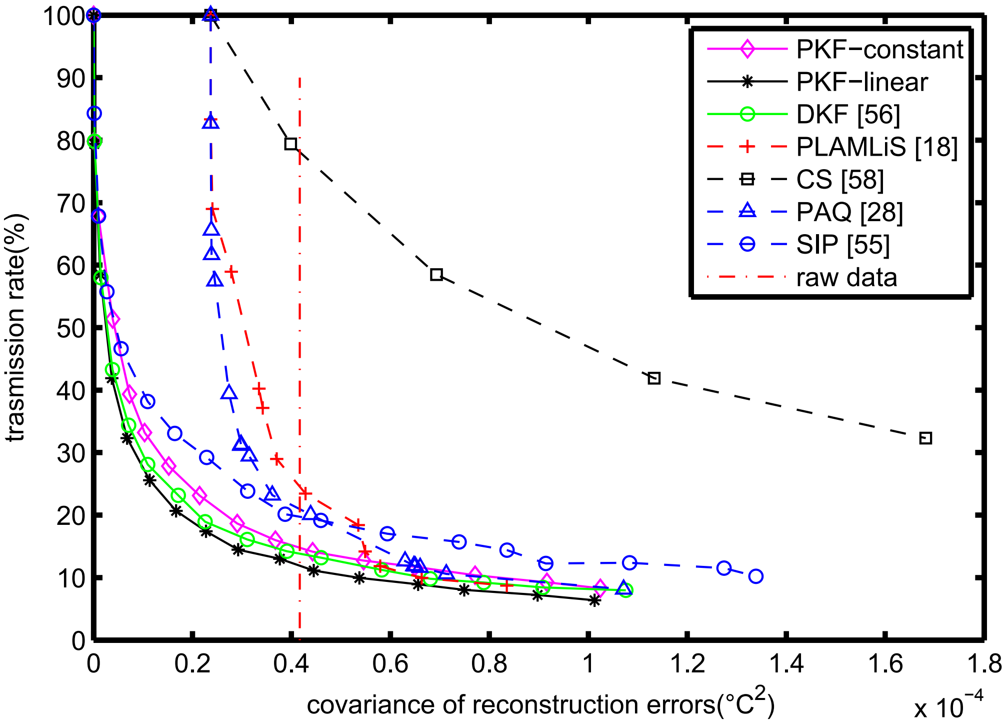


Figure 6.2: Performance comparisons of PKF, DKF [56], PAQ [28], PLAMLiS [18], CS [58] and SIP [55] using dataset 2.

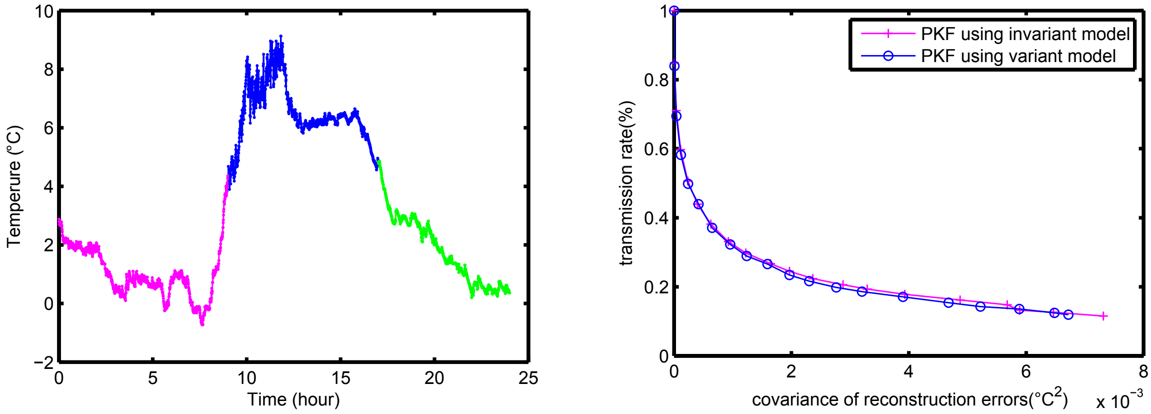


Figure 6.3: (a) Raw data of a node collected in one day can be separated into three segments according to the measurement noise with time variant models;
(b) Performance of PKF in the time variant system using offline stored system parameters.

6.3 Estimation of Math Analysis Using WSN Datasets

We have validated our analysis using the simulated systems in Chapter 4. This section evaluates the robustness of our mathematical analysis using dataset 1 and dataset 2.

The transmission rate decreases as the threshold increases. The analyzed results using Eq. (4.17) for both PKF-constant and PKF-linear follow the experimental measurements as shown in Fig. 6.4a and Fig. 6.4b using dataset 1. The similar results hold also when the node using dataset 2 as shown in Fig. 6.7a and Fig. 6.7b. The comparison of the covariance of reconstruction errors among experimental measurements, mathematical analysis using Eq. (4.22), and approximation using Eq. (4.40) are depicted in Figs. 6.5a, 6.5b, 6.8a and 6.8b when the node executes PKF-constant and PKF-linear using dataset 1 and dataset 2, respectively. As usual, the approximated covariance of reconstruction errors using Eq. (4.40) becomes less accurate as the threshold increases. Combining the results of transmission rate and reconstruction quality, we obtain the trade-offs between them. The comparison of experimental results and mathematical analysis are reported in Figs. 6.6a, 6.6b, 6.9a and 6.9b. The node needs to spend more energy on communication for better reconstruction quality. Although there are some deviations between the mathematical analysis and the experimental measurements, they are reasonable and acceptable, because the system models are uncertain and the noise distribution may not perfectly satisfy Gaussian distribution.

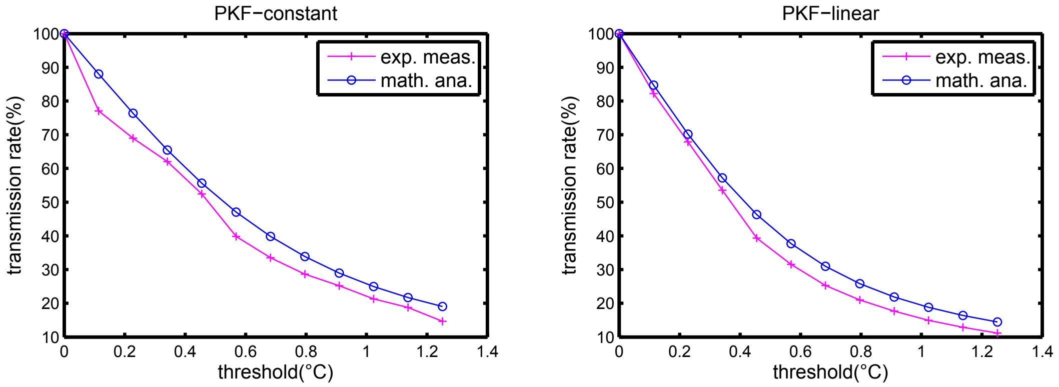


Figure 6.4: Comparison of transmission rate between experimental measurements and mathematical analysis Eq. (4.17) in (a) PKF-constant and (b) PKF-linear using dataset 1.

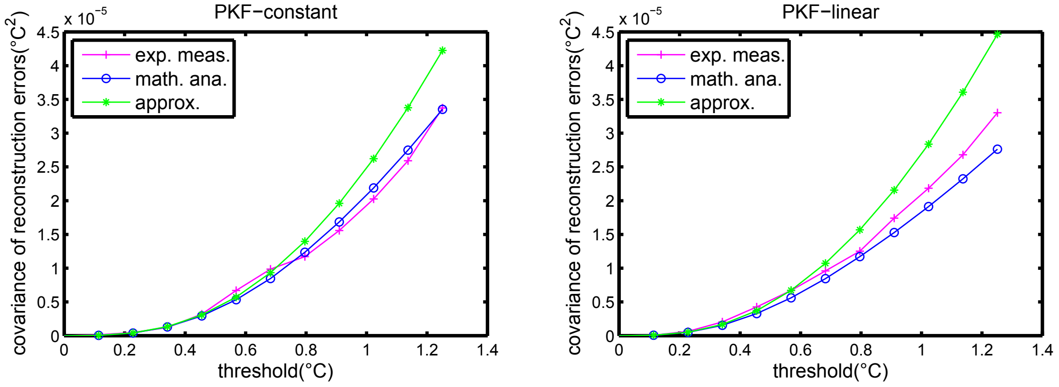


Figure 6.5: Comparison of covariance of reconstruction errors among experimental measurements, mathematical analysis Eq. (4.22), and approximation Eq. (4.40) in (a) PKF-constant and (b) PKF-linear using dataset 1.

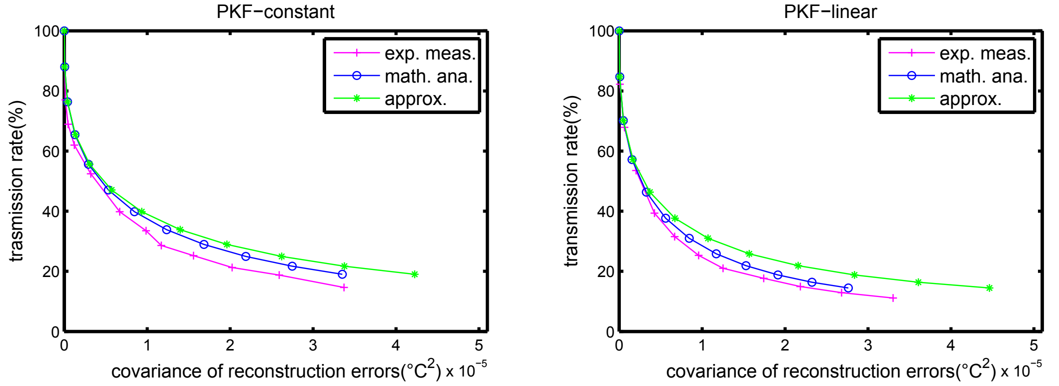


Figure 6.6: Comparison of the trad-off between transmission rate and covariance of reconstruction errors among experimental measurements, mathematical analysis Eqs. (4.17) and (4.22), and approximation Eqs. (4.17) and (4.40) in (a) PKF-constant and (b) PKF-linear using dataset 1.

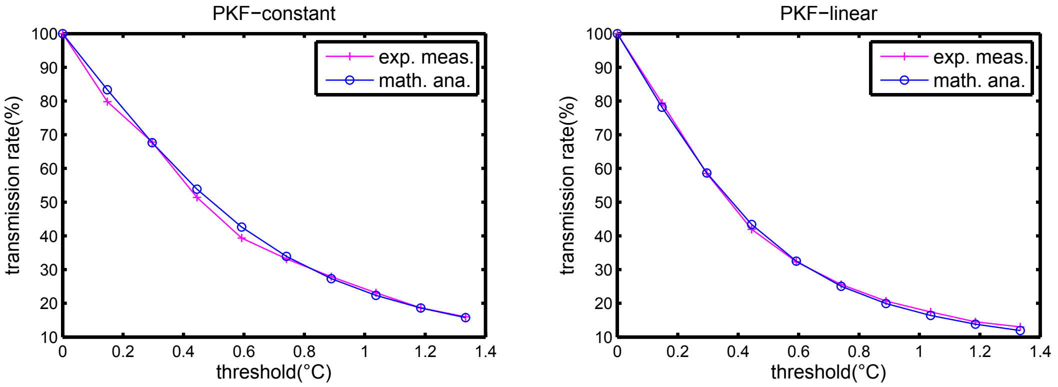


Figure 6.7: Comparison of transmission rate between experimental measurements and mathematical analysis Eq. (4.17) in (a) PKF-constant and (b) PKF-linear using dataset 2.

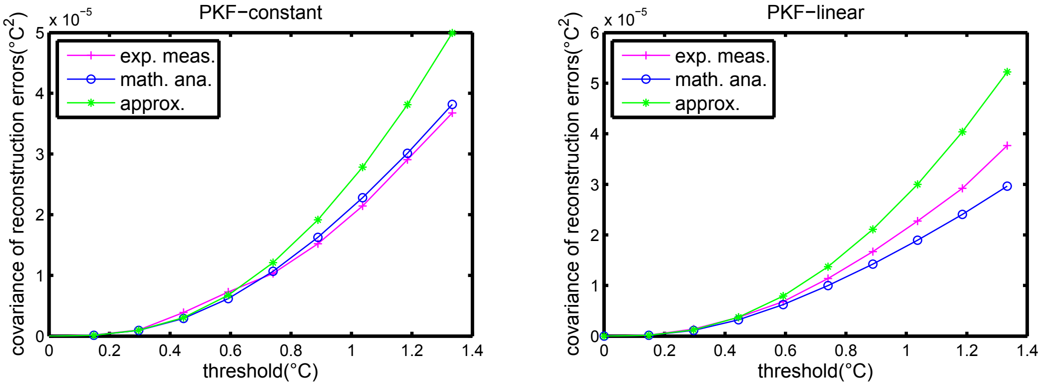


Figure 6.8: Comparison of covariance of reconstruction errors among experimental measurements, mathematical analysis Eq. (4.22), and approximation Eq. (4.40) in (a) PKF-constant and (b) PKF-linear using dataset 2.

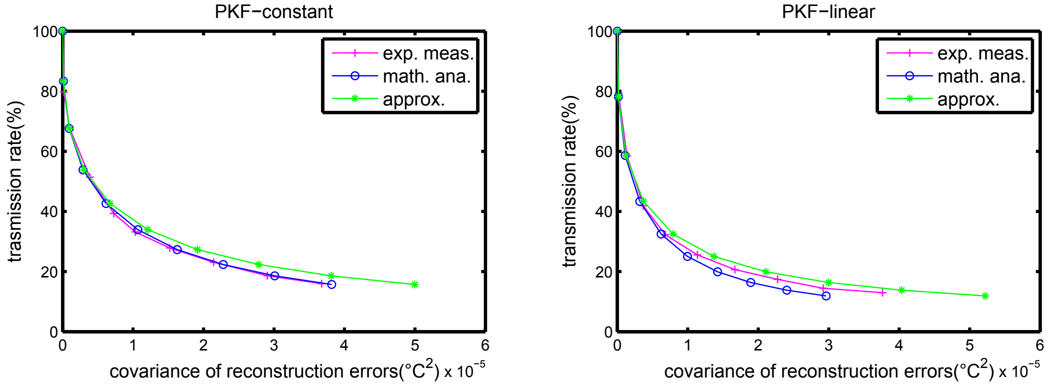


Figure 6.9: Comparison of the trad-off between transmission rate and covariance of reconstruction errors among experimental measurements, mathematical analysis Eqs. (4.17) and (4.22), and approximation Eqs. (4.17) and (4.40) in (a) PKF-constant and (b) PKF-linear using dataset 2.

6.4 Estimation of EPKF Using WSN Datasets

This section estimates the performance of EPKF using the real WSN datasets with different sizes of the cluster. In addition to the comparison among the PKF-based approaches, Rand-ST, EPKF-simp, EPKF-norm, EPKF-mix, we compare it with two popular techniques using both temporal and spatial correlations: EEDC [18] and CS [58]. The former one is a classical clustering approach that selects active nodes as representatives in a period after using temporal correlation with PlAMLiS method; the later one based on the new sampling theory is very popular as introduced in Chapter 2.

The datasets are taken from LUCE WSN testbed [119], which are collected by Shockfish TinyNode at intervals of 30 seconds across the EPFL campus. We use the temperature and humidity values as the data types and group the nodes into different clusters according to the correlation coefficient. Two clusters with different sizes are presented here: the first one consists of 4 nodes with 4×2103 temperature values and the second one has 15 nodes with 15×856 relative humidity values. To initiate PKF-based approaches, the system model for each dataset is found by the Matlab system identification toolbox [116] [117] as usual. The estimations of the global KF generated with all nodes measurements are assumed to be the real state here. For EEDC [18], we assume the buffer size is big enough to store all values, since it uses the PLAMLiS method in the temporal domain. In order to implement CS [58], we exploit the discrete cosine transform (DCT) as the representation basis, Ψ, for these two datasets, which can sparsify the original signals sufficiently. An independent and identically distributed (i.i.d.) Gaussian matrix is used for random projection, Φ, which is incoherent with Ψ. The signals are reconstructed using ℓ_1 minimization method [118].

Since there are too many nodes in the system, we present the average covariance of reconstruction errors and the average transmission rate. The trade-offs between them using each approach with two datasets are depicted in Fig. 6.10a and Fig. 6.11a, respectively. EPKF-norm, EPKF-mix and EPKF-simp nearly have the same performance, which could further reduce the transmission rate of the leaf node by more than 90% under the same reconstruction quality as PKF. Rand-ST could degrade the reconstruction quality of PKF, which is consistent with the results in the artificial system. Compared with other techniques, the advantages of EPKF methods are more significant. Only 5% transmission is required to achieve the same reconstruction quality as CS [58] using the full transmission. The best reconstruction of EEDC, denoted as EEDC-max, appears when only temporal correlation is used, namely using PLAMLiS method. Each node approximates the raw data as the line segment and transmits the end points of each segment inde-

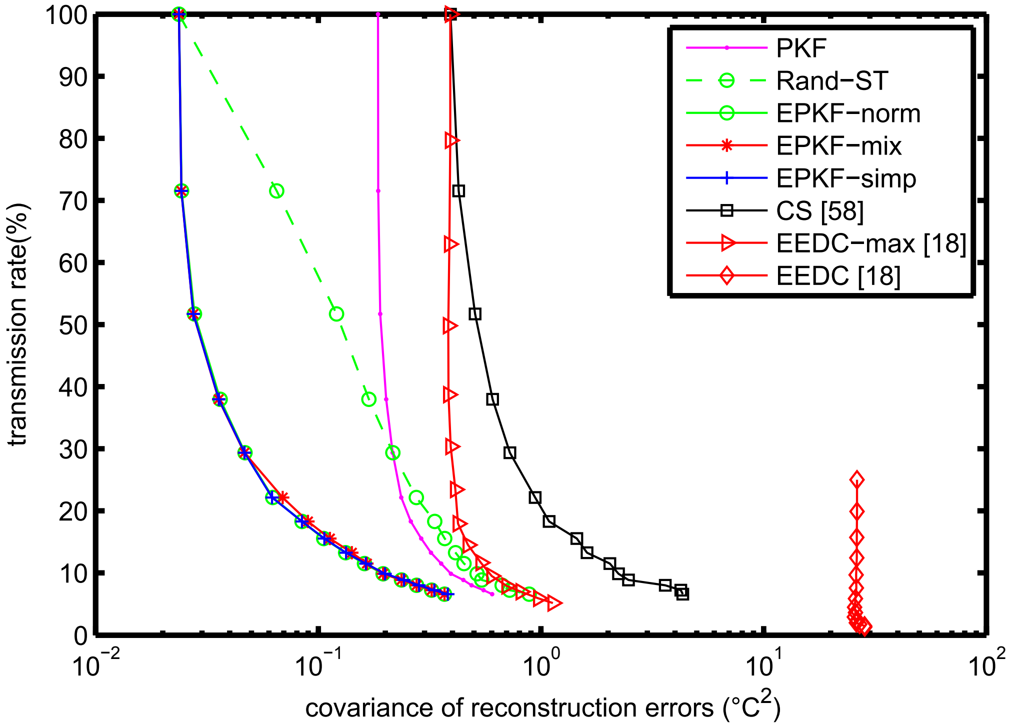


Figure 6.10: Performance comparison of PKF, Rand-ST, EPKF-norm, EPKF-mix, EPKF-simp, CS [58] and EEDC [18] using real dataset 1 with four leaf nodes.

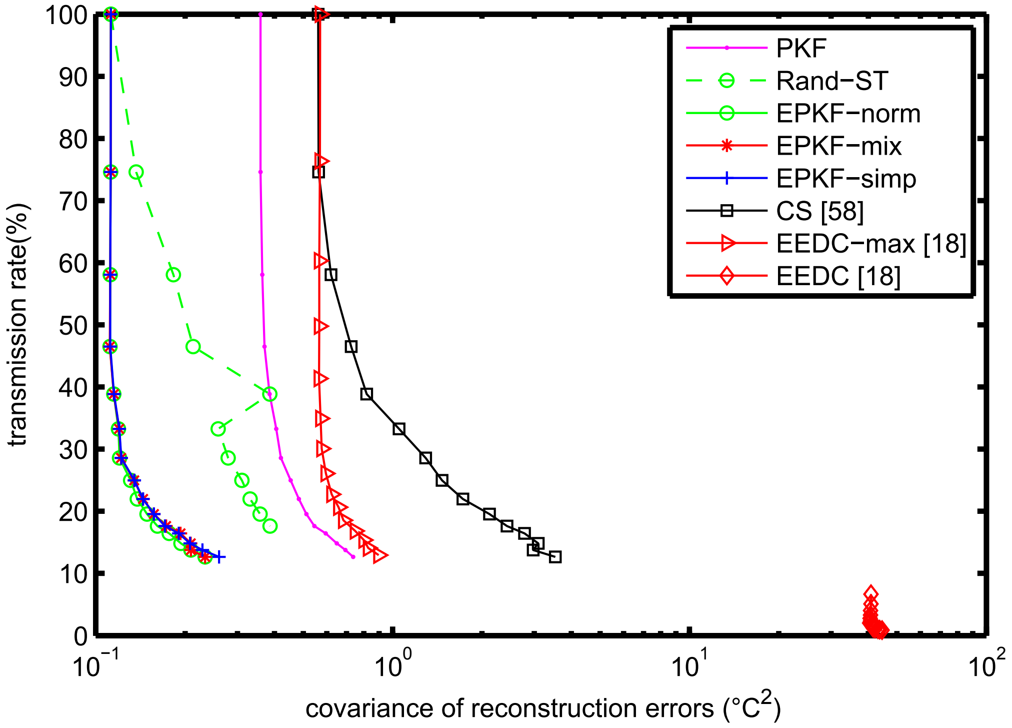


Figure 6.11: Performance comparison of PKF, Rand-ST, EPKF-norm, EPKF-mix, EPKF-simp, CS [58] and EEDC [18] using dataset 2 with 15 nodes.

pendently without using any representative node. Consistent with the results in Fig. 6.1 and Fig. 6.2, the reconstruction quality of EEDC-max is worse than PKF as reported in Fig. 6.10a and Fig. 6.11a under the same transmission rate. When it further uses spatial correlation, where each node is uniformly selected as the representative node, the performance of EEDC degenerates. The reconstruction error of EEDC is even dozens of times more than EPKF. Note that, when the transmission rate is 100%, the advantage of PKF-based approaches reflects the effort of KF. Using the filtered data produces better results than directly using the noisy measurements.

6.5 Physical Implementation with Openmote

This section assesses the proposed algorithms using the hardware implementation. A simple WSN monitoring the indoor temperature is formed with four leaf nodes and 1 master node to measure the transmission rate of the leaf node using PKF and the reconstruction quality improvement by further exploiting EPKF methods in the master node. Then the energy consumption of PKF and communication energy consumption of the leaf node are compared to examine how much energy can be saved. The current profile is visualized on an oscilloscope by measuring the voltage drop over a fixed resistor. Combining the overall per-day current consumption of the node including sensing, computation, communication and OS related activities, and the obtained transmission rate, the lifetime improvements of each node using PKF are estimated.

OpenMote-CC2538 [113] is used as the hardware in our experiments as shown in Fig. 6.12. It is based on the Ti CC2538 System on Chip (SoC) [120], which combines a 32-bit ARM Cortex-M3 with an IEEE 802.15.4 compliant RF transceiver in one chip [120]. It is connected to the OpenBattery board [121] as our leaf node, which is powered by 2 AAA batteries. The mater node uses Openmote-CC2538 with the Openbase board [122], which is connected to PC. The nodes run Contiki OS, which is an open source, highly portable, multi-tasking operating system for memory-efficient networked embedded systems and wireless sensor networks [114, 123]. The RIME communication stack is used, which provides a set of custom lightweight communication primitives designed for low-power wireless networks [124] [125]. The node accesses media under the control of CSMA/CA scheme. To attain low-power operation of the radio, ContikiMAC [126] [127] is used. The node is required to keep the radio off as much as possible and periodically wake up to check for radio activity. The channel check rate (CCR) is given in Hz, specifying the number of channel checks per second, and the default CCR is 8 Hz. CCRs

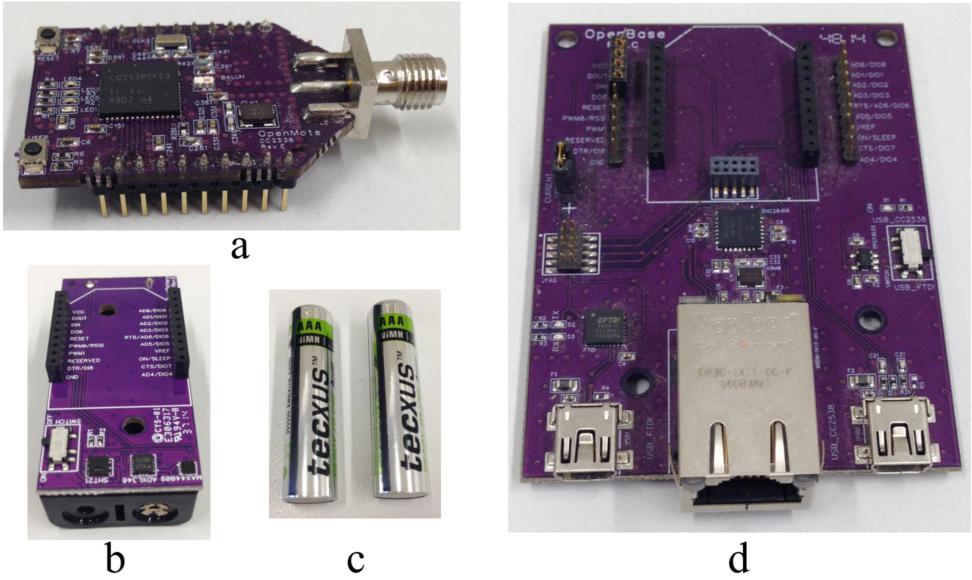


Figure 6.12: (a) Openmote-CC2538; (b) OpenBattery board; (c) AAA batteries; (d) Openbase board.

are given in powers of two, and typical settings are 2, 4, 8, and 16 Hz [128]. If a packet transmission is detected, the receiver stays awake to receive the next packet and sends a link layer acknowledgment (ACK). To send a packet, the sender repeatedly sends the same packet until a link layer ACK is received [126]. The leaf node collects the temperature using SHT21 sensor every six seconds and transmits it to the master node using single-hop unicast scheme.

Table 6.1: The reconstruction quality improvement using EPKF methods in the cluster head w.r.t. PKF for each leaf node.

Methods	Improvement (%)			
	node 1	node 2	node 3	node 4
EPKF-simp	95.41	81.84	98.29	91.09
EPKF-norm	95.41	81.85	98.29	91.10
EPKF-mix	95.41	81.85	98.29	91.10

We randomly setup a wireless sensor network with four leaf nodes and 1 master node in the laboratory to measure the transmission rate of each node running PKF. The positions of each node are measured afterwords and depicted in Fig. 6.13. Each node firstly collects the raw data for three days. We obtain the system model using these data offline with the help of Matlab. Then each node runs PKF with these models online for one week. The threshold of each node is set to 0.01°C. The obtained transmission rates of each node are 0.8%, 0.72%, 0.81% and 0.81%, respectively. Compared with the typical used

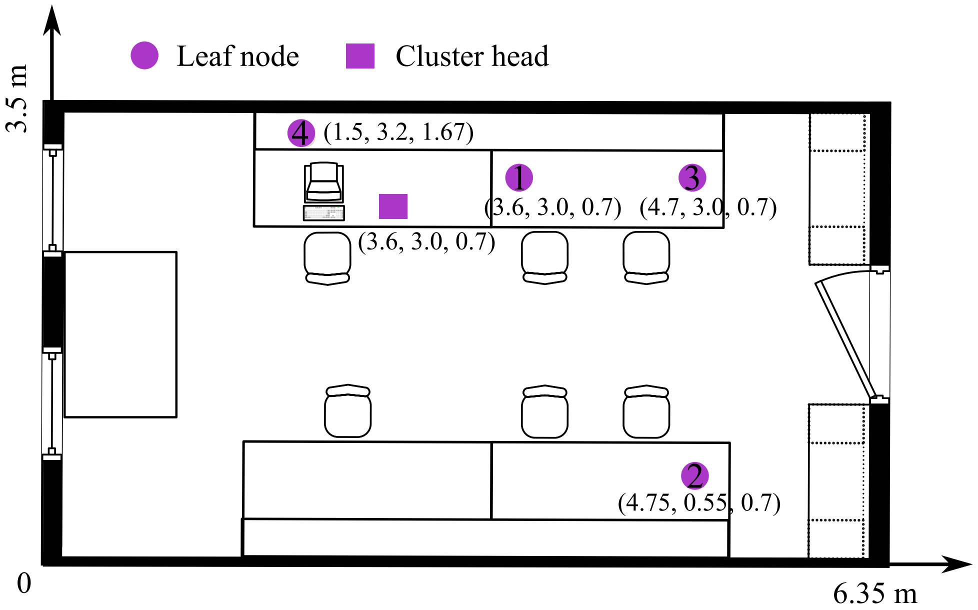


Figure 6.13: The setup of a simple WSN with 4 leaf nodes and 1 master node in the Lab.

threshold 0.5°C in the literatures, the reconstructions using PKF have very small errors. The reconstruction quality of PKF in the leaf node is further improved using EPKF in the cluster head as expected. The improvement for each node is listed in Table 6.1. EPKF-norm and EPKF-mix have the same improvements, which are slightly higher than EPKF-simp.

Now we measure the computation energy consumption of PKF, E_{pkf}, and the communication energy consumption, E_{cmn}, of the leaf node to observe how much energy can be saved. The general idea is to visualize the current profile on an oscilloscope by measuring the voltage drop over a fixed resistor. The measurement setup is shown in Fig. 6.14. Instead using the battery, the leaf node is powered by DC power supply with 3.0 V to obtain more stable power input. The step-down DC-DC converter TPS62730 in the mote regulates the input voltage down to 2.1 V in the regulated mode [129]. A 10 Ω resistor is connected in series with the mote. The master node is powered by PC and communicates with the leaf node through the RF radio. The oscilloscope provides a graphical representation of the voltage drop over the resistor, which is the same as the current consumed by the system because of the Ohm's Law. The detailed current profile of the node during computation and communication when the CCR is 128, 32 and 8 Hz are depicted in the Section 8.3 in Figs. 8.3 to 8.5. The corresponding current consumption and the duration during each process are summarized in Tables 8.1 to 8.3. Here we present the final results.

Because the execution time of PKF is very small, the PKF algorithm is repeated 20

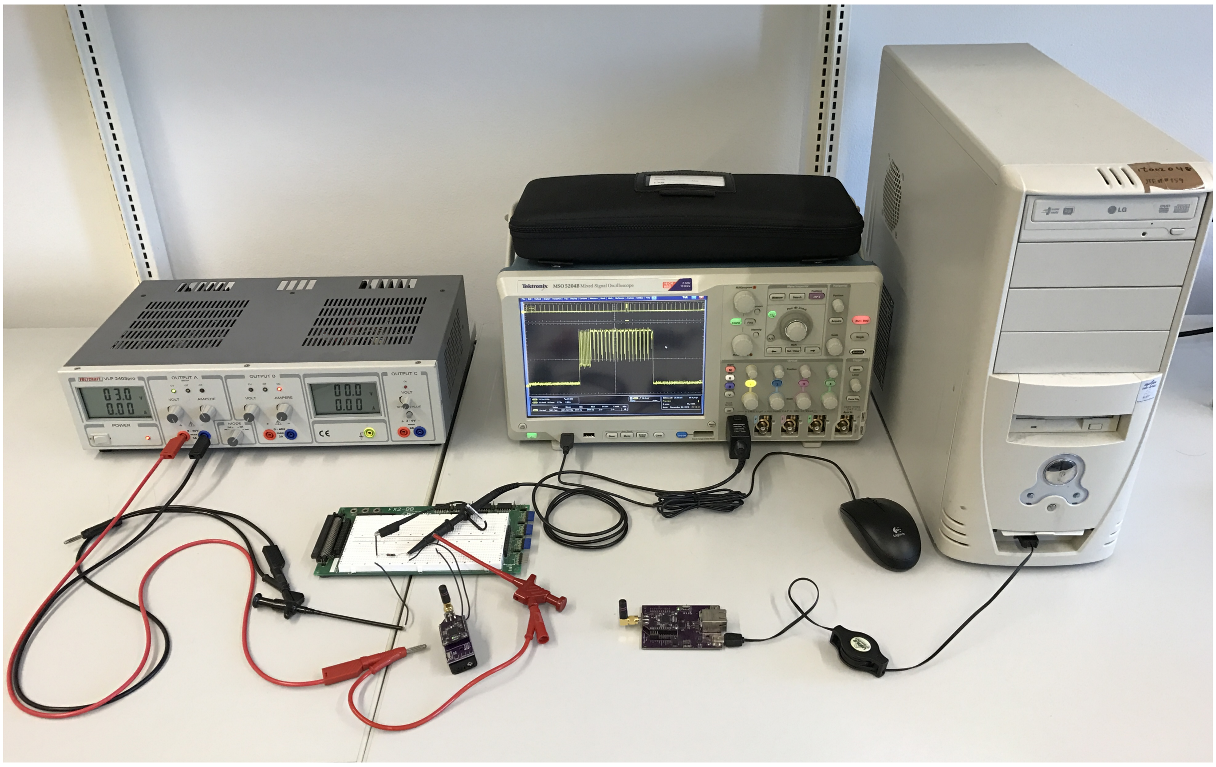


Figure 6.14: Measurement setup for analyzing the power consumption of the leaf node during each process.

times in the node. It costs 4.55 ms in total, which corresponds to 0.2275 ms for each execution of PKF. The voltage is 101.76 mV and corresponds to 10.176 mA of the current in the 10 Ω resistor. Thus the per-time current consumption[2] by executing PKF is $C_{pkf} = 10.176 * 0.2275 = 2.315 mAms$. The corresponding energy consumption of PKF is the product of the electric charge and the regulated voltage (2.1 V), i.e., $E_{pkf} = 2.315 \times 10^{-6} * 2.1 = 4.86 \mu J$.

To measure the energy consumption of the communication, E_{cmn}, the CCR of the leaf node is assigned to 2 Hz to keep the radio sleep as much as possible and the CCR of the master node is set to different values ranging from 2 to 128 Hz to obtain the current profiles of the communication in different scenarios. From the observations, there are eight processes involved in the radio activities: regular channel check, CSMA/CA, switch from RX to TX, transmitting, switch from TX to RX, waiting for ACK, receiving ACK and RF in RX to process ACK. The radio wakes up to firstly detect, if there is an incoming transmission. Two successive clear channel assessments (CCA) are performed for this purpose. Then a collision avoidance mechanism is conducted before data transmission,

[2] Also known as the electric charge with the unit coulomb C and $1C = 1A \cdot 1S$.

Table 6.2: The communication energy consumption of the leaf node, when the CCR of the master node increases from 2 to 128 Hz.

CCR (Hz)	2	4	8	16	32	64	128
E_{cmn} (mJ)	19.27	7.44	4.45	2.02	0.92	0.58	0.58

where several successive CCAs are performed to check the availability of the channel. The average currents consumed by the leaf node during these processes, under different CCRs of the master node, are nearly the same. After that, the node starts to transmit the data packet consisting of 39 Bytes packet headers and 4 Bytes temperature values. Then the RF is switched from transmission to receive the ACK. Because the absence of ACK, the radio is switched to transmission again to retransmit the data packet until it receives the ACK after several times retransmission. The number of retransmissions increases as the CCR of the master node decreases, since it can not promptly detect the communication and respond the leaf node. For example, there are only 2 times retransmissions, when the CCR of the master node is 128 Hz; the number increases to 5 and 38, when CCR decreases to 32 Hz and 8 Hz, respectively.

Table 6.2 summarizes the decrease of the communication energy consumption of the leaf node as the CCR of the master node increases from 2 to 128 Hz. Compared with the computation energy consumption of PKF, the energy cost of communication is hundreds or thousands of times larger than the computation energy as shown in Fig. 6.15a. For example, when the master node uses the default CCR 8 Hz, it is 915.8 times larger than the computation power consumption of PKF. Thus, PKF with very few computation cost can significantly reduce the communication energy consumption of the node.

Then we examine the lifetime improvement using PKF method considering the overall per-day current consumption of each node. Besides the communication and computation current consumption, the node also spends current on sensing and OS related activities. From the observation, it costs 65,09 ms and 11.364 mA to wait for MCU stable and reading sensor from I^{2}C. We call this electric charge C_{sens} =739.65 mAms. The leaf node checks the channel twice per second and each time consumes 14.98 mAms. Thus, it costs 179.76 mAms during each period of 6 s. This cost is named C_{cca}. Then the overall per-day current consumption of a node without using PKF, C_{no}, can be calculated using Eq. (6.1), where N is the number of transmissions. Using the obtained transmission rate T_r, the overall per-day current consumption of the leaf node using PKF, C_{with}, can be calculated

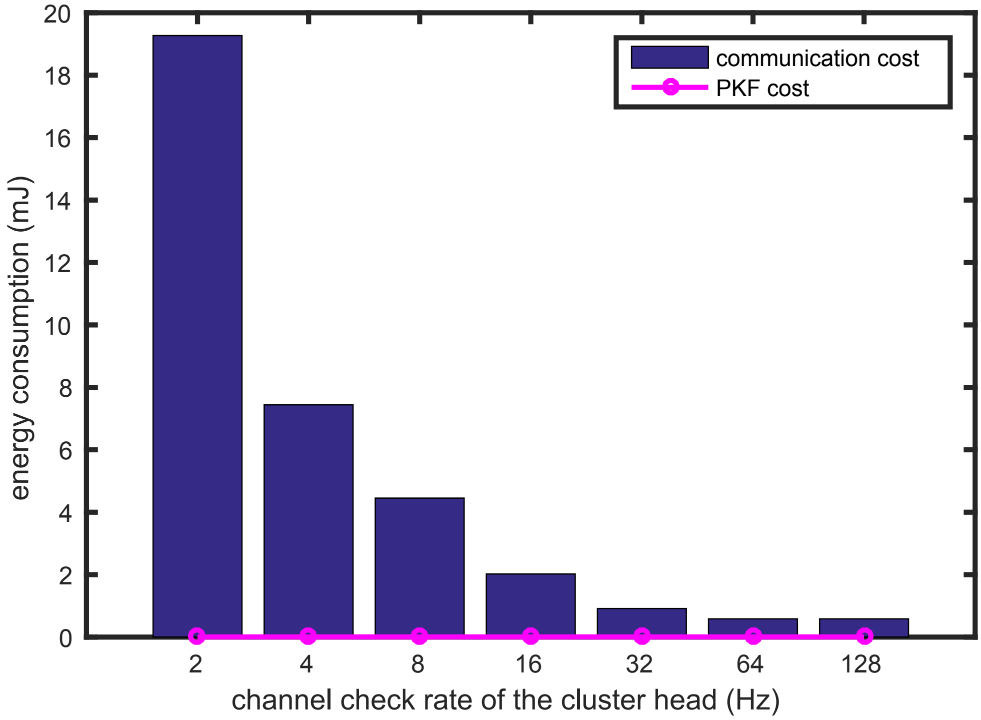


Figure 6.15: The ratio between the communication and computation energy consumption of the leaf node, when the CCR of the master node increases from 2 to 128 Hz.

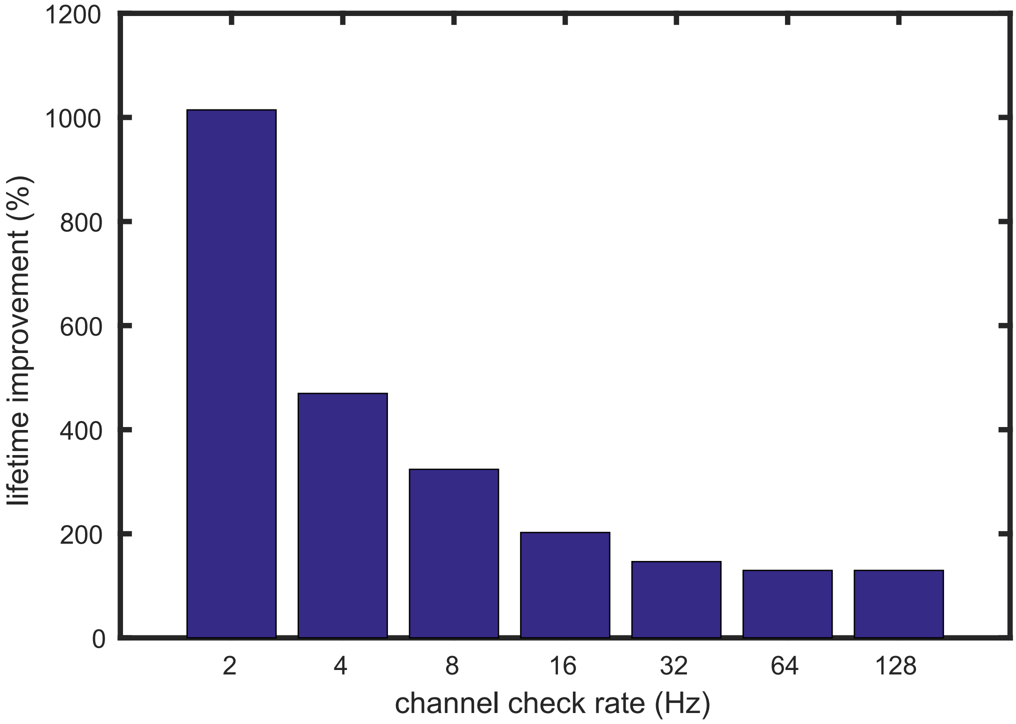


Figure 6.16: The lifetime improvement using PKF w.r.t. no PKF considering the overall electric charge, when the CCR of the master node increases from 2 to 128 Hz.

Table 6.3: The total per-day current consumption of node 1 without and with PKF using the obtained transmission rate 0.8%, when the CCR of the master node increases from 2 to 128 Hz.

CCR (Hz)	2	4	8	16	32	64	128
C_{no} (mAh)	40.38	17.85	12.16	7.53	5.42	4.79	4.79
C_{with} (mAh)	3.98	3.80	3.75	3.72	3.70	3.69	3.69

using Eq. (6.2).

$$C_{no} = (C_{sen} + C_{cmn} + C_{cca}) * N \tag{6.1}$$

$$C_{with} = (C_{sen} + C_{pkf} + C_{cca}) * N + C_{cmn} * N * T_r \tag{6.2}$$

Table 6.3 summarizes the total per-day current consumption of node 1 without and with PKF using the obtained transmission rate 0.8%. C_{no} and C_{with} decrease as the CCR of the master node increases and finally converges due to the fact that the current consumption during sensing and CCA make more contribution to the overall cost of the leaf node. The capacity of the battery, C_{bat}, is 800 mAh in our experiments. Then the corresponding lifetime without and with PKF can be calculated as:

$$T_{no} = C_{bat}/C_{no} \tag{6.3}$$

$$T_{with} = C_{bat}/C_{with} \tag{6.4}$$

The lifetime improvement using PKF compared with no PKF is calculated by T_{with}/T_{no}. The comparison is shown in Fig. 6.16a. The improvement increases as the CCR of the master node decreases, since the communication cost takes a greater proportion of the overall cost and PKF can efficiently reduce this cost. When the CCR is 128 Hz, there are no very big improvement, where the lifetime is extended from 167.01 days to 216.8 days. While when the CCR of the master node decreases to default value 8 Hz, the lifetime is extended from 65.79 days to 213.33 days with 323.80% improvement. In the best case that CCR=2 Hz, the lifetime of the leaf node can be extended by 10.14 times.

6.6 Summary

This chapter evaluates the performance of PKF and EPKF using the real WSN datasets and the hardware implementation in Openmotes. We firstly compare PKF with other techniques using two kinds of real temperature datasets taken from typical WSN scenarios with a single node. Two techniques with KF for transmission rate reduction are selected, namely, DKF [56] and SIP [55]. Another three selected methods are PAQ [28], PLAMLiS [18] and CS [58], which are the popular techniques without KF to achieve transmission rate compression. Compared with DKF [56] and SIP [55], PKF requires the fewest transmission rate under the same covariance of reconstruction errors. Without data degradation, PKF-linear only needs 11.11% and 11.91% transmission for dataset 1 and dataset 2, which in turn saves the transmission of DKF by 23.11% and 13.93%, respectively. SIP requires around 37.5% and 29.25% transmission to achieve the same reconstruction quality as the raw data, which are around 3 times as much as PKF. Compared with the techniques without KF, the advantages of PKF are even more significant. Without quality degradation, PKF-linear decreases the transmission rate of PAQ [28] by 37.98% and 29.08% for dataset 1 and dataset 2, respectively. To achieve the same quality as the raw measurements, PLAMLiS [18] and CS [58] require 2.14 and 6.60 times as much transmission as PKF for dataset 1; and 2.06 and 6.56 times for dataset 2, respectively. In addition, we present one example to illustrate how to use PKF in time variant systems. The model variations are stored in both leaf node and the cluster head. They update the model in time. Compared with using the time invariant model, the reconstruction quality of PKF is improved. The mathematical analyzed results of PKF follow the experimental measurements. The small deviations between them are reasonable and acceptable, because the system model is uncertain and the noise distribution may not perfectly satisfy Gaussian distribution.

To estimate the performance of EPKF methods, two WSN datasets are used with different sizes of the cluster. Besides the comparison among the PKF-based approaches, Rand-ST, EPKF-norm, EPKF-mix and EPKF-simp, another two popular techniques using both temporal and spatial correlations are compared, namely EEDC [18] and CS[58]. EEDC [18] uses PLAMLiS method in the temporal domain and selects active nodes as representatives in the spatial domain. CS [58] is the popular method using the new sampling theory in both time and spatial domain. Since there are too many nodes in the system, we present the average covariance of reconstruction errors and the average transmission rate. EPKF-norm, EPKF-mix and EPKF-simp nearly have the same performance, which could further reduce the transmission rate of the leaf node by more than 90% under the same reconstruction quality as PKF. Rand-ST could degrade the reconstruction quality of PKF, which is consistent with the results in the artificial system. Compared with other techniques, the advantages of EPKF methods are more significant. Only 5% transmission is required to achieve the same reconstruction quality as CS [58] using the full transmission. The best reconstruction of EEDC, denoted as EEDC-max, appears when only temporal correlation is used, namely using PLAMLiS method. Each node approximates the raw data as the line segment and transmits the end points of each segment independently without using any representative node. The reconstruction quality of EEDC-max is worse than PKF under the same transmission rate. When it further uses spatial correlation, where each node is uniformly selected as the representative node, the performance of EEDC degenerates. The reconstruction error of EEDC is even dozens of times more than EPKF.

To measure the energy consumption and lifetime improvement using our proposed approaches, the algorithms are implemented in the WSN motes, Openmote [113], running on Contiki OS [114]. An arbitrary network is formed with four leaf nodes and one master node. Each node firstly collects the raw data for three days. We obtain the system model using these data offline with the help of Matlab. Then each node runs PKF with the model online for one week. The threshold of each node is set to $0.01°\mathrm{C}^2$. The obtained transmission rates of each node are 0.8%, 0.72%, 0.81% and 0.81%, respectively. By using EPKF methods in the cluster head, the reconstruction quality of PKF is further improved by at least 81.84%. Then the energy consumption of PKF and communication energy consumption of the leaf node are compared. The method is to visualize the current profile on an oscilloscope by measuring the voltage drop over a fixed resistor. Compared with the computation power consumption of PKF, 2.315 mAms, the energy consumption of communication is hundreds or thousands of times more than the computation energy cost. For example, when the master node uses the default CCR 8 Hz, it is 915.8 times as big as the computation energy consumption of PKF. Thus, PKF with very few computation energy consumption can significantly reduce the communication cost of the node. At last, the lifetime improvement using PKF is studied. The total per-day current consumption of the leaf node with and without PKF are calculated using the obtained transmission rate. The smaller the CCR of the master node is, the higher lifetime improvement can be achieved using PKF, since the communication cost takes a greater proportion of the overall energy cost. In the default case that the CCR of the master node is 8 Hz, the lifetime can be extended by 323.80%.

7 Conclusion and Outlook

The goal of a WSN is to monitor the physical system using the sensor nodes. Higher sampling rate may provide better characterization of the system, while consuming more power of the node. This dissertation has proposed the PKF approach to suppress the transmission between the leaf node and the cluster head, while reconstructing the system in the best way using the compressed information for a single node. It has been used for leakage detection in pipelines [130] and thermal monitoring in photonic network-on-chip [23]. Based on the thorough analysis of the approach, it is further extended to exploit spatial correlation, when there are multi-nodes monitoring the system. The reduction of communication energy cost and lifetime improvement by using the proposed approaches are measured using the real hardware implementation. This chapter summarizes our contribution to the state of the art in Section 7.1 and presents the future research directions in Section 7.2.

7.1 Contribution to the State of the Art and Restrictions of the Proposed Approaches

PKF aims to reconstruct the internal state of the system, instead of providing the approximations of the noisy raw data in most of the existing methods. It allows multiple sensor types to be encoded in a single state vector and the reconstructed signal based on the compressed transmission to be even more precise than transmitting all of the raw measurements without processing. Compared with the techniques using KF for data compression, PKF using a KF-predictor provides the optimal reconstruction solution. It achieves data filtering, state estimation, data compression and reconstruction within one KF framework.

The detailed mathematical analysis provides the solid theoretical support for the proposed approach, which is absent in many techniques. It also supplies a common framework to analyze the underlying process of prediction-based schemes. The obtained formulas describe how the system parameters affect the trade-off between energy consumption and reconstruction quality and could be used to estimate the gain by using PKF before the physical implementation. The transmission of the leaf node using PKF not only determines the current optimal estimate of the system state, but also indicates the range and the transmission probability of the k-step ahead prediction of the cluster head.

The extension of PKF exploits spatial correlation without any intra-communication or a coordinator. Each leaf node executes a PKF independently. The reconstruction quality is further improved by the cluster head using the received information, which is equivalent to further reducing the transmission rate of the node under the guaranteed reconstruction quality. Compared with the available techniques, EPKF methods not only ensure an error bound of the reconstruction for each node, but also allow them to report the emergency event in time, which avoids the loss of penitential important information.

The limitations of the proposed approaches are the assumptions that the system is linear and the model is known in advance by the leaf node and the cluster head. How to overcome these problems brings a new topic for the future work.

7.2 Outlook

The proposed approaches compress the transmission rate and reconstruct the system state given a linear state-space model. Several directions for the future work can be foreseen as presented in the following.

WSNs, as the bridge between the real world and the internet, play an important role in Internet of things (IoT). With the expected deployment of trillions of wireless sensor nodes and the expected millions of terabytes of traffic generated annually [131], IoT is emerging as the next technology megatrend. It brings however, an unprecedented technical challenge: the vast amount of information and the associated energy consumption produced by the IoT infrastructure. This data-overflow problem needs to be addressed at all abstraction levels from cloud infrastructure to motes. Instead of processing and analyzing the information mainly in the cloud, smart nodes should be created which can understand data and process it into useful information. This will not only reduce the amount of collected and transmitted data, but also the related energy consumption. To achieve this, motes in IoT should become more **intelligent** and **autonomous**.

I. **Model Learning and Update**
   In the PKF-based schemes, each node requires to know the state space model, whose accuracy affects the energy consumption of the node. However, the environment and even the requirements of the sensor nodes may change over time, it requires a model

learning and update phase. Considering the energy constraint of the sensor nodes, the algorithm should be light enough. For example, when the node realizes that the radio is turned on more often, it can store enough samples and identify the new system parameters with the fixed model structure as done in [29]. Alternatively, a new model structure may need to be provided by the machine learning techniques with low computation effort.

II. **Autonomous Sensing**
Although the original idea of PKF is to suppress the transmission of a leaf node at a time step, it could be extended to further decrease the sensing rate and optimize the network protocols based on its solid mathematical foundation. For example, the sensing units are able to predict the error of k-step ahead based on the analysis of the reconstruction error. They do not need to sense the data if the error is tolerable and only wake up after the predicted maximum step. Moreover, we have extended PKF to exploit spatial correlation. Nodes with fewer samples can be compensated by using neighbors' information. Combining with PKF and EPKF methods, the sensing unit of the node can decide autonomously the optimal sampling rate and reduce the energy consumption of sensing, processing and transmission simultaneously.

III. **Autonomous Communication**
The existing synchronous and asynchronous network protocols [132] waste significant energy to ensure that the receiver of the cluster is turned on when a leaf node wants to transmit data. This can also be concluded from our experimental results. The channel check rate significantly affects the power consumption of the node. With our approach, the cluster head is able to estimate the transmission probability of each node. This grants PKF the ability to cooperate with the MAC protocols to reduce the idle time of the radio. It is expected that the sender-initiated MAC protocols are more efficient in these acyclic transmission cases by slightly configuring the related parameters. Besides, the nodes with higher error bounds are expected to have more residual energy since they transmit less. They should have higher probability to become the cluster head in the next round to balance the network energy consumption.

IV. **Physical World Applications**
After finishing the above three phases, the sensor node has the ability to understand and interpret the monitored systems. Then physical implementations in the real WSN applications are needed to estimate the performance.

Here are the general ideas for the future research directions and I will investigate deeply in my Postdoc project supported by the Central Research Development Fund of University of Bremen.

8 Appendix

8.1 Calculation Complexity Reduction of F_i^+

This section reduces the calculation complexity of $F_i^+(\tau)$, when $1 < i < k$.

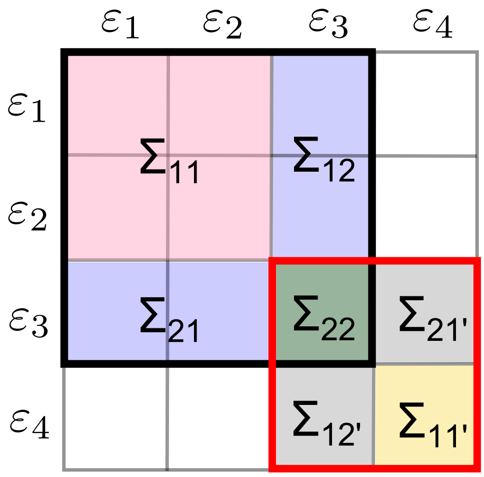


Figure 8.1: One example of two independent MVN distributions of $\boldsymbol{v\varepsilon_2} = [\varepsilon_1, \varepsilon_2]^T$ and ε_4 when $i = 3$ and $k = 4$.

When $1 < i < k$, the distribution of the first $i - 1$ variables in $\boldsymbol{v\varepsilon_k}$, $\boldsymbol{v\varepsilon_{i-1}} = [\varepsilon_1, \cdots, \varepsilon_{i-1}]^T$, and the last $k - i$ variables, $\boldsymbol{v\varepsilon_{k-i}} = [\varepsilon_{i+1}, \cdots, \varepsilon_k]^T$ conditional on $\varepsilon_i = \tau$ are independent. They have independent MVN distributions. One example with $k = 4$ and $i = 3$ is depicted in Fig. 8.1. The distribution of $\boldsymbol{v\varepsilon_2} = [\varepsilon_1, \varepsilon_2]^T$ conditional on $\varepsilon_3 = \tau$ is independent of the distribution of ε_4 conditional on $\varepsilon_3 = \tau$. The mean $\boldsymbol{\tilde{\mu}_1}$ and the covariance $\boldsymbol{\tilde{\Sigma}_1}$ of $\boldsymbol{v\varepsilon_{i-1}}$ can be calculated using the first $i \times i$ elements in the covariance matrix, $\boldsymbol{S_k}(1 : i, 1 : i)$; the mean $\boldsymbol{\tilde{\mu}_2}$ and the covariance $\boldsymbol{\tilde{\Sigma}_2}$ of $\boldsymbol{v\varepsilon_{k-i}}$ can be calculated using the last $(k - i + 1) \times (k - i + 1)$ elements in the covariance matrix , $\boldsymbol{S_k}(i : k, i : k)$. The rearranged matrices of $\boldsymbol{S_k}(1 : i, 1 : i)$ and $\boldsymbol{S_k}(i : k, i : k)$ (see Fig. 8.1) are

$$\boldsymbol{\Sigma_i} = \begin{bmatrix} \boldsymbol{\Sigma_{11}} & \boldsymbol{\Sigma_{12}} \\ \boldsymbol{\Sigma_{21}} & \Sigma_{22} \end{bmatrix} \quad \boldsymbol{\Sigma_{k-i+1}} = \begin{bmatrix} \boldsymbol{\Sigma_{11'}} & \boldsymbol{\Sigma_{12'}} \\ \boldsymbol{\Sigma_{21'}} & \Sigma_{22} \end{bmatrix}$$

Then according to Eq. (3.14) we can obtain that

$$\begin{aligned} \tilde{\boldsymbol{\mu}}_{\mathbf{1}} &= \boldsymbol{\Sigma}_{\mathbf{12}}\Sigma_{22}^{-1}\tau & \tilde{\boldsymbol{\Sigma}}_{\mathbf{1}} &= \boldsymbol{\Sigma}_{\mathbf{11}} - \boldsymbol{\Sigma}_{\mathbf{12}}\Sigma_{22}^{-1}\boldsymbol{\Sigma}_{\mathbf{21}} \\ \tilde{\boldsymbol{\mu}}_{\mathbf{2}} &= \boldsymbol{\Sigma}_{\mathbf{12'}}\Sigma_{22}^{-1}\tau & \tilde{\boldsymbol{\Sigma}}_{\mathbf{2}} &= \boldsymbol{\Sigma}_{\mathbf{11'}} - \boldsymbol{\Sigma}_{\mathbf{12'}}\Sigma_{22}^{-1}\boldsymbol{\Sigma}_{\mathbf{21'}} \end{aligned}$$

In this case, the calculation of $F_i^+(\tau)$ can be reduced from a $(k-1)$ dimensional integral to the multiplication of one $(i-1)$ and one $(k-i)$ dimensional integral, namely,

$$F_i^+(\tau) = \Phi_{\boldsymbol{v\varepsilon_{i-1}}}(\tau)\Phi_{\boldsymbol{v\varepsilon_{k-i}}}(\tau)\varphi_{\varepsilon_i}(\tau) \tag{8.1}$$

■

8.2 Remove the Colored Noise of the KF Estimate

This section aims to decolor the noise of the estimates $\hat{x}_k$ from the local KF using [89]. Combining Eqs. (3.4), (3.5), (3.9) and (3.12), the estimates of the KF is:

$$\hat{x}_k = (I - K_kH_k)A_{k-1}\hat{x}_{k-1} + K_kH_kA_{k-1}x_{k-1} + K_kH_kw_{k-1} + K_kv_k + B_{k-1}u_{k-1} \tag{8.2}$$

The estimate error is:

$$\begin{aligned} \Delta x_k &= \hat{x}_k - x_k \\ &= (I - K_kH_k)A_{k-1}\Delta x_{k-1} + (K_kH_k - I)w_{k-1} + K_kv_k \end{aligned} \tag{8.3}$$

We create an auxiliary signal y_k to remove the correlation, such that

$$\begin{aligned} y_k &= \hat{x}_{k+1} - \psi_k\hat{x}_k \\ &= (x_{k+1} + \Delta x_{k+1}) - \psi_k(x_k + \Delta x_k) \\ &= K_{k+1}H_{k+1}A_kx_k + B_ku_k + K_{k+1}H_{k+1}w_k + K_{k+1}v_{k+1} \\ &= H_k^*x_k + B_ku_k + v_k^* \end{aligned}$$

where $\psi_k = (I - K_{k+1}H_{k+1})A_k$, $H_k^* = K_{k+1}H_{k+1}A_k$ and $v_k^* = K_{k+1}H_{k+1}w_k + K_{k+1}v_{k+1}$. The new but equivalent system can therefore be written as:

$$\begin{aligned} x_k &= A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + w_{k-1} \\ y_k &= H_k^*x_k + B_ku_k + v_k^* \end{aligned}$$

$$R_k^* = E[v_k^* v_k^{*T}] = K_{k+1} H_{k+1} Q_k H_{k+1}^T K_{k+1}^T + K_{k+1} R_{k+1} K_{k+1}^T$$
$$M_k = E[w_k v_k^{*T}] = E\big[w_k (K_{k+1} H_{k+1} w_k + K_{k+1} v_{k+1})'\big] = Q_k H_{k+1}^T K_{k+1}^T$$

■

8.3 Current Profile of Computation and Communication

This section presents the detailed current profile of the leaf node during computation and communication processes.

Figure 8.2 depicts the current profile of the leaf node executing 20 times PKF with total 4.55 ms duration and 101.76 mV voltage drop over the 10 Ω resistor. Thus the per-time current consumption by executing PKF is $C_{pkf} = 10.176 * 0.2275 = 2.315 mAms = 2.315 \times 10^{-6} C$, where C is the unit of electric charge coulomb.

Figures 8.3 to 8.5 record the current consumption of the leaf node during communication, when the CCR of the master node is 128, 32 and 8 Hz, respectively. There are eight processes involved in the radio activities: regular channel check, CSMA/CA, switch from RX to TX, transmitting, switch from TX to RX, waiting for ACK, receiving ACK and RF in RX to process ACK. The corresponding current consumption and the duration during each process are summarized in Tables 8.1 to 8.3.

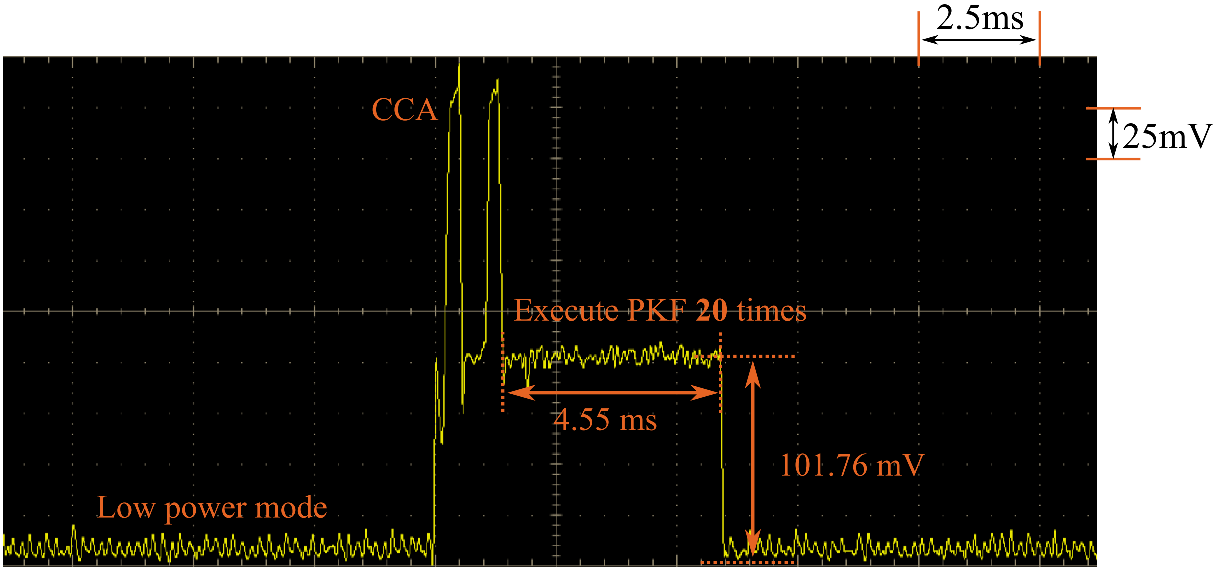


Figure 8.2: Current profile of the leaf node running PKF 20 times.

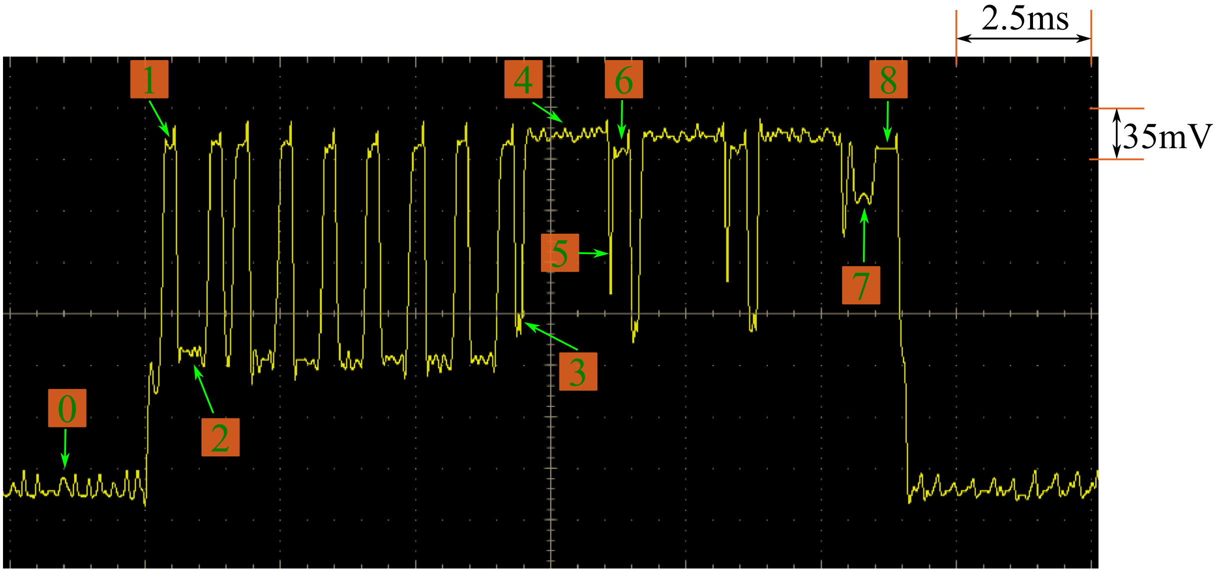


Figure 8.3: The current profile of the leaf node during communication, when the channel check rate of the master node is 128 Hz.

Table 8.1: The current consumption and the duration spent on each process during communication, when the channel check rate of the master node is 128 Hz.

Process	Unit Opertation Description	Voltage (mV)	Current (mA)	Time (ms)	# of Units	Cost (mA*ms)
0	Low power mode		0.002			
1	CCA	241.55	24.155	0.31	9	67.39
2	CCA interval	98.62	9.862	0.50	8	39.45
3	Switch from RX to TX	118.23	11.823	0.20	3	7.09
4	Transmit data packet	250.31	25.031	1.61	3	120.90
5	Switch from TX to RX	154.16	15.416	0.10	3	4.63
6	Wait to receive ACK	240.74	24.074	0.35	2	16.85
7	Receive ACK	208.07	20.807	0.45	1	9.37
8	RF in RX and process ACK	240.74	24.074	0.51	1	12.28
						277.96

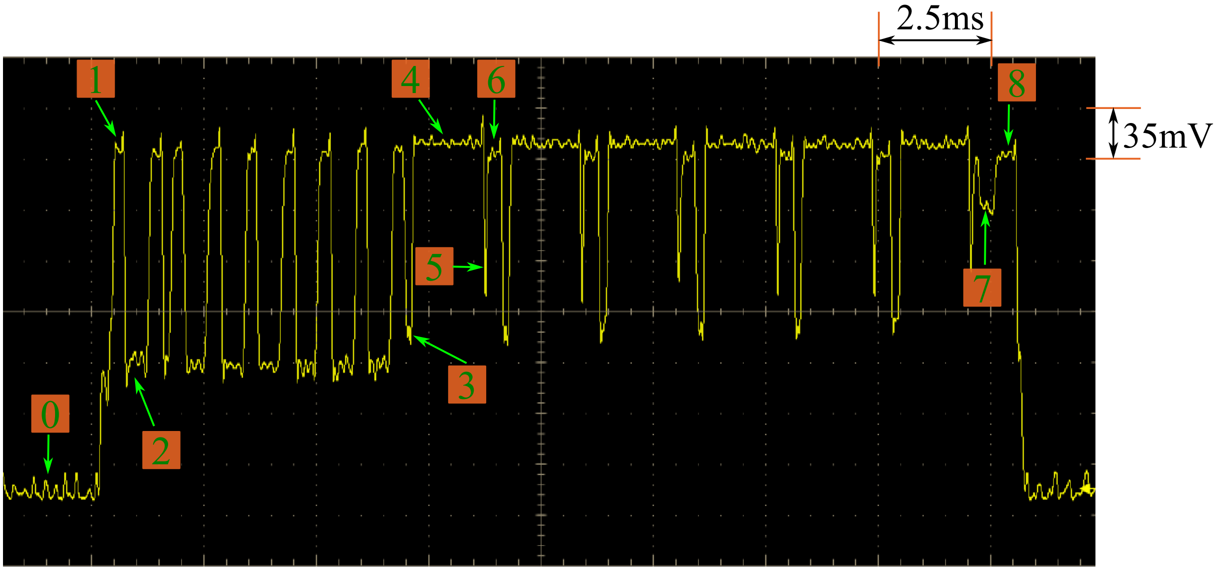


Figure 8.4: The current profile of the leaf node during communication, when the channel check rate of the master node is 32 Hz.

Table 8.2: The current consumption and the duration spent on each process during communication, when the channel check rate of the master node is 32 Hz.

Process	Unit Opertation Description	Voltage (mV)	Current (mA)	Time (ms)	# of Units	Cost (mA*ms)
0	Low power mode		0.002			
1	CCA	241.55	24.155	0.31	9	67.39
2	CCA interval	98.62	9.862	0.50	8	39.45
3	Switch from RX to TX	118.23	11.823	0.20	6	14.19
4	Transmit data packet	250.31	25.031	1.61	6	241.80
5	Switch from TX to RX	154.16	15.416	0.10	6	9.25
6	Wait to receive ACK	240.74	24.074	0.35	5	42.13
7	Receive ACK	208.07	20.807	0.45	1	9.37
8	RF in RX and process ACK	240.74	24.074	0.51	1	12.28
						435.86

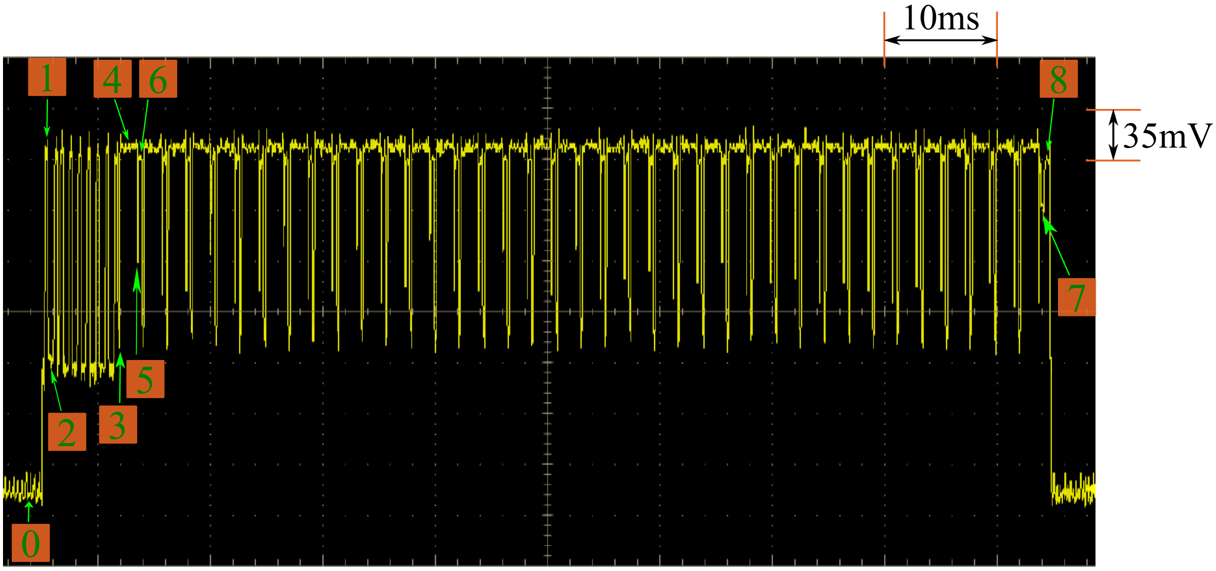


Figure 8.5: The current profile of the leaf node during communication, when the channel check rate of the master node is 8 Hz.

Table 8.3: The current consumption and the duration spent on each process during communication, when the channel check rate of the master node is 8 Hz.

Process	Unit Opertation Description	Voltage (mV)	Current (mA)	Time (ms)	# of Units	Cost (mA*ms)
0	Low power mode		0.002			
1	CCA	241.55	24.155	0.31	9	67.39
2	CCA interval	98.62	9.862	0.50	8	39.45
3	Switch from RX to TX	118.23	11.823	0.20	38	89.85
4	Transmit data packet	250.31	25.031	1.61	38	1531.40
5	Switch from TX to RX	154.16	15.416	0.10	38	58.58
6	Wait to receive ACK	240.74	24.074	0.35	37	311.76
7	Receive ACK	208.07	20.807	0.45	1	9.37
8	RF in RX and process ACK	240.74	24.074	0.51	1	12.28
						2120.08

List of Notations and Acronyms

Notation

x_k	the system state at time instant k.
z_k	the observation at time instant k.
w_k	the system noise at time instant k.
v_k	the observation noise at time k.
R_k	the covariance of the observation noise.
$\hat{x}_k^-$	the *a priori* estimate of the KF.
P_k^-	the covariance of the *a priori* estimate.
K_k	the Kalman gain.
$\hat{x}_k$	the *a posteriori* estimate of the KF.
P_k	the covariance of the *a posteriori* estimate.
$\boldsymbol{Z_k}$	the sequence of the measurements till time k.
$\boldsymbol{U_k}$	the sequence of the inputs from time 0 to time $k-1$.
H_k	the observation matrix at time k.
$\boldsymbol{\hat{X}_k}$	the sequence of the optimal estimates till time k.
$\boldsymbol{\hat{X}_{k\_s}}$	a subset of $\boldsymbol{\hat{X}_k}$.
$\boldsymbol{\hat{X}^i_{k\_s}}$	the collection of the local estimates of node i till time k.
ϵ_k	prediction error at time instant k.
ε_k	k-step ahead prediction error.
$\hat{e}_k$	*a posteriori* estimate of KF at time instant k.
$\tilde{e}_k$	state prediction error w.r.t. real state at time instant k.
σ_k^2	the covariance of the k-step ahead prediction error.
e_k	prediction error of the state w.r.t. the KF-optimal at time instant k.
$\varphi_{\varepsilon_k}(\chi)$	the probability distribution function of ε_k.
$\boldsymbol{v\varepsilon_k}$	the vector of k steps ahead prediction errors.
$\boldsymbol{S_k}$	the covariance of k steps ahead prediction errors.

$\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})$	the probability distribution function of $\boldsymbol{v\varepsilon_k}$.
$\boldsymbol{\chi}$	a real k-dimensional column vector.
Υ_n	the number of most recent consecutive successes that have been observed at the nth trial in the PKF process.
$\boldsymbol{R_k(\tau)}$	k-dimensional region.
$\Phi_{\boldsymbol{v\varepsilon_k}}(\tau)$	the probability of ε_k located in $\boldsymbol{R_k(\tau)}$.
$H_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})$	the Hessian matrix of $\varphi_{\boldsymbol{v\varepsilon_k}}(\boldsymbol{\chi})$.
$\boldsymbol{v\varepsilon_{k-i}}$	extracting the i-step ahead error from $\boldsymbol{v\varepsilon_k}$.
$\Phi_{\boldsymbol{v\varepsilon_{k-i}}}(\tau)$	the probability of $\boldsymbol{v\varepsilon_{k-i}}$ located in $\boldsymbol{R_k(\tau)}$.
z_k^i	the observation of node i at time k.
H_k^i	the observation matrix of node i at time k.
$\boldsymbol{Z_k^i}$	the collection of the observations of node i till time k.
$\hat{x}_k^i$	the local KF estimate of node i at time k.
$\boldsymbol{\hat{X}_k^i}$	the collection of the local estimates of node i till time k.
$\tilde{X}_k$	the *aposteriori* estimate of the KF for the expanded state vector in the cluster..
$\tilde{X}_k^-$	the *apriori* estimate of the KF for the expanded state vector in the cluster..
Γ_n	the number of most recent consecutive successes that have been observed at the nth trial in the random transmission process.
$\boldsymbol{\bar{X}_k^i}$	the collection of the reconstructions in the cluster head using PKF for node i till time k.
E_{pkf}	Computation energy cost of PKF..
E_{cmn}	Communication energy cost..

Acronym

WSN	wireless sensor network.
MVN	multivariate normal.
WMSN	wireless multimedia sensor network.
MMSE	minimum mean square error.
MAP	maximum *a posteriori* estimation.
PDF	probability density function.
CDF	cumulative density function.
CCR	channel check rate.

ACK	acknowledgment.
CCA	clear channel access.
IoT	Internet of things.

Abbreviation of Proposed Approach

PKF	Optimal reconstruction solution using temporal correlation..
Rand-ST	Reconstruction solution using spatial and temporal correlation under random transmission..
EPKF	Extensions of PKF using spatial correlation including EPKF-simp, EPKF-norm and EPKF-mix..
EPKF-simp	Reconstruction solution using spatial and temporal correlation by always approximating the reconstruction of PKF as the optimal estimates of local KF without noise..
EPKF-norm	Reconstruction solution using spatial and temporal correlation by always approximating the reconstruction of PKF as the optimal estimates of local KF with normal distributed noise..
EPKF-mix	Reconstruction solution using spatial and temporal correlation by approximating the reconstruction of PKF as the optimal estimates of local KF without noise if the reconstruction error is smaller than the *a posteriori* estimation covariance; normal distributed noise, otherwise..
Rand-idp	Reconstruction solution using spatial and temporal correlation under random transmission..
Rand-ST-dec	Reconstruction solution using spatial and temporal correlation under random transmission..

List of Figures

List of Tables

Bibliography

[1] Jun Zheng and Abbas Jamalipour. *Wireless Sensor Networks: A Networking Perspective*. Wiley-IEEE Press, 2009.

[2] Tossaporn Srisooksai, Kamol Keamarungsi, Poonlap Lamsrichan, and Kiyomichi Araki. Practical data compression in wireless sensor networks: A survey. *J. Netw. Comput. Appl.*, 35(1):37–59, January 2012.

[3] P. COY and N. et al. GROSS. 21 ideas for the 21st century. *Business Week Online*, pages 78–167, 1999.

[4] Manuel Delamo, Santiago Felici-Castell, Juan J Pérez-Solano, and Andrew Foster. Designing an open source maintenance-free environmental monitoring application for wireless sensor networks. *Journal of Systems and Software*, 103:238–247, 2015.

[5] Nadeem Javaid, Mohsin Raza Jafri, Zahoor Ali Khan, Nabil Alrajeh, Muhammad Imran, and Athanasios Vasilakos. Chain-based communication in cylindrical underwater wireless sensor networks. *Sensors*, 15(2):3625–3649, 2015.

[6] Z. Sheng, C. Mahapatra, C. Zhu, and V. C. M. Leung. Recent advances in industrial wireless sensor networks toward efficient management in iot. *IEEE Access*, 3:622–637, 2015.

[7] C. Lu, A. Saifullah, B. Li, M. Sha, H. Gonzalez, D. Gunatilaka, C. Wu, L. Nie, and Y. Chen. Real-time wireless sensor-actuator networks for industrial cyber-physical systems. *Proceedings of the IEEE*, 104(5):1013–1024, May 2016.

[8] Debiao He, Neeraj Kumar, Jianhua Chen, Cheng-Chi Lee, Naveen Chilamkurti, and Seng-Soo Yeo. Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks. *Multimedia Systems*, 21(1):49–60, 2015.

[9] Nicolás E Cortez, Jozué Vieira Filho, and Fabricio G Baptista. Design and implementation of wireless sensor networks for impedance-based structural health monitoring using zigbee and global system for mobile communications. *Journal of Intelligent Material Systems and Structures*, 26(10):1207–1218, 2015.

[10] Reiner Jedermann, Mike Nicometo, Ismail Uysal, and Walter Lang. Reducing food losses by intelligent food logistics. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 372(2017), 2014.

[11] Sobhi Mejjaouli and Radu F Babiceanu. Rfid-wireless sensor networks integration: Decision models and optimization of logistics systems operations. *Journal of Manufacturing Systems*, 35:234–245, 2015.

[12] María Gabriela Calle Torres. *Energy consumption in wireless sensor networks using GSP*. PhD thesis, University of Pittsburgh, 2006.

[13] V. Raghunathan, C. Schurgers, Sung Park, and M.B. Srivastava. Energy-aware wireless microsensor networks. *Signal Processing Magazine, IEEE*, 19(2):40–50, Mar 2002.

[14] Yanqiu Huang, Wanli Yu, and Alberto Garcia-ortiz. Accurate energy-aware workload distribution for wireless sensor networks using a detailed communication energy cost model. *Journal of Low Power Electronics*, 10(2):183–193(11), June 2014.

[15] Francesco Marcelloni and Massimo Vecchio. An efficient lossless compression algorithm for tiny nodes of monitoring wireless sensor networks. *Comput. J.*, 52(8):969–987, November 2009.

[16] Christopher M. Sadler and Margaret Martonosi. Data compression algorithms for energy-constrained devices in delay tolerant networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, SenSys '06, pages 265–278, New York, NY, USA, 2006. ACM.

[17] Yao Liang and Wei Peng. Minimizing energy consumptions in wireless sensor networks via two-modal transmission. *SIGCOMM Comput. Commun. Rev.*, 40(1):12–18, January 2010.

[18] Chong Liu, Kui Wu, and Jian Pei. An energy-efficient data collection framework for wireless sensor networks by exploiting spatiotemporal correlation. *Parallel and Distributed Systems, IEEE Transactions on*, 18(7):1010–1023, July 2007.

[19] D. Chu, A Deshpande, J.M. Hellerstein, and Wei Hong. Approximate data collection in sensor networks using probabilistic models. In *Data Engineering, 2006. ICDE '06. Proceedings of the 22nd International Conference on*, pages 48–48, April 2006.

[20] Y. Huang, W. Yu, C. Osewold, and A. Garcia-Ortiz. Analysis of PKF: A communication cost reduction scheme for wireless sensor networks. *IEEE Transactions on Wireless Communications*, 15(2):843–856, Feb 2016.

[21] Yanqiu Huang, Wanli Yu, and Alberto Garcia-Ortiz. PKF-ST: A communication cost reduction scheme using spatial and temporal correlation for wireless sensor networks. In *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*, EWSN '16, pages 47–52, USA, 2016. Junction Publishing.

[22] W. Yu, Y. Huang, and A. Garcia-Ortiz. Modeling optimal dynamic scheduling for energy-aware workload distribution in wireless sensor networks. In *2016 International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 116–118, May 2016.

[23] W. Büter, Y. Huang, D. Gregorek, and A. García-Ortiz. A decentralised, autonomous, and congestion-aware thermal monitoring infrastructure for photonic network-on-chip. In *2015 10th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC)*, pages 1–8, June 2015.

[24] Wanli Yu, Yanqiu Huang, and Alberto García-Ortiz. An altruistic compression-scheduling scheme for cluster-based wireless sensor networks. In *Sensing, Communication, and Networking (SECON), 2015 12th Annual IEEE International Conference on*, pages 73–81, June 2015.

[25] Yanqiu Huang, Wanli Yu, and Alberto Garcia-Ortiz. PKF: A communication cost reduction schema based on kalman filter and data prediction for wireless sensor networks. In *Proceedings of the 26th IEEE Inernational system-on-chip conference*, pages 73–78. CAS, 2013.

[26] Jamal N. Al-karaki and Ahmed E. Kamal. Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, 11:6–28, 2004.

[27] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi. In-network aggregation techniques for wireless sensor networks: a survey. *Wireless Communications, IEEE*, 14(2):70–87, April 2007.

[28] Daniela Tulone and Samuel Madden. PAQ: Time series forecasting for approximate query answering in sensor networks. In *Proceedings of the Third European Conference on Wireless Sensor Networks*, EWSN'06, pages 21–37, Berlin, Heidelberg, 2006.

[29] Daniela Tulone and Samuel Madden. An energy-efficient querying framework in sensor networks for detecting node similarities. In *Proceedings of the 9th ACM International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '06, pages 191–300, New York, NY, USA, 2006.

[30] You-Chiun Wang. Data compression techniques in wireless sensor networks. *Pervasive Computing, New York: Nova Science Publishers, Inc*, 2012.

[31] Giuseppe Anastasi, Marco Conti, Mario Di Francesco, and Andrea Passarella. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Netw.*, 7(3):537–568, May 2009.

[32] Kenneth C. Barr and Krste Asanović. Energy-aware lossless data compression. *ACM Trans. Comput. Syst.*, 24(3):250–291, August 2006.

[33] Francesco Marcelloni and Massimo Vecchio. An efficient lossless compression algorithm for tiny nodes of monitoring wireless sensor networks. *The Computer Journal*, 52(8):969–987, 2009.

[34] Jonathan Gana Kolo, S Anandan Shanmugam, David Wee Gin Lim, Li-Minn Ang, and Kah Phooi Seng. An adaptive lossless data compression scheme for wireless sensor networks. *Journal of Sensors*, (2012), 2012.

[35] M. Vecchio, R. Giaffreda, and F. Marcelloni. Adaptive lossless entropy compressors for tiny iot devices. *IEEE Transactions on Wireless Communications*, 13(2):1088–1100, February 2014.

[36] David Slepian and Jack Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on information Theory*, 19(4):471–480, 1973.

[37] J. Chou, D. Petrovic, and Kannan Ramachandran. A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, volume 2, pages 1054–1062 vol.2, March 2003.

[38] J. Zheng, P. Wang, and C. Li. Distributed data aggregation using slepian-wolf coding in cluster-based wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 59(5):2564–2574, Jun 2010.

[39] V.K. Goyal. Theoretical foundations of transform coding. *Signal Processing Magazine, IEEE*, 18(5):9–21, Sep 2001.

[40] A. Ciancio, S. Pattem, A. Ortega, and B. Krishnamachari. Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm. In *2006 5th International Conference on Information Processing in Sensor Networks*, pages 309–316, April 2006.

[41] P. Chithra and P. Thangavel. A fast and efficient memory image codec (encoding/decoding) based on all level curvelet transform co-efficients with spiht and run length encoding. In *Recent Advances in Space Technology Services and Climate Change 2010 (RSTS CC-2010)*, pages 174–178, Nov 2010.

[42] D. M. Pham and S. M. Aziz. An energy efficient image compression scheme for wireless sensor networks. In *2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 260–264, April 2013.

[43] G. A. Ruiz, J. A. Michell, and A. M. Buron. Parallel pipeline 8 times 8 forward 2-d ict processor chip for image coding. *IEEE Transactions on Signal Processing*, 53(2):714–723, Feb 2005.

[44] Mohamed Abdelaal and Oliver Theel. An efficient and adaptive data compression technique for energy conservation in wireless sensor networks. In *Wireless Sensor (ICWISE), 2013 IEEE Conference on*, pages 124–129, 2013.

[45] S. Pudlewski, A. Prasanna, and T. Melodia. Compressed-sensing-enabled video streaming for wireless multimedia sensor networks. *IEEE Transactions on Mobile Computing*, 11(6):1060–1072, June 2012.

[46] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst. Compressed sensing for real-time energy-efficient ecg compression on wireless body sensor nodes. *IEEE Transactions on Biomedical Engineering*, 58(9):2456–2466, Sept 2011.

[47] Z. Zhang, T. P. Jung, S. Makeig, and B. D. Rao. Compressed sensing of eeg for wireless telemonitoring with low energy consumption and inexpensive hardware. *IEEE Transactions on Biomedical Engineering*, 60(1):221–224, Jan 2013.

[48] Yann-Aël Le Borgne, Silvia Santini, and Gianluca Bontempi. Adaptive model selection for time series prediction in wireless sensor networks. *Signal Process.*, 87(12):3010–3020, December 2007.

[49] U. Raza, A. Camerra, A. L. Murphy, T. Palpanas, and G. P. Picco. Practical data prediction for real-world wireless sensor networks. *IEEE Transactions on Knowledge and Data Engineering*, 27(8):2231–2244, Aug 2015.

[50] Carlos Carvalho, Danielo G. Gomes, Nazim Agoulmine, and José Neuman de Souza. Improving prediction accuracy for wsn data reduction by applying multivariate spatio-temporal correlation. *Sensors*, pages 10010–10037, November 2011.

[51] C. Sarkar, V. S. Rao, R. Venkatesha Prasad, S. N. Das, S. Misra, and A. Vasilakos. VSF: An energy-efficient sensing framework using virtual sensors. *IEEE Sensors Journal*, 16(12):5046–5059, June 2016.

[52] Amol Deshpande, Carlos Guestrin, Samuel R. Madden, Joseph M. Hellerstein, and Wei Hong. Model-driven data acquisition in sensor networks. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, VLDB '04, pages 588–599. VLDB Endowment, 2004.

[53] B. Kanagal and A Deshpande. Online filtering, smoothing and probabilistic modeling of streaming data. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 1160–1169, April 2008.

[54] Derek Rowell. State-space representation of LTI systems. 2002.

[55] Daniel Goldsmith and James Brusey. The spanish inquisition protocol-model based transmission reduction for wireless sensor networks. In *Sensors, 2010 IEEE*, pages 2043–2048, 2010.

[56] Ankur Jain, Edward Y. Chang, and Yuan-Fang Wang. Adaptive stream resource management using kalman filters. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, SIGMOD '04, pages 11–22, New York, NY, USA, 2004.

[57] Guiyi Wei, Yun Ling, Binfeng Guo, Bin Xiao, and Athanasios V. Vasilakos. Prediction-based data aggregation in wireless sensor networks: Combining grey model and kalman filter. *Computer Communications*, 34(6):793 – 802, 2011.

[58] E.J. Candes and M.B. Wakin. An introduction to compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):21–30, March 2008.

[59] C. Caione, D. Brunelli, and L. Benini. Distributed compressive sampling for lifetime optimization in dense wireless sensor networks. *IEEE Transactions on Industrial Informatics*, 8(1):30–40, Feb 2012.

[60] Chong Luo, Feng Wu, Jun Sun, and Chang Wen Chen. Compressive data gathering for large-scale wireless sensor networks. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, MobiCom '09, pages 145–156, New York, NY, USA, 2009.

[61] C. Luo, F. Wu, J. Sun, and C. W. Chen. Efficient measurement generation and pervasive sparsity for compressive data gathering. *IEEE Transactions on Wireless Communications*, 9(12):3728–3738, December 2010.

[62] Baoming Sun, Yan Guo, Ning Li, Laixian Peng, and Dagang Fang. Tdl: Two-dimensional localization for mobile targets using compressive sensing in wireless sensor networks. *Computer Communications*, 78:45 – 55, 2016.

[63] Rajarshi Middya, Nabajit Chakravarty, and Mrinal Kanti Naskar. Compressive sensing in wireless sensor networks: a survey. *IETE Technical Review*, pages 1–13, Oct. 2016.

[64] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8*, HICSS '00, page 8020, Washington, DC, USA, 2000.

[65] Qin Wang, M. Hempstead, and Woodward Yang. A realistic power consumption model for wireless sensor network devices. In *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*, volume 1, pages 286–295, 2006.

[66] M. Goyal, D. Rohm, W. Xie, S.H. Hosseini, K.S. Trivedi, Y. Bashir, and A. Divjak. A stochastic model for beaconless IEEE 802.15.4 MAC operation. *Computer Communications*, 34(12):1460 – 1474, 2011.

[67] Eduardo Casilari, Jose M Cano-García, and Gonzalo Campos-Garrido. Modeling of current consumption in 802.15. 4/ZigBee sensor motes. *Sensors*, 10(6):5443–5468, 2010.

[68] Daniel Goldsmith. *Model-based Transmission Reduction and Virtual Sensing in Wireless Sensor Networks*. PhD thesis, Coventry University, 2013.

[69] Mohammad S. Obaidat and Sudip Misra. *Principles of Wireless Sensor Networks*. Cambridge University Press, 2014.

[70] Steven Dubowsky David Trumper. Modeling dynamics and control: Natural response. 2005.

[71] Zhe Chen. Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, 182(1):1–69, 2003.

[72] V. Vaidehi, S. Vasuhi, K.S. Ganesh, C. Theanammai, N.T. Naresh Babu, N. Uthiravel, P. Balamuralidhar, and G. Chandra. Person tracking using kalman filter in wireless sensor network. In *Advanced Computing (ICoAC), 2010 Second International Conference on*, pages 60–65, Dec 2010.

[73] Meng Shuai, Kunqing Xie, Guanhua Chen, Xiuli Ma, and Guojie Song. A kalman filter based approach for outlier detection in sensor networks. In *Computer Science and Software Engineering, 2008 International Conference on*, volume 4, pages 154–157, Dec 2008.

[74] A. Gasparri and F. Pascucci. An interlaced extended information filter for self-localization in sensor networks. *Mobile Computing, IEEE Transactions on*, 9(10):1491–1504, Oct 2010.

[75] Eduardo F. Nakamura and Antonio A. F. Loureiro. Information fusion in wireless sensor networks. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1365–1372, New York, NY, USA, 2008. ACM.

[76] A. Ribeiro, G.B. Giannakis, and S.I. Roumeliotis. SOI-KF: Distributed kalman filtering with low-cost communications using the sign of innovations. *Signal Processing, IEEE Transactions on*, 54(12):4782–4795, Dec 2006.

[77] Emil Kalman, Rudolph. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D.):35–45, 1960.

[78] Karel J. Keesman. *System Identification.* Springer, 2011.

[79] Derek Rowell. Linear graph modeling: One-port elements. 2003.

[80] Ton van den Boom. Discrete-time systems analysis. 2006.

[81] Jan Willem Polderman and Jan C Willems. Introduction to the mathematical theory of systems and control, 1998.

[82] Allen L Barker, Donald E Brown, and Worthy N Martin. Bayesian estimation and the kalman filter. *Computers & Mathematics with Applications*, 30(10):55–77, 1995.

[83] Steven M Kay. *Fundamentals of Statistical Signal Processing: Practical Algorithm Development*, volume 3. Pearson Education, 2013.

[84] Morris L. Eaton. *Multivariate Statistics: a Vector Space Approach.* John Wiley and Sons, 1983.

[85] John Tsitsiklis David Gamarnik. Fundamentals of probability: continuous random variables. 2008.

[86] Max Welling. The kalman filter. *Lecture Note*, 2010.

[87] Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches.* John Wiley & Sons, 2006.

[88] Guobin Chang. On kalman filter for linear system with colored measurement noise. *Journal of Geodesy*, 88(12):1163–1170, 2014.

[89] A.E. Bryson and L. J. Henrikson. Estimation using sampled data containing sequentially correlated noise. *Journal of Spacecraft and Rockets*, 5(6):662–665, June 1968.

[90] M. G. Petovello, K. O'Keefe, G. Lachapelle, and M. E. Cannon. Consideration of time-correlated errors in a kalman filter applicable to gnss. *Journal of Geodesy*, 83(1):51–56, 2009.

[91] A. Bryson and D. Johansen. Linear filtering for time-varying systems using measurements containing colored noise. *IEEE Transactions on Automatic Control*, 10(1):4–10, Jan 1965.

[92] K. Wang, Y. Li, and C. Rizos. Practical approaches to kalman filtering with time-correlated measurement errors. *IEEE Transactions on Aerospace and Electronic Systems*, 48(2):1669–1681, APRIL 2012.

[93] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, MobiCom '00, pages 56–67, New York, NY, USA, 2000. ACM.

[94] Sunhee Yoon and Cyrus Shahabi. The clustered aggregation (CAG) technique leveraging spatial and temporal correlations in wireless sensor networks. *ACM Trans. Sen. Netw.*, 3(1), March 2007.

[95] J. Michael Steele. *Stochastic Calculus and Financial Applications.* Springer Science & Business Media, 2012.

[96] Corrinne Pellillo Brase Charles Henry Brase. *Understanding Basic Statistics.* Cengage Learning, 2015.

[97] Y. L. Tong. *The Multivariate Normal Distribution.* Spinger, New York, 1990.

[98] J. F. Kenney and E. S. Keeping. *Mathematics of Statistics.* 1954.

[99] Gregory F. Lawler. *Introduction to Stochastic Processes.* CRC Press, 2006.

[100] Scott Miller and Donald Childers. *Probability and Random Processes With Applications to Signal Processing and Communications.* 2004.

[101] James R Norris. *Markov chains.* Number 2. Cambridge university press, 1998.

[102] Richard Serfozo. *Basics of applied stochastic processes.* Springer Science & Business Media, 2009.

[103] Marco Muselli. Simple expressions for success run distributions in bernoulli trials. *Statistics and Probability Letters*, 31(2):121 – 128, 1996.

[104] Konrad Jacobs. *Discrete Stochastics.* 1992.

[105] Daniel Zwillinger and Stephen Kokoska. *CRC Standard Probability and Statistics Tables and Formulae.* CRC Press, December 27, 1999.

[106] Norman L. Johnson Samuel Kotz, N. Balakrishnan. *Continuous Multivariate Distributions, Models and Applications.* New York: John Wiley & Sons., 2000.

[107] Manjunath B G and Stefan Wilhelm. Moments calculation for the doubly truncated multivariate normal density. *ArXiv e-prints*, June 2012.

[108] Neudecker. Jan R, Magnus; Heinz. *Matrix Differential Calculus with Applications in Statistics and Econometrics.* New York: John Wiley & Sons, 1988.

[109] Bruno Sinopoli, Luca Schenato, Massimo Franceschetti, Kameshwar Poolla, Michael I Jordan, and Shankar S Sastry. Kalman filtering with intermittent observations. *IEEE transactions on Automatic Control*, 49(9):1453–1464, 2004.

[110] Xiangheng Liu and Andrea Goldsmith. Kalman filtering with partial observation losses. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 4, pages 4180–4186. IEEE, 2004.

[111] Athanasios Papoulis. *"Bernoulli Trials". Probability, Random Variables, and Stochastic Processes (2nd e.).* Number pp.57-63. New York: McGraw-Hill, 1984.

[112] McBrewster John Frederic P. Miller, Agnes F. Vandome, editor. *Bernoulli Process.* VDM Publishing, 2010.

[113] OpenMote Technologies. Openmote features. `http://www.openmote.com/hardware/openmote-cc2538-en.html`, 2015. [Online], Accessed Nov, 30, 2016.

[114] Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. Contiki-a lightweight and flexible operating system for tiny networked sensors. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pages 455–462. IEEE, 2004.

[115] S. Suthaharan, M. Alzahrani, S. Rajasegarar, C. Leckie, and M. Palaniswami. Labelled data collection for anomaly detection in wireless sensor networks. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2010 Sixth International Conference on*, pages 269–274, Dec 2010.

[116] Lennart Ljung. System identification toolbox getting started guide. `https://www.mathworks.com/help/pdf_doc/ident/ident_gs.pdf` [Online], Sept. 2016. Accessed Nov. 30. 2016.

[117] Lennart Ljung. System identification toolbox users guide. https://www.mathworks.com/help/pdf_doc/ident/ident.pdf [Online], Sep 2016. Accessed Nov. 30. 2016.

[118] Emmanuel Candes and Justin Romberg. l1-magic: Recovery of sparse signals via convex programming. 2005.

[119] François Ingelrest, Guillermo Barrenetxea, Gunnar Schaefer, Martin Vetterli, Olivier Couach, and Marc Parlange. Sensorscope: Application-specific sensor network for environmental monitoring. *ACM Trans. Sen. Netw.*, 6(2):17:1–17:32, March 2010.

[120] Texas Instruments Incorporated. CC2538 powerful wireless micro-controller system-on-chip for 2.4-GHz IEEE 802.15.4, 6LoWPAN, and ZigBee applications. http://www.ti.com/lit/ds/swrs096d/swrs096d.pdf, April 2015. [Online], Accessed Nov, 30, 2016.

[121] OpenMote Technologies. Openbattery features. http://www.openmote.com/hardware/openbattery.html, 2015. [Online], Accessed Nov, 30, 2016.

[122] OpenMote Technologies. Openbase features. http://www.openmote.com/hardware/openbase.html, 2015. [Online], Accessed Nov, 30, 2016.

[123] Thingsquare. Contiki os homepage. http://www.contiki-os.org/index.html, 2015. [Online], Accessed Nov, 30, 2016.

[124] Adam Dunkels, Fredrik Österlind, and Zhitao He. An adaptive communication architecture for wireless sensor networks. In *Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 335–349. ACM, 2007.

[125] Adam Dunkels. Rime-a lightweight layered communication stack for sensor networks. In *Proceedings of the European Conference on Wireless Sensor Networks (EWSN), Poster/Demo session, Delft, The Netherlands*. Citeseer, 2007.

[126] Adam Dunkels. The contikimac radio duty cycling protocol. 2011.

[127] Mathieu Michel and Bruno Quoitin. Technical report: Contikimac vs x-mac performance analysis. *arXiv preprint arXiv:1404.3589*, 2014.

[128] Andre Hahn Pereira. Change mac or radio duty cycling protocols. `https://github.com/contiki-os/contiki/wiki/Change-mac-or-radio-duty-cycling-protocols`, Aug 2015. [Online], Accessed 02. Dec. 2016.

[129] Texas Instruments Incorporated. TPS6273x step-down converter with bypass mode for ultra low-power wireless applications. `http://www.ti.com/lit/ds/symlink/tps62730.pdf`, December 2014. [Online], Accessed 10, December, 2016.

[130] Fatma Karray, Alberto Garcia-Ortiz, Mohamed W Jmal, Abdulfattah M Obeid, and Mohamed Abid. Earnpipe: A testbed for smart water pipeline monitoring using wireless sensor network. *Procedia Computer Science*, 96:285–294, 2016.

[131] Francois Botman Guerric de Streel Sebastien Bernard David Bol, Julien De Vos. Green socs for a sustainable internet-of-things. In *IEEE Faible Tension Faible Consommation (FTFC)*, June 2013.

[132] Xenofon Fafoutis, Alessio Di Mauro, Madava D Vithanage, and Nicola Dragoni. Receiver-initiated medium access control protocols for wireless sensor networks. *Computer Networks*, 76:55–74, 2015.