Communications in Mobile Wireless Networks: A Finite Time-Horizon Viewpoint

Yirui Cong

February 2018

A Thesis Submitted for the Degree of
Doctor of Philosophy
of The Australian National University

Research School of Engineering
College of Engineering and Computer Science
The Australian National University

Declaration

The contents of this thesis are the results of original research and have not been submitted for a higher degree to any other university or institution.

Much of the work in this thesis has been published or has been submitted for publication as journal papers or conference proceedings.

The research work presented in this thesis has been performed jointly with Dr. Xiangyun Zhou (The Australian National University), Prof. Rodney A. Kennedy (The Australian National University), and Prof. Vincent Lau (The Hong Kong University of Science and Technology). The substantial majority of this work was my own.

Yirui Cong
Research School of Engineering,
College of Engineering and Computer Science,
The Australian National University,
Canberra, ACT 2601,
AUSTRALIA

Acknowledgments

When I was a little child, I always asked my father questions about the "myths" of nature and science. Gratefully, my father is so knowledgeable and patient that every question I asked was answered by him; and every time I asked a good question, I got a lot of praises from him. With his help, I saw the first inverted image of a candle in my life from a self-made pinhole camera (when I was 5), and I witnessed the sunlight consisting of different colors by just using a normal mirror and a basin of water (when I was 7). I think my father is the main reason why I love doing research so much from deep inside!

After becoming a PhD student, I always asks myself a question – what the real meaning of a PhD is. I thought there would have been a perfect answer provided by someone, but I gradually realized that there is no perfect answer, and every PhD student has to seek his/her own answer individually and independently – that turns out to be the "perfect" answer!

Thankfully, I meet an excellent teacher and important friend that accompanies me on my PhD journey to seek the real meaning of my PhD study. This man is my primary supervisor Dr. Xiangyun (Sean) Zhou. Sean has great talent for discovering the advantages and shortcomings of his students. With his help, I keep challenging my limit and improving myself throughout my entire PhD journey. Under his wing, I am becoming an independent and well-trained researcher. I have always been feeling so lucky to be one of his students, and I would like to express my sincere gratitude to him for his caring supervision, patient guidance, and countless encouragements. I really appreciate the time we spent together, and it will be one of the most treasurable memories in my whole life!

I would like to express my heartfelt thanks to Prof. Rodney A. Kennedy for his great support and help. I have learned a lot from his insightful technical judgements on our co-authored papers, where I witnessed the extremely strong power from a highly experienced researcher and scientist. As the chair of my supervisory panel, Rod spent a lot of time in evaluating my annual progress and providing guidance on many aspects of my PhD study, which is greatly appreciated!

I would like to thank Prof. Vincent Lau at Hong Kong University of Science and Technology (HKUST), and Prof. Xiangke Wang at National University of Defense Technology (NUDT) for their supports during my visits. Thanks also go to Prof. Changbin (Brad) Yu for providing me advice on my application to the ANU PhD program before I came to Australia.

I take this opportunity to thank everyone in our research group for making the group friendly and relaxing research environment. Special thanks to Dr. Nan Yang, Dr. Shihao Yan,

Dr. Salman Durrani, and Dr. Gerard Borg. Thank you to my colleagues, both past and present, for helpful discussions and friendships, especially, Dr. Biao He, Dr. Jing Guo, Dr. Wanchun Liu, Xiaohui Zhou, Yuting Fang, Chunhui Li, Dr. Yifei Huang, Abbas Koohian, Noman Akbar, Khurram Shahzad, Mohammad Shahedul Karim, Jianwei Hu, Jihui Zhang, Jianping Yao, Xian Li, Xiaofang Sun, Nicole Sawyer, Simin Xu, and Yiran Wang.

I will thank my friends from other research groups in the RSISE building (now Brian Anderson Building), especially, Yi Zhou, Dr. Ni Ding, Xiang Wu, Dr. Hanchi Chen, Usama Elahi, Prof. Parastoo Sadeghi, Prof. Thushara D. Abhayapala, Dr. Wen Zhang, Fei Ma, Dr Prasanga Samarasinghe, Zohair Abu-Shaban, Dr. Alice Bates, Dr. Zhiyong Sun, Gao Zhu, Dr. Yun Hou, Dr. Xiaolei hou, Dr. Junming Wei, Xin Yu, Tong Zhang, Yonggang Hu, and Abdullah Fahim. Their companions make my PhD journey so colorful and enjoyable. Thanks also go to my friends from other universities for providing me beautiful memories during my visits, especially, Shimin Wang (Chinese University of Hong Kong), Prof. Huayong Zhu (NUDT), Dr. Qingjie Zhang (Aviation University of Air Force), Prof. Xin Xu (NUDT), Dr. Bingpeng Zhou (HKUST), Dr. Zhenghua Zhou (HKUST), Dr. Shengde Jia (NUDT), Zhaowei Ma (NUDT), Shulong Zhao (NUDT), Hao Chen (NUDT), Yangguang Yu (NUDT), Liyao Zhao (HKUST), Ho Man Lee (HKUST), and Kun Xiong (HKUST).

Thanks must go to the Chinese Government and the Australian National University for providing me the PhD scholarship and stipend.

Last but not least, I would like to express my gratitude, appreciation and love to my family for always standing with me for so many years. My parents Bin Cong and Suyan Liu, my grandparents Changhou Cong and Tianyun He, and my dear wife Xuena Lu, who are always the most important part in my life. This thesis is dedicated to my family, especially to my wife.

Abstract

In mobile wireless networks (MWNs), short-term communications carry two key features: 1) Different from communications over a large time window where the performance is governed by the long-term average effect, the short-term communications in MWNs are sensitive to the instantaneous location and channel condition caused by node mobility. 2) The short-term communications in MWNs have the finite blocklength coding effect which means it is not amenable to the well-known Shannon's capacity formulation.

To deal with the short-term communications in MWNs, this thesis focuses on three main issues: how the node mobility affects the instantaneous interference, how to reduce the uncertainty in the locations of mobile users, and what is the maximal throughput of a multi-user network over a short time-horizon.

First, we study interference prediction in MWNs by proposing and using a general-order linear model for node mobility. The proposed mobility model can well approximate node dynamics of practical MWNs. Unlike previous studies on interference statistics, we are able through this model to give a best estimate of the time-varying interference at any time rather than long-term average effects. In particular, we propose a compound Gaussian point process functional (CGPPF) in a general framework to obtain analytical results on the mean value and moment-generating function of the interference prediction.

Second, to reduce the uncertainty in nodal locations, the cooperative localization problem for mobile nodes is studied. In contrast to previous works, which highly rely on the synchronized time-slotted systems, this cooperative localization framework we establish does not need any synchronization for the communication links and measurement processes in the entire wireless network. To solve the cooperative localization problem in a distributed manner, we first propose the centralized localization algorithm based on the global information, and use it as the benchmark. Then, we rigorously prove when a localization estimation with partial information has a small performance gap from the one with global information. Finally, by applying this result at each node, the distributed prior-cut algorithm is designed to solve this asynchronous localization problem.

Finally, we study the throughput region of any MWN consisting of multiple transmitter-receiver pairs where interference is treated as noise. Unlike the infinite-horizon throughput region, which is simply the convex hull of the throughput region of one time slot, the finite-horizon throughput region is generally non-convex. Instead of directly characterizing all achievable rate-tuples in the finite-horizon throughput region, we propose a metric termed the rate margin, which not only determines whether any given rate-tuple is within the throughput

region (i.e., achievable or unachievable), but also tells the amount of scaling that can be done to the given achievable (unachievable) rate-tuple such that the resulting rate-tuple is still within (brought back into) the throughput region.

This thesis advances our understanding in communications in MWNs from a finite-time horizon viewpoint. It establishes new frameworks for tracking the instantaneous behaviors, such as interference and nodal location, of MWNs. It also reveals the fundamental limits on short-term communications of a multi-user mobile network, which sheds light on communications with low latency.

List of Publications

Journal Articles

J1. **Y. Cong**, X. Zhou, and R. A. Kennedy, "Interference Prediction in Mobile Ad Hoc Networks with a General Mobility Model," *IEEE Trans. Wireless Commun.*, vol. 14, no. 8, pp. 4277–4290, Aug. 2015.

J2. **Y. Cong**, X. Zhou, and R. A. Kennedy, "Finite-Horizon Throughput Region for Wireless Multi-User Interference Channels," *IEEE Trans. Wireless Commun.*, vol. 16, no. 1, pp. 634–646, Jan. 2017.

J3. **Y. Cong** and X. Zhou, "Event-Trigger Based Robust-Optimal Control for Energy Harvesting Transmitter," *IEEE Trans. Wireless Commun.*, vol. 16, no. 2, pp. 744–756, Feb. 2017. (not included in this thesis)

J4. **Y. Cong**, X. Zhou, B. Zhou, and V. Lau, "Cooperative Localization in Wireless Networks for Mobile Nodes with Asynchronous Measurements and Communications," submitted, Feb. 2018.

J5. **Y. Cong**, X. Zhou, and R. A. Kennedy, "Finite Blocklength Entropy-Achieving Coding for Linear System Stabilization," submitted, Dec. 2017. (not included in this thesis)

Conference Papers

C1. **Y. Cong**, X. Zhou, and R. A. Kennedy, "Rate-Achieving Policy in Finite-Horizon Capacity Region for Multi-User Interference Channels," in *Proc. IEEE GLOBECOM*, Washington, DC, USA, Dec. 2016, pp. 1–6.

C2. **Y. Cong** and X. Zhou, "Offline Delay-Optimal Transmission for Energy Harvesting Nodes," in *Proc. IEEE GLOBECOM*, Washington, DC, USA, Dec. 2016, pp. 1–6. (not included in the thesis)

C3. S. Yan, B. He, **Y. Cong**, and X. Zhou, "Covert Communication with Finite Blocklength in AWGN Channels," in *Proc. IEEE ICC*, Paris, France, May 2017, pp. 1–6. (not included in the thesis)

Acronyms

AEBR	average effective branching ratio
AIN	average iteration number
AMS	admissible measurement-state
AWGN	additive white Gaussian noise
BP	belief propagation
BPP	binomial point process
CGPPF	compound Gaussian point process functional
CSMA	carrier-sense multiple access
CV	coefficient of variance
DGPS	differential global positioning system
EBR	effective branching ratio
GLC	general-order linear continuous-time
GPS	global positioning system
GWN	Gaussian white noise
i.i.d.	independent and identically distributed
MAC	media access control
MWN	mobile wireless network
PP	point process
PPP	Poisson point process
pdf	probability density function
QoS	quality of service
RMSE	root-mean-square error

SINR	signal-to-interference-plus-noise ratio
SNCAMS	state-number-constraint admissible measurement-state
SPAWN	sum-product algorithm over a wireless network
UAV	unmanned aerial vehicle
UCM	uniform circular motion
UKF	unscented Kalman filter

Notations

$\mathbb{Z}$	set of integers
$\mathbb{Z}_+$	set of positive integers
$\overline{\mathbb{Z}}_+$	set of non-negative integers
$\mathbb{R}$	set of real numbers
$\mathbb{R}_+$	set of positive real numbers
$\overline{\mathbb{R}}_+$	set of non-negative real numbers
$\mathbb{R}^n$	n dimensional Euclidean space
$\emptyset$	empty set
$\forall$	universal quantification
$\exists$	existential quantification
$\wedge$	logical conjunction
$\vee$	logical disjunction
$\uparrow$	undefined term
$\in, \notin$	is an element of, is not an element of
$\subseteq, \subset$	is a subset of, is a proper subset of
$\bigcup, \bigcap$	union, intersection
$\#\mathcal{A}$	cardinality of finite or countably infinite set $\mathcal{A}$
$\mathcal{A} \setminus \mathcal{B}$	relative complement of $\mathcal{A}$ with respect to $\mathcal{B}$
$\mathcal{A} \times \mathcal{B}$	Cartesian product of $\mathcal{A}$ and $\mathcal{B}$
$A := B$	B is defined as A
$A =: B$	A is defined as B
$\|\cdot\|_q$	q-norm or $\mathcal{L}^q$-norm

$\|\cdot\|$	Euclidean norm or $\mathcal{L}^2$-norm
$O(\cdot)$	$f(x) = O(g(x))$ if and only if $\lim_{x\to\infty} f(x)/g(x) = \text{constant}$
$p(x)$	pdf of a random variable/vector x
$N(m,\Sigma)$	Gaussian distribution with mean vector m and covariance matrix Σ
$\mathbb{E}[\cdot]$	expectation operator
$\mathbb{E}_x[\cdot]$	expectation operator with respect to x
$\mathrm{Var}[\cdot]$	variance of a random variable
$\mathrm{std}[\cdot]$	standard deviation of a random variable
$\mathrm{Cov}[\cdot]$	covariance matrix of a random vector
$\mathrm{Cov}[\cdot,\cdot]$	covariance between two random variables/vectors
$\{x(t)\}_{t\in\mathcal{T}}$	stochastic process with respect to $t \in \mathcal{T}$
$\dot{x}(t)$	derivative of $\{x(t)\}_{t\in\mathcal{T}}$ in the mean square sense
$\log(\cdot)$	logarithm to base 2
$\ln(\cdot)$	natural logarithm
$\lfloor\cdot\rfloor$	floor operator
$\max\{\cdot\}$	maximum value of a totally ordered set
$\min\{\cdot\}$	minimum value of a totally ordered set
A^{T}	transpose of matrix A
$(\cdot)^+$	$(a)^+ = [\max\{a^{(1)},0\},\ldots,\max\{a^{(n)},0\}]^{\mathrm{T}}$ for $a = [a^{(1)},\ldots,a^{(n)}]^{\mathrm{T}}$
$a_1 \succeq (\succ,\preceq,\prec) a_2$	$a_1^{(1)} \geq (>,\leq,<) a_2^{(1)} \wedge \cdots \wedge a_1^{(n)} \geq (>,\leq,<) a_2^{(n)}$
$\overline{\mathbb{R}}_+^n (\mathbb{R}_+^n)$	$\{a \in \mathbb{R}^n \colon a \succeq (\succ) 0\}$

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Mobile wireless networks (MWNs) are present almost everywhere. From the information exchange between traditional mobile users to the communications for smart vehicles [1] and unmanned aerial vehicles (UAVs) [2], the MWN is a general concept representing communication devices having mobility (see Fig. 1.1). Due to its broad applications and the increasing demands, the MWN has already been acting as a pivotal role in supporting the communications for civil and military uses, and will show its great potentials in the future.

With the arrival of 5G networks, the MWN is facing the new standards/challenges from a wide range of quality of service (QoS) requirements, where one of the most important requirements is the low latency [3]. From the perspective of the physical layer, the low latency requirement means each communication packet should be transmitted and received within a very short time period. As a result, the design of communication systems is inevitably associated with the short-term effect.

Note that due to node mobility, the short-term effect behaves very differently from the well-studied long-term effect. For example, in Fig. 1.2 the signal-to-interference-plus-noise ratio (SINR) fluctuates with time. If we focus on the short-term SINR within the highlighted

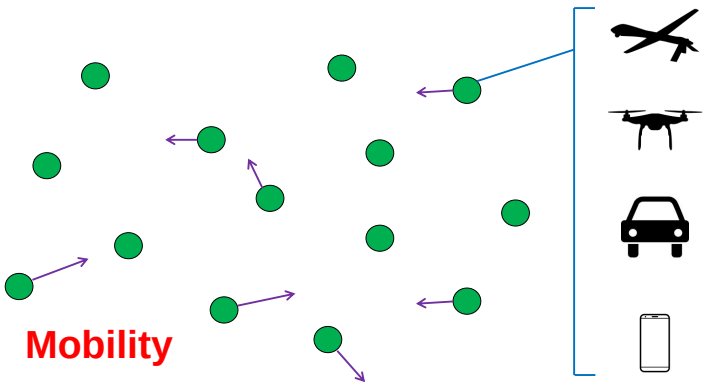


Figure 1.1: Mobile wireless networks.

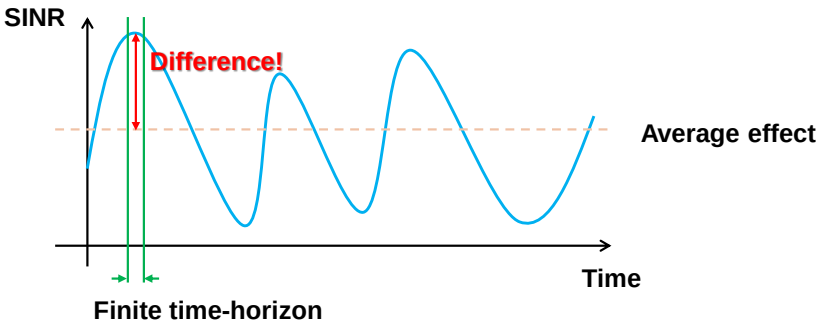


Figure 1.2: SINR changes in mobile wireless networks.

time window, then the short-term SINR is very different from the long-term average SINR. This motivates us to focus on the short-term behavior and performance of MWNs. To this end, this thesis focuses on the following aspects: 1) the mobility of the nodes need to be well modeled to understand the real-time effect as opposed to the long-term averaged effect. 2) the locations of nodes, if can be accurately estimated in a real-time manner, are beneficial for specifying the short-term behavior of a MWN. 3) Focusing on a short time-horizon, the communication performance of a MWN needs to be characterized.

1.1.1 Research Challenges

1.1.1.1 Node Mobility Modeling

Compared to the static wireless network, the MWN carries a key different feature – node mobility. Actually, without node mobility, the MWN will become a static wireless network. Thus, how to model the node mobility is of the first importance.

Current mobility models in MWNs are usually long-term oriented, e.g., the high mobility model [4, 5] and the random walk model [6] using Poisson point process (PPP) as it initial location distribution. Those models cannot reflect how the nodal locations change in real time. In other words, the nodal location distributions they returned are with no difference between different time instants. Consequently, they cannot provide sufficient information for short-term communications (see Fig. 1.2 for an example). Therefore, it is necessary to establish a new framework to model the node mobility in MWNs, which can accurately describe the motion of mobile nodes.

1.1.1.2 Time-Varying Interference Analysis

In MWNs, interference plays a pivotal role, through the SINR, in determining the communication performance. In contrast to static wireless networks, the distances between interferers and receiver change dynamically during the times of communications because of the mobilities of

interferers and receiver. As a result, the received signal is affected by fluctuating interference generated by these mobile nodes. This fact directly links the interference analysis and node mobility modeling (see Section 1.1.1.1) together.

Similar to the node mobility modeling, if the interference analysis is long-term oriented, then it cannot tell how the interference varies with time. Unfortunately, in the literature the interference analysis is based on the long-term oriented mobility models, e.g., [4, 6], and thus is not applicable for short-term communications in MWNs. Therefore, it is desirable to propose a new analysis framework in dealing with the time-varying interference.

1.1.1.3 Real-Time Localization

If we can track the nodal locations in real time, then it will be very beneficial to the communication analyses and designs in MWNs. An extreme/ideal example is that the location trajectories of all nodes in a MWN are known exactly. In that case, the communications in MWNs are only dependent on the radio propagation environment, which means the effect of node mobility is minimized. Practically, even though it is not possible to obtain such ideal location trajectories, we can use real-time localization technology to largely reduce the uncertainties in tracking the nodal locations.

Unlike the localizations in static wireless networks [7], the real-time localization is much more challenging in MWNs, since the measurements should be updated in a timely manner. This becomes even worse when the global positioning system (GPS) is not accessible to every node or the GPS cannot provide sufficiently accurate location information. In those cases, the real-time cooperative localization provides a potential solution [8]. However, this technology is still not complete in theory, and a series of problems like the synchronization and time delay are not well addressed.

1.1.1.4 Finite-Horizon Network Capacity

Seeking for the fundamental limit on transmissions in any MWN is important for communication designs. Traditionally, the network capacity refers to the one with infinite number of channel uses, and we call it the infinite-horizon network capacity in this thesis to distinguish it from the finite-horizon network capacity with finite number of channel uses. Even for the infinite-horizon network capacity, it is unknown in general, expect for some special cases, e.g., some multiple access channels and broadcast channels [9]. For the finite-horizon network capacity, it is largely an open research problem, where the finite blocklength effect [10] has to be taken into account.

The tremendous difficulty in deriving the network capacity is due to the potential complexity of cooperative coding schemes between nodes. If we only consider the point-to-point coding schemes, i.e., all the nodes just focus on the directly related transmitted/received in-

formation and regard the other information as noise, then the network capacity will become a network layer notion of capacity [11]. We call such a network capacity as the throughput region or stability region. Note that point-to-point coding schemes are usually considered in communication designs, and the throughput region plays an important role in deriving the fundamental limit on these communication designs.

Similar to the network capacity, the existing results for throughput region are mostly based on infinite time-horizon. Therefore, it is desirable to study the finite-horizon throughput region so that the fundamental limit on short-term communications in MWNs can be determined. A major challenge is that we need to deal with a finite number of channel uses and hence the classical Shannon's capacity cannot be used.

1.2 Background

In this subsection, we provide the background information of the techniques used in this thesis which largely makes the thesis self-contained.

1.2.1 Nodal Location and Mobility

Mathematically speaking, every point in our world corresponds to a unique vector y in $\mathbb{R}^2$ or $\mathbb{R}^3$ once an origin is selected. In MWNs, the altitude is often neglected, in which case the point is expressed in 2D space. When the altitude is considered, the point is modeled in 3D space. If the location of a node in MWN is perfectly known at time t, then its location is a point $y(t) \in \mathbb{R}^d$, where $d \in \{2,3\}$. If we do not have the perfect location information of a node at time t, then its true location can be one of the possible points with some possibilities. In that case, the location is a random vector $y(t)$ in 2D or 3D space. For example, when $y(t)$ is uniformly distributed in a 100m $\times$ 100m area, the location $y(t)$ follows the uniform distribution on $[-50,\ 50] \times [-50,\ 50]$ if we regard the centroid as the origin.

Since a MWN contains more than one node (at least two nodes), we use different indices $i \in \mathcal{I} := \{1,\ldots,\bar{I}\}$ to label different nodes, where $\mathcal{I}$ is the set of indices and $\bar{I} = \#\mathcal{I}$ is the number of nodes.[1] Then, the nodal locations in the MWN form a set $\{y_i(t) : i \in \mathcal{I}\} =: \mathcal{P}(t)$, where each $y_i(t)$ can be deterministic or stochastic. In this thesis, we call $\mathcal{P}(t)$ as the location pattern of the MWN at time t. Note that the location pattern is also known as the point process in stochastic geometry [12, 13, 14].

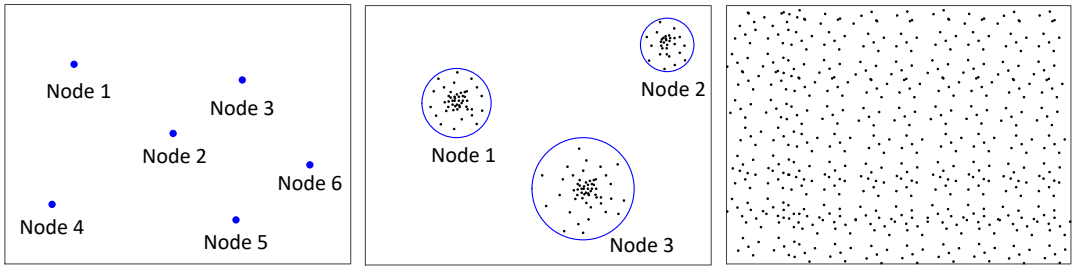
For a MWN, the location pattern $\mathcal{P}(t)$ contains all its nodal location information at time t. We classify the location patterns into three categories in Fig. 1.3:

[1] Note that I can be infinity, which means the MWN contains infinite number of nodes. This seems to be unrealistic, but it can reduce the complexity in mathematical modeling.

- **Deterministic pattern.** The locations of all nodes are known precisely, i.e., every location $y_i(t)$ in $\mathcal{P}(t)$ is constant [see Fig. 1.3(a)]. If a location pattern is not known exactly, we call it the stochastic pattern, which can be further classified into heterogeneous and homogeneous patterns as follows.

- **Heterogeneous pattern.** The location distributions are different for different nodes, i.e., $y_i(t)$ $(i \in \mathcal{I})$ are not identical [see Fig. 1.3(b)]. For heterogeneous pattern, it is possible to distinguish one node to the other by their location distributions.

- **Homogeneous pattern.** The location distributions are the same for all the nodes, i.e., $y_i(t)$ $(i \in \mathcal{I})$ are identical [see Fig. 1.3(c)]. For homogeneous pattern, it is impossible to distinguish any two nodes by their location distributions, e.g., the Poisson point process (PPP) and binomial point process (BPP) [12].



(a) (b) (c)

Figure 1.3: Classification of location patterns. For (a), it is an example of deterministic pattern, and a point is the true location of one node. For (b), it is an example of heterogeneous pattern, and a blue circle represents the region of all possible locations for one node or stands for the region of the locations for one node with high probability. In the blue circle of node i $(i \in \{1,2,3\})$, the black points are some realizations of $y_i(t)$ which reflect the probability density of $y_i(t)$. For (c), it is an example of homogeneous pattern, and the black points are some realizations of all nodes.

From Fig. 1.3(a) to Fig. 1.3(c), the uncertainty increases in turn, i.e., from the totally deterministic pattern to distinguishably uncertain pattern, to indistinguishably uncertain pattern.

As time goes by, one kind of patterns can become other the two kinds by increasing or decreasing uncertainties. The deterministic pattern can convert to the heterogeneous or homogeneous pattern when the node mobility is with uncertainty. For the heterogeneous pattern, it can develop into homogeneous pattern if the node mobility is with uncertainty; and it can convert to the deterministic pattern by reducing the location uncertainty, e.g., using a localization technique. For the homogeneous pattern, it can change to the heterogeneous or deterministic pattern, e.g., using a localization technique.

For the mobility of one node, it determines how the location changes over time due to the motion of this node. If $y_i(t)$ for node $i \in \mathcal{I}$ is deterministic, then the node mobility gives

a location trajectory $y_i(t)$ along time $t \in \mathcal{T}$, where the time set $\mathcal{T} \subseteq \overline{\mathbb{R}}_+$ can be discrete or continuous. In that case, the node mobility can be modeled by the difference equation (for discrete-time mobility) or differential equation (for continuous-time mobility). If $y_i(t)$ for node $i \in \mathcal{I}$ is stochastic, then the node mobility returns a stochastic process $\{y_i(t)\}_{t\in\mathcal{T}}$, which can be modeled by stochastic difference equation (for discrete-time mobility) or stochastic differential equation (for continuous-time mobility).

For the mobility of all nodes in a MWN, it leads to the set of $y_i(t)$ $(t \in \mathcal{T})$ or $\{y_i(t)\}_{t\in\mathcal{T}}$ for all $i \in \mathcal{I}$. Equivalently, it determines the location pattern process $\{\mathcal{P}(t)\}_{t\in\mathcal{T}}$.

1.2.2 Location Based Channel Gain, Interference, and SINR

In MWNs, the nodal locations change over time. Thus, the nodal location is the key feature in characterizing the property of communication channel and interference.

1.2.2.1 Time-Varying Power Gain of Communication Channels

The power gain of a communication channel is the ratio of the received signal power to the transmitted signal power. It plays an important role in designing communication systems.

For any two deterministic nodal locations $y_i(t)$ and $y_j(t)$ $(i, j \in \mathcal{I}, i \neq j)$, the power gain $H_{ij}(t)$ of the communication channel is determined by the radio propagation environment. Even though the accurate radio wave propagation model in MWNs is difficult to build up, we can use stochastic models to approximate $H_{ij}(t)$, which is usually considered in the communication theory. In this thesis, we employ the commonly used model as follows:

$$H_{ij}(t) = h_{ij}(t)g_{ij}(t), \tag{1.1}$$

where $h_{ij}(t)$ represents the power gain from (small-scale) fading, i.e., the fading gain, and $g_{ij}(t)$ is the (large-scale) path loss. The fading gain $h_{ij}(t)$ is a random variable for any given t which is typically assumed to be independent of the locations. For example, for Rayleigh fading, $h_{ij}(t)$ follows exponential distribution (see Chapter 5 in [15]). On the contrary, the path loss $g_{ij}(t)$ is dependent on nodal locations $y_i(t)$ and $y_j(t)$, and has the following form

$$g_{ij}(t) = g\left(\|y_i(t) - y_j(t)\|\right), \tag{1.2}$$

where $\|y_i(t) - y_j(t)\|$ returns the distance between node i and node j, and $g : \overline{\mathbb{R}}_+ \to \overline{\mathbb{R}}_+$ is called the path loss function which depends on the distance, e.g., the singular path loss function $g\left(\|y_i(t) - y_j(t)\|\right) = \|y_i(t) - y_j(t)\|^{-\alpha}$ with path loss exponent α.

Note that if $y_i(t)$ or $y_j(t)$ is not deterministic, then the distance $\|y_i(t) - y_j(t)\|$ is stochastic, which means the power gain $H_{ij}(t)$ depends on the distribution of $\|y_i(t) - y_j(t)\|$.

Now, we can give an important observation that: at time $t \in \mathcal{T}$, all the power gains $H_{ij}(t)$ $(i \neq j)$ are determined by the fading gains $h_{ij}(t)$, the path loss function g, and the location pattern $\mathcal{P}(t)$ (see Section 1.2.1). Therefore, in MWNs, the time-varying power gains of communication channels are fully characterized by the fading gains $h_{ij}(t)$, the path loss function g, and the location pattern process $\{\mathcal{P}(t)\}_{t\in\mathcal{T}}$.

1.2.2.2 Time-Varying Interference and SINR

Based on the power gain $H_{ij}(t)$ discussed in Section 1.2.2.1, if node i transmit messages to node j with transmit power $P_i^{\text{tx}}(t)$, then the received power from node j's side is

$$P_j^{\text{rx}}(t) = H_{ij}(t)P_i^{\text{tx}}(t). \tag{1.3}$$

It should be noted that in any MWN, node i is probably not the only communication device transmitting the signal using the same bandwidth. For any other transmitter $k \neq i$, if its transmitting signal is not wanted by node j, then node k is an interferer and the signal is an interference with the following power

$$I_{jk}(t) = H_{kj}(t)P_k^{\text{tx}}(t), \tag{1.4}$$

where $I_{jk}(t)$ is the interference power received by node j from node k, and we call it the interference from k to j for short. In (1.4), $P_k^{\text{tx}}(t)$ is the transmit power of interferer k, and $H_{kj}(t)$ is the power gain from k to j.

The aggregated interference at node j represents the sum of the received interference powers from all the interferers, and it has the following form

$$I_j(t) = \sum_{k\in\mathcal{I}_j^{\text{Int}}} I_{jk}(t) = \sum_{k\in\mathcal{I}_j^{\text{Int}}} H_{kj}(t)P_k^{\text{tx}}(t) \overset{(a)}{=} \sum_{k\in\mathcal{I}_j^{\text{Int}}} h_{kj}(t)g\left(\|y_k(t) - y_j(t)\|\right)P_k^{\text{tx}}(t), \tag{1.5}$$

where $\mathcal{I}_j^{\text{Int}}$ is the set of all interferers, and (a) follows from (1.1) and (1.2). If the transmit powers in the MWN are equal, then by normalizing the transmit power to 1,

$$I_j(t) = \sum_{k\in\mathcal{I}_j^{\text{Int}}} h_{kj}(t)g\left(\|y_k(t) - y_j(t)\|\right). \tag{1.6}$$

We can see that the aggregated interference at time t is highly dependent on the location pattern $\mathcal{P}(t)$, and the time-vary aggregated interference is highly related to the location pattern process $\{\mathcal{P}(t)\}_{t\in\mathcal{T}}$.

With the aggregated interference, we can define the SINR which is directly linked to the

maximal achievable rate in transmission. For node j, the SINR at time t is

$$\begin{aligned}\gamma_j(t) &= \frac{H_{ij}(t)P_i^{\text{tx}}(t)}{W_j + \sum_{k\in\mathcal{I}_j^{\text{tnt}}} H_{kj}(t)P_k^{\text{tx}}(t)} \\ &= \frac{h_{ij}(t)g\left(\|y_k(t)-y_j(t)\|\right)P_i^{\text{tx}}(t)}{W_j + \sum_{k\in\mathcal{I}_j^{\text{tnt}}} h_{kj}(t)g\left(\|y_k(t)-y_j(t)\|\right)P_k^{\text{tx}}(t)},\end{aligned} \tag{1.7}$$

where W_j is the power of white Gaussian noise received at node j. Note that the SINR at time t [i.e., $\gamma_j(t)$] is highly dependent on the location pattern $\mathcal{P}(t)$, and the time-varying SINR is highly related to the location pattern process $\{\mathcal{P}(t)\}_{t\in\mathcal{T}}$.

1.2.3 Real-Time Localization and Bayesian Filtering

From Section 1.2.2, we know that the communication channel, interference, and SINR are highly dependent on the location pattern process $\{\mathcal{P}(t)\}_{t\in\mathcal{T}}$. From Section 1.2.1, we see that if the node mobility is with uncertainties, then the location pattern can change from the deterministic pattern to heterogeneous pattern or homogeneous pattern; or from heterogeneous pattern to homogeneous pattern. It means that node mobility causes the location pattern to become more and more uncertain as time goes by. To reduce the uncertainty, we consider real-time localization which decreases the uncertainty by taking measurements.

Note that after localization is performed at a particular time instant to reduce location uncertainty, as time goes by the uncertainty will increase again due to the node mobility. As a result, the uncertainty will increase and decrease in turn, which implies the localization method is in a sequential/recursive manner (see Fig. 1.4). Mathematically, the probability density function (pdf) of nodal location $y_i(t)$ is $p(y_i(t))$. As t becomes larger, the uncertainty of $y_i(t)$ will increase, e.g., the diagonal entries in covariance matrix $\text{Cov}[y_i(t)]$ will increase. After taking measurement $z_i(t_1)$ from an anchor[2] at time $t_1 > t$, the location pdf is conditioned on this measurement, i.e., $p(y_i(t_1)|z_i(t_1))$ which corresponds to less uncertainty compared to $p(y_i(t_1))$. To calculate the $p(y_i(t_1)|z_i(t_1))$ from $p(y_i(t_1))$ and $z_i(t_1)$, we need Bayes' rule:

$$p(y_i(t_1)|z_i(t_1)) = \frac{p(z_i(t_1)|y_i(t_1))p(y_i(t_1))}{p(z_i(t_1))}, \tag{1.8}$$

where $p(y_i(t_1))$, $p(y_i(t_1)|z_i(t_1))$, and $p(z_i(t_1)|y_i(t_1))$ are called the prior, the posterior, and the likelihood function[3], respectively; and $p(z_i(t_1)) = \int p(z_i(t_1)|y_i(t_1))p(y_i(t_1))\text{d}y_i(t_1)$ is a constant that makes sure the integral of the posterior $p(y_i(t_1)|z_i(t_1))$ is 1. Since the posterior $p(y_i(t_1)|z_i(t_1))$ is fully determined by the nominator $p(z_i(t_1)|y_i(t_1))p(y_i(t_1))$, (1.8) is conven-

[2] Anchor means its location and/or other location related information are known.

[3] The likelihood function $p(z_i(t_1)|y_i(t_1))$ gives the pdf of $z_i(t)$ given $y_i(t)$, and it is determined by the measurement model, i.e., the relationship between the location $y_i(t)$ and the measurement $z_i(t)$.

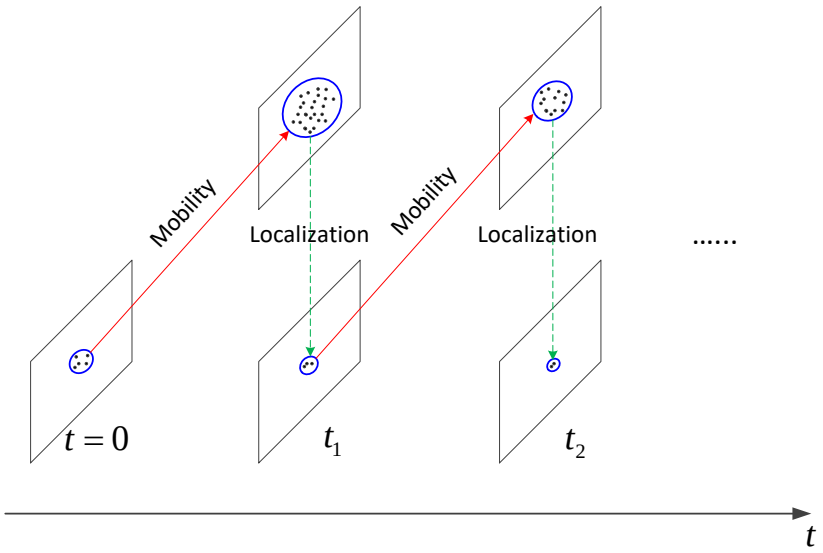


Figure 1.4: Illustrations of real-time localization for one node, where the blue circles and black points have the same meaning as that in Fig. 1.3. Time $t = 0$ gives the initial location distribution. Due to the node mobility, the uncertainty increases for $t \in (0,\ t_1)$ (see the red arrow from $t = 0$ to $t = t_1$). At time $t = t_1$, the localization is conducted so that the uncertainty is reduced (see the dashed green arrow at t_1). After $t = t_1$, the location uncertainty increases again during $t \in (t_1,\ t_2)$ which is followed the localization at t_2 to reduce the uncertainty once more. Likewise, this process will proceed forward for time instants t_k $(k = 3,4,5,\ldots)$.

tionally written as

$$p(y_i(t_1)|z_i(t_1)) \propto p(z_i(t_1)|y_i(t_1))p(y_i(t_1)). \tag{1.9}$$

For $t > t_1$, the nodal location provided by $p(y_i(t)|z_i(t_1))$ will become more and more uncertain again. If node j takes a measurement $z_i(t_2)$ at time $t_2 > t_1$, then the uncertainty in the nodal location will be reduced in $p(y_i(t_2)|z_i(t_1), z_i(t_2))$, which can be derived by Bayes' law in (1.8) with substituting $p(y_i(t_2)|z_i(t_1))$, $p(z_i(t_2)|y_i(t_2))$, and $p(z_i(t_2)|z_i(t_1))$ for the prior, the likelihood function, and the denominator, respectively. Likewise, the $k+1^{\text{th}}$ measurement $z_i(t_{k+1})$ can reduce the location uncertainty at t_{k+1} by employing Bayes' rule on $p(y_i(t_{k+1})|z_i(t_1),\ldots,z_i(t_k))$. The whole process is called the Bayesian filtering. This paragraph gives the key idea of the Bayesian filtering, and the readers are highly recommended to read the related books (e.g., [16, 17, 18]) for more details.

The localization method given above is in an ideal situation that node i can directly link to an anchor. This thesis considers more practical scenarios that most of non-anchor nodes cannot directly take measurement from an anchor, and hence all the nodes have to work cooperatively to perform self-localization.

1.2.4 Communications over Finite-Time Horizon

Traditionally, the maximal achievable rate μ in point-to-point transmission is described by Shannon's capacity C [19, 20]. Shannon's theorem states that for any transmission rate smaller than Shannon's capacity (i.e., $\mu < C$), the data can be transmitted with arbitrarily low (non-zero) error probability if the blocklength is sufficiently large. Large blocklength works fine in long-term communications, but for the short-term transmissions, the blocklength L is usually upper bounded (i.e., it cannot exceed a given number) and cannot be sufficiently large. Therefore, the constrained blocklength L is the very important aspect to be considered in short-term communications. One may ask, given a fixed blocklength L and an error probability ε which meets the requirements of a short-term transmission, what is the maximal achievable rate μ? This question is very interesting and practical, and serves as the foundation of the finite-horizon network throughput region which is established on point-to-point coding schemes.

To formally answer this question, we need to define (L,M,ε)-code [10], where M is the codebook size, i.e., the number of codewords one codebook contains. An (L,M,ε)-code means it can carry $\log M$ bits information with blocklength L and error probability ε. If we fix L and ε, M can be different for different coding schemes, where the largest M is

$$M^*(L,\varepsilon) = \max\{M\colon \exists \text{ an } (L,M,\varepsilon)-\text{code}\}, \tag{1.10}$$

which is called as the maximal code size. Then, the maximal achievable rate given L and ε is expressed as

$$\mu_{\max}(L,\varepsilon) = \frac{1}{L}\log M^*(L,\varepsilon). \tag{1.11}$$

Even though the exact form of $\mu_{\max}(L,\varepsilon)$ is still unknown in general, an accurate approximation is given by [10] that

$$\mu_{\max}(L,\varepsilon) = C - \sqrt{\frac{V}{L}}\tilde{Q}^{-1}(\varepsilon) + O\left(\frac{\log n}{n}\right), \tag{1.12}$$

where V is the channel dispersion, $\tilde{Q}$ is the complementary Gaussian cumulative distribution function. For AWGN channel, the channel dispersion is

$$V = \frac{\log^2 e}{2}\left[1 - \frac{1}{(1+\gamma)^2}\right], \tag{1.13}$$

where γ is the SINR.

With above discussions on the maximal achievable rate and its approximation in finite blocklength coding schemes, the fundamental limit on the transmission rate for each link in a MWN can be determined. Therefore, the finite-horizon network throughput region can be studied by optimally allocating the transmit powers and designing the routing algorithms in a

given time window. In this thesis, we focus on an important case of the finite-horizon network throughput region that N transmitter-receiver pairs transmit data in an Gaussian interference channel, i.e., the finite-horizon throughput region for multi-user Gaussian interference channels It helps us to understand the difference between the finite and infinite throughput regions and serves as the first step to study the finite-horizon network throughput region.

1.3 Literature Review

In this thesis, we mainly focus on the following three important problems:

- How to predict the time-varying interference with the node mobility model?
- How to localize mobile nodes cooperatively?
- What is the finite-horizon throughput region for multi-user Gaussian interference channels?

The related prior work is described in this section. The limitations of the existing work are also discussed.

1.3.1 Interference Analysis in Mobile Wireless Networks

1.3.1.1 Existing Works

Interference analysis can help to discover and exploit the regularity of time-vary interference. For example, it helps to understand the statistical performance of communication under such interference, e.g., outage probability. Due to disturbances on nodes' movement or our incomplete information of the node locations, the trajectories of these nodes are often accompanied with uncertainties when analyzing interference in MWNs. Stochastic Geometry [12, 13] is a powerful tool for describing the random pattern of mobile nodes as a point process (PP) at each time instant. For mobile networks with high mobility nodes, the authors in [4] analyzed the local delay, which is a functional of the interference moment generating function (MGF), by modeling node locations as a Poisson point process (PPP). Indeed, if one just focuses on one time instant, the existing results on interference analysis in static networks can be directly used in highly mobile networks (e.g., for PPP networks [4, 5, 21, 22] or for binomial point process (BPP) networks [23, 24]). Highly mobile networks imply that the node locations at two different time instants have almost no correlation, if the node velocities are sufficiently large. Nevertheless, for most practical cases, we are more interested in interference analysis in MWNs with finite nodal velocities.

In order to analyze interference in MWNs with finite node velocities, mobility models are required to capture the node locations at each time instant. A summary of mobility models and

Table 1.1: Summary of Mobility Models and Their Corresponding Point Processes in Previous Studies in MWNs

Reference	Mobility Model	Point Process
[4, 5, 21, 22, 25, 26, 27, 28, 29, 30]	Highly mobile networks	PPP (time independent)
[23, 24]	Highly mobile networks	BPP (time independent)
[6, 31, 32]	Constrained i.i.d. mobility	PPP (time homogenous, non-Markov [6])
[6, 32, 33]	Random walk	PPP (time homogenous)
[6, 32]	Brownian motion	PPP (time homogenous)
[6, 32, 33]	Random waypoint	PPP (time homogenous) with quadratic polynomial intensity

Time independent: the locations of nodes change independently from one time instant to the next.
Time homogenous: the locations of nodes are correlated between different time instants but have the same pdf.

their corresponding point processes in previous studies is provided in Table 1.1. By employing random walk, Brownian motion and random waypoint models[4], the statistics of interference were analyzed in [6, 32, 33]. For random walk and Brownian motion models, the approximate distribution of aggregate interference was given in [6, 33]. The mean of the aggregate interference was analyzed in [6], and upper bounds in temporal correlation for aggregate interference and outage probability were given in [32] and [6]. For the random waypoint model, similarly, the approximated distribution of aggregate interference was given in [6, 33], and the mean of the aggregate interference was analyzed in [6].

1.3.1.2 Limitation of Existing Interference Analysis Studies

The existing analysis methods only considered special or limiting scenarios where statistics of interference are identical at every time instant. For the random walk and Brownian motion models in [6, 33], the means or pdfs (probability density functions) of aggregate interference at every time instant are the same due to the assumption that the initial node distribution is uniform. Consequently, node dynamics do not provide any useful information in the interference analysis. For the random waypoint model in [6, 33], the idea was to wait an infinite time for the node distribution to converge to one limiting distribution that can be regarded as a PPP with a quadratic polynomial intensity [34]. Based on that kind of PPP, mean, outage and approximated distributions for aggregate interference were given. As a result, not only do node dynamics not contribute to the analysis, but also the results are only valid in infinite time. It should be noted that in many applications what we would like to know are the statistics of aggregate interference within a finite time window rather than an infinite one.

[4]The constrained i.i.d. mobility model in [31] is similar to the highly mobile network, thus we mainly discuss the interference analysis in MWNs under random walk, Brownian motion and random waypoint.

Furthermore, the existing analysis on interference between multiple time instants typically focused on the temporal correlation [6, 25, 26, 27, 32], which provides no further information beyond a linear relation. To design a communication strategy (e.g., transmission power control), we are interested in knowing and exploiting the statistics of interference at a future time of interest, which cannot be derived from a simple temporal correlation. We assert that interference prediction can provide us more effective information than temporal correlation. Beyond the simple temporal correlation, [30] proposed a joint temporal characteristic function of interference for multiple time instants, and [25, 28, 29] investigated the conditional/joint outage/success probabilities over time. Despite the comprehensive characterizations of the temporal statistics, the studies in [25, 28, 29] assumed networks with either fixed or independent node locations from one time slot to the next. Hence, their results cannot be used for interference prediction in realistic network with practical mobility models.

In fact, another limitation of current studies on interference in MWNs is the mobility model. For example, the random walk, Brownian motion and random waypoint models are often inadequate to describe many kinds of mobilities in the real world, like mobilities constrained by the physical laws of acceleration and velocity [35]. Modelling should include all kinds of communicating objects that have the ability to move. With the development of automation, the need for communication between robots is increasing. For military use, UAVs (which can be regarded as a group of intelligent flight robots) deliver new and enhanced battlefield capabilities to warfare. For civil use, unmanned aircrafts offer new ways for commercial enterprises and public operators to increase operational efficiency, decrease costs, and enhance safety. The node mobilities in such scenarios have complex dynamics and cannot be captured by random walk, Brownian motion, random waypoint mobility models or even the Gauss-Markov mobility model [35, 36] that takes acceleration and velocity into account. Therefore, it is desirable to develop a more general mobility model that is capable to describe mobile nodes governed by complex mobility dynamics, and then use it for interference prediction.

1.3.2 Cooperative Localization in Mobile Wireless Networks

1.3.2.1 Existing Works

Wireless network based cooperative localization has attracted significant attention in recent years. To locate the position, each wireless device (node) works cooperatively with other nodes in the same wireless network. Generally speaking, there are three key elements for a cooperative localization algorithm: 1) receiving location-related information from other nodes (information collecting); 2) estimating location based on the received information so far (information inferring); 3) transmitting new location-related information to other nodes (information sharing). The information collecting-inferring-sharing framework is very helpful to understand the existing works and develop new techniques.

We review the Belief-Propagation (BP) based cooperative localization which is a commonly used method in locating mobile nodes. The BP localization is based on the factor graph [37] in the Bayesian filtering framework [18]. It marginalizes the joint posterior into approximated marginal posteriors (i.e., beliefs) by iteratively interchanging local information (i.e., scheduled message passing) among a given cluster of nodes, where each marginal posterior refers to the posterior of the location-related state[5] in the corresponding node. Employing the BP method in localization problems can date back to the study on static nodes [39], which was then generalized to the famous sum-Product algorithm over a wireless network (SPAWN) method for mobile nodes by the celebrated work [8]. Note that for SPAWN the information is collected, inferred, and shared multiple times (by iterations) in each time slot, which is not cheap in communications. To solve this problem, a series of improvements were studied to reduce the computational complexity and communication (iteration) times [40, 41, 42, 43, 44]. In [40], a Gaussian mixture model replaced the particles to describe the beliefs, and the communication scheme was modified to only include beliefs (rather than include both beliefs and messages) where Kullback Leibler divergence was employed to censor if a belief is good enough to transmit. In [41], the sigma point BP was proposed to meet the low communication requirements. Based on the sigma point BP, [42] designed a greedy algorithm for further reducing the communications. In [43, 44], the BP method was combined with the mean-field approximation to reduce the message size in transmission.

1.3.2.2 Limitation of Existing Cooperative Localization Studies

For mobile nodes, the cooperative localization techniques, e.g., the BP based localizations, are usually based on the sequential Bayesian filtering technique in time-slotted systems. That means all the information in the wireless network must be collected, inferred, and shared in a synchronized manner corresponding to the time-slotted system. Otherwise, the internodal information would be meaningless, e.g., one node's first time slot would cross another node's first and second time slots, in which case the collected information (e.g., the measurements) cannot provide a correct information inferring. Thus, synchronization is important (see also [45, 46, 47]), and without any synchronization, the existing cooperative localization algorithms for mobile nodes cannot work properly.

In practical scenario, it is not an easy thing to synchronize all the communication links and sensor measurements, especially in the large-scale wireless networks. This is because the synchronization not only refers to the synchronous clocks, but also requires the communications and measurements are synchronized in the same way. This fact motivates us to rethink the sequential Bayesian filtering framework: how to model and solve the cooperative localization

[5]For example, states can include the location, velocity, or acceleration, etc., which depends on the mobility model. More details can be found in [8, 38].

problem for mobile nodes in an asynchronous way. In this thesis, we aim to provide an answer to this problem.

1.3.3 Finite-Horizon Throughput Region

Since this is the first work that rigorously studies the finite-horizon throughput region, the most related prior works are the ones on infinite-horizon throughput region. Specifically, the seminal work in [48, 49] introduced the infinite-horizon throughput region and gave two important results: the infinite-horizon throughput region is the convex hull of one-slot throughput region; and the max-weight algorithm can achieve any given rate-tuple in the throughput region. In [11, 50], the infinite-horizon throughput region was generalized and applied to time-varying wireless networks, and a max-weight algorithm based transmission policy was designed. We recommend the tutorials in [51, 52, 53] to readers who are interested in the infinite-horizon throughput region.

Some recent studies focused on reducing the delay by shrinking the infinite-horizon throughput region [54, 55, 56, 57], where the average delay was studied in [54, 55], and the worst-case delay was analyzed in [56, 57]. It was observed by [54, 55, 56, 57] that choosing a rate-tuple closer to the boundary of the infinite-horizon throughput region causes a larger delay, and hence, shrinking the throughput region removes those rate-tuples near the boundary corresponding to large delays. Furthermore, the effect of finite buffer size was consider in [58, 59, 60], and it turned out that the required buffer size increases with the rate-tuple. Indeed, by Little's law, the average length of data queue is proportional to the delay. Hence, this line of work also demonstrated the delay caused by the rate-tuples in the infinite-horizon throughput region.

We stress that in light of the studies on infinite-horizon throughput region, a small number of studies have introduced the concept of finite-horizon throughput region. However, they did not analyze any property of the finite-horizon throughput region: The work in [61] proposed a T-slot lookahead utility which helped to analyze the short-term performance for the proposed opportunistic scheduling algorithm, but no analysis on finite-horizon throughput region was presented; In [56], the rate-tuple over a finite time horizon was defined, but it was only employed to derive the infinite-horizon throughput region when the number of time slots goes to infinity. A possible reason for the lack of study on the finite-horizon throughput region might be that the finite-horizon throughput region was thought to have similar properties as its infinite-horizon counterpart. As we will discuss in this thesis, however, the finite-horizon throughput region behaves very differently as compared with the infinite-horizon throughput region.

Despite the significant efforts made on studying the achievable rate-tuple and infinite-horizon throughput region [53, 62, 63], significantly less is known about the throughput re-

gion over a finite horizon. As mentioned in Section 1.2.2, the channel, interference, and SINR change with time due to the node mobility. Therefore, it is desirable to design transmission for a finite time duration such that the network and channel information used in the design is timely and matches with the condition during the actual transmission. Another important reason for considering the finite-horizon throughput region is the guaranteed delay (low latency, as mentioned in Section 1.1). For example, if a rate-tuple is achievable in a five-slot throughput region, then the time delay for the transmitted packets is at most five time slots. On the contrary, any rate-tuple in an infinite-horizon throughput region can possibly cause an unacceptably large delay. This also motivates us to study the finite-horizon throughput region of a MWN. To the best of our knowledge, the finite-horizon throughput region has not yet been investigated.

1.4 Thesis Outline and Contributions

Chapter 2 – Interference Prediction with a General Mobility Model

In Chapter 2, we focus on interference prediction in MWNs with nodes following a very general class of mobilities. Our interference prediction provides the best estimate of the interference level at a future time instant based on the knowledge at the current time. By developing a general mobility model, the predictions can be used in a wide range of MWNs.

The main contributions of this work are:

- We propose a general-order linear continuous-time (GLC) mobility model to describe the dynamics of moving nodes in practical applications. The random walk, Brownian motion and Gauss-Markov mobility models in [6, 33, 35] can be regarded as special cases of the GLC mobility model with discretizations. In this framework, the random walk and Brownian motion turn out to be first-order linear mobility models, and Gauss-Markov model is a second-order linear mobility model.

- Based on the GLC mobility model, the mean and moment-generating function (MGF) of interference prediction on a mobile reference point at any finite time into the future are derived and analyzed. In order to simplify the expression for mean and MGF under different path loss functions and multipath fading, a compound Gaussian point process functional (CGPPF) is defined and expressed in a series form.

- Apart from interference prediction at finite time into the future, we also give the necessary and sufficient condition for when our predictions converge to those from a Gaussian BPP as time goes to infinity. This result provides a guideline on when the previous studies on interference statistics with BPP are relevant.

The results in this chapter have been presented in [64], which is listed again for ease of reference:
[64] **Y. Cong**, X. Zhou, and R. A. Kennedy, "Interference Prediction in Mobile Ad Hoc Networks with a General Mobility Model," *IEEE Trans. Wireless Commun.*, vol. 14, no. 8, pp. 4277–4290, Aug. 2015.

Chapter 3 – Cooperative Localization with Asynchronous Measurements and Communications

In Chapter 3, we establish the mathematical model of the asynchronous cooperative localization problem for mobile nodes, and solve this problem by our proposed prior-cut algorithm.

Before giving the detailed contributions, we highlight the key idea of prior-cut algorithm here. Different from the BP method which marginalizes the posterior (see Section 1.3.2), the prior-cut algorithm is based on cutting the prior into two parts at each node: one part is closely related to the self-location, and the other part is not. Then, every node refines its prior from the received messages by prior cutting, and discards less useful information accordingly (i.e., only collecting the useful information for self-localization).

The main contributions are given as follows:

- The asynchronous cooperative localization problem is modeled in the continuous-time domain. To be more specific, the mobility models of mobile nodes are modeled by stochastic differential equations, where a measurement can be taken and a communication can happen at arbitrary time instants without any time-slotted constraint.

- We give the centralized localization algorithm by the continuous-discrete Bayesian filtering theory, where the global information in the entire network is utilized. Then, we strictly prove an important property of the centralized algorithm that: if we cut the prior properly, i.e., the two cut parts are independent enough, then a localization algorithm with partial information can get very close to the centralized algorithm.

- Following the analysis of the centralized localization algorithm, the prior-cut algorithm is proposed. This algorithm has a memory at each node to record the history information which is different from the existing algorithms in the Bayesian framework. When a node receives the related information from other nodes (information collecting), the memory update is triggered (information inferring). Specifically, the history information is updated by the continuous-discrete Bayesian filter, and the priors are refined by prior cutting. After updating the memory, the node broadcasts its refined information to other nodes (information sharing), but only the nodes, who think this information is useful, will use this information.

The results in this chapter have been presented in [65], which is listed again for ease of reference:
[65] **Y. Cong**, X. Zhou, B. Zhou, and V. Lau, "Cooperative Localization in Wireless Networks for Mobile Nodes with Asynchronous Measurements and Communications," to be submitted, Sept. 2017.

Chapter 4 – Finite-Horizon Throughput Region for Multi-User Interference Channels

In Chapter 4, we investigate the finite-horizon throughput region of a wireless network consisting of multiple transmitter-receiver pairs. It should be noted that studying the finite-horizon throughput region is far more challenging than its infinite-horizon counterpart for the following reasons: (i) Unlike the convex throughput region for the infinite horizon, the finite-horizon throughput region is non-convex and the computational complexity for determining it is exponentially increasing with the number of time slots. (ii) As we will show, a rate-tuple that is achievable in T_1 time slots may not be achievable in T_2 $(T_2 > T_1)$ time slots. This is in contrast with the fact that any achievable rate-tuple over a finite horizon is also achievable over the infinite horizon. This property prevents us from using a result for one throughput region to obtain a result for another throughput region with a different number of time slots.

Instead of directly characterizing the throughput region by finding the set of all achievable rate-tuples, we provide an efficient method to determine whether an arbitrary given rate-tuple is achievable or not. More specifically:

- We propose a metric termed the *rate margin*. By computing the rate margin of any given rate-tuple, we are able to tell whether a given rate-tuple is achievable within the considered finite horizon. Furthermore, the rate margin also provides information to the system designer on: (i) how much one can scale up the given achievable rate-tuple so that the resulting rate-tuple is still within the finite-horizon throughput region; (ii) how much one can scale down the given unachievable rate-tuple so that the resulting rate-tuple is brought back into the finite-horizon throughput region.

- We provide the rate-achieving policy for any achievable rate-tuple in a finite-horizon throughput region by determining the transmit power and rate for each communication pair in each time slot.

- We formulate an optimization problem for computing the rate margin and deriving the rate-achieving policy. The solution inevitably requires a search. To reduce the complexity while maintain the optimality of the search, we use three techniques among which the most important one is the proposed admissible heuristic function that allows the highly-efficient A* search algorithm to be employed. The simulation result demonstrates the

computational efficiency of our algorithm.

The results in this chapter have been presented in [66] and [67], which is listed again for ease of reference:

[66] **Y. Cong**, X. Zhou, and R. A. Kennedy, "Finite-Horizon Throughput Region for Wireless Multi-User Interference Channels," *IEEE Trans. Wireless Commun.*, vol. 16, no. 1, pp. 634–646, Jan. 2017.

[67] **Y. Cong**, X. Zhou, and R. A. Kennedy, "Rate-Achieving Policy in Finite-Horizon Capacity Region for Multi-User Interference Channels," in *Proc. IEEE GLOBECOM*, Washington, DC, USA, Dec. 2016, pp. 1–6.

Chapter 2

Interference Prediction with a General Mobility Model

2.1 Introduction

In a MWN, effective prediction of time-varying interference can enable adaptive transmission designs and therefore improve the communication performance.

This chapter investigates interference prediction in MWNs with a finite number of nodes by proposing and using a general-order linear model for node mobility. The proposed mobility model can well approximate node dynamics of practical MWNs. In contrast to previous studies on interference statistics, we are able through this model to give a best estimate of the time-varying interference at any time rather than long-term average effects. Specifically, we propose a compound Gaussian point process functional (CGPPF) as a general framework to obtain analytical results on the mean value and moment-generating function of the interference prediction. With a series form of this functional, we give the necessary and sufficient condition for when the prediction is essentially equivalent to that from a binomial point process (BPP) network in the limit as time goes to infinity. These conditions permit one to rigorously determine when the commonly used BPP approximations are valid. Finally, our simulation results corroborate the effectiveness and accuracy of the analytical results on interference prediction and also show the advantages of our method in dealing with complex mobilities.

This chapter is organized as follows. In Section 2.2, we present the case of uniform circular motion (UCM) as an example to motivate our general model. In Section 2.3 we define and analyze the GLC mobility model. Additionally, the dynamic reference point is defined to study the relative node locations of interferers relative to the mobile receiver. In Section 2.4, the mean and MGF of the interference prediction are analyzed. The numerical examples are given in Section 2.5 to illustrate the effectiveness of our approach and corroborate our analytical results. Finally, the concluding remarks of this chapter are given in Section 2.6.

2.2 Interference in MWNs with Uniform Circular Motions

In this section, we investigate the interference that arises in nodes exhibiting uniform circular motion (UCM) to model UAVs carrying out a scanning task. This simple model will be seen to be a special case of the general mobility model that we develop in Section 2.3.

Consider $N \in \mathbb{Z}^+$ UAVs carrying out a scanning task for a target, see Fig. 2.1(a), which has many applications such as rescue operations, monitoring applications, etc. This target (the central pink circle) could be a point or an area. Due to limitations of their visual fields, each UAV can only acquire partial information about the target. Therefore, in order to have a better understanding of the target, they need to share information using wireless communication.

Even though the dynamic model of the UAVs when scanning a target can be highly nonlinear and possibly complex, the mobility model can be approximated by UCM in the 2-D plane as two coupled differential equations

$$\begin{cases} \dot{x}_i^{(1)}(t) = \omega_i x_i^{(2)}(t) + w_i^{(1)}(t) \\ \dot{x}_i^{(2)}(t) = -\omega_i x_i^{(1)}(t) + w_i^{(2)}(t) \end{cases} \tag{2.1}$$

where the subscript i denotes the ith node, and $i = 1,2,\ldots,N$, where N is the number of nodes in the MWN. $(x_i^{(1)}(t), x_i^{(2)}(t))$ is the location of node i in the 2D plane at time t, and ω_i is the UCM angular velocity, and the initial vector $x_i(t_0) = [x_i^{(1)}(t_0), x_i^{(2)}(t_0)]^T$ determines the initial location and radius of node i. $w_i^{(1)}(t)$ and $w_i^{(2)}(t)$ stand for additive disturbance due to airflow.

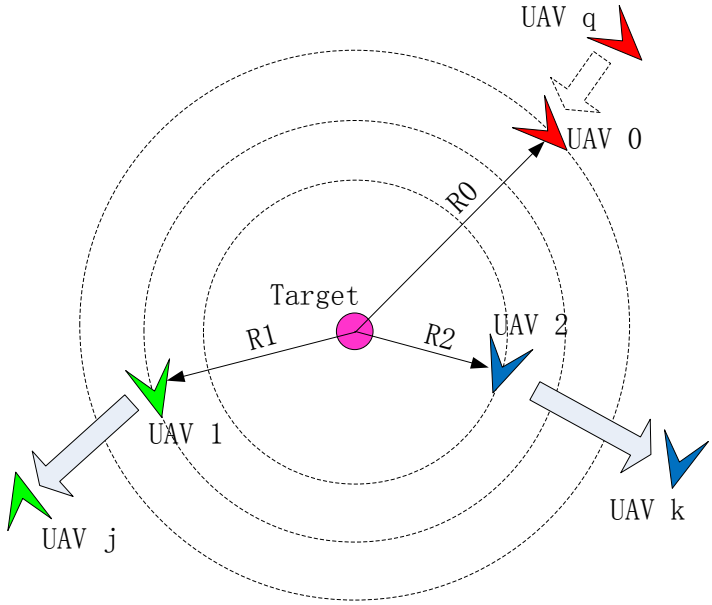
Consider a duration of time in which UAV 1 and 2 communicate with UAV j and k, respectively. For UAV 0, it tries to receive information from UAV q. Therefore, there are two interferers that affect the signal reception at UAV 0.

The aggregate interference at UAV 0 is shown in Fig. 2.1(b),[1] from which we see that the received aggregate interference at UAV 0 is periodically changing and does not converge to a constant value independent of time t. Interference predictions should be adaptive to the node dynamics and therefore make use of the characteristics of mobility models.

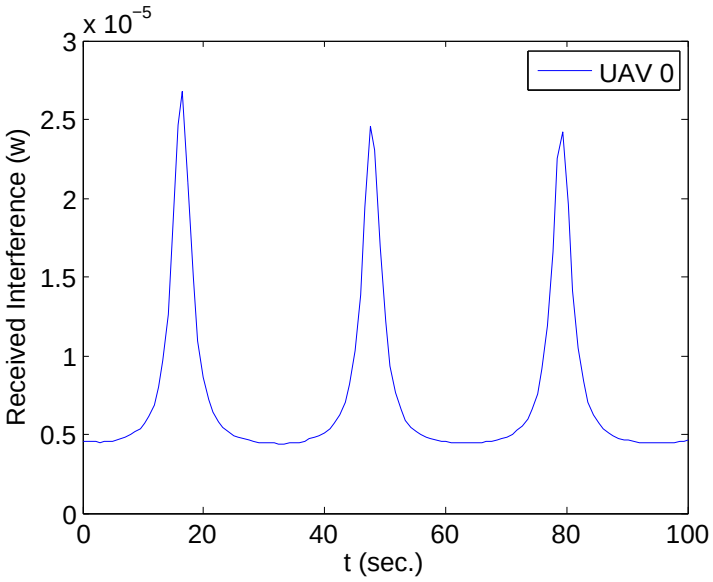
2.3 General-order Linear Mobility Model

In this section, the general-order linear continuous-time (GLC) Mobility Model is proposed and the corresponding statistics of node distribution is given.

[1] Simulation Parameters: Signal power is 1 for each UAV, and the mobility model follows (2.1) with $\omega_0 = \omega_2 = 0.1\text{rad/s}$ and $\omega_1 = -0.1\text{rad/s}$. The initial locations for UAV $0-2$ are $(500,500)$, $(-400,-300)$, $(400,0)$ (selecting the target center as the origin), thus according to (2.1) average radii are $R_0 = 500\sqrt{2}\text{m}$, $R_1 = 500\text{m}$, and $R_2 = 400\text{m}$. The powers of $w_i^{(1)}(t)$ and $w_i^{(2)}(t)$ are assumed to be unit. We assume the Path loss function is r^{-2}, since UAVs often hover in a free space (i.e., no obstacles between them), where r is the Euclid distance between transmitter (UAV) and the point whose interference is needed to be calculated.



(a)



(b)

Figure 2.1: (a) Three UAVs scan a target by using vision sensors and share information through a wireless network. (b) The received interference UAV 0 versus time t.

2.3.1 General-order Linear Mobility Model and Node Distribution

Consider a network having N nodes in d-dimensional space (e.g., $d = 2$ means nodes move in a 2D plane). For each node i, where $i \in \{1,2,\ldots,N\}$, we model it employing the state-space model with additive uncertainties given by the stochastic differential and algebraic equations [68]

$$\dot{x}_i(t) = A_i x_i(t) + w_i(t) \tag{2.2}$$

$$y_i(t) = C_i x_i(t), \tag{2.3}$$

where the state vector $x_i(t)$ is a random vector in $\mathbb{R}^n$, which can contain the velocity, acceleration or angular velocity, etc., and it depends on the way the mobilities of nodes are modelled. The additive uncertainties $w_i(t) \in \mathbb{R}^n$ (assumed to be a second-order moment process) can represent the airflow for aircrafts (see Section 2.2), velocity uncertainty for mobile phone users, etc. The location of node i in the d-dimensional space is denoted as $y_i(t) \in \mathbb{R}^d$. The constant matrices $A_i \in \mathbb{R}^{n\times n}$, and $C_i \in \mathbb{R}^{d\times n}$ are the model parameters determined by the node dynamics. The initial vector for the differential equation (2.2) is $x_i(s)$ at time s and is independent from $w_i(t)$, $\forall t > s$, because $w_i(t)$ only affects the future behavior of $x_i(t)$.

For the UCM example in Section 2.2, we have

$$A_i = \begin{bmatrix} 0 & \omega_i \\ -\omega_i & 0 \end{bmatrix}, \quad C_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{2.4}$$

and indeed $y_i(t) = x_i(t)$. This shows UCM is a second-order linear mobility model.

Definition 2.1 (Homogenous Mobility)**.** If pairs (A_i, C_i) for all the nodes are equal, i.e.,

$$A_i = A, \quad C_i = C, \quad i \in \{1,\ldots,N\} \tag{2.5}$$

and $w_i(t)$ for all nodes are i.i.d., then the node mobilities are homogenous.

Remark 2.1. The random walk and discrete-time Brownian motion models given in [6, 33] are homogenous and can be regarded as a special case when (2.2) is discretized and $A = 0$. Thus, they are homogenous first-order linear mobility models. The one-dimensional homogenous continuous-time mobility model with

$$A = \begin{bmatrix} 0 & 1 \\ 0 & \ln(1-\alpha) \end{bmatrix}, C = [1 \quad 0], w(t) = \begin{bmatrix} 0 \\ w^{(2)}(t) \end{bmatrix}, \tag{2.6}$$

can be discretized to recover the Gauss-Markov mobility model given in [35], where $\alpha \in (0,1)$ and the mean of $w^{(2)}(t)$ is the asymptotic velocity.

Remark 2.2. For any second-order moment process, the mean vector $\mathbb{E}\left[y_i(t)\right]$ and covariance matrix $\mathrm{Cov}\left[y_i(t)\right]$ of $y_i(t)$ can be calculated, even though the pdf of $y_i(t)$ has no closed-form expression in general. For a fixed covariance matrix $\mathrm{Cov}\left[y_i(t)\right]$, the entropy is maximized when $y_i(t)$ is Gaussian [20]. It implies that for all kinds of second-order moment process $w_i(t)$, Gaussian process contains the most uncertainties. Practically, it is difficult to determine what kind of process $w_i(t)$ is, and hence the best choice is to conservatively (since it contains the most uncertainties) regard $w_i(t)$ as a Gaussian process. Furthermore, $w_i(t)$ is independent on time t in practice, we often consider it as a white noise. Therefore, $w_i(t)$ can be regarded as Gaussian white noise (GWN). Actually, GWN is widely used in modeling uncertainties like disturbances [69, 70, 71]. In the rest of this chapter, we investigate the statistics of $y_i(t)$ when $w_i(t)$ is GWN. It can be proved that, if $w_i(t)$ is Gaussian in (2.2), then $y_i(t)$ in (2.3) is still Gaussian. Thus, the pdf of $y_i(t)$ can be determined by its mean and variance.

Lemma 2.1 (PDF of Node Distribution)**.** Assume that $w_i(t)$ is GWN, the pdf of $y_i(t)$ at time t is a Gaussian distribution with parameters

$$\mathbb{E}\left[y_i(t)\right] = C_i e^{A_i(t-s)} x_i(s), \tag{2.7}$$

$$\mathrm{Cov}\left[y_i(t)\right] = C_i \Theta_{x_i}(t) C_i^T, \tag{2.8}$$

where

$$\Theta_{x_i}(t) = \int_s^t e^{A_i(t-\tau)}\, \mathrm{Cov}[w_i(\tau)]\, e^{A_i^T(t-\tau)}\, \mathrm{d}\tau. \tag{2.9}$$

Proof: See Appendix *A.1*. ■

2.3.2 Dynamic Reference Point

A static reference point can be used when we investigate the interference statistics at a fixed point, like at a fixed base station. However, if we want to analyze the interference statistics of a mobile node in a MWN, e.g., a moving robot or a UAV (see Fig. 2.1(b)), the reference point should be dynamic with possible uncertainties.

We assume the dynamic reference point, denoted as $y_0(t)$, satisfies

$$\dot{x}_0(t) = A_0 x_0(t) + w_0(t) \tag{2.10}$$

$$y_0(t) = C_0 x_0(t), \tag{2.11}$$

and the relative location of node i from this reference point is given by

$$\bar{y}_i(t) := y_i(t) - y_0(t). \tag{2.12}$$

Let $f_y(y_i(t))$ denote the pdf of $y_i(t)$. The following Lemma derives the pdf of $\bar{y}_i(t)$, denoted by $f_y(\bar{y}_i(t))$.

Lemma 2.2 (Node Distribution w.r.t. a Dynamic Reference Point)**.** Assuming the mobility model of nodes and reference point are GLC with GWN, the location of the ith node relative to the dynamic reference point is Gaussian distributed at time t with mean

$$\mathbb{E}\left[\bar{y}_i(t)\right] = C_i e^{A_i(t-s)} x_i(s) - C_0 e^{A_0(t-s)} x_0(s) \tag{2.13}$$

and variance

$$\mathrm{Cov}\left[\bar{y}_i(t)\right] = C_i \Theta_{x_i}(t) C_i^T + C_0 \Theta_{x_0}(t) C_0^T. \tag{2.14}$$

Proof: Lemma 2.2 follows from Lemma 2.1 and (2.12). ■

2.4 Interference Prediction

The main problem of study in this work is to characterize the interference received at a reference point at a future time instant given the interferers' mobility and location information at the current time instant, i.e., interference prediction from the current time into the future. We use the GLC mobility model defined in the previous section to describe the mobility of the interferers and the reference node. Specifically, the quantity $y_i(t)$ $i \in \{0,\ldots,N\}$ in (2.3) or (2.11) is the random variables that describe the locations of nodes (interferers or reference node) at time t with a known initial condition at time s, where t can be viewed as a future time instant and s can be viewed as the current time instant. To make the time-dependency more explicit, we rewrite it as $y_i(t|s)$ in the remainder of this chapter.

2.4.1 Problem Description

Suppose reference node 0 is receiving information from its transmitter. Assume that there are N mobile interferers in this network whose mobilities are modeled by (2.2) and (2.3), and their interference lasts for time duration $[t_0, t_f]$. The aggregate interference on the reference node at time $t \in [t_0, t_f]$, conditioned on knowing the interferers' node dynamics and locations at time s with $s \leq t$, can be defined as

$$I(t|s) = \sum_{i=1}^{N} h_i \, g\big(\|\bar{y}_i(t|s)\|\big), \tag{2.15}$$

where h_i is the multipath fading gain with $\mathbb{E}[h_i] = 1$, which is independent of $\bar{y}_i(t|s)$ and time t. $g(\cdot)$ denotes the path loss function, and $\|\cdot\|$ is the Euclidean norm. For the general case of $s < t$, the quantity $I(t|s)$ represents the interference prediction at a future time instant t based on the information available at the current time instant s, and hence, it is a random variable

due to the uncertainty in the mobility over the time duration from s to t. For the very special case of $s = t$, the quantity $I(t|t)$ represents the actual interference received at time t which is a constant instead of a random variable. In fact, $I(t|t)$ can be viewed as a realization of the random variable $I(t|s)$ for $s < t$. In this chapter, we simply refer to $I(t|s)$ as interference prediction.

We consider the problem of predicting the statistics of interference received by reference node 0 at time t with the available information from current time $s \leq t$. Denote

$$\mathbb{S}\big[I(t|s)\big] = \mathbb{S}\Big[\sum_{i=1}^{N} h_i g\big(\|\overline{y}_i(t|s)\|\big)\Big], \tag{2.16}$$

where $\mathbb{S}[\cdot]$ can be any statistics of the interference prediction, such as the mean value, and $\overline{y}_i(t|s)$ is a conditioned random vector which represents the relative location of interferer i from node 0. In most cases, evaluating $\mathbb{S}\big[I(t|s)\big]$ requires the pdf of $\overline{y}_i(t|s)$, which can be determined by Lemma 2.2.

In the remainder of this section, the mean value and MGF of the interference prediction will be considered in Section 2.4.2 and Section 2.4.3, respectively. In Section 2.4.4, we will propose a compound Gaussian point process functional as a general framework to study these statistics. In Section 2.4.5, the information decay in interference predictions will be discussed, and the prediction has close ties to BPP modelling when time goes to infinity.

2.4.2 Interference Prediction Mean

The mean of the interference prediction is given by the following theorem.

Theorem 2.1 (The Mean of the Interference Prediction)**.** The mean of the interference prediction $\mathbb{E}[I(t|s)]$ is

$$\mathbb{E}\big[I(t|s)\big] = \sum_{i=1}^{N} \int_{\mathbb{R}^d} g\big(\|\overline{y}_i(t|s)\|\big) f_y\left(\overline{y}_i(t|s)\right) \mathrm{d}\big(\overline{y}_i(t|s)\big), \tag{2.17}$$

where $f_y\big(\overline{y}_i(t|s)\big)$ is the pdf of $\overline{y}_i(t|s)$, and can be obtained from Lemma 2.2.

Proof: The mean value of interference prediction is

$$\mathbb{E}\big[I(t|s)\big] = \sum_{i=1}^{N} \mathbb{E}\big[I_i(t|s)\big]. \tag{2.18}$$

According to (2.15), $\mathbb{E}[h_i] = 1$ and the independence of h_i and $\overline{y}_i(t|s)$, we can derive

$$\mathbb{E}\big[I_i(t|s)\big] = \int_{\mathbb{R}^d} g\big(\|\overline{y}_i(t|s)\|\big) f_y\left(\overline{y}_i(t|s)\right) \mathrm{d}\left(\overline{y}_i(t|s)\right). \tag{2.19}$$

Thus, (2.17) can be obtained. ■

The calculation for the mean of the interference prediction will be discussed in Section 2.4.4.

2.4.3 Interference Prediction MGF

Similar to the mean interference, the MGF of the interference prediction can be derived in the following theorem.

Theorem 2.2 (The MGF of the Interference Prediction)**.** The MGF of the interference prediction is

$$\mathbb{E}\left[e^{\beta I(t|s)}\right] = \prod_{i=1}^{N} \int_{\mathbb{R}^d} \mathbb{E}_h\left[e^{\beta h_i g(\|\overline{y}_i(t|s)\|)}\right] f_y\big(\overline{y}_i(t|s)\big)\,\mathrm{d}\big(\overline{y}_i(t|s)\big), \tag{2.20}$$

where $f_y\big(\overline{y}_i(t|s)\big)$ is the pdf of $\overline{y}_i(t|s)$, and can be obtained from Lemma 2.2.

Proof: With (2.15) and independence of h_i and $\overline{y}_i(t)$, Theorem 2.2 can be proved. ■

If the power fading is Nakagami-m ($m = 1$ gives the Rayleigh fading), i.e.,

$$f_h(x) = \frac{m^m x^{(m-1)} e^{-mx}}{\Gamma(m)}, \tag{2.21}$$

then the MGF of the interference prediction can take a more specific form stated as follows.

Corollary 2.1 (The MGF of the interference prediction w.r.t. Nakagami-m Fading)**.** With Nakagami-m Fading (2.21), $\mathbb{E}[e^{\beta I(t|s)}]$ in (2.20) becomes

$$\prod_{i=1}^{N} \int_{\mathbb{R}^d} \Big[\frac{m}{m - \beta g\big(\|\overline{y}_i(t|s)\|\big)}\Big]^m f_y\left(\overline{y}_i(t|s)\right) \mathrm{d}\left(\overline{y}_i(t|s)\right), \tag{2.22}$$

where $\beta\, g(\|\overline{y}_i(t)\|) < 1$.

Remark 2.3. As already discussed, $I(t|s)$ is a random variable that represents the interference prediction at a future time instant t based on the information available at the current time instant s. The uncertainty of the random variable can be computed using its MGF. For instance, one can compute how much the realizations of $I(t|s)$ deviates from its mean value using the variance

$$\mathrm{Var}[I(t|s)] = \mathbb{E}[I^2(t|s)] - \left(\mathbb{E}[I(t|s)]\right)^2, \tag{2.23}$$

where the first and second moments of $I(t|s)$ are used. Since the actual interference received at time t, i.e., $I(t|t)$, is a realization of the interference prediction $I(t|s)$, the variance computed above tells on average how much the actual interference received at time t deviates from the predicted value at time s using the mean prediction.

The calculation for the MGF of the interference prediction is discussed in Section 2.4.4.

Although the expressions of either the mean or MGF of the interference prediction do not usually admit any closed form due to the generality of the GLC mobility model, exceptions are found in some special cases where closed-form expressions are obtained. These results are discussed in Remark 2.5 in the next section.

2.4.4 Compound Gaussian Point Process Functional and its Series Form

In both Section 2.4.2 and Section 2.4.3, the mean and MGF have similar integrals to evaluate. This suggests there may be some general methods to compute these quantities. Here, we define the compound Gaussian point process functional (CGPPF) as a general framework for computing these statistics. Its series form is also given, which will also be useful in analyzing a limiting property of interference prediction in Section 2.4.5.

Definition 2.2 (Compound Gaussian Point Process Functional)**.** The CGPPF is a functional $G\colon \mathcal{V} \to \mathbb{R}$ of the form

$$G[\nu] = \mathbb{E}_y\big[\nu\big(\|y\|\big)\big] = \int_{\mathbb{R}^d} \nu\big(\|y\|\big)\, f_y\big(y\big)\, \mathrm{d}y, \tag{2.24}$$

where $\nu \in \mathcal{V}$ is a Lebesgue Integrable function, and $f_y\big(y\big)$ is a pdf of d-dimensional Gaussian distribution given by

$$f_y\big(y\big) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(y-\mu)^T\Sigma^{-1}(y-\mu)}, \tag{2.25}$$

where $\mu \in \mathbb{R}^d$ and $\Sigma \in \mathbb{R}^{d\times d}$ are mean vector and covariance matrix of location vector y. For example, $\mu = \mathbb{E}\big[\bar{y}_i(t|s)\big]$ and $\Sigma = \mathrm{Cov}\big[\bar{y}_i(t|s)\big]$ are mean and covariance of $\bar{y}_i(t|s)$.

Remark 2.4. For interferer i and dynamic reference point 0, the following are derived: If $\nu(\|\cdot\|) = g(\|\bar{y}_i(t|s)\|)$, then (2.24) returns the mean of the interference prediction from interferer i. If $\nu(\|\cdot\|) = \mathbb{E}_h\left[e^{\beta h_i(t) g(\|\bar{y}_i(t|s)\|)}\right]$, then (2.24) gives the MGF of the interference prediction from interferer i. Note that the path loss function, $g(\cdot)$, can be arbitrary.

Remark 2.5. In most cases, this functional cannot be simplified to a closed-form expression, and numerical integration is needed. Nevertheless, if

$$\mu = 0,\ \Sigma = \mathrm{diag}\{\sigma, \ldots, \sigma\} \tag{2.26}$$

is satisfied, we can get closed-form expressions for the first and second moments of the interference prediction, where $\sigma > 0$ is the std (standard deviation) of all components in location vector y. These expressions are given in Appendix A.4. The usefulness of condition (2.26) will be further discussed in Section 2.4.5.

If the integral in (2.24) exists, the CGPPF can be expanded into a series form, which is the cornerstone for analyzing the limit properties of interference predictions in Section 2.4.5.

Theorem 2.3 (Series Form for Compound Gaussian Point Process Functional)**.** Let P be an orthogonal matrix such that $P^T\mu = \eta = [\eta_i]_{d\times 1}$ and $P^T\Sigma^{-1}P = \mathrm{diag}\{1/\sigma_1^2,\ldots,1/\sigma_d^2\}$, then $G[v]$ can be written as

$$\frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}}\lim_{R\to\infty}\sum_{n=0}^{\infty}\frac{(-1)^n}{2^n n!}\sum_{k_1+k_2+k_3=n}\binom{n}{k_1,k_2,k_3}\Omega\,\Psi[v]. \tag{2.27}$$

The parameters of (2.27) are listed as follows:

$$\Psi[v] = \int_0^R v(r)\, r^{2k_1+k_2+d-1}\mathrm{d}r, \tag{2.28}$$

$$\Omega = \Big[\sum_{a=1}^{d}\Big(\frac{\eta_a}{\sigma_a}\Big)^2\Big]^{k_3}\sum_{\substack{k_1^{(1)}+\cdots+k_1^{(d)}=k_1\\ k_2^{(1)}+\cdots+k_2^{(d)}=k_2}}\binom{k_1}{k_1^{(1)},\ldots,k_1^{(d)}}\binom{k_2}{k_2^{(1)},\ldots,k_2^{(d)}}\Xi, \tag{2.29}$$

$$\Xi = \prod_{\substack{1\le a\le d\\ 1\le b\le d}}\Big(\frac{1}{\sigma_a}\Big)^{2k_1^{(a)}}\Big(-2\frac{\eta_b}{\sigma_b^2}\Big)^{k_2^{(b)}}\int_\Theta(\Phi_a)^{2k_1^{(a)}}(\Phi_b)^{k_2^{(b)}}V(\phi)\,\mathrm{d}\phi, \tag{2.30}$$

where the integration is with respect to the $d-1$ dimensional vector $\phi = \Big[\phi_1,\phi_2,\ldots,\phi_{d-1}\Big]^T$ over the domain $\Theta = \big\{\phi : \phi \in \underbrace{[0,\ \pi]\times\cdots\times[0,\ \pi]}_{d-2}\times[0,\ 2\pi]\big\}$, and

$$[\Phi_1,\ldots,\Phi_d]^T = \Big[\cos\phi_1,\sin\phi_1\cos\phi_2,\ldots,\prod_{p=1}^{d-2}\sin\phi_p\cos\phi_{d-1},\prod_{p=1}^{d-1}\sin\phi_p\Big]^T, \tag{2.31}$$

$$V(\phi)\,\mathrm{d}\phi = \prod_{q=1}^{d-1}(\sin\phi_l)^{d-q-1}\mathrm{d}\phi_q. \tag{2.32}$$

Proof: See Appendix A.2. ■

Remark 2.6. It should be noted that there are two required integrations in Theorem 2.3. For (2.28), it has a closed-form expression when calculating the mean and MGF of the interference predictions with the widely-used path-loss function of the form

$$g\big(\|y\|\big) = \frac{1}{\varepsilon + \|y\|^\alpha} \tag{2.33}$$

where $\varepsilon \ge 0$ (here $\varepsilon = 0$ refers to a singular path loss), and α is the path-loss exponent. The expression of (2.28) is given in Appendix A.3.

To evaluate Ω in (2.29), we need to compute the integral in (2.30), i.e.,

$$\int_{\Theta} (\Phi_a)^{2k_1^{(a)}} (\Phi_b)^{k_2^{(b)}} V(\phi)\, \mathrm{d}\phi, \tag{2.34}$$

the calculation of which is rather complex. However, in the real world, $1 \leq d \leq 3$, and it is relatively easy to deal with. In Appendix A.3, we give the closed-form expression of (2.29) for the cases of $d = 1$ (e.g., a vehicular network on a highway) and $d = 2$ (which is the most common scenario of interest).

2.4.5 Gaussian BPP Approximations for Interference Prediction

The interference prediction method discussed in previous subsections is naturally based on the mobility model of individual interferers. Although prior work on interference analysis for MWNs did not explicitly study the interference prediction problem, one useful idea from them is to use a certain point process to approximate the node distribution in the network from which the time-invariant interference statistics can be derived. For networks where its node distribution (in distant future) can be well approximated using a point process, the information on the initial positions of nodes and node mobilities becomes unnecessary in determining the interference statistics. Clearly, not all mobile networks can have such a nice property. In this subsection, we study the condition under which the interference prediction at a time of far future can be well approximated based on a simple point process. We will see that the series form of the CGPPF will be useful in determining such a condition.

Firstly we give the definition of Gaussian binomial point process (Gaussian BPP).

Definition 2.3 (Gaussian BPP)**.** Let f_y be a pdf of Gaussian distribution with support $\mathbb{R}^d$. A Gaussian BPP with N points on $\mathbb{R}^d$ is a set of i.i.d. random vectors $\{y_1, \ldots, y_N\}$, each with pdf f_y.

Due to the effect of Gaussian white noise, i.e., $w_i(t)$ in (2.2), the uncertainty of our prediction will increase with time, which implies the information available for prediction will decay. In this section, we will focus on the prediction into the far future with N interferers governed by homogenous mobilities. The following theorem gives a necessary and sufficient condition that the statistics of interference predictions can be approximated as those generated by interferers whose locations follow a Gaussian BPP when t becomes large enough.

Theorem 2.4 (Necessary and Sufficient Condition for Gaussian BPP Approximation)**.** Assume that all interferers' mobilities are homogenous, as defined in Definition 2.1, and the integral in (2.24) exists, $\forall \nu \in \mathcal{V}$, the predictions satisfy

$$\lim_{t\to\infty} G_i[\nu] - G_j[\nu] = 0,\ \forall i, j \in \{1, \ldots, N\} \tag{2.35}$$

if and only if

$$\left[\lim_{t\to\infty}\frac{\eta_a^{(i)}}{\sigma_a^{(i)}}-\lim_{t\to\infty}\frac{\eta_a^{(j)}}{\sigma_a^{(j)}}\right]=0,\quad \forall a=\{1,\ldots,d\},\quad \forall i,j\in\{1,\ldots,N\} \tag{2.36}$$

holds, where $\eta_a^{(i)}$ and $\sigma_a^{(i)}$ are defined in Theorem 2.3 with superscripts (i) for ith interferer and

$$G_i[v]=\int_{\mathbb{R}^d} v(\|\bar{y}_i\|)\frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_i|^{\frac{1}{2}}}e^{-\frac{1}{2}(\bar{y}_i-\mu_i)^T\Sigma_i^{-1}(\bar{y}_i-\mu_i)}\mathrm{d}\bar{y}_i. \tag{2.37}$$

In (2.37), μ_i, Σ_i and $\bar{y}_i$ stand for $\mathbb{E}\big[\bar{y}_i(t|s)\big]$, $\mathrm{Cov}\big[\bar{y}_i(t|s)\big]$ and $\bar{y}_i(t|s)$, respectively.

Proof: If the mobilities of all interferers are homogenous, from Lemma 2.2, the covariance matrix $\mathbf{\Sigma}_i$ in (2.37) for all the interferers are the same. Thus, if the integral in (2.24) exists, $\lim_{t\to\infty}\big\{G_i[v]-G_j[v]\big\}$ can be written as

$$\frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_1|^{\frac{1}{2}}}\lim_{R\to\infty}\sum_{n=1}^{\infty}\Big\{\frac{(-1)^n}{2^n n!}\sum_{\substack{k_1+k_2+k_3=n\\ k_3\neq 0}}\binom{n}{k_1,k_2,k_3}\lim_{t\to\infty}(\Omega_i-\Omega_j)\Psi[v]\Big\}. \tag{2.38}$$

Obviously, formula (2.38) equals 0 if and only if (2.35) holds.

Necessity. By the contrapositive, if the (2.36) does not hold, $\lim_{t\to\infty}(\Omega_i-\Omega_j)\neq 0$ (Otherwise, the quadratic forms in (A.7) will be the same for i and j, which implies (2.36) is satisfied). As a result, the (2.38) does not equal 0. Therefore, the contrapositive is established, and the necessity of (2.36) is proved.

Sufficiency. If (2.36) is satisfied, then the (2.38) equals 0, so as for (2.35). Thus the sufficiency of (2.36) is established. ■

Remark 2.7. Theorem 2.4 tells us the interference predictions, from interferers with homogenous mobility asymptotically converge to the predictions from a Gaussian BPP. Nevertheless, if (2.36) does not hold, we cannot use the BPP approximation. In Section 2.5, we will see examples in both scenarios. It should be noted that $\eta_a^{(i)}$ and $\sigma_a^{(i)}$ in (2.36) can be easily calculated for a given GLC mobility model (see Section 2.5 for examples).

Remark 2.8. Gaussian BPP approximation implies that the mean and MGF of the interference prediction become

$$\mathbb{E}\big[I(t|s)\big]\approx N G_i\big[g\big(\|\bar{y}_i(t|s)\|\big)\big],\quad \forall i \tag{2.39}$$

$$\mathbb{E}\big[e^{\beta I(t|s)}\big]\approx G_i\Big[\Big(\frac{m}{m-\beta g(\|\bar{y}_i(t|s)\|)}\Big)^m\Big]^N,\quad \forall i, \tag{2.40}$$

when $t\gg s$. Therefore, we can calculate the mean or MGF of the interference prediction using

just one (arbitrary) interferer's information instead of all interferers, and the computation is significantly simplified. Note that (2.39) and (2.40) change with time t, because the distribution of the predicted location $\overline{y}(t|s)$ varies with t.

Corollary 2.2 (Necessary and Sufficient Condition for Gaussian BPP Approximation with $\mu = 0$)**.** Assume that all interferers' mobilities are homogenous and the integral in (2.24) exists, $\forall v \in \mathcal{V}$, the predictions satisfy

$$\lim_{t\to\infty} G_i[v] - G_o[v] = 0,\ \forall i \in \{1,\ldots,N\} \tag{2.41}$$

if and only if

$$\lim_{t\to\infty} \frac{\eta_a^{(i)}}{\sigma_a^{(i)}} = 0,\ \forall a \in \{1,\ldots,d\},\ \forall i \in \{1,\ldots,N\} \tag{2.42}$$

holds, and

$$G_o[v] = \int_{\mathbb{R}^d} v(\|y\|) \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}y^T\Sigma^{-1}y}\mathrm{d}y, \tag{2.43}$$

where $\mu = \mu_1 = \ldots = \mu_N$ ($\mu_i = \mathbb{E}\left[\overline{y}_i(t|s)\right]$) and $\Sigma = \Sigma_1 = \ldots = \Sigma_N$ ($\Sigma_i = \mathrm{Cov}[\overline{y}_i(t|s)]$).

Remark 2.9. Corollary 2.2 implies that we can use the Gaussian BPP with $\mu = 0$ to approximate the mean or MGF of the interference prediction when condition (2.42) holds. It should be noted that it is easy to test condition (2.42) by calculating $\eta_a^{(i)}$ and $\sigma_a^{(i)}$ for a given GLC mobility model. Furthermore, if $\Sigma = \mathrm{diag}\{\sigma,\ldots,\sigma\}$ also holds, we can use the result stated in Remark 2.5 to obtain closed-form expressions for first and second moments of interference prediction given in Appendix A.4. An example of mobility model that has such nice properties is Brownian motion.

2.5 Simulation Examples

In order to corroborate our theoretical results, simulations are presented to illustrate the effectiveness of interference prediction. We consider three different mobile networks with the mobility model given by: (1) the basic 2-dimensional Brownian motion, (2) 2-dimensional Brownian motion with inertia in the velocity and (3) UCM in 3-dimensional space. The basic Brownian motion is a simple and commonly-used mobility model while Brownian motion with inertia is a new mobility model that has not been considered in the literature. We will study and compare the interference prediction results for these two examples. Specifically, we will see that the existence of inertia completely changes the limiting behavior of the interference prediction. After that, we move from 2-dimensional examples to a 3-dimensional example of

UCM which can represent the scenario of UAV target scanning similar to that in Section II. Note that all three mobility models are special cases of the GLC mobility model developed in this work.

To make predictions, the parameters for path loss function (e.g., the path-loss exponent α) and multipath fading (e.g., the value of m for the Nakagami-m fading channel) are necessary. In addition the following parameters are assumed to be known at time s:

- For 2D Brownian motion (Section 2.5.1), the location for each node.
- For 2D Brownian motion with inertia (Section 2.5.2), the location and velocity for each node.
- For UCM (Section 2.5.3), the location and angular velocity for each node.

Note that the node location and velocity at only one time instant (not frequently updated) can be obtained in many practical scenarios [72, 73, 74]. For mobile users (Section 2.5.1), their locations can be updated by GPS [72]. For vehicles (Section 2.5.2) or UAVs (Section 2.5.3), their locations or velocities can be obtained by Differential GPS (DGPS), e.g., [73, 74], and the angular velocity can be calculated by the node location, node velocity and the center location of UCM. In our simulations, all the units are normalized.

2.5.1 2D Brownian Motion

We consider a network having 7 nodes with mobility governed by Brownian motion, which can be used to describe human mobility [75]. As a special case of our GLC mobility model, 2D Brownian motion has the parameters

$$A_i = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad C_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{2.44}$$

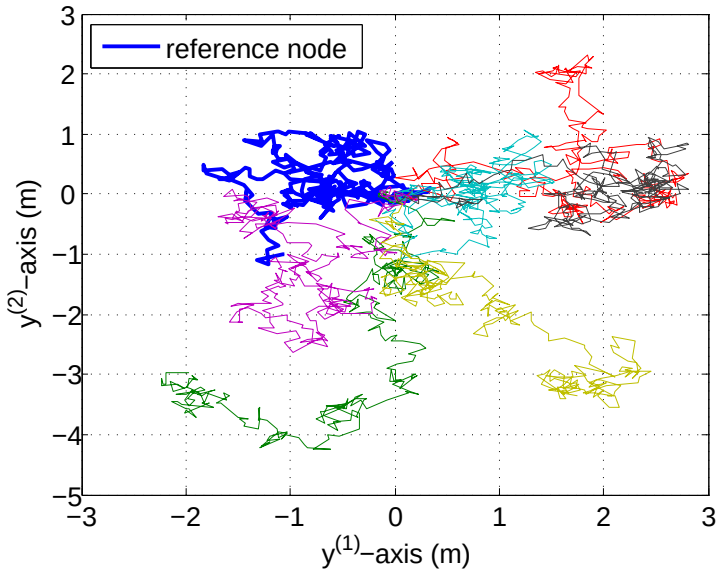
where $i = 0, 1, \ldots, 6$ (that is, 6 interferers and 1 reference point). Note that the reference node also does Brownian motion. Furthermore, all the nodes (including the reference node) start moving from the origin at time $t_0 = 0$, i.e., $y_i(0) = x_i(0) = 0$.

The uncertainty w_i follows

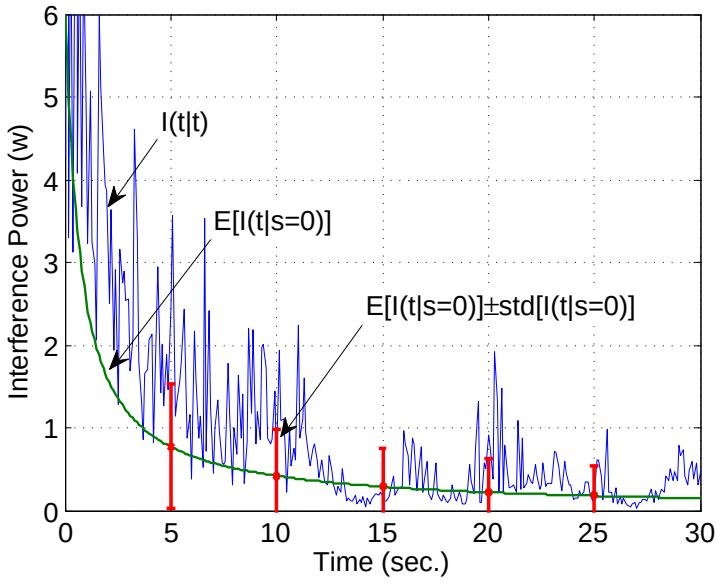
$$w_i(t) = \left[w_i^{(1)}(t), w_i^{(2)}(t)\right]^T, \tag{2.45}$$

where all $w_i(t)$ for all i are identical GWNs with unit power. A realization of Brownian motion is shown in Fig. 2.2(a).

Since the 7 nodes walk in an urban area, the path loss exponent $\alpha = 4$ and the fading severity parameter $m = 2$ are chosen. Considering $\varepsilon = 1$, we can predict the mean and standard deviation of the aggregate interference at node 0 from $s = 0$, i.e., $\mathbb{E}[I(t|s=0)]$ and std$[I(t|s=$



(a)



(b)

Figure 2.2: (a) Node motions. (b) Aggregate interference at reference node and the corresponding predictions. $\mathbb{E}[I(t|s=0)]$ stands for the mean value of interference prediction based on information available at $s=0$, and $\mathrm{std}\,[I(t|s=0)]$ is the standard deviation of prediction. They are both dependent on time t and s. Each error bar $\mathbb{E}[I(t|s=0)] \pm \mathrm{std}\,[I(t|s=0)]$ is symmetric about $\mathbb{E}[I(t|s=0)]$, and the negative part of $\mathbb{E}[I(t|s=0)] - \mathrm{std}\,[I(t|s=0)]$ is omitted.

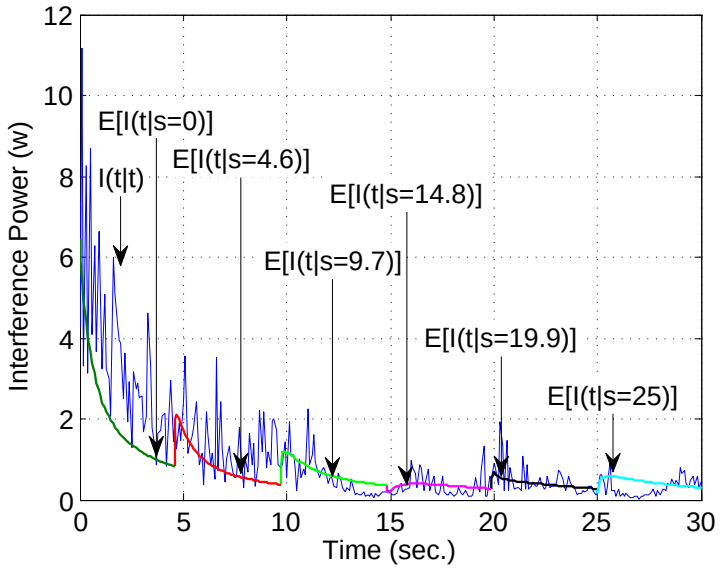


Figure 2.3: Predictions on the mean of aggregate interference based on frequently updated location information at time $s = 0, 4.6, 9.7, 14.8, 19.9, 25$.

$0)]$, see Fig. 2.2(b). The $\mathrm{std}[I(t|s=0)]$ measures how much the realization of $I(t|s=0)$, i.e., $I(t|t)$, deviates from its predicted mean value, and thus $\mathrm{std}[I(t|s=0)]$ reveals the uncertainty of interference prediction. In this case, both $\mathbb{E}[I(t|s=0)]$ and $\mathrm{std}[I(t|s=0)]$ have closed-form expressions (because (2.26) in Remark 2.5 is satisfied), which can be derived by calculating $\mathbb{E}[I_i(t|s=0)]$ and $\mathrm{std}[I_i(t|s=0)]$, $\forall i$, from (A.22) and (A.23) in Appendix A.4.

From Fig. 2.2(b) we can see that the mean of the interference prediction does not perform well, and the Coefficient of Variation (CV) $\mathrm{std}[I_i(t|s=0)]/\mathbb{E}[I_i(t|s=0)]$, i.e., normalized standard deviation, increases with t. This is due to the fact that the uncertainties $w_i(t)$ in velocities $\bar{x}_i(t)$ dominate the behavior of the mobilities of nodes. In Fig. 2.2(a) the node locations are far away from the origin, while prediction tells us $\mathbb{E}[y_i(t|s=0)] = y_i(0) = 0$ [see Fig. 2.2(a)].

In Fig. 2.2(b), $s = 0$ means the information on node locations is updated at time 0. If we want a better prediction for near future, the information on node locations need to be updated more frequently. Interference prediction is then done with updated location information until the next update (see Fig. 2.3).

We end this subsection with a discussion on the Gaussian BPP approximations for these predictions in Fig. 2.3. Actually, these predictions can be approximated by a Gaussian BPP at origin. This is because the condition (2.42) in Corollary 2.2 is satisfied.

It should be noted that $\eta_a^{(i)}$ and $\sigma_a^{(i)}$ can be easily calculated to test the condition (2.42): By Lemma 2.2, the mean and variance of the location prediction at t from s for ith node are

$$\mu = y_i(s) - y_0(s), \quad \Sigma = \mathrm{diag}\left\{2(t-s), 2(t-s)\right\}, \tag{2.46}$$

where $y_i(s) - y_0(s)$ is finite. As Σ is naturally a diagonal matrix (which means the orthogonal transformation $P^T\Sigma^{-1}P = \text{diag}\{1/\sigma_1^2, \ldots, 1/\sigma_d^2\}$ is not required), we have

$$\eta_1 = \mu_1 = y_i^{(1)}(s) - y_0^{(1)}(s), \quad \eta_2 = \mu_2 = y_i^{(2)}(s) - y_0^{(2)}(s), \sigma_1 = \sigma_2 = \sqrt{2(t-s)}. \quad (2.47)$$

Thus, the condition (2.36) in Theorem 2.4 is satisfied, which implies the predictions made from different s in Fig. 2.3 can be approximated by the corresponding prediction from the Gaussian BPP with 6 nodes whose location pdf has $\mu = 0$ when $t-s$ is large enough. The approximation has the following closed-form expression

$$\mathbb{E}[I(t-s|s)] = 6\,\mathbb{E}[I_i(t-s|s)] \approx \frac{6\text{Ci}\left(\frac{\sqrt{\varepsilon}}{2\sigma^2}\right)\sin\frac{\sqrt{\varepsilon}}{2\sigma^2} + 3\cos\frac{\sqrt{\varepsilon}}{2\sigma^2}\left[\pi - 2\text{Si}\left(\frac{\sqrt{\varepsilon}}{2\sigma^2}\right)\right]}{2\sqrt{\varepsilon}\sigma^2}, \quad (2.48)$$

where $\mathbb{E}[I_i(t-s|s)]$ is derived from (A.22) in Appendix A.4, and $\sigma = \sigma_1 = \sigma_2 = \sqrt{2(t-s)}$. Since $\mathbb{E}[I_i(t-s|s)]$ is time-invariant, it is exactly the $\mathbb{E}[I(t|s=0)]$. Therefore, if we translate the time-axis by $s-t$, these prediction $\mathbb{E}[I(t|s)]$ ($s = 4.6, 9.7, 14.8, 19.9, 25$) will asymptotically converge to $\mathbb{E}[I(t|s=0)]$ as $t-s$ increases (see Table 2.1).

Table 2.1: Convergence Behavior for the Mean of the Interference Prediction in Networks with Brownian Motion Mobility models

	$t-s=10$	$t-s=50$	$t-s=100$	$t-s=500$
$\mathbb{E}[I(t\|s=0)]$	0.2201	0.0463	0.0233	0.0047
$\mathbb{E}[I(t\|s=4.6)]$	0.2115	0.0459	0.0232	0.0047
$\mathbb{E}[I(t\|s=9.7)]$	0.2032	0.0455	0.0231	0.0047
$\mathbb{E}[I(t\|s=14.8)]$	0.1772	0.0442	0.0228	0.0047
$\mathbb{E}[I(t\|s=19.9)]$	0.1768	0.0441	0.0228	0.0047
$\mathbb{E}[I(t\|s=25)]$	0.1850	0.0446	0.0229	0.0047

2.5.2 2D Brownian Motion with Inertia

The basic Brownian motion in Section 2.5.1 is dominated by velocity uncertainty, thus our predictions cannot be expected to offer great accuracy. In this section, we focus on the Brownian motion with velocity inertia in 2D space, and it can be employed to describe the motion for vehicle or pedestrian with destination.

The mobility model has the parameters

$$A_i = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad C_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad (2.49)$$

where $i = 0, 1, \ldots, 6$, and with the initial condition

$$x_i(0) = \left[0, x_i^{(2)}(0), 0, x_i^{(4)}(0)\right]^T, \tag{2.50}$$

where $x_i^{(1)}(0) = 0$ and $x_i^{(3)}(0) = 0$. Note that the ith node location is given by $y_i(0) = [x_i^{(1)}(0), x_i^{(3)}(0)]^T$. The non-zero states, $x_i^{(2)}(0)$ and $x_i^{(4)}(0)$, which represent the ith node velocity components (i.e., $\dot{y}_i(0) = [x_i^{(2)}(0), x_i^{(4)}(0)]^T$), are randomly generated in $[-1, 1]$ and assumed to be known. The w_i follows

$$w_i(t) = \left[w_i^{(1)}(t), 0, w_i^{(3)}(t), 0\right]^T, \tag{2.51}$$

where all $w_i^{(1)}(t)$ are identical GWN with unit power, which implies the node velocities are fluctuating due to these uncertainties. The parameters of the path loss function and multipath fading are the same as those in Section 2.5.1 (i.e., $\varepsilon = 1$, $\alpha = 4$, $m = 2$). A realization for Brownian motion with inertia is shown in Fig. 2.4(a). The mean and standard deviation of interference prediction from $s = 1.8$ are shown in Fig. 2.4(b).

From Fig. 2.4(b) we can see that the prediction performs much better than that for Brownian motion without inertia, and the standard deviation is smaller than that in Brownian motion without inertia for the same interference level. It also should be noted that, similar to the Brownian motion without inertia, the coefficient of variance (CV) becomes larger with increasing t.

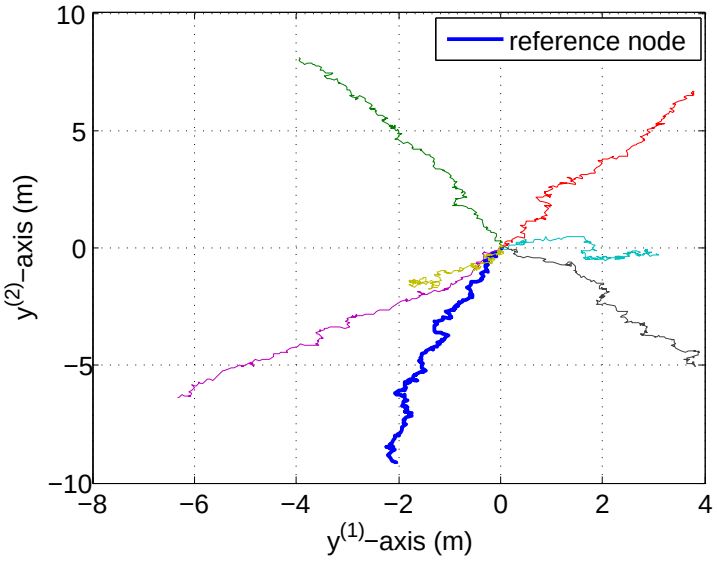
In terms of the limiting behavior of the prediction, unfortunately, it cannot be approximated by the prediction from a Gaussian BPP in Theorem 2.4 no matter how large t is. It can be calculated that

$$\begin{aligned} \eta_1 &= x_i^{(1)}(s) - x_0^{(1)}(s) + \left[x_i^{(2)}(s) - x_0^{(2)}(s)\right] \cdot (t-s) \\ \eta_2 &= x_i^{(3)}(s) - x_0^{(3)}(s) + \left[x_i^{(4)}(s) - x_0^{(4)}(s)\right] \cdot (t-s) \\ \sigma_1 &= \sigma_2 = \sqrt{2(t-s)}. \end{aligned} \tag{2.52}$$

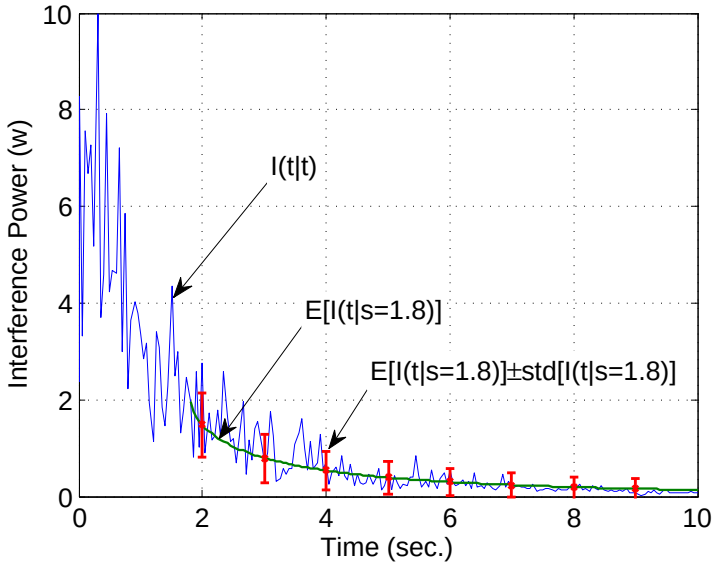
Because $\left[x_i^{(2)}(s) - x_0^{(2)}(s)\right] \neq \left[x_i^{(4)}(s) - x_0^{(4)}(s)\right]$ at $s = 1.8$, condition (2.36) in Theorem 2.4 is not satisfied. The prediction based on the actual mobility model and that based on BPP approximation are shown in Table 2.2, and it turns out that there is a big difference between them.

2.5.3 Uniform Circular Motion in 3D Space

In this section, we revisit the UAV target scanning problem discussed in Section 2.2. In the real world, UAVs that execute the scanning task seldom hover in the same 2D plane (their flight altitudes differ) for collision avoidance. Thus, UCM should be modeled in 3D rather than 2D.



(a)



(b)

Figure 2.4: (a) Node motions. (b) Aggregate interference at reference node and the mean of the interference prediction.

Table 2.2: Comparison of the Mean of the Interference Prediction in Networks with BPP Approximation

	$t-s=10$	$t-s=50$	$t-s=100$	$t-s=500$
$\mathbb{E}[I(t\|s=1.8)]$	2.679×10^{-2}	1.837×10^{-4}	3.025×10^{-5}	8.665×10^{-7}
BPP Approximation	0.2201	0.0463	0.0233	0.0047

Assume there is one receiver (reference node, UAV 0) and two interferers (UAVs 1 and 2), whose parameters of mobility models are

$$\begin{aligned} A_0 = A_2 &= \begin{bmatrix} 0 & -0.1 & 0 \\ 0.1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad C_0 = C_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ A_1 &= \begin{bmatrix} 0 & 0.1 & 0 \\ -0.1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad C_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \end{aligned} \tag{2.53}$$

which implies the hover angular velocities for UAVs 0 and 2 are both -0.1rad/s, and for node 1 is 0.1rad/s. The initial locations $y_i(0) = x_i(0)$, $i = 0,1,2$ are

$$y_0(0) = \begin{bmatrix} 500 \\ 500 \\ 500 \end{bmatrix}, \; y_1(0) = \begin{bmatrix} -400 \\ -300 \\ 700 \end{bmatrix}, \; y_0(0) = \begin{bmatrix} 400 \\ 0 \\ 300 \end{bmatrix}, \tag{2.54}$$

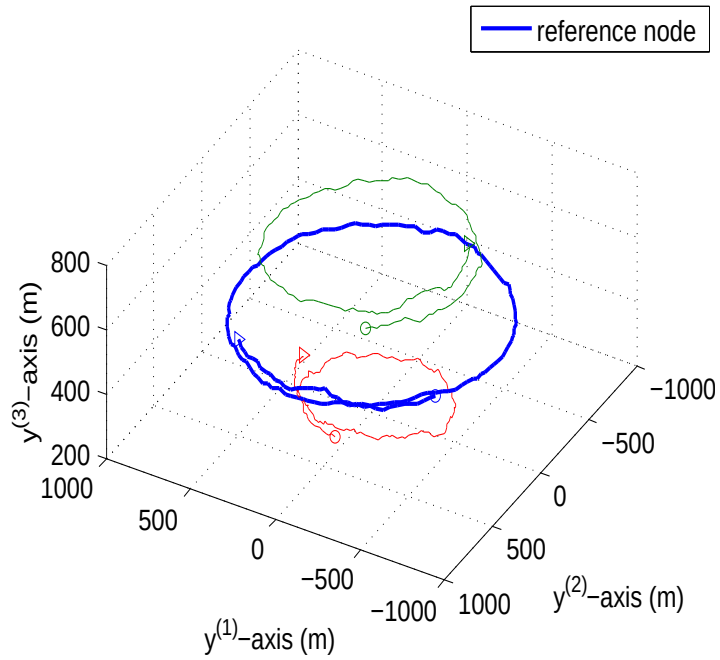
where $y_0^{(3)}(0) = 500$, $y_1^{(3)}(0) = 700$ and $y_2^{(3)}(0) = 300$ are the initial altitudes for UAVs 0, 1, and 2. The w_i follows

$$w_i(t) = \left[w_i^{(1)}(t), w_i^{(2)}(t), w_i^{(3)}(t)\right]^T, \tag{2.55}$$

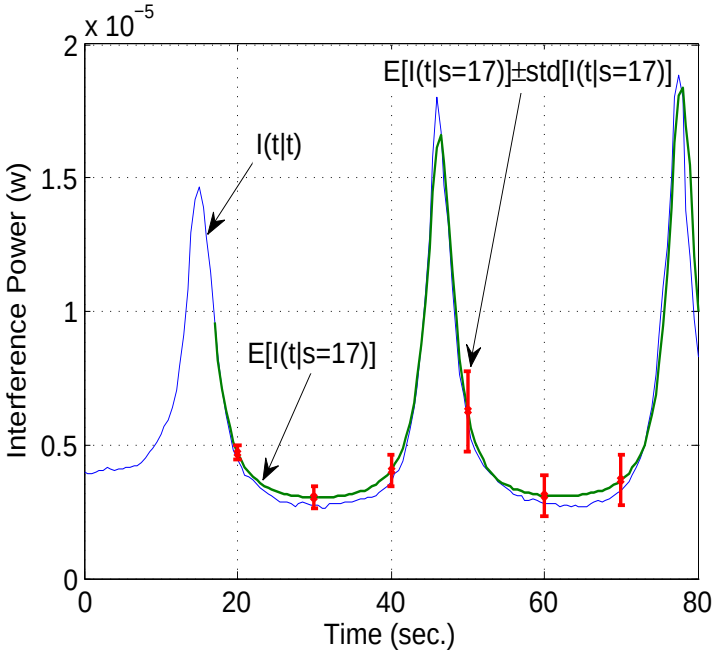
where all $w_i^{(j)}(t)$, $j = 1,2,3$ are identical GWN with power $\sigma^2 = 100$. The parameters of path loss function are the same as those in Sections 2.5.1 and 2.5.2 (i.e., $\varepsilon = 1$, $\alpha = 2$). Because there are few obstacles in airspace, we assume that there is no multipath fading. The UAVs' motion-trajectories are shown in Fig. 2.5(a). The aggregate interference and $\mathbb{E}\left[I(t|s = 17)\right]$ are shown in Fig. 2.5(b). We see that the interference prediction is very close to the actual interference. It is interesting to see that the CV is not always increasing with t. However, if we consider the interference at the same level, the CV does increase with time, e.g., $t = 30$ and $t = 60$.

2.6 Summary

In this chapter, the interference prediction problem in MWNs has been investigated. The GLC mobility model has been proposed to describe or approximate a large class of mobilities in the real world. The statistics of interference prediction with respect to a dynamic reference point have been analyzed, including mean value and the MGF. We have defined the CGPPF as a general framework to compute the mean and MGF, and discussed important special cases where closed-form expressions can be obtained. By expressing the CGPPF in series form, we have analyzed the limiting behavior of the statistics of the interference prediction, and given the necessary and sufficient for when the node locations can be regarded as a Gaussian BPP when analyzing the statistics of interference prediction.



(a)



(b)

Figure 2.5: (a) UAV locations versus time t, starting from the triangular marks and ending at the circular marks. (b) Aggregate interference at reference node and the mean of the interference prediction.

Chapter 3

Cooperative Localization with Asynchronous Measurements and Communications

3.1 Introduction

In Chapter 2, the location-based interference prediction has been studied, where the current nodal locations information are perfectly known at the time when the prediction is made. However, even if the nodal locations are known at some points in time, the uncertainty in nodal locations increases due to node mobility, which results in loss of accuracy in the next interference prediction. To solve this problem, localization methods should be considered to reduce the location uncertainty in a real-time manner. Note that the real-time localization is very challenging, since the the measurements should be updated timely.

This chapter studies the cooperative localization problem for mobile nodes, where the communications and measurements can arrive in a completely asynchronous way. Different from the previous works which highly rely on the synchronized time-slotted systems, the cooperative localization framework we establish does not need any synchronization for the communication links and measurement processes in the entire wireless networks. The mobility of nodes is modeled by the stochastic differential equation, which enables each node to utilize the asynchronously received messages and measurements in the continuous-time domain. To solve the cooperative localization problem, we first propose the centralized localization algorithm based on the global information as the benchmark. Then, we rigorously prove when a localization estimation with partial information has a small performance gap from the one with global information. Finally, by applying this result at each node, the prior-cut algorithm is designed to solve this asynchronous localization problem. Two cooperative localization case studies (for mobile users and unmanned aerial vehicles, respectively) are presented to corroborate the effectiveness of the prior-cut algorithm in dealing with asynchronous communications and measurements.

This chapter is organized as follows. In Section 3.2, the system model (including node mobility model and information exchanging model) is given, where the measurements and communications are asynchronous. Then the cooperative localization framework is provided and the cooperative localization problem is formulated in Section 3.3. In Section 3.4, we propose and analyze the centralized localization algorithm. Base on the analysis in Section 3.4, we design the distributed prior-cut algorithm in Section 3.5. In Section 3.6, simulation results are shown to illustrate the effectiveness of the distributed prior-cut algorithm, where two cooperative localization case studies for the mobile users and UAVs are presented, respectively. Finally, the concluding remarks are given in Section 3.7.

3.2 System Model

Consider a wireless network with I nodes. For each node $i \in \{1,\ldots,I\} =: \mathcal{I}$, its location trajectory $y_i(t)$ follows a stochastic process $\{y_i(t)\}_{t\in\mathcal{T}}$ ($\mathcal{T} := \overline{\mathbb{R}}_+$) described by a state-space model:

$$\mathrm{d}x_i(t) = f_i(x_i(t),t)\mathrm{d}t + \mathrm{d}\beta_i(t), \tag{3.1}$$

$$y_i(t) = g_i(x_i(t)), \tag{3.2}$$

where (3.1) and (3.2) are called the state equation and the location equation, respectively. The state equation is a stochastic differential equation, and $t \in \mathcal{T}$ denotes the global time. $x_i(t) \in \mathcal{X}_i = \mathbb{R}^{n_i}$ stands for the location-related state (or system state) of node i, which can contain the location, velocity, acceleration, etc. We assume the initial condition $x_i(0)$ for different $i \in \mathcal{I}$ are mutually independent. The nonlinear map $f_i : \mathcal{X}_i \times \mathcal{T} \to \mathcal{X}_i$ is the system function, and it captures the key feature for how the system state evolves with time (a simple example is given in Remark 3.1). $\{\beta_i(t)\}_{t\in\mathcal{T}}$ is a n_i-dimensional Brownian motion with diffusion matrix $\bar{Q}_i \in \mathbb{R}^{n_i\times n_i}$,[1] which describes the additive noise for state evolutions. We assume $\{\beta_i(t)\}_{t\in\mathcal{T}}$ ($i \in \mathcal{I}$) are mutually independent. The location equation determines the nodal location $y_i(t) \in \mathcal{Y}_i \subseteq \mathbb{R}^d$ ($d \in \{1,2,3\}$), where $g_i : \mathcal{X}_i \to \mathcal{Y}_i$ is the location function, which reflects the relationship between the state and location. Note that if $x_i(t)$ refers to the location of the node i, then $g_i(x_i(t)) = x_i(t)$ is the identity map.

Remark 3.1. A simple example for the mobility model described by (3.1) and (3.2) is a 1-dimensional Brownian motion, where $x_i(t)$ gives the location of node i, and

$$f_i(x_i(t),t) = 0, \quad g_i(x_i(t)) = x_i(t), \quad \bar{Q}_i = 1, \quad i \in \mathcal{I}. \tag{3.3}$$

The initial condition is $x_i(0) \sim \mathrm{N}(0,1)$, i.e., it follows the standard Gaussian distribution. Thus,

[1] That means $\mathrm{d}\beta_i(t)$ satisfies $\mathbb{E}[\mathrm{d}\beta_i(t)\mathrm{d}\beta_i^T(t)] = \bar{Q}_i\mathrm{d}t$.

at each time t, the location $y_i(t)$ is a realization of $y_i(t) \sim \mathrm{N}(0, t+1)$. Note that the Brownian motion is a commonly used model for describing human mobilities in wireless networks (see [35]). Our mobility model not only embraces the standard Brownian motion as a special case but also describes more complicated and realistic mobilities [64], e.g., for UAVs in Section 3.6.2.

For localization problems, location $y_i(t)$ is usually not obtainable by node i itself, except for anchor nodes $i \in \mathcal{A}$, where $\mathcal{A}$ is called the anchor set. For the other nodes $i \in \mathcal{I} \setminus \mathcal{A} =: \mathcal{L}$, they are unable to get their locations $y_i(t)$, and we call them non-anchor nodes. Without loss of generality, we set $\mathcal{L} = \{1, \ldots, L\}$ and $\mathcal{A} = \{L+1, \ldots, I\}$. We assume that anchor node $a \in \mathcal{A}$ can accurately get its true state and location at any given time $t \in \mathcal{T}$, which are labeled as $x_a^*(t)$ and $y_a^*(t) = g_a(x_a^*(t))$, respectively. Note that for each anchor node $a \in \mathcal{A}$, its state $x_a(t)$ of (3.1) is degenerated to a deterministic vector; but for each non-anchor node $l \in \mathcal{I} \setminus \mathcal{A} =: \mathcal{L}$, it does not know its exact state or location by itself, and the only way to estimate its own location is to cooperate with other nodes.

In this work, the cooperation is based on exchanging two kinds of information: the direct information (see Section 3.2.1) and the indirect information (see Section 3.2.2).

3.2.1 Direct information

The direct information exchanged between nodes is the distance-based measurements directly taken by a node. For example, node i can take a measurement of the distance between node j and node i by using the TOA or TDOA estimation. The detailed description is given as follows.

At time $t_{i,j,k}^{\mathrm{ms}} \in \{t_{i,j,k}^{\mathrm{ms}} \in \mathcal{T} \setminus \{0\} \colon k \in \mathcal{K}_{i,j}^{\mathrm{ms}} \subseteq \mathbb{Z}_+\} =: \mathcal{T}_{i,j}^{\mathrm{ms}}$ $(j \in \mathcal{I})$, node $i \in \mathcal{I}$ measures the distance between nodes j and i via

$$z_{i,j}(t_{i,j,k}^{\mathrm{ms}}) = \|y_i(t_{i,j,k}^{\mathrm{ms}}) - y_j(t_{i,j,k}^{\mathrm{ms}})\| + v_{i,j}(t_{i,j,k}^{\mathrm{ms}}). \tag{3.4}$$

Equation (3.4) is called the measurement equation, and $z_{i,j}(t_{i,j,k}^{\mathrm{ms}}) \in \mathcal{Z}_{i,j} \subseteq \mathbb{R}$ is called the measurement $i \leftarrow j$ at time $t_{i,j,k}^{\mathrm{ms}}$. Random variable $v_{i,j}(t_{i,j,k}^{\mathrm{ms}})$ represents the measurement noise and the process $\{v_{i,j}(t_{i,j,k}^{\mathrm{ms}})\}_{k \in \mathcal{K}_{i,j}^{\mathrm{ms}}}$ is white[2]. The measurement equation in (3.4) is a general model for a variety of distance-based measurement methods. The specific measurement method is not the focus of this work. In this chapter, we assume that $x_i(0)$, $\{\beta_i(t)\}_{t \in \mathcal{T}}$, and $\{v_{i,j}(t_{i,j,k}^{\mathrm{ms}})\}_{k \in \mathcal{K}_{i,j}^{\mathrm{ms}}}$ are mutually independent. We can see that $t_{i,j,k}^{\mathrm{ms}}$ for different node i are not required to be synchronous.

The assumption of taking internodal measurements is given as follows.

Assumption 3.1. No measurements are taken between any two anchor nodes.

[2]It means $v_{i,j}(t_{i,j,k}^{\mathrm{ms}})$ $(t_{i,j,k}^{\mathrm{ms}} \in \mathcal{T}_{i,j}^{\mathrm{ms}})$ are mutually independent and have zero mean.

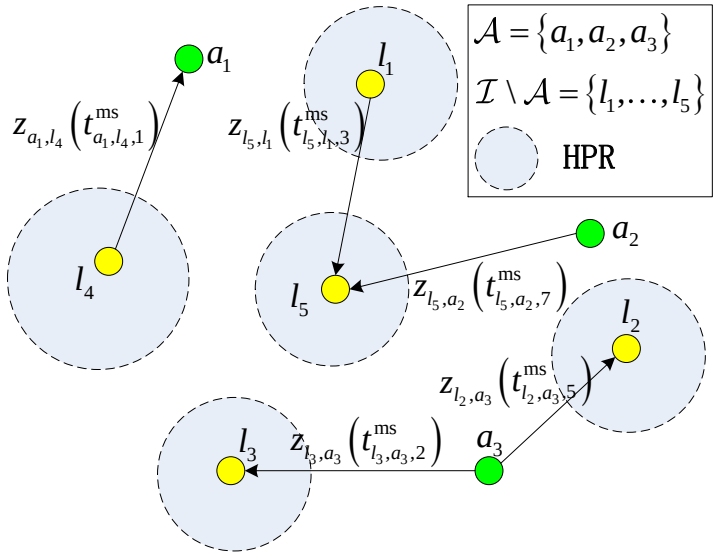


Figure 3.1: Illustrations of direct information. There are 8 nodes in this network, where points a_1, a_2, a_3 stand for the anchor nodes, and points $l_1, \ldots, l_5$ denote the non-anchor nodes. The shaded circles with dashed boundary are high probability regions (HPRs) for nodes $l_1, \ldots, l_5$, where an HPR for a node is the region that the true location falls in with high probability. For anchor nodes a_1, a_2, a_3, they know their own true locations. The straight arrows $l_1 \to l_5$, $a_2 \to l_5$, $l_4 \to a_1$, $a_3 \to l_3$, and $a_3 \to l_2$ represent the measurements $z_{l_5,l_1}(t^{\mathrm{ms}}_{l_5,l_1,3})$, $z_{l_5,a_2}(t^{\mathrm{ms}}_{l_5,a_2,7})$, $z_{a_1,l_4}(t^{\mathrm{ms}}_{a_1,l_4,1})$, $z_{l_3,a_3}(t^{\mathrm{ms}}_{l_3,a_3,2})$, and $z_{l_2,a_3}(t^{\mathrm{ms}}_{l_2,a_3,5})$ at time instants $t^{\mathrm{ms}}_{l_5,l_1,3}$, $t^{\mathrm{ms}}_{l_5,a_2,7}$, $t^{\mathrm{ms}}_{a_1,l_4,1}$, $t^{\mathrm{ms}}_{l_4,l_3,1}$, $t^{\mathrm{ms}}_{l_3,a_3,2}$, and $t^{\mathrm{ms}}_{l_2,a_3,5}$, respectively.

The measurements can be taken between two non-anchor nodes, or one non-anchor node and one anchor node. However, for anchor nodes, since the states and the locations are exactly known, there is no need to take measurements between anchor nodes. Thus, Assumption 3.1 is reasonable.

An example for measurements is given in Fig. 3.1.

For the internodal measurements, we provide some notations for the convenience of discussion in the rest of the chapter. At time t, the set of all the measurements $i \leftarrow j$ during $[0,\ t]$ is labeled as $Z_{i,j}[t] := \{z_{i,j}(t^{\mathrm{ms}}_{i,j,k}) \colon t^{\mathrm{ms}}_{i,j,k} \in \mathcal{T}^{\mathrm{ms}}_{i,j}[t]\}$, where $\mathcal{T}^{\mathrm{ms}}_{i,j}[t] := \mathcal{T}^{\mathrm{ms}}_{i,j} \bigcap [0,\ t]$, i.e., $\mathcal{T}^{\mathrm{ms}}_{i,j}[t]$ is the set of measurement time instants (in $\mathcal{T}^{\mathrm{ms}}_{i,j}$) within time window $[0,\ t]$. We define $Z_i[t] =: \bigcup_{j\in\mathcal{I}} Z_{i,j}[t]$ which stands for the set of all the measurements $i \leftarrow \cdot$, i.e., taken by node i, during $[0,\ t]$. Furthermore, we define $Z[t] := \bigcup_{i\in\mathcal{I}} Z_i[t]$ to represent the set of all the measurements in the whole network within time window $[0,\ t]$.

3.2.2 Indirect information

The indirect information refers to the messages exchanged between nodes. A message can include the measurements and other information related to the nodal locations. For example, after node i obtaining direct information $z_{i,j}(t^{\mathrm{ms}}_{i,j,k})$, it can transmit this measurement as a mes-

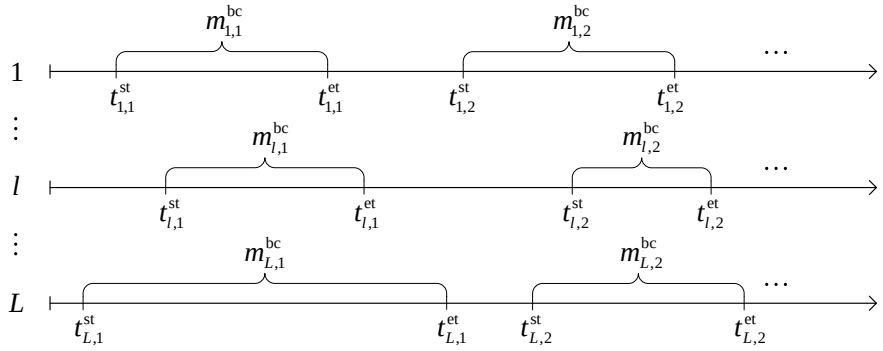


Figure 3.2: Illustrations of broadcasting time.

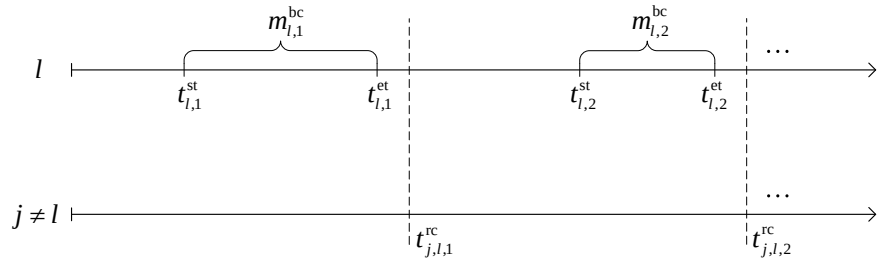


Figure 3.3: Illustrations of receiving time.

sage (indirect information) to other nodes. Node i can include its location information $y_i(t^{\mathrm{ms}}_{i,j,k})$ as well to the message to help other nodes' localization.

In this work, each non-anchor node $l \in \mathcal{L}$ transmits messages to other nodes by broadcasting, where the broadcasting time instants are illustrated in Fig. 3.2. In this figure, $m^{\mathrm{bc}}_{l,k}$ refers to message $k \in \mathcal{K}^{\mathrm{bc}}_l$ broadcasted by node l, and the broadcast starts from $t^{\mathrm{st}}_{l,k}$ and ends at $t^{\mathrm{et}}_{l,k}$. When receiving the broadcast messages, each node spends time to finish decoding and get the message. This is illustrated in Fig. 3.3, where broadcast messages $m^{\mathrm{bc}}_{l,1}$ and $m^{\mathrm{bc}}_{l,2}$ are received by node $j \neq l$ ($j \in \mathcal{L}$) at time instants $t_{j,l,1}$ and $t_{j,l,2}$, respectively. For general cases, at time $t^{\mathrm{rc}}_{l,j,k} \in \{t^{\mathrm{rc}}_{l,j,k} \in \mathcal{T} \colon k \in \mathcal{K}^{\mathrm{rc}}_{l,j} \subseteq \mathbb{Z}_+\} =: \mathcal{T}^{\mathrm{rc}}_{l,j}$ ($l, j \in \mathcal{L}$), node l receives message $m^{\mathrm{rc}}_{l,j}(t^{\mathrm{rc}}_{l,j,k})$ from node j. Note that not all the broadcast message can be successfully received by node l, because the communication channel is not always in good condition. We can see that this communication framework does not need any synchronization.

Similar to Section 3.2.2, we give some notations for the indirect information. $M^{\mathrm{rc}}_{l,j}[t] := \{m_{l,j}(t^{\mathrm{rc}}_{l,j,k}) \colon t^{\mathrm{rc}}_{l,j,k} \in \mathcal{T}^{\mathrm{rc}}_{l,j}[t]\}$ denotes the set of all the messages received from node j within time window $[0,\, t]$, where $\mathcal{T}^{\mathrm{rc}}_{l,j}[t] := \mathcal{T}^{\mathrm{rc}}_{l,j} \bigcap [0,\, t]$. $M^{\mathrm{rc}}_{l}[t] := \bigcup_{j \in \mathcal{L}} M^{\mathrm{rc}}_{l,j}[t]$ represents the set of all the messages received by node l during $[0,\, t]$.

3.3 Cooperative Localization Framework and Design Problem

3.3.1 Cooperative Localization Framework

In this section, we describe the proposed cooperative localization framework and define the localization design problem. We consider a practical scenario that each node $l \in \mathcal{L}$ does not know when the next measurement $z_{l,j}(t^{\text{ms}}_{l,j,k+1})$ ($j \in \mathcal{I}$) or the next message $m_{l,j}(t^{\text{rc}}_{l,j,k+1})$ ($j \in \mathcal{L}$) will come,[3] and all the information one node can use is the received measurements and messages at hand.

To conduct the cooperative localization, all the nodes (including both anchor nodes and non-anchor nodes) work cooperatively.

3.3.1.1 Anchor nodes

- For each anchor node $a \in \mathcal{A}$, it can take measurement from non-anchor node $l \in \mathcal{L}$, say $z_{a,l}(t^{\text{ms}}_{a,l,k})$. After taking this measurement, it transmits its state $x^*_a(t^{\text{ms}}_{a,l,k})$ as well as the measurement $z_{a,l}(t^{\text{ms}}_{a,l,k})$ to node l immediately.

- For each anchor node $a \in \mathcal{A}$, it can be measured by non-anchor node $l \in \mathcal{L}$,[4] and we label the corresponding measurement as $z_{l,a}(t^{\text{ms}}_{l,a,k})$. When anchor node a realizes that it is being measured, it sends its state $x^*_a(t^{\text{ms}}_{l,a,k})$ to node l at once.

Assumption 3.2. The time spent on transmission for $z_{a,l}(t^{\text{ms}}_{a,l,k})$, $x^*_a(t^{\text{ms}}_{a,l,k})$ and $x^*_a(t^{\text{ms}}_{l,a,k})$ can be neglected.

Since the anchor state $x^*_a(t^{\text{ms}}_{a,l,k})$ or $x^*_a(t^{\text{ms}}_{l,a,k})$ just contains one node's information, and $z_{a,l}(t^{\text{ms}}_{a,l,k})$ is merely a one-dimensional number, the transmission can be completed timely. Hence, Assumption 3.2 is appropriate.

Similar to Section 3.2.1 and Section 3.2.2, we provide some notations for the sets of received anchor states. For non-anchor node l, the set of all the received anchor states from anchor node a during $[0,\, t]$ is

$$W_{l,a}[t] := \{x_a(t^{\text{ms}}_{l,a,k}) \colon t^{\text{ms}}_{l,a,k} \in \mathcal{T}^{\text{ms}}_{l,a}[t]\} \bigcup \{x_a(t^{\text{ms}}_{a,l,k}) \colon t^{\text{ms}}_{a,l,k} \in \mathcal{T}^{\text{ms}}_{a,l}[t]\},$$

where $\mathcal{T}^{\text{ms}}_{l,a}[t]$ and $\mathcal{T}^{\text{ms}}_{a,l}[t]$ correspond to the sets of measurement times (see Section 3.2.1), since the anchor states are sent when anchor node a takes measurements or is measured. We define $W_l[t] := \bigcup_{a\in\mathcal{A}} W_{l,a}[t]$ as the set of all the received anchor states within time window

[3] One reason for it is that the measurement/meassge cannot be always successfully taken/received.

[4] That means the distance between nodes a and l is measured by node l.

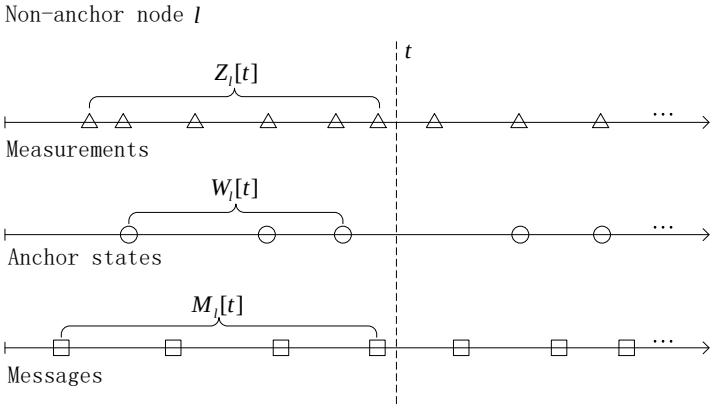


Figure 3.4: Illustrations of cooperative localization at non-anchor node $l \in \mathcal{L}$. The triangles, circles, and squares represent the measurements, anchor states, and messages, respectively. Note that at time t, node l utilizes the information from $Z_l[t]$, $W_l[t]$, and $M_l[t]$ to estimate the location $y_l(t)$.

$[0,\ t]$. Furthermore, $W[t] := \bigcup_{l\in\mathcal{L}} W_l[t]$ stands for the set of all the received anchor states in the whole network.

3.3.1.2 Non-anchor nodes

For each non-anchor node $l \in \mathcal{L}$, it wants to estimate its location $y_l(t)$ for the current time instant t with the help of the other nodes. This is called the cooperative localization, which is briefly illustrated in Fig. 3.4.

From Fig. 3.4, we can see that all the information a non-anchor node $l \in \mathcal{L}$ can obtain by time instant t is from $Z_l[t]$, $W_l[t]$, and $M_l[t]$, and we call it the external information. As a results, different cooperative localization algorithms are indeed based on utilizing different parts of the external information.

3.3.2 Cooperative Localization Design Problem

To give a formal description of the cooperative localization problem, we need to introduce a concept – local probability, which is given as follows. Let $X(t) := \left[x_1^{\mathrm{T}}(t),\ldots,x_L^{\mathrm{T}}(t)\right]^{\mathrm{T}}$, i.e., the vector of all non-anchor nodes' states at time t. For the simplicity of analysis, we rewrite $X(t)$ as $\left[x_l^{\mathrm{T}}(t)\right]_{l\in\mathcal{L}}^{\mathrm{T}}$. Hence, a sub-vector of $X(t)$ with part of non-anchor nodes can be simply expressed as $\left[x_l^{\mathrm{T}}(t)\right]_{l\in\bar{\mathcal{L}}}^{\mathrm{T}} =: \bar{X}(t)$, where $\bar{\mathcal{L}} \subseteq \mathcal{L}$. Due to the limited communication bandwidth, global information $X(t)$ can hardly be exchanged among all non-anchor nodes. As a result, each non-anchor node $l \in \mathcal{L}$ at time $t \in \mathcal{T}$ can only know a portion of the global information as denoted by $\bar{X}_l(t)$ based on its received measurements and messages during $[0,\ t]$, where the subscript l is employed to distinguish the different sub-vectors $\bar{X}(t)$ for different non-anchor nodes. In this chapter, the inference of $\bar{X}_l(t)$ refers to deduce the conditional prob-

ability $p(\bar{X}_l(t)|Z_l[t],W_l[t],M_l[t])$ which is conditioned on the received measurements, anchor states, and messages. Since the storage and computational capability of node l are limited, it is impossible to utilize all the information from $Z_l[t]$, $W_l[t]$, and $M_l[t]$ when t goes to sufficiently large.[5] That means only a portion of $Z_l[t]$, $W_l[t]$, and $M_l[t]$ can be utilized to infer $\bar{X}_l(t)$ at node l. To represent this "local" inference of $\bar{X}_l(t)$, we use the notation $\hat{p}_l(\bar{X}_l(t)|Z_l[t],W_l[t],M_l[t])$ [called the local probability of $\bar{X}_l(t)$], where $\hat{p}_l(\cdot)$ means it is deduced from the perspective of node l, and the local probability must be properly designed at node l to largely approximate $p(\bar{X}_l(t)|Z_l[t],W_l[t],M_l[t])$. As a special case of local probabilities, the initial condition of non-anchor node l is $\hat{p}_l(x_l(0))$ which reflects the initial guess of state $x_l(0)$ and is only known to node i itself, while the initial condition $\hat{p}_l(x_j(0))$ is not available (undefined) for other node $j \neq l$ (where $j \in \mathcal{L}$). Note that local probabilities are the first part to be designed in our cooperative localization algorithm.

With local probabilities, we can define the local location estimation $\hat{y}_l(t)$ from the perspective of non-anchor nodes $l \in \mathcal{L}$ as follows:

$$\hat{y}_l(t) = \hat{\mathbb{E}}^{\bar{X}_l(t)}_{y_l(t)}[y_l(t)|Z_l[t],W_l[t],M_l[t]] := \int_{\bar{\mathcal{X}}_l(t)} g_l(x_l(t))\hat{p}_l(\bar{X}_l(t)|Z_l[t],W_l[t],M_l[t])\mathrm{d}\bar{X}_l(t), \tag{3.5}$$

where the superscript $\bar{X}_l(t)$ of $\hat{\mathbb{E}}[\cdot]$ means the estimation is based on the local probability of $\bar{X}_l(t)$, and $\bar{\mathcal{X}}_l(t) \ni \bar{X}_l(t)$ is the support of $\bar{X}_l(t)$. In this chapter, we assume

$$\int_{\bar{\mathcal{X}}_l(t)} \|g_l(x_l(t))\|\hat{p}_l(\bar{X}_l(t)|Z_l[t],W_l[t],M_l[t])\mathrm{d}\bar{X}_l(t) < \infty. \tag{3.6}$$

The second part to be designed is what to exchange between different nodes, i.e., designing the broadcast messages. The broadcast messages can include the measurements and local probabilities from other nodes, or any useful information for location estimation. Note that whether these messages are successfully transmitted is dependent on the physical layer and the MAC layer of the network which are considered in Section 3.6.

Now, it is readily to formally introduce the cooperative localization problem.

Problem 3.1 (Cooperative Localization)**.** At time t, each non-anchor node $l \in \mathcal{L}$ calculates the location estimation $\hat{y}_l(t)$ by (3.5), where the prior $\hat{p}(x_l(0))$, the measurement set $Z_l[t]$, the anchor state set $W_l[t]$, and the message set $M_l[t]$ are known to node l. With the help of anchor nodes (see Section 3.3.1.1), a cooperative localization algorithm for each non-anchor node l is to:

i) design the local probability $\hat{p}_l(\bar{X}_l(t)|Z_l[t],W_l[t],M_l[t])$ at node $l \in \mathcal{L}$;

ii) design the broadcast message $m^{\mathrm{bc}}_{l,k}$ ($k \in \mathcal{K}^{\mathrm{bc}}_l$) for each node $l \in \mathcal{L}$;

[5] One important reason is that the Markov property cannot be guaranteed when the information is collected in a distributed manner.

such that the mean squared error $\mathbb{E}\|\hat{y}_l(t) - y_l(t)\|^2$ is minimized at each $t \in \mathcal{T}$.

If we do not consider the communication constraints, the local probability at each non-anchor node $l \in \mathcal{L}$ will become

$$\hat{p}_l(\bar{X}_l(t)|Z_l[t], W_l[t], M_l[t]) = p(X(t)|Z[t], W[t]), \tag{3.7}$$

i.e., it contains the global information since ideal communications can make $Z[t]$ and $W[t]$ known to every non-anchor node. In this case, the solution to the cooperative localization in Problem 3.1 is equivalent to run the centralized localization algorithm at each non-anchor node where the global information can be always obtained in estimating locations. This centralized algorithm is given in Lemma 3.1 in Section 3.4.1. Note that the centralized algorithm can serves as a benchmark for distributed algorithms in which the communication constraints are considered.

Since the optimal solution to Problem 3.1 is hard to derive when communications constraints are considered, in this work, we are aim to find a suboptimal solution to Problem 3.1 by analyzing the performance gap between the distributed and centralized algorithms. The analysis of performance gap is given in Section 3.4, and the distributed algorithm is proposed in Section 3.5.

3.4 Inspiration From Centralized Localization

In this section, we propose a centralized localization algorithm, as a benchmark for distributed algorithms, which can utilize all the states of nodes and all the measurements in the whole network (see Section 3.4.1), and then show that the centralized localization algorithm has a key property which is very helpful in the designs of cooperative algorithms (see Section 3.4.2).

3.4.1 Centralized Localization Algorithm

The centralized localization algorithm is to derive the location estimation in a centralized manner. That means the set of all measurements $Z[t] = Z[t_k^{\text{ms}}]$ and the set of all anchor states $W[t] = W[t_k^{\text{ms}}]$ in this network are used, where t_k^{ms} is the largest time instant in $\mathcal{T}^{\text{ms}} := \bigcup_{i,j\in\mathcal{I}} \mathcal{T}_{i,j}^{\text{ms}}$ satisfying $t \geq t_k^{\text{ms}}$.[6] The centralized location estimation $\hat{y}_l^*(t)$ is given by

$$\hat{y}_l^*(t) = \mathbb{E}_{y_l(t)}[y_l(t)|Z[t_k^{\text{ms}}], W[t_k^{\text{ms}}]] = \int_{\mathcal{X}} g_l(x_l(t)) p(X(t)|Z[t_k^{\text{ms}}], W[t_k^{\text{ms}}]) \mathrm{d}X(t), \tag{3.8}$$

[6]The subscript k in t_k^{ms} means that t_k^{ms} is the k^{th} smallest element in $\mathcal{T}^{\text{ms}}$.

where $l \in \mathcal{L}$ and $X(t) \in \mathcal{X}$. In this work, we assume

$$\int_{\mathcal{X}} \|g_l(x_l(t))\| p(X(t)|Z[t_k^{\text{ms}}], W[t_k^{\text{ms}}]) \mathrm{d}X(t) < \infty. \tag{3.9}$$

The centralized algorithm can be implemented in the recursive Bayesian filtering framework as shown in Lemma 3.1.

Lemma 3.1 (Centralized Localization Algorithm)**.** At time instant t, the estimated location $\hat{y}_l^*(t)$ for non-anchor node $l \in \mathcal{L}$ is calculated by (3.8), where $p(X(t)|Z[t_k^{\text{ms}}], W[t_k^{\text{ms}}])$ is derived by

$$\int_{\mathcal{X}} p(X(t)|X(t_k^{\text{ms}})) p(X(t_k^{\text{ms}})|Z[t_k^{\text{ms}}], W[t_k^{\text{ms}}]) \mathrm{d}X(t_k^{\text{ms}}), \tag{3.10}$$

in which $p(X(t_k^{\text{ms}})|Z[t_k^{\text{ms}}])$ is computed by the following recursive equations:

- **Initialization.** The recursion starts from the prior distribution $p(X(0)) = \prod_{l \in \mathcal{L}} \hat{p}_l(x_l(0))$.
- **Prediction.** At time instant $t_k^{\text{ms}} \in \mathcal{T}^{\text{ms}}$, the prior distribution $p(X(t_k^{\text{ms}})|Z[t_{k-1}^{\text{ms}}], W[t_{k-1}^{\text{ms}}])$ is derived by the Chapman-Kolmogorov equation

  $$\int_{\mathcal{X}} p(X(t_k^{\text{ms}})|X(t_{k-1}^{\text{ms}})) p(X(t_{k-1}^{\text{ms}})|Z[t_{k-1}^{\text{ms}}], W[t_{k-1}^{\text{ms}}]) \mathrm{d}X(t_{k-1}^{\text{ms}}), \tag{3.11}$$

  where $t_0^{\text{ms}} := 0$ and $p(X(0)|Z[0], W[0]) := p(X(0))$.
- **Update.** Given the measurement set and the received anchor state set at t_k^{ms}, i.e., $Z(t_k^{\text{ms}}) := Z[t_k^{\text{ms}}] \setminus Z[t_{k-1}^{\text{ms}}]$ and $W(t_k^{\text{ms}}) := W[t_k^{\text{ms}}] \setminus W[t_{k-1}^{\text{ms}}]$, respectively, the posterior distribution $p(X(t_k^{\text{ms}})|Z[t_k^{\text{ms}}], W[t_k^{\text{ms}}])$ can be computed by

  $$\frac{p(Z(t_k^{\text{ms}})|X(t_k^{\text{ms}}), W(t_k^{\text{ms}})) p(X(t_k^{\text{ms}})|Z[t_{k-1}^{\text{ms}}], W[t_{k-1}^{\text{ms}}])}{\int_{\mathcal{X}} p(Z(t_k^{\text{ms}})|X(t_k^{\text{ms}}), W(t_k^{\text{ms}})) p(X(t_k^{\text{ms}})|Z[t_{k-1}^{\text{ms}}], W[t_{k-1}^{\text{ms}}]) \mathrm{d}X(t_k^{\text{ms}})}, \tag{3.12}$$

  where the likelihood function $p(Z(t_k^{\text{ms}})|X(t_k^{\text{ms}}), W(t_k^{\text{ms}}))$ is derived from the measurement equation (3.4).

  Proof: See Appendix B.1. ■

Remark 3.2. In Lemma 3.1, all the probabilities are defined except for $p(X(t)|X(t_k^{\text{ms}}))$ in (3.10) and $p(X(t_k^{\text{ms}})|X(t_{k-1}^{\text{ms}}))$ in (3.11). These two probabilities can be derived by solving the well-known Fokker-Planck equation (see Chapter 4.9 in [16]) of the group of state equations (3.1) for $l \in \mathcal{L}$, when the system function is linear w.r.t. the system state $x_l(t)$. If the system function is nonlinear, solving the Fokker-Planck equation is difficult, but we can numerically solve the state equations with different initial values instead.[7] These solutions can work as the propagated sigma points in Gaussian filters [18, 77], or the particles in particle filters [78, 79].

[7] For example, we can use the Euler-Maruyama method or high-order methods [76, 77, 78, 79] to solve the state equations with initial values at t_k^{ms} for deriving $p(X(t)|X(t_k^{\text{ms}}))$.

From Lemma 3.1, we can see that when new measurements come, the algorithm should make the prediction [through (3.11)] and update [through (3.12)] to derive the posterior distribution $p(X(t_k^{\mathrm{ms}})|Z[t_k^{\mathrm{ms}}],W[t_k^{\mathrm{ms}}])$. Thus, the distribution $p(X(t)|Z[t],W[t])$ is derived by (3.10), and the estimated location is $\hat{y}_l^*(t)$ is calculated by (3.8).

It is, $Z(t_k^{\mathrm{ms}})$, the set of all the new measurements who updates the estimated location $\hat{y}_l^*(t) = \mathbb{E}_{y_l(t)}[y_l(t)|Z[t],W[t]]$. However, not all the new measurements and system states have notable contributions to the estimated location. Actually, we can utilize part of the measurements and the states without losing too much performance of location estimation. This property will be analyzed comprehensively in Section 3.4.2, and we stress that it plays a pivotal role in designing a cooperative localization algorithm, since not all the non-anchor nodes need to know all the new measurements and the states as the centralized algorithm does.

3.4.2 Estimation Gap Under Partial Information

In this subsection, we analyze the performance gap between the centralized location estimation and the location estimation when the information of measurements and anchor states is partially known (see Theorem 3.1), which plays a pivotal role in designing the cooperative localization algorithms.

Recall that the centralized location estimation $\mathbb{E}_{y_l(t)}[y_l(t)|Z[t_k^{\mathrm{ms}}],W[t_k^{\mathrm{ms}}]]$ is defined in (3.8). For the location estimation with partial information at t_k^{ms}, it has the following form

$$\mathbb{E}_{y_l(t)}^{\bar{X}(t)}[y_l(t)|Z[t_{k-1}^{\mathrm{ms}}],\bar{Z}(t_k^{\mathrm{ms}}),W[t_{k-1}^{\mathrm{ms}}],\bar{W}(t_k^{\mathrm{ms}})] = \int_{\bar{\mathcal{X}}(t)} g_l(x_l(t))p(\bar{X}(t)|Z[t_{k-1}^{\mathrm{ms}}],\bar{Z}(t_k^{\mathrm{ms}}),W[t_{k-1}^{\mathrm{ms}}],\bar{W}(t_k^{\mathrm{ms}}))\mathrm{d}\bar{X}(t), \quad (3.13)$$

where $\bar{Z}(t_k^{\mathrm{ms}})$ and $\bar{W}(t_k^{\mathrm{ms}})$ are the partial known measurement set and anchor set, respectively. Thus, the estimation gap is define as follows

$$\left\| \mathbb{E}_{y_l(t)}[y_l(t)|Z[t_k^{\mathrm{ms}}],W[t_k^{\mathrm{ms}}]] - \mathbb{E}_{y_l(t)}^{\bar{X}(t)}[y_l(t)|Z[t_{k-1}^{\mathrm{ms}}],\bar{Z}(t_k^{\mathrm{ms}}),W[t_{k-1}^{\mathrm{ms}}],\bar{W}(t_k^{\mathrm{ms}})] \right\|. \quad (3.14)$$

We can see that the smaller the estimation gap is, the closer the location estimation with partial information to the centralized location estimation will be.

Before analyzing the estimation gap, we propose two important concepts: the measurement graph and the measurement cluster, which are given in Definition 3.1.

Definition 3.1 (Measurement Graph and Measurement Cluster)**.** At time instant t_k^{ms}, directed graph $\mathcal{G}_k^{\mathrm{ms}} := \big(\mathcal{V}(Z(t_k^{\mathrm{ms}})),\mathcal{E}(Z(t_k^{\mathrm{ms}}))\big)$ is the measurement graph, where vertex set $\mathcal{V}(Z(t_k^{\mathrm{ms}}))$ represents the set of nodes directly related to the measurements in $Z(t_k^{\mathrm{ms}})$, and has the following form

$$\mathcal{V}(Z(t_k^{\mathrm{ms}})) := \{i\colon z_{i,j}(t_k^{\mathrm{ms}}) \in Z(t_k^{\mathrm{ms}}) \vee z_{j,i}(t_k^{\mathrm{ms}}) \in Z(t_k^{\mathrm{ms}})\}, \quad (3.15)$$

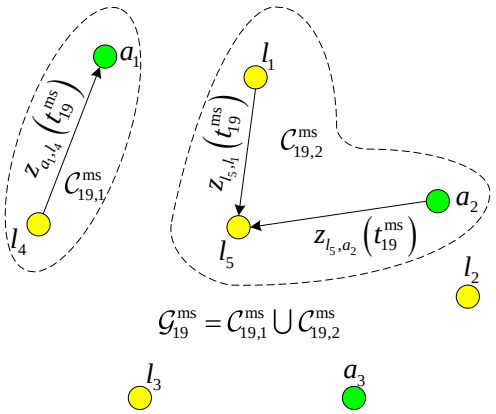


Figure 3.5: Illustration of measurement graph and measurement cluster. We continue using the network in Fig. 3.1, where $t_{19}^{\text{ms}} = t_{l_5,l_1,3}^{\text{ms}} = t_{l_5,a_2,7}^{\text{ms}} = t_{a_1,l_4,1}^{\text{ms}}$. The measurement graph is $\mathcal{G}_{19}^{\text{ms}} = (\{a_1,a_2,l_1,l_4,l_5\},\{(l_4,a_1),(l_1,l_5),(a_2,l_5)\})$. Note that nodes a_3, l_2, and l_3 are not included, since their corresponding measurement times are not equal to t_{19}^{ms}. In $\mathcal{G}_{19}^{\text{ms}}$, there are two measurement clusters, i.e., $\mathcal{C}_{19,1}^{\text{ms}} = (\{a_1,l_4\},\{(l_4,a_1)\})$ and $\mathcal{C}_{19,2}^{\text{ms}} = (\{a_2,l_1,l_5\},\{(a_2,l_5),(l_1,l_5)\})$.

in which $\vee$ is the logical "or"; and edge set $\mathcal{E}(Z(t_k^{\text{ms}}))$ records the pairs of node related to the measurements which is

$$\mathcal{E}(Z(t_k^{\text{ms}})) := \{(j,i)\colon z_{i,j}(t_k^{\text{ms}}) \in Z(t_k^{\text{ms}}) \vee z_{j,i}(t_k^{\text{ms}}) \in Z(t_k^{\text{ms}})\}. \tag{3.16}$$

The measurement clusters are the connected components[8] of $\mathcal{G}_k^{\text{ms}}$ when ignoring the direction of all the edges. We label the measurement clusters as $\mathcal{C}_{r,k}^{\text{ms}}$, where $r \in \{1,\ldots,R_k\} =: \mathcal{R}_k$.

Remark 3.3. The measurement graph tells which nodes are related to the considered measurements (i.e., vertices), and how the measurements connect to these nodes (i.e., edges). An example of the measurement graph is given in Fig. 3.5, where we use the same network as that in Fig. 3.1. The vertices of $\mathcal{G}_{19}^{\text{ms}}$ are $\{a_1,a_2,l_1,l_4,l_5\}$ and the edges are $\{(l_4,a_1),(l_1,l_5),(a_2,l_5)\}$. Note that edge (i,j) corresponds to the measurement $z_{j,i}(t_{19}^{\text{ms}})$.

In terms of the measurement clusters, they reflect which nodes are directly connected by the measurements. Between any two clusters, there are no nodes linked by the measurements. From Fig. 3.5, we can see that the measurement graph $\mathcal{G}_{19}^{\text{ms}}$ contains two disjoint subgraphs $(\{a_1,l_4\},\{(l_4,a_1)\})$ and $(\{a_2,l_1,l_5\},\{(a_2,l_5),(l_1,l_5)\})$ which are exactly the measurement clusters $\mathcal{C}_{19,1}^{\text{ms}}$ and $\mathcal{C}_{19,2}^{\text{ms}}$, respectively.

With the measurement graph and measurement cluster, we define the admissible measurement-state (AMS) triple as follows.

[8] A formal definition of connected component for an undirected graph is from [80] that: A path is a sequence of vertices where there is an edge connecting each vertex to the next vertex in the path. If there exists a path from vertex u to w, then we say that vertex w is reachable from vertex u. A connected component is a group of vertices in an undirected graph that are reachable from one another.

Definition 3.2 (AMS Triple)**.** Let $\bar{Z}(t_k^{\text{ms}}) \subseteq Z(t_k^{\text{ms}})$, $\bar{W}(t_k^{\text{ms}}) \subseteq W(t_k^{\text{ms}})$, and $\bar{X}(t_k^{\text{ms}}) \subseteq X(t_k^{\text{ms}})$. The measurement-state triple $\langle \bar{Z}(t_k^{\text{ms}}), \bar{X}(t_k^{\text{ms}}), \bar{W}(t_k^{\text{ms}}) \rangle$ is admissible, if the following three conditions hold:

$$(\mathcal{V}(\bar{Z}(t_k^{\text{ms}})), \mathcal{E}(\bar{Z}(t_k^{\text{ms}}))) = \bigcup_{r \in \mathcal{R}'_k} \mathcal{C}_{k,r}^{\text{ms}}, \tag{3.17}$$

$$\mathcal{V}(\bar{Z}(t_k^{\text{ms}})) \bigcap \mathcal{L} \subseteq \mathcal{L}_{\bar{X}(t_k^{\text{ms}})} \subseteq \mathcal{L} \setminus \mathcal{V}(\check{Z}(t_k^{\text{ms}})), \tag{3.18}$$

$$\mathcal{V}(\bar{Z}(t_k^{\text{ms}})) \bigcap \mathcal{A} \subseteq \mathcal{A}_{W(t_k^{\text{ms}})} \subseteq \mathcal{A} \setminus \mathcal{V}(\check{Z}(t_k^{\text{ms}})), \tag{3.19}$$

where $\mathcal{R}'_k \subseteq \mathcal{R}_k$, $\check{Z}(t_k^{\text{ms}}) = Z(t_k^{\text{ms}}) \setminus \bar{Z}(t_k^{\text{ms}})$, $\mathcal{L}_{\bar{X}(t_k^{\text{ms}})} := \{l \colon x_l(t_k^{\text{ms}}) \in \bar{X}(t_k^{\text{ms}})\}$,[9] and $\mathcal{A}_{\bar{W}(t_k^{\text{ms}})} := \{a \colon x_a(t_k^{\text{ms}}) \in \bar{W}(t_k^{\text{ms}})\}$.

Remark 3.4. The AMS triple contains three elements. The first one is the measurement set $\bar{Z}(t_k^{\text{ms}})$ which satisfies condition (3.17). This condition tells that graph $(\mathcal{V}(\bar{Z}(t_k^{\text{ms}})), \mathcal{E}(\bar{Z}(t_k^{\text{ms}})))$ is exactly the union of connected components of graph $\mathcal{G}_k^{\text{ms}}$. That means any node linked by $\bar{Z}(t_k^{\text{ms}})$ cannot be connected by any other measurements in $Z(t_k^{\text{ms}})$. For example, in Fig. 3.5, $\bar{Z}(t_{19}^{\text{ms}})$ can be $\{z_{a_1,l_4}(t_{19}^{\text{ms}})\}$, $\{z_{l_5,l_1}(t_{19}^{\text{ms}}), z_{l_5,a_2}(t_{19}^{\text{ms}})\}$, or $\{z_{a_1,l_4}(t_{19}^{\text{ms}}), z_{l_5,l_1}(t_{19}^{\text{ms}}), z_{l_5,a_2}(t_{19}^{\text{ms}})\}$.

The second element is the state set $\bar{X}(t_k^{\text{ms}})$ which satisfies condition (3.18). This condition states that the set of non-anchor nodes $\mathcal{L}_{\bar{X}(t_k^{\text{ms}})}$ must contain non-anchor node set $\mathcal{V}(\bar{Z}(t_k^{\text{ms}})) \bigcap \mathcal{L}$ and must be contained in non-anchor node set $\mathcal{L} \setminus \mathcal{V}(\check{Z}(t_k^{\text{ms}}))$. That means non-anchor node set $\mathcal{L}_{\bar{X}(t_k^{\text{ms}})}$ must contain the non-anchor nodes linked by $\bar{Z}(t_k^{\text{ms}})$, and cannot be associated with other measurements [i.e., measurements in $\check{Z}(t_k^{\text{ms}})$]. For example, in Fig. 3.5, if $\bar{Z}(t_{19}^{\text{ms}}) = \{z_{a_1,l_4}(t_{19}^{\text{ms}})\}$, then a valid $\bar{X}(t_k^{\text{ms}})$ can be $\{x_{l_3}(t_k^{\text{ms}}), x_{l_4}(t_k^{\text{ms}})\}$, since condition (3.18) holds, i.e.,

$$\mathcal{V}(\bar{Z}(t_{19}^{\text{ms}})) \bigcap \mathcal{L} = \{l_4\} \subseteq \mathcal{L}_{\bar{X}(t_k^{\text{ms}})} = \{l_3, l_4\} \subseteq \mathcal{L} \setminus \mathcal{V}(\check{Z}(t_{19}^{\text{ms}})) = \{l_2, l_3, l_4\}. \tag{3.20}$$

However, for $t \in (t_{19}^{\text{ms}}, t_{20}^{\text{ms}}]$, state set $\{x_{l_2}(t_k^{\text{ms}})\}$ is not a valid $\bar{X}(t_k^{\text{ms}})$, since $\mathcal{V}(\bar{Z}(t_{19}^{\text{ms}})) \bigcap \mathcal{L} \nsubseteq \mathcal{L}_{\bar{X}(t_k^{\text{ms}})}$ [i.e., $\{l_2\}$ does not contain any non-anchor node linked by $\bar{Z}(t_{19}^{\text{ms}})$]. Also, state set $\{x_{l_1}(t_k^{\text{ms}}), x_{l_3}(t_k^{\text{ms}})\}$ is not a valid $\bar{X}(t_k^{\text{ms}})$, since $\mathcal{L}_{\bar{X}(t_k^{\text{ms}})} \nsubseteq \mathcal{L} \setminus \mathcal{V}(\check{Z}(t_{19}^{\text{ms}}))$ [i.e., $\{x_{l_1}(t_k^{\text{ms}}), x_{l_3}(t_k^{\text{ms}})\}$ is associated with measurement $z_{l_5,l_1}(t_{19}^{\text{ms}}) \notin \bar{Z}(t_{19}^{\text{ms}})$)].

The third element is the anchor state set $\bar{W}(t_k^{\text{ms}})$ satisfying condition (3.19). It says that anchor set $\mathcal{A}_{W(t_k^{\text{ms}})}$ should include anchor set $\mathcal{V}(\bar{Z}(t_k^{\text{ms}})) \bigcap \mathcal{A}$. This implies any anchor node associated with measurements in $\bar{Z}(t_k^{\text{ms}})$ must be in set $\mathcal{A}_{W(t_k^{\text{ms}})}$. For example, in Fig. 3.5, if we choose $\bar{Z}(t_{19}^{\text{ms}}) = \{z_{a_1,l_4}(t_{19}^{\text{ms}})\}$, then a valid $W(t_k^{\text{ms}})$ can be $\{a_1\}$ or $\{a_1, a_3\}$, since it contains $\mathcal{V}(\bar{Z}(t_{19}^{\text{ms}})) \bigcap \mathcal{A} = \{a_1\}$ and is included in $\mathcal{A} \setminus \mathcal{V}(\check{Z}(t_{19}^{\text{ms}})) = \{a_1, a_3\}$.

Given prior $p(X(t_k^{\text{ms}}) | Z[t_{k-1}^{\text{ms}}], W[t_{k-1}^{\text{ms}}])$, each AMS triple determines a prior cut

$$p(\bar{X}(t_k^{\text{ms}}) | Z[t_{k-1}^{\text{ms}}], W[t_{k-1}^{\text{ms}}]) p(\check{X}(t_k^{\text{ms}}) | Z[t_{k-1}^{\text{ms}}], W[t_{k-1}^{\text{ms}}]), \tag{3.21}$$

[9] Strictly speaking, $\mathcal{L}_{\bar{X}(t_k^{\text{ms}})}$ refers to the set of nodes such that $[x_l(t_k^{\text{ms}})]_{l \in \mathcal{L}_{\bar{X}(t_k^{\text{ms}})}}$.

where $\check{X}(t_k^{\mathrm{ms}}) = \left[x_j^{\mathrm{T}}(t_k^{\mathrm{ms}})\right]^{\mathrm{T}}_{j\in\mathcal{L}_{X(t_k^{\mathrm{ms}})}\setminus\mathcal{L}_{\bar{X}(t_k^{\mathrm{ms}})}}$. In (3.21), the first term and the second term are called the related sub-prior and unrelated sub-prior, respectively. The cut gap is defined as

$$\|p(X(t_k^{\mathrm{ms}})|Z[t_{k-1}^{\mathrm{ms}}],W[t_{k-1}^{\mathrm{ms}}]) - p(\bar{X}(t_k^{\mathrm{ms}})|Z[t_{k-1}^{\mathrm{ms}}],W[t_{k-1}^{\mathrm{ms}}])p(\check{X}(t_k^{\mathrm{ms}})|Z[t_{k-1}^{\mathrm{ms}}],W[t_{k-1}^{\mathrm{ms}}])\|_{\infty}, \tag{3.22}$$

which measures the independence between the prior and prior cut. If the cut gap is small, then $\bar{X}(t_k^{\mathrm{ms}})$ and $\check{X}(t_k^{\mathrm{ms}})$ tend to be independent of each other; and if the cut gap is zero, then they are independent. Theorem 3.1 says that the estimation gap defined in (3.14) is continuous with the cut gap, if the following two conditions are satisfied

$$\int_{\mathcal{X}} \left\| \mathbb{E}_{x_l(t)}[g_l(x_l(t))|x_l(t_k^{\mathrm{ms}})] \right\| p(Z(t_k^{\mathrm{ms}})|X(t_k^{\mathrm{ms}}),W(t_k^{\mathrm{ms}}))\mathrm{d}X(t_k^{\mathrm{ms}}) < \infty, \tag{3.23}$$

$$\int_{\mathcal{X}} p(Z(t_k^{\mathrm{ms}})|X(t_k^{\mathrm{ms}}),W(t_k^{\mathrm{ms}}))\mathrm{d}X(t_k^{\mathrm{ms}}) < \infty, \tag{3.24}$$

where

$$\mathbb{E}_{x_l(t)}[g_l(x_l(t))|x_l(t_k^{\mathrm{ms}})] = \int_{\mathcal{X}_l} g(x_l(t))p(x_l(t)|x_l(t_k^{\mathrm{ms}}))\mathrm{d}x_l(t). \tag{3.25}$$

Theorem 3.1 (Continuity of Estimation Gap)**.** Let $\langle \bar{Z}(t_k^{\mathrm{ms}}),\bar{X}(t_k^{\mathrm{ms}}),\bar{W}(t_k^{\mathrm{ms}})\rangle$ be an AMS triple, and conditions (3.23) and (3.24) are satisfied. For node $l \in \mathcal{L}_{\bar{X}(t_k^{\mathrm{ms}})} = \mathcal{L}_{\bar{X}(t)}$, $\forall \varepsilon > 0$, there exists a $\delta > 0$ such that if the cut gap defined in (3.22) is not greater than δ, the the estimation gap defined in (3.14) is not greater than ε.

Proof: See Appendix B.2. ■

Remark 3.5 (Implications from the Key Property in Theorem 3.1)**.** Theorem 3.1 provides an important implication for designing cooperative localization algorithms that: If the cut gap is small (i.e., the two sub-priors in (3.21) are nearly independent of each other), then the estimation gap in each node $l \in \mathcal{L}_{\bar{X}(t)}$ is also small. Also, it means the related nodes should share their information to estimate their locations cooperatively.

3.5 Distributed Prior-Cut Algorithm

In this section, we propose a cooperative localization algorithm termed the distributed prior-cut algorithm, where the prior refers to the local prior defined in Definition 3.3.

Before defining the local prior and posterior, we illustrate that in the distributed prior-cut algorithm, the local probability $\hat{p}_l(\bar{X}_l(t)|Z_l[t],W_l[t],M_l[t])$ is actually with the form $\hat{p}_l(\bar{X}_l(t)|\bar{Z}[t],\bar{W}[t])$, where $\bar{Z}[t] \subseteq Z[t]$ and $\bar{W}[t] \subseteq W[t]$. This is because, in the distributed prior-cut algorithm, the received messages during $[0,\, t]$ contain the information of the measurements and anchor states from other non-anchor nodes (see Section 3.5.4).

Definition 3.3 (Local Prior and Posterior)**.** For the local probability $\hat{p}_l(\bar{X}_l(t)|\bar{Z}[t],\bar{W}[t])$:

- If all the measurements in $\bar{Z}[t]$ are within the time window $[0,\ t)$, i.e., not including t, then we label this measurement set as $\bar{Z}^{\mathrm{a}}[t]$. Similarly, $\bar{W}^{\mathrm{a}}[t]$ represents the anchor state set whose elements are within $[0,\ t)$. We call $\hat{p}_l(\bar{X}_l(t)|\bar{Z}^{\mathrm{a}}[t],\bar{W}^{\mathrm{a}}[t])$ as the local prior.

- If at least one measurement in $\bar{Z}[t]$ is at time instant t, then we label this measurement set as $\bar{Z}^{\mathrm{b}}[t]$. Likewise, $\bar{W}^{\mathrm{b}}[t]$ stands for the anchor state set which contains at least one anchor state at time instant t. We call $\hat{p}_l(\bar{X}_l(t)|\bar{Z}^{\mathrm{b}}[t],\bar{W}^{\mathrm{b}}[t])$ as the local posterior.

3.5.1 Methodology

This subsection gives the main ideas of designing the distributed prior-cut algorithm mainly based on the main property (which is given in Theorem 3.1 and explained in Remark 3.5) derived in Section 3.4.2.

Firstly, the broadcast-message design in Section 3.2 is beneficial to the information sharing, since the information in one node is potentially useful for all nodes. Specifically, the more one node knows, the closer its local localization algorithm to the centralized algorithm (see Lemma 3.1) will be.

However, if all the nodes broadcast all their information without proper reduction, then the communication cost would be very large and cannot be afforded by a wireless communication network. Theorem 3.1 (see also Remark 3.5) provides an effective resolution to this problem: every node only needs to consider and transmit the messages related to the AMS triple (see Definition 3.2) with a small cut gap [defined in (3.22)]. It should be noted that the AMS triple used in each node is similar to the analysis in Section 3.4.2 but based on the local information in each node, since the global information is usually unable to be accessed.

3.5.2 Algorithm Structure

In this subsection, we propose the structure of the distributed prior-cut algorithm. The main ideas are:

- Each non-anchor node $l \in \mathcal{L}$ contains a database $\mathcal{B}_l(t)$ which contains all the knowledge on the whole network from the perspective of node l.

- After receiving measurements, anchor states, or messages, each non-anchor node l updates its database $\mathcal{B}_l(t)$ through these newly arrived information.

- Based on the newly updated database $\mathcal{B}_l(t)$, the broadcast messages are properly designed.

- Based on the newly updated database, each non-anchor node l estimates its location continuously[10].

[10]Practically, the update of estimated location depends on the clock of the onboard chip.

We can see that the database $\mathcal{B}_l(t)$ plays a pivotal role in our algorithm.

Now, we begin to propose the structure for the cooperative localization algorithm, which is given in Algorithm 3.1.

Algorithm 3.1 Cooperative Localization Algorithm of Non-anchor Node l

1: **Initialization:** $\hat{p}_l(x_l(0))$.
2: **loop**
3: **if** $Z_l(t) \neq \emptyset \,||\, W_l(t) \neq \emptyset \,||\, M_l(t) \neq \emptyset$ **then**
4: Update the database $\mathcal{B}_l(t)$;
5: Design broadcast messages;
6: **end if**
7: Calculate $\hat{p}_l(\bar{X}_l(t)|Z_l[t], W_l[t], M_l[t])$;
8: Estimate $\hat{y}_l(t)$ by using (3.5);
9: **end loop**

In Algorithm 3.1, the initialization gives the local probability $\hat{p}_l(x_l(0))$ for non-anchor node l. Recall that at time $t = 0$, non-anchor node l does not know $\hat{p}_j(x_j(0))$ from any other nodes $j \neq l$ in $\mathcal{L}$ (see Problem 3.1). Line 3 is a condition to trigger the database update and the broadcast message design (see Lines 4 and 5, respectively). For Line 7, it calculates the local probability of $\bar{X}_l(t)$ based on the database $\mathcal{B}_l(t)$. Then, Line 8 gives the local estimated location $\hat{y}_l(t)$.

Above completes the description of the structure of the cooperative localization. Note that the parts to be design are Lines 4 and 5 in Algorithm 3.1.

3.5.3 Database Structure

Recall that for each non-anchor node $l \in \mathcal{L}$, the database is updated after taking measurements, anchor states or messages (see Section 3.5.2), and thus it just makes changes at time instants $t_{l,k}^{\text{upd}} \in \mathcal{T}_l^{\text{upd}} := \mathcal{T}_l^{\text{ms}} \bigcup \mathcal{T}_l^{\text{ac}} \bigcup \mathcal{T}_l^{\text{rc}}$, where $\mathcal{T}_l^{\text{ms}} := \bigcup_{j\in\mathcal{I}} \mathcal{T}_{l,j}^{\text{ms}}$, $\mathcal{T}_l^{\text{ac}} := \bigcup_{a\in\mathcal{A}} \mathcal{T}_{a,l}^{\text{ms}}$, and $\mathcal{T}_l^{\text{rc}} := \bigcup_{j\in\mathcal{L}} \mathcal{T}_{l,j}^{\text{rc}}$ are the time sets of measurements, anchor measurements, and messages, respectively. At any given time instant t, there is the largest $t_{l,k}^{\text{upd}} \in \mathcal{T}_l^{\text{upd}}$ satisfying $t \geq t_{l,k}^{\text{upd}}$. Then, the database $\mathcal{B}_l(t)$ ($l \in \mathcal{L}$) has the following form:

$$\mathcal{B}_l(t) = \mathcal{B}_l(t_{l,k}^{\text{upd}}), \quad t_{l,k}^{\text{upd}} = \max \mathcal{T}_l^{\text{upd}} \bigcap [0,\, t], \tag{3.26}$$

which means the database keep unchanged before the next update time $t_{l,k+1}^{\text{upd}}$. Thus, we just need to focus on $\mathcal{B}_l(t_{l,k}^{\text{upd}}) =: \mathcal{B}_{l,k}$.

For $\mathcal{B}_{l,k}$, it contains the local priors and posteriors (see Definition 3.3), local measurement sets, and local anchor state sets over time window $[0,\, t_{l,k}^{\text{upd}}]$. The detailed descriptions are given as follows.

For the local priors, they have the following form

$$\hat{p}_{l,k}^{\text{a}}(t_{l,k,s}^{\text{db,a}}) := \hat{p}_l(X_{l,k}^{\text{db,a}}(t_{l,k,s}^{\text{db,a}})|Z_{l,k}^{\text{db,a}}[t_{l,k,s}^{\text{db,a}}], W_{l,k}^{\text{db,a}}[t_{l,k,s}^{\text{db,a}}]), \quad t_{l,k,s}^{\text{db,a}} \in \mathcal{T}_{l,k}^{\text{db,a}}, \tag{3.27}$$

where $\mathcal{L}_{X_{l,k}^{\mathrm{db,a}}(t_{l,k,s}^{\mathrm{db,a}})} \subseteq \mathcal{L}$ represents the local prior node set, and $Z_{l,k}^{\mathrm{db,a}}[t_{l,k,s}^{\mathrm{db,a}}] \subseteq Z[t_{l,k,s}^{\mathrm{db,a}}]$ is the local prior measurement set, and $W_{l,k}^{\mathrm{db,a}}[t_{l,k,s}^{\mathrm{db,a}}] \subseteq Z[t_{l,k,s}^{\mathrm{db,a}}]$ is the local prior anchor state set.

Similarly, for the local posteriors, they have the following form

$$\hat{p}_{l,k}^{\mathrm{b}}(t_{l,k,s}^{\mathrm{db,b}}) := \hat{p}_l(X_{l,k}^{\mathrm{db,b}}(t_{l,k,s}^{\mathrm{db,b}})|Z_{l,k}^{\mathrm{db,b}}[t_{l,k,s}^{\mathrm{db,b}}], W_{l,k}^{\mathrm{db,b}}[t_{l,k,s}^{\mathrm{db,b}}]), \quad t_{l,k,s}^{\mathrm{db,b}} \in \mathcal{T}_{l,k}^{\mathrm{db,b}}, \tag{3.28}$$

where $\mathcal{L}_{X_{l,k}^{\mathrm{db,b}}(t_{l,k,s}^{\mathrm{db,b}})} \subseteq \mathcal{L}$, $Z_{l,k}^{\mathrm{db,b}}[t_{l,k,s}^{\mathrm{db,b}}] \subseteq Z[t_{l,k}^{\mathrm{upd}}]$, and $W_{l,k}^{\mathrm{db,b}}[t_{l,k,s}^{\mathrm{db,b}}] \subseteq Z[t_{l,k}^{\mathrm{upd}}]$ are the local posterior node set, local posterior measurement set, and local posterior anchor state set, respectively.

For local measurement sets, they not only refer to those measurements taken by node l, but also include the measurements between other nodes which are obtained by received messages or anchor measurements. We label the measurement set in database $\mathcal{B}_{l,k}$ as

$$Z_{l,k}^{\mathrm{db,ms}}(t_{l,k,s}^{\mathrm{db,ms}}) \ni z_{i,j}(t_{l,k,s}^{\mathrm{db,ms}}), \quad t_{l,k,s}^{\mathrm{db,ms}} \in \mathcal{T}_{l,k}^{\mathrm{db,ms}}, \tag{3.29}$$

where $i, j \in \mathcal{I}$, and the local measurement set $Z_{l,k}^{\mathrm{db,ms}}(t_{l,k,s}^{\mathrm{db,ms}})$ is a subset of the measurement set $Z(t_{l,k,s}^{\mathrm{db,ms}})$.

Similarly, for local anchor state sets, they not only refer to those only related to node l, but also contain the anchor states associated with other nodes. We label the anchor state set in database $\mathcal{B}_{l,k}$ as

$$W_{l,k}^{\mathrm{db,ms}}(t_{l,k,s}^{\mathrm{db,ms}}) \ni x_a(t_{l,k,s}^{\mathrm{db,ms}}), \quad t_{l,k,s}^{\mathrm{db,ms}} \in \mathcal{T}_{l,k}^{\mathrm{db,ms}}. \tag{3.30}$$

For a clearer description of the database, we label

$$\mathcal{T}_{l,k}^{\mathrm{db}} = \mathcal{T}_{l,k}^{\mathrm{db,a}} \bigcup \mathcal{T}_{l,k}^{\mathrm{db,b}} \bigcup \mathcal{T}_{l,k}^{\mathrm{db,ms}} \tag{3.31}$$

as the local timeline in database $\mathcal{B}_{l,k}$. At each time $t_{l,k,s}^{\mathrm{db}} \in \mathcal{T}_{l,k}^{\mathrm{db}}$, we use a quadruple to contain the local prior, posterior, measurement set, and anchor state set, i.e.,

$$b_{l,k}(t_{l,k,s}^{\mathrm{db}}) = \left\langle \hat{p}_{l,k}^{\mathrm{a}}(t_{l,k,s}^{\mathrm{db}}), \hat{p}_{l,k}^{\mathrm{b}}(t_{l,k,s}^{\mathrm{db}}), Z_{l,k}^{\mathrm{db,ms}}(t_{l,k,s}^{\mathrm{db}}), W_{l,k}^{\mathrm{db,ms}}(t_{l,k,s}^{\mathrm{db}}) \right\rangle. \tag{3.32}$$

If at $t_{l,k,s}^{\mathrm{db}}$, there is no prior or posterior, then the prior or posterior is undefined. Likewise, if at $t_{l,k,s}^{\mathrm{db}}$, the measurement or anchor state set does not exist, then it is empty. To sum up, $\mathcal{B}_{l,k}$ has the following structure:

$$\mathcal{B}_{l,k} = \left\{ b_{l,k}(t_{l,k,s}^{\mathrm{db}}) : t_{l,k,s}^{\mathrm{db}} \in \mathcal{T}_{l,k}^{\mathrm{db}} \right\}. \tag{3.33}$$

3.5.4 Broadcast Message

In the distributed prior-cut algorithm, only local priors, measurement sets, and anchor state sets are transmitted. Thus, the broadcast messages does not include any local posterior from

the database. To be more specific, each broadcast message has the following form

$$m_{l,k}^{\mathrm{bc}} = \left\{ b_{l,k}^{\mathrm{bc}}(t) \colon t \in \bar{\mathcal{T}}_{l,k}^{\mathrm{db}} \right\}, \tag{3.34}$$

where $b_{l,k}^{\mathrm{bc}}(t) = \left\langle \hat{p}_{l,k}^{\mathrm{a}}(t), Z_{l,k}^{\mathrm{db,ms}}(t), W_{l,k}^{\mathrm{db,ms}}(t) \right\rangle$. Also, $\bar{\mathcal{T}}_{l,k}^{\mathrm{db}} \subseteq \mathcal{T}_{l,k}^{\mathrm{db}}$ such that $b_{l,k}(t_{l,k,s}^{\mathrm{db}}) \neq b_{l,k-1}(t_{l,k,s}^{\mathrm{db}})$. That means each $m_{l,k}^{\mathrm{bc}}$ only contains the information different from that in the previous database.

3.5.5 Received Message

Since not all the parts of a broadcast message can be successfully received, the received message is a part of the broadcast message. Recall that at $t_{l,j,k}^{\mathrm{rc}}$, node l receives message $m_{l,j}^{\mathrm{rc}}(t_{l,j,k}^{\mathrm{rc}})$ from node j (see Section 3.2). To unify the time index, we still consider the time index $t_{l,k}^{\mathrm{upd}}$, since $\mathcal{T}_l^{\mathrm{rc}} \subseteq \mathcal{T}_l^{\mathrm{upd}}$ (see Section 3.5.3). At $t_{l,k}^{\mathrm{upd}}$, if node l receives message from node j, then we label it as

$$m_{l,j}^{\mathrm{rc}}(t_{l,k}^{\mathrm{upd}}) = \{ \mu_{l,j}^{\mathrm{rc}}(t_{l,j,k,s}^{\mathrm{mg}}) \colon t_{l,j,k,s}^{\mathrm{mg}} \in \mathcal{T}_{l,j,k}^{\mathrm{mg}} \}, \tag{3.35}$$

where $\mu_{l,j}^{\mathrm{rc}}(t_{l,j,k,s}^{\mathrm{mg}})$ is a triple containing the local prior, measurement set, and anchor state set from non-anchor node j, i.e.,

$$\mu_{l,j}^{\mathrm{rc}}(t_{l,j,k,s}^{\mathrm{mg}}) = \left\langle \hat{p}_{j,k'}^{\mathrm{a}}(t_{l,j,k,s}^{\mathrm{mg}}), Z_{j,k'}^{\mathrm{mg,ms}}(t_{l,j,k,s}^{\mathrm{mg}}), W_{j,k'}^{\mathrm{mg,ms}}(t_{l,j,k,s}^{\mathrm{mg}}) \right\rangle, \tag{3.36}$$

where $\hat{p}_{j,k'}^{\mathrm{a}}(t_{l,j,k,s}^{\mathrm{mg}}) = \hat{p}_j(X_{j,k'}^{\mathrm{db,a}}(t_{l,j,k,s}^{\mathrm{mg}}) | Z_{j,k'}^{\mathrm{db,a}}[t_{l,j,k,s}^{\mathrm{mg}}], W_{j,k'}^{\mathrm{db,a}}[t_{l,j,k,s}^{\mathrm{mg}}])$.[11]

We define $M_{l,j}^{\mathrm{rc}}(t_{l,k}^{\mathrm{upd}}) := \{m_{l,j}^{\mathrm{rc}}(t_{l,k}^{\mathrm{upd}})\}$, and let $M_i^{\mathrm{rc}}(t_{l,k}^{\mathrm{upd}}) := \bigcup_{j \in \mathcal{I}} M_{l,j}^{\mathrm{rc}}(t_{l,k}^{\mathrm{upd}})$, and also set $\mathcal{T}_{l,k}^{\mathrm{mg}} = \bigcup_{j \in \mathcal{I}} \mathcal{T}_{l,j,k}^{\mathrm{mg}}$. Then, it is readily to design the database update in Section 3.5.6.

3.5.6 Database Update

At each time instant $t_{l,k}^{\mathrm{upd}}$, the database of each non-anchor node $l \in \mathcal{L}$ is updated from $\mathcal{B}_{l,k-1}$ to $\mathcal{B}_{l,k}$ by the newly arrived measurements, anchor states, or messages, which is $Z_l(t_{l,k}^{\mathrm{upd}}) := \bigcup_{j \in \mathcal{I}} Z_{l,j}(t_{l,k}^{\mathrm{upd}})$, $W_l(t_{l,k}^{\mathrm{upd}}) := \bigcup_{a \in \mathcal{A}} W_{l,a}(t_{l,k}^{\mathrm{upd}})$ or $M_i^{\mathrm{rc}}(t_{l,k}^{\mathrm{upd}})$, respectively. The database update refreshes the local timeline [see (3.31)], measurement set, anchor state set, prior, and posterior, respectively.

We provide the local timeline update first, since it is the foundation of the other updates. Due to the limited computational capability and the storage space, $\mathcal{B}_{l,k}$ cannot include every information in the past, especially when k goes large. We use $N_l^{\max}$, the largest number of time instants to constrain the size of $\mathcal{B}_{l,k}$, which means the total number of time instants included in the database cannot exceed this number. As a result, the updated timeline $\mathcal{T}_{l,k}^{\mathrm{db}}$ [see (3.31)]

[11] The local prior, measurement set, and anchor state set correspond to those in (3.27), (3.29), and (3.30), respectively.

contains $N_l^{\max}$ most recent time instants in time set $\mathcal{T}_{l,k}^{\text{all}}$ defined as follows

$$\mathcal{T}_{l,k}^{\text{all}} := \begin{cases} \mathcal{T}_{l,k-1}^{\text{db}} \bigcup \{t_{l,k}^{\text{upd}}\} \bigcup \bar{\mathcal{T}}_{l,k}^{\text{mg}}, & \text{if } Z_l(t_{l,k}^{\text{upd}}) \neq \emptyset \text{ or } W_l(t_{l,k}^{\text{upd}}) \neq \emptyset, \\ \mathcal{T}_{l,k-1}^{\text{db}} \bigcup \bar{\mathcal{T}}_{l,k}^{\text{mg}}, & \text{otherwise,} \end{cases} \tag{3.37}$$

where $\bar{\mathcal{T}}_{l,k}^{\text{mg}}$ is the subset of $\mathcal{T}_{l,k}^{\text{mg}}$ satisfying at least one of the following three conditions:

1) $\forall t_{l,j,k,s}^{\text{mg}} \in \bar{\mathcal{T}}_{l,k}^{\text{mg}}$, $\mathcal{L}_{X_{j,k'}^{\text{db,a}}(t_{l,j,k,s}^{\text{mg}})} \bigcap \mathcal{L}_{X_{l,k}^{\text{db,a}}(\max \mathcal{T}_{l,k-1}^{\text{db}})} \neq \emptyset$;

2) $\forall t_{l,j,k,s}^{\text{mg}} \in \bar{\mathcal{T}}_{l,k}^{\text{mg}}$, $\mathcal{V}(Z_{j,k'}^{\text{mg,ms}}(t_{l,j,k,s}^{\text{mg}})) \bigcap \mathcal{L}_{X_{l,k}^{\text{db,a}}(\max \mathcal{T}_{l,k-1}^{\text{db}})} \neq \emptyset$;

3) $\#\mathcal{L}_{X_{l,k}^{\text{db,a}}(\max \mathcal{T}_{l,k-1}^{\text{db}})} \leq L_l^{\max}$.

Condition 1) tells that for each time instant $t_{l,j,k,s}^{\text{mg}}$ in $\bar{\mathcal{T}}_{l,k}^{\text{mg}}$, the corresponding local node set $\mathcal{L}_{X_{j,k'}^{\text{db,a}}(t_{l,j,k,s}^{\text{mg}})}$ in $\hat{p}_{j,k'}^{\text{a}}(t_{l,j,k,s}^{\text{mg}})$ from a received message must has a non-empty intersection with $\mathcal{L}_{X_{l,k}^{\text{db,a}}(\max \mathcal{T}_{l,k-1}^{\text{db}})}$ in node l's database. It implies that we should utilize those messages whose local prior is related to the most recent local prior in the previous database. Similarly, Condition 2) means we should consider those messages whose measurement set is correlated to the most recent local prior in the previous database. Condition 3) says that if the number of nodes stored in the previous database is smaller than a level $L_l^{\max}$,[12] the received message should be unconditionally included in the database update. Above complete the description of timeline update. For the other updates, they are given in Algorithm 3.2.

In Algorithm 3.2, the database $\mathcal{B}_{l,k}$ is updated element by element from $t_{l,k,s}^{\text{db}} = \min \mathcal{T}_{l,k}^{\text{db}}$ to $\max \mathcal{T}_{l,k}^{\text{db}}$. Lines between 1 and 21 provide the detailed update of $b_{l,k}(t_{l,k,s}^{\text{db}})$, which contain five building blocks: local measurement and anchor state update, prior fusion, prior cut, posterior update, and prior prediction. Note that the database for each non-anchor node $l \in \mathcal{L}$ is initialized as

$$\mathcal{B}_{l,0} := \mathcal{B}_l(0) = \{\langle \hat{p}_l(x_l(0)), \uparrow, \uparrow, \uparrow \rangle\}, \tag{3.38}$$

which further specifies the initial condition in Algorithm 3.1. In (3.38), each $\uparrow$ stands for an undefined term. For $\mathcal{B}_{l,0}$, the local posterior, measurement set, and anchor state set are undefined.

3.5.6.1 Local measurement and anchor state update

From Line 2 to Line 8 of Algorithm 3.2, the local measurement set $Z_{l,k}^{\text{db,ms}}(t_{l,k,s}^{\text{db}})$ and the local anchor state set $W_{l,k}^{\text{db,ms}}(t_{l,k,s}^{\text{db}})$ in $b_{l,k}(t_{l,k,s}^{\text{db}})$ are updated. If $t_{l,k,s}^{\text{db}}$ equals $t_{l,k}^{\text{upd}}$, then the updated local measurement set is composed by the set of newly taken measurements $Z_l(t_{l,k}^{\text{upd}})$ and the set of measurements from the old database $Z_{l,k-1}^{\text{db,ms}}(t_{l,k,s}^{\text{db}})$ (see Line 3); and the updated local anchor state set consists of newly received local anchor state set $W_l(t_{l,k}^{\text{upd}})$ and the local anchor

[12] It is a parameter in this algorithm which constrains the sizes of prior and posterior, see Section 3.5.6.3.

Algorithm 3.2 Distributed Prior-Cut Algorithm: Database Update

1: **for** $t^{\mathrm{db}}_{l,k,s} := \min \mathcal{T}^{\mathrm{db}}_{l,k}$ **to** $\max \mathcal{T}^{\mathrm{db}}_{l,k}$ **do**
2: **if** $t^{\mathrm{db}}_{l,k,s} == t^{\mathrm{upd}}_{l,k}$ **then**
3: $Z^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db}}_{l,k,s}) = Z_l(t^{\mathrm{upd}}_{l,k}) \bigcup Z^{\mathrm{db,ms}}_{l,k-1}(t^{\mathrm{db}}_{l,k,s})$;
4: $W^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db}}_{l,k,s}) = W_l(t^{\mathrm{upd}}_{l,k}) \bigcup W^{\mathrm{db,ms}}_{l,k-1}(t^{\mathrm{db}}_{l,k,s})$;
5: **else**
6: $Z^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db}}_{l,k,s}) = Z^{\mathrm{mg,ms}}_{l,k}(t^{\mathrm{mg}}_{l,k,s}) \bigcup Z^{\mathrm{db,ms}}_{l,k-1}(t^{\mathrm{db}}_{l,k,s})$;
7: $W^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db}}_{l,k,s}) = W^{\mathrm{mg,ms}}_{l,k}(t^{\mathrm{mg}}_{l,k,s}) \bigcup W^{\mathrm{db,ms}}_{l,k-1}(t^{\mathrm{db}}_{l,k,s})$;
8: **end if**
9: **for each** valid l' **do**
10: $j^*_{l,l',k,s} = \arg\min_j \left\{ \widehat{\mathrm{Var}}^{\mathrm{a}}_j \left[x_{l'}(t^{\mathrm{db}}_{l,k,s}) \right] : j \in \mathcal{J}^{\mathrm{rc}}_{l,k,s} \bigcup \{l, -1\} \right\}$;
11: **end for**
12: $\tilde{p}^{\mathrm{a}}_{l,k}(t^{\mathrm{db,a}}_{l,k,s}) = \prod_{j \in \mathcal{J}^*_{l,k,s}} \hat{p}^{\mathrm{a}}_{j,k'} \left(t^{\mathrm{db}}_{l,k,s}, \mathcal{J}^*_{l,k,s}(j) \right)$;
13: $\langle \bar{Z}^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db}}_{l,k,s}), X^{\mathrm{db,a}}_{l,k}(t^{\mathrm{db}}_{l,k,s}) \rangle = \arg\min \Delta^{\mathrm{db}}_{l,k,s}$;
14: $\hat{p}^{\mathrm{a}}_{l,k}(t^{\mathrm{db}}_{l,k,s}) \leftarrow$ (3.46);
15: **if** $Z^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db}}_{l,k,s}) \neq \emptyset || W^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db}}_{l,k,s}) \neq \emptyset$ **then**
16: $\hat{p}_l(X^{\mathrm{db,a}}_{l,k}(t^{\mathrm{db}}_{l,k,s}) | Z^{\mathrm{db,b}}_{l,k}[t^{\mathrm{db}}_{l,k,s}]) \leftarrow$ (3.47);
17: **end if**
18: **if** $t^{\mathrm{db}}_{l,k,s} < \max \mathcal{T}^{\mathrm{db}}_{l,k}$ **then**
19: $\hat{p}_{-1}(X^{\mathrm{db,a}}_{l,k}(t^{\mathrm{db}}_{l,k,s+1}) | Z^{\mathrm{db,a}}_{l,k}[t^{\mathrm{db}}_{l,k,s}]) \leftarrow$ (3.48);
20: **end if**
21: **end for**

state set in the old database $W^{\mathrm{db,ms}}_{l,k-1}(t^{\mathrm{db}}_{l,k,s})$ (see Line 4). If $t^{\mathrm{db}}_{l,k,s}$ is not equal to $t^{\mathrm{upd}}_{l,k}$, then the updated measurement set includes the set of measurements provided by the newly arrived messages $Z^{\mathrm{mg,ms}}_{l,k}(t^{\mathrm{mg}}_{l,k,s})$ and the set of measurements from the old database $Z^{\mathrm{db,ms}}_{l,k-1}(t^{\mathrm{db}}_{l,k,s})$ (see Line 6), where $Z^{\mathrm{mg,ms}}_{l,k}(t^{\mathrm{mg}}_{l,k,s}) := \bigcup_{j \in \mathcal{I}} Z^{\mathrm{mg,ms}}_{j,k'}(t^{\mathrm{mg}}_{l,k,s})$; and similarly the updated local anchor state set consists of the local anchor state sets provided by the newly arrived messages and the old database, respectively (see Line 7), where $W^{\mathrm{mg,ms}}_{l,k}(t^{\mathrm{mg}}_{l,k,s}) := \bigcup_{j \in \mathcal{I}} W^{\mathrm{mg,ms}}_{j,k'}(t^{\mathrm{mg}}_{l,k,s})$ in Line 7.

3.5.6.2 Local prior fusion

Lines 9-14 of Algorithm 3.2 return the fused local prior

$$\tilde{p}^{\mathrm{a}}_{l,k}(t^{\mathrm{db,a}}_{l,k,s}) := \hat{p}_l(\tilde{X}^{\mathrm{db,a}}_{l,k}(t^{\mathrm{db,a}}_{l,k,s}) | Z^{\mathrm{db,a}}_{l,k}[t^{\mathrm{db,a}}_{l,k,s}], W^{\mathrm{db,a}}_{l,k}[t^{\mathrm{db,a}}_{l,k,s}]), \quad t^{\mathrm{db,a}}_{l,k,s} \in \mathcal{T}^{\mathrm{db,a}}_{l,k}. \tag{3.39}$$

Firstly, we define the local prior variance $\widehat{\mathrm{Var}}^{\mathrm{a}}_j \left[x_{l'}(t^{\mathrm{db}}_{l,k,s}) \right]$ as follows:

$$\widehat{\mathrm{Var}}^{\mathrm{a}}_j \left[x_{l'}(t^{\mathrm{db}}_{l,k,s}) \right] := \int_{\mathcal{X}^{\mathrm{db,a}}_{j,k',s}} \left\{ \widehat{\mathbb{E}}^{\mathrm{a}}_j \left[x_{l'}(t^{\mathrm{db}}_{l,k,s}) \right] - x_{l'}(t^{\mathrm{db}}_{l,k,s}) \right\}^2 \hat{p}^{\mathrm{a}}_{j,k'}(t^{\mathrm{db}}_{l,k,s}) \mathrm{d} X^{\mathrm{db,a}}_{j,k'}(t^{\mathrm{db}}_{l,k,s}), \tag{3.40}$$

where $k' = k$ for $j = l$, $X^{\mathrm{db,a}}_{j,k'}(t^{\mathrm{db}}_{l,k,s}) \in \mathcal{X}^{\mathrm{db,a}}_{j,k',s}$, and

$$\widehat{\mathbb{E}}^{\mathrm{a}}_{j}\left[x_{l'}(t^{\mathrm{db}}_{l,k,s})\right] := \int_{\mathcal{X}^{\mathrm{db,a}}_{j,k',s}} x_{l'}(t^{\mathrm{db}}_{l,k,s})\hat{p}^{\mathrm{a}}_{j,k'}(t^{\mathrm{db}}_{l,k,s})\mathrm{d}X^{\mathrm{db,a}}_{j,k'}(t^{\mathrm{db}}_{l,k,s}). \tag{3.41}$$

Variance $\widehat{\mathrm{Var}}^{\mathrm{a}}_{j}\left[x_{l'}(t^{\mathrm{db}}_{l,k,s})\right]$ reflects the uncertainty of $x^{\mathrm{a}}_{l'}(t^{\mathrm{db}}_{l,k,s})$ under local prior $\hat{p}^{\mathrm{a}}_{j,k'}(t^{\mathrm{db}}_{l,k,s})$ which comes from: $b_{l,k-1}(t^{\mathrm{db}}_{l,k,s})$ (i.e., $j = l$), $\mu^{\mathrm{rc}}_{l,j}(t^{\mathrm{db}}_{l,k,s})$ (i.e., $j \notin \{l,-1\}$, and all such j form the set $\mathcal{J}^{\mathrm{rc}}_{l,k,s}$) or the predicted local prior from the previous time instant (i.e., $j = -1$, see Section 3.5.6.5). For different j but the same state $x_{l'}(t^{\mathrm{db}}_{l,k,s})$, the variance $\widehat{\mathrm{Var}}^{\mathrm{a}}_{j}\left[x_{l'}(t^{\mathrm{db}}_{l,k,s})\right]$ differs. The smaller the variance is, the less uncertainty the state $x_{l'}(t^{\mathrm{db}}_{l,k,s})$ should have. Thus, the marginal pdf of $\hat{p}^{\mathrm{a}}_{j,k'}(t^{\mathrm{db}}_{l,k,s})$ w.r.t. state $x_{l'}(t^{\mathrm{db}}_{l,k,s})$ with the smallest variance should be chosen as the updated marginal pdf. This process is conducted from Line 9 to Line 11 for each valid state $x_{l'}(t^{\mathrm{db}}_{l,k,s})$, i.e., it should be included in at least one of the message $m^{\mathrm{rc}}_{l,j}(t^{\mathrm{upd}}_{l,k})$, or in the old database $\mathcal{B}_{i,k-1}$, or in the predicted prior from $t^{\mathrm{db}}_{i,k,s-1}$. Note that if $s = 1$, the predicted pdf is derived from the nearest time instant in the old database. For each valid l', the best marginal prior's index[13] $j^*_{l,l',k,s}$ is selected by Line 10 among indices $j \in \mathcal{J}^{\mathrm{rc}}_{l,k,s} \bigcup \{l,-1\}$. All such $j^*_{l,l',k,s}$ (for all valid l') form the set $\mathcal{J}^*_{l,k,s}$, and we set $\mathcal{J}^*_{l,k,s}(j) = \{j' \colon j^*_{l,l',k,s} = j\}$, and

$$\hat{p}^{\mathrm{a}}_{j,k'}\left(t^{\mathrm{db}}_{l,k,s}, \mathcal{J}^*_{l,k,s}(j)\right) = \int_{\check{\mathcal{X}}^*_{j,k,s}} \hat{p}^{\mathrm{a}}_{j,k'}\left(t^{\mathrm{db}}_{l,k,s}\right) \mathrm{d}\check{X}^*_{j,k,s}, \tag{3.42}$$

where $\mathcal{L}_{\check{X}^*_{j,k,s}} = \mathcal{L}_{X^{\mathrm{db,a}}_{j,k'}(t^{\mathrm{db}}_{l,k,s})} \setminus \mathcal{J}^*_{l,k,s}(j)$, and $\check{X}^*_{j,k,s} \in \check{\mathcal{X}}^*_{j,k,s}$. Then, we reconstruct the local prior (i.e., the fused local prior) from the selected marginal priors, which is given in Line 12: For the marginal priors from node j, they still remain the same joint distribution $\hat{p}^{\mathrm{a}}_{j,k'}\left(t^{\mathrm{db}}_{l,k,s}, \mathcal{J}^*_{l,k,s}(j)\right)$. Since the relationship among the priors in different nodes j are unknown, we assume they are independent of each other, which means $\tilde{p}^{\mathrm{a}}_{l,k}(t^{\mathrm{db,a}}_{l,k,s})$ is the product of different $\hat{p}^{\mathrm{a}}_{j,k'}\left(t^{\mathrm{db}}_{l,k,s}, \mathcal{J}^*_{l,k,s}(j)\right)$.

Remark 3.6. Note that the marginal priors can be from all possible nodes who broadcast messages in this network, and as a result, the fused local prior would contain a large number of nodes' information. Thus, it is impractical if

- we put the fused local prior in the broadcast message, since the communication cost would be very large;
- the whole fused local prior is used to do further calculations (including Bayesian inference and local prior prediction), because the computational capability of a node is limited;
- we store the fused local prior in the new database, as the storage space is limited.

[13] This means the best marginal prior is provided by node $j^*_{l,l',k,s}$.

Therefore, it is necessary to reduce the size of the fused local prior. This size reduction is based on the local prior cut (see Section 3.4.2) and given in Lines 13 and 14 of Algorithm 3.2.

3.5.6.3 Local prior cut

We explain the meaning of Line 13 as follows. Firstly, we define the state-number-constraint AMS (SNCAMS) triple. Similar to the AMS triple in Definition 3.2, we need the local measurement graph based on the local measurement set $Z_{l,k}^{\mathrm{db,ms}}(t_{l,k,s}^{\mathrm{db}})$, i.e.,

$$\mathcal{G}_{l,k,s}^{\mathrm{db}} := \left(\mathcal{V}(Z_{l,k}^{\mathrm{db,ms}}(t_{l,k,s}^{\mathrm{db}})), \mathcal{E}(Z_{l,k}^{\mathrm{db,ms}}(t_{l,k,s}^{\mathrm{db}}))\right). \tag{3.43}$$

Then the local measurement cluster can be defined as $\mathcal{C}_{l,k,s,r}^{\mathrm{db}}$, where $r \in \{1,\ldots,R_{l,k,s}^{\mathrm{db}}\} =: \mathcal{R}_{l,k,s}^{\mathrm{db}}$. Thus, the (local) AMS triple $\langle \bar{Z}_{i,k}^{\mathrm{db,ms}}(t_{l,k,s}^{\mathrm{db}}), X_{l,k}^{\mathrm{db,a}}(t), \bar{W}_{i,k}^{\mathrm{db,ms}}(t_{l,k,s}^{\mathrm{db}})\rangle$ is defined according to Definition 3.2, where $\bar{Z}_{l,k}^{\mathrm{db,ms}}(t_{l,k,s}^{\mathrm{db}}) \subseteq Z_{l,k}^{\mathrm{db,ms}}(t_{l,k,s}^{\mathrm{db}})$, $X_{l,k}^{\mathrm{db,a}}(t_{l,k,s}^{\mathrm{db}}) \subseteq \check{X}_{l,k}^{\mathrm{db,a}}(t_{l,k,s}^{\mathrm{db}})$, and $\bar{W}_{l,k}^{\mathrm{db,ms}}(t_{l,k,s}^{\mathrm{db}}) \subseteq W_{l,k}^{\mathrm{db,ms}}(t_{l,k,s}^{\mathrm{db}})$. To give the definition of SNCAMS triple, we introduce the state-generating node set $\mathcal{L}_{l,k,s}^{\mathrm{gen}}$, and it has the following relationship with $X_{l,k}^{\mathrm{db,a}}(t_{l,k,s}^{\mathrm{db}})$:

$$\mathcal{L}_{X_{l,k}^{\mathrm{db,a}}(t_{l,k,s}^{\mathrm{db}})} = \left\{ l' \colon (l',j) \in \mathcal{E}(Z_{l,k}^{\mathrm{db,ms}}(t_{l,k,s}^{\mathrm{db}})) \vee (j,l') \in \mathcal{E}(Z_{l,k}^{\mathrm{db,ms}}(t_{l,k,s}^{\mathrm{db}})), j \in \mathcal{L}_{l,k,s}^{\mathrm{gen}} \right\}. \tag{3.44}$$

If we only consider the state-generating node set with number constraint $L_l^{\max}$, i.e., $\#\mathcal{L}_{l,k,s}^{\mathrm{gen}} = L_l^{\max}$, then the generated AMS triple is called $L_l^{\max}$-SNCAMS triple.

For $L_l^{\max}$-SNCAMS triples, we have the similar results as those in Theorem 3.1: Cut local prior $\tilde{p}_{l,k}^{\mathrm{a}}(t_{l,k,s}^{\mathrm{db}})$ into two sub-priors, i.e., $\hat{p}_{l,k}^{\mathrm{a}}(t_{l,k,s}^{\mathrm{db}})$ and $\tilde{p}_{l,k,s}^{\mathrm{left}}$ which are the local priors of $X_{l,k}^{\mathrm{db,a}}(t)$ and $\check{X}_{l,k}^{\mathrm{db,a}}(t)$, respectively. From the perspective of node l, if the cut gap is small enough, then there is no big difference between the original local location estimation and the estimation only considering the $L_l^{\max}$-SNCAMS triple.

Let $\mathcal{U}_{l,k,s}^{\mathrm{db}}(L_l^{\max})$ be the set of all possible $L_l^{\max}$-SNCAMS triples at $t = t_{l,k,s}^{\mathrm{db}}$, which means it is the set of $\langle \bar{Z}_{l,k}^{\mathrm{db,ms}}(t_{l,k,s}^{\mathrm{db}}), X_{l,k}^{\mathrm{db,a}}(t_{l,k,s}^{\mathrm{db}}), \bar{W}_{l,k}^{\mathrm{db,ms}}(t_{l,k,s}^{\mathrm{db}})\rangle$ with states-number constraint $L_l^{\max}$, labeled as $u_{l,k,s}^{\mathrm{db}}$. Then, we define the set of cut gaps corresponding to $\mathcal{U}_{l,k,s}^{\mathrm{db}}(L_l^{\max})$ as follows

$$\Delta_{l,k,s}^{\mathrm{db}} = \left\{ \left\| \tilde{p}_{l,k}^{\mathrm{a}}(t_{l,k,s}^{\mathrm{db}}) - \hat{p}_{l,k}^{\mathrm{a}}(t_{l,k,s}^{\mathrm{db}}) \tilde{p}_{l,k,s}^{\mathrm{left}} \right\|_{\infty} \colon u_{l,k,s}^{\mathrm{db}} \in \mathcal{U}_{l,k,s}^{\mathrm{db}}(L_l^{\max}) \right\}. \tag{3.45}$$

Line 13 selects the SNCAMS triple corresponding to the smallest element in $\Delta_{l,k,s}^{\mathrm{db}}$, and finally Line 14 gives the local prior $\hat{p}_{l,k}^{\mathrm{a}}(t_{l,k,s}^{\mathrm{db}})$ stored in the database, specifically:

$$\hat{p}_{l,k}^{\mathrm{a}}(t_{l,k,s}^{\mathrm{db}}) = \int_{\check{\mathcal{X}}_{l,k,s}^{\mathrm{db,a}}} \tilde{p}_{l,k}^{\mathrm{a}}(t_{l,k,s}^{\mathrm{db}}) \mathrm{d}\check{X}_{l,k}^{\mathrm{db,a}}(t_{l,k,s}^{\mathrm{db}}), \tag{3.46}$$

where $\check{X}_{l,k}^{\mathrm{db,a}}(t_{l,k,s}^{\mathrm{db}}) \in \check{\mathcal{X}}_{l,k,s}^{\mathrm{db,a}}$.

3.5.6.4 Local posterior update

Lines 15-17 provide the local posterior update. If the local measurement set or anchor state set is not empty, then the local posterior can be derived by Bayes' rule

$$\hat{p}^{\mathrm{a}}_{l,k}(t^{\mathrm{db,b}}_{l,k,s}) = \frac{p(\bar{Z}^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db}}_{l,k,s})|X^{\mathrm{db,a}}_{l,k}(t^{\mathrm{db}}_{l,k,s}))\hat{p}^{\mathrm{a}}_{l,k}(t^{\mathrm{db}}_{l,k,s})}{\int_{\mathcal{X}^{\mathrm{db,a}}_{l,k,s}} p(\bar{Z}^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db}}_{l,k,s})|X^{\mathrm{db,a}}_{l,k}(t^{\mathrm{db}}_{l,k,s}))\hat{p}^{\mathrm{a}}_{l,k}(t^{\mathrm{db}}_{l,k,s})\mathrm{d}X^{\mathrm{db,a}}_{l,k}(t^{\mathrm{db}}_{l,k,s})}. \tag{3.47}$$

3.5.6.5 Local prior prediction

From Line 18 to Line 20, the updated local posterior is employed to predict the local prior at the next time instant $t^{\mathrm{db}}_{l,k,s+1}$, if $t^{\mathrm{db}}_{l,k,s}$ is not the last time instant in $\mathcal{T}^{\mathrm{db}}_{l,k}$. This prediction is given in Line 19, where

$$\hat{p}^{\mathrm{a}}_{-1,k}(t^{\mathrm{db}}_{l,k,s+1}) = \int_{\mathcal{X}^{\mathrm{db,a}}_{l,k,s}} p(X^{\mathrm{db,a}}_{l,k}(t^{\mathrm{db}}_{l,k,s+1})|X^{\mathrm{db,a}}_{l,k}(t^{\mathrm{db}}_{l,k,s}))\hat{p}^{\mathrm{a}}_{l,k}(t^{\mathrm{db}}_{l,k,s})\mathrm{d}X^{\mathrm{db,a}}_{l,k}(t^{\mathrm{db}}_{l,k,s}). \tag{3.48}$$

3.5.7 Algorithm Summary

We give a summary of the distributed prior-cut algorithm in this subsection. For each non-anchor node $l \in \mathcal{L}$, it has an initial prior $\hat{p}_l(x(0))$ on its state $x_l(0)$ at $t = 0$. Since node l is not able to obtain its location $y_l(t)$ (see Section 3.2), it takes measurements from other nodes (see Section 3.2.1), and broadcasts/receives messages to/from other nodes (see Section 3.2.2). Note that node l does not know when the next measurement or the next message will come (see Section 3.3.1). Based on the measurements and messages, the distributed prior-cut algorithm estimates $y_l(t)$ in a real-time manner (the structure of this algorithm is given in Algorithm 3.1):

- If node l takes a measurement from a non-anchor node, then the database (whose structure is shown in Section 3.5.3) is updated according to this measurement.
- If node l takes a measurement from an anchor node, then it will receive the anchor state from that anchor node immediately (see Section 3.3.1.1), and the database is updated according to this measurement and the anchor state.
- If an anchor node takes a measurement from node l, then it will receive this measurement as well as the anchor state immediately (see Section 3.3.1.1), and the database is updated according to this measurement and the anchor state.
- If node l receives a message (the structure of the received message is given in Section 3.5.5), then the database is updated according to this message.

Note that these four situations can happen simultaneously. The detailed database update, including the design of local probability, is given in Section 3.5.6. After updating the database,

node l designs the broadcast message accordingly (see Section 3.5.4). We stress that the designed message is not necessarily to be broadcasted directly due to the limited bandwidth, and whether it can be successfully broadcasted is dependent on the MAC layer (an example is given in Appendix B.3 which is employed by our simulation examples in Section 3.6). After designing the broadcast message, node l calculates the local prior based on the updated database, and estimates its location $y_l(t)$ correspondingly. Note that even if no measurements or messages arrive, node l still needs to calculate the local prior and estimate its location $y_l(t)$ but based on the current database (without update). Measurements and messages play the role in updating the database and designing broadcast message.

3.6 Simulation Results

In this section, simulations are given to corroborate the effectiveness of our proposed distributed prior-cut algorithm. We consider two different localization problems: the mobile user cooperative localization (Section 3.6.1), and the UAV localization in scanning task (Section 3.6.2). Note that the simulations are conducted under practical wireless network settings, where the physical and MAC layers are properly modeled.

3.6.1 Mobile User Cooperative Localization

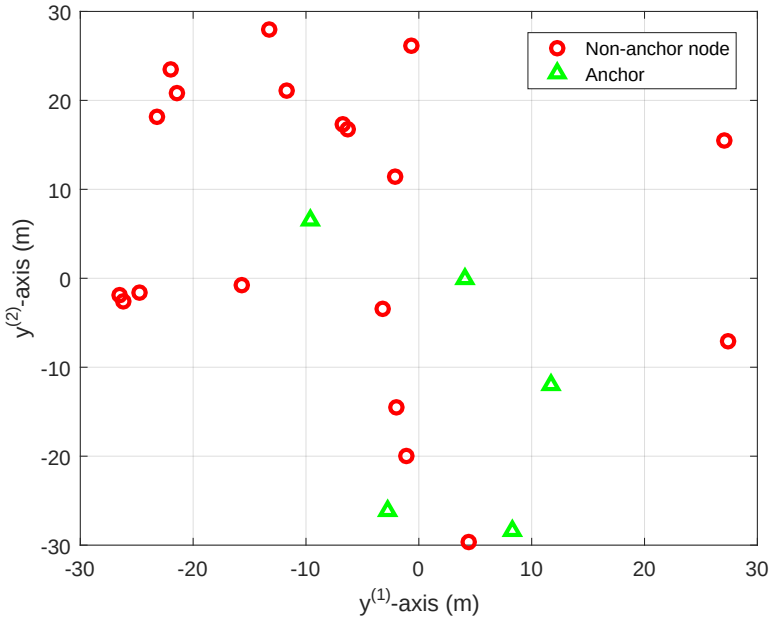
Consider a network with 25 mobile users ($I = 25$), where only 5 users (anchor nodes) can access their real-time locations timely. For example, these locations can be obtained from the GPS signals, base station measurements for outdoor positioning systems; or WiFi based localization for indoor positioning systems. For the other 20 users (non-anchor nodes), they conduct the self-localization cooperatively by using our proposed distributed prior-cut algorithm during time duration $[0,\ 50]$.

3.6.1.1 Mobility Model

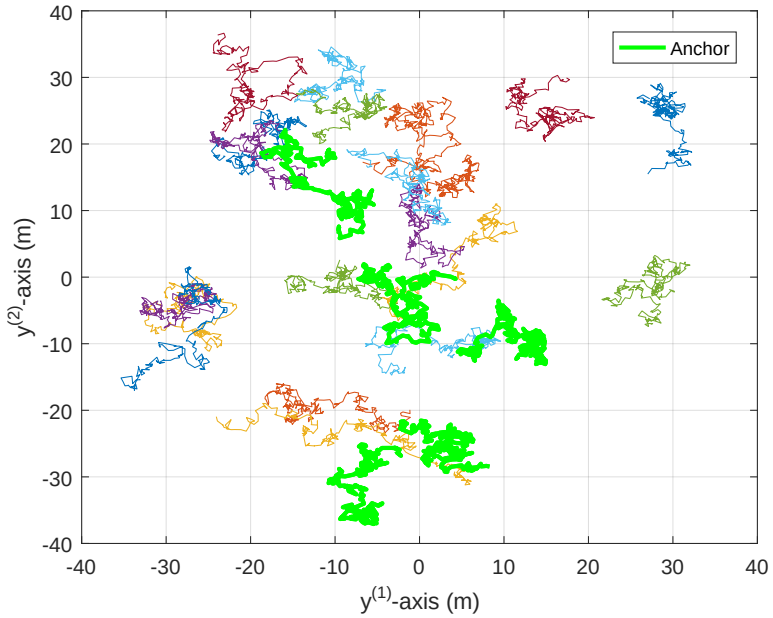
The initial locations of these 25 users are uniformly distributed in a 60m $\times$ 60m area, see Fig. 3.6(a) for one realization. Each node does not know the initial location accurately, but we assume its guess on the initial location is within a 10m $\times$ 10m area centered at the true location. The mobility model of mobile users is govern by a 2-dimensional Browian motion, which can be described by (3.1) and (3.2) with

$$f_i(x_i(t),t) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad g_i(x_i(t)) = x_i(t), \quad \bar{Q}_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad i \in \mathcal{I}, \tag{3.49}$$

where $x_i(t), y_i(t) \in \mathbb{R}^2$. One realization of the location trajectories is given in Fig. 3.6(b).



(a)



(b)

Figure 3.6: An example of the mobile users' motions: (a) initial locations, where the circles and triangles are the locations of non-anchor nodes and anchor nodes, respectively; (b) location trajectories, where the thin and thick lines represent the trajectories of non-anchor nodes and anchor nodes, respectively.

3.6.1.2 Measurement model

Each node can take measurements from at most 5 neighbors at one time instant, and we assume these 5 neighbors are nearest to the node.[14] For each measurement link, the measuring time instant follows the Poisson process with rate $\lambda = 5$, i.e., a measurement is taken every 0.2 second on average. We assume the measurement noise in (3.4) follows a zero-mean Gaussian distribution with standard deviation 0.1, i.e., the measurement accuracy is at the level of 0.1m. Note that the neighbors are changing over time due to the mobility of users.

3.6.1.3 Communication model

In the distributed prior-cut algorithms, the non-anchor nodes interchange their information through broadcasting in wireless communications. We assume all the non-anchor nodes share a $B = 40$MHz communication bandwidth. To present our simulation in a more explicit way, we move the communication details in Appendix B.3, where the physical layer power control is with the on-off structure and the media access control (MAC) layer protocol is carrier-sense multiple access (CSMA). It should be noted that the communications are fully asynchronous, i.e., they are not scheduled to happen exactly in each time slot synchronized by the whole network.

3.6.1.4 Prior-cut algorithm

This algorithm just has two parameters, one is the memory length $N_l^{\max}$, and the other is the constraint for state-generating node set $L_l^{\max}$ which determines the size of local prior. In this problem, we set $N_l^{\max} = 5$ and $L_l^{\max} = 8$. For the prior prediction and posterior update, we use the continuous-discrete unscented Kalman filter (UKF).[15]

3.6.1.5 Results and comparisons

We compare the average RMSE among the distributed prior-cut algorithm, the localization without cooperation, and the centralized localization algorithm for 100 simulation runs. The results are shown in Fig. 3.7. We can see that on average, each node's localization error is within 0.5m. This result is very close to the centralized algorithm and much better than the localization algorithm without cooperation.

[14] It should be noted that the node never knows its distances between their neighbors. This assumption just implies that the measurement can be taken more easily between closeby nodes, and the other measurements corresponding to longer distances are neglected in the simulation.

[15] Since the local priors and posteriors are approximated by Gaussian distribution in the UKF, the cut gap is not easy to calculate (for particle filters, the cut gap is easy to calculate), and for the local prior cut in Section 3.5.6.3, we use the linear correlation (averaged on each component of a state) instead. This is reasonable, because the correlation reflects the dependence of two Gaussian variables.

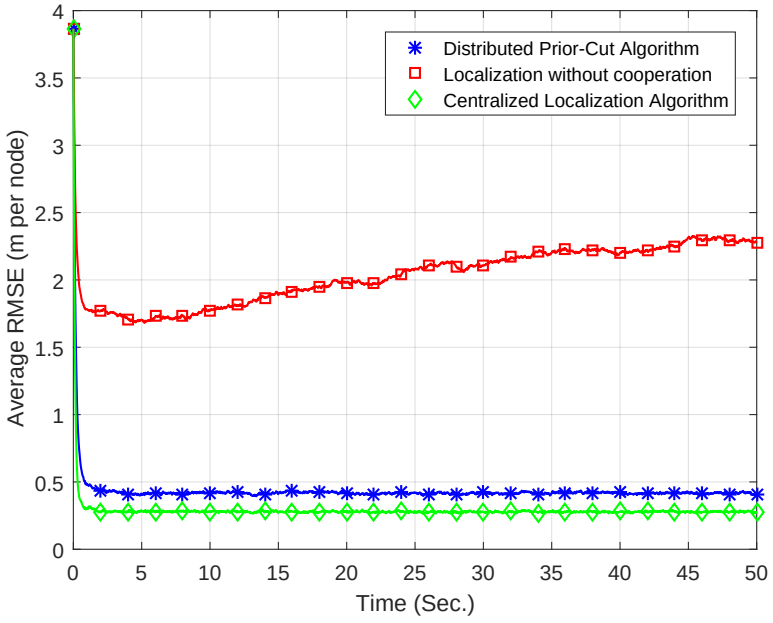


Figure 3.7: Comparisons of the distributed prior-cut algorithm, the localization without cooperation, and the centralized localization algorithm for mobile user cooperative localization. The average root-mean-square error (RMSE) means the square root of $\mathbb{E}\|\hat{y}_l(t) - y_l(t)\|$ (see Problem 3.1) averaged by all nodes $l \in \mathcal{L}$. The average RMSE of the distributed prior-cut algorithm converges to 0.43m/node within 2 seconds, which is comparable to the centralized algorithm. For the localization without cooperation, even though the average RMSE experienced a decrease (around 1.7m/node) during time interval $[0, 5]$, it goes worse and reach 2.3m/node at $t = 50$sec.

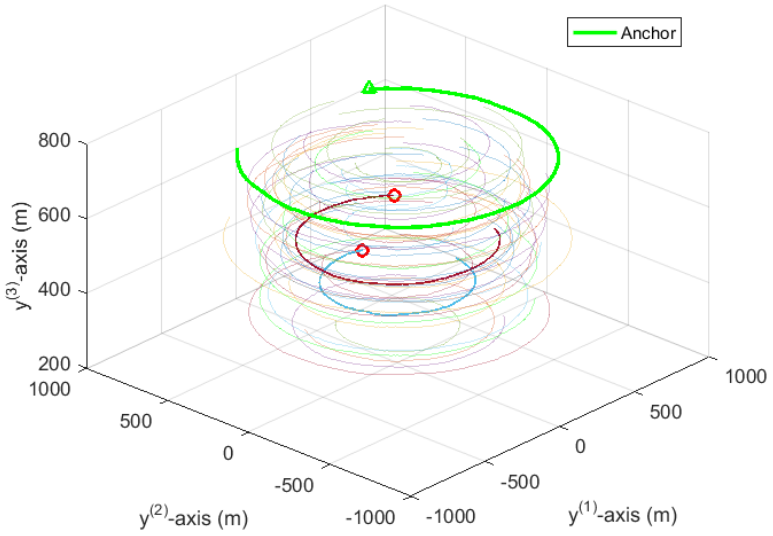


Figure 3.8: Location trajectories of UAVs. From the top to bottom, three highlighted UAVs are UAV 42 (non-anchor nodes), UAVs 7 and 3 (anchor nodes), respectively. For these three UAVs, the triangle and circles label the initial locations of anchor and non-anchor nodes, respectively.

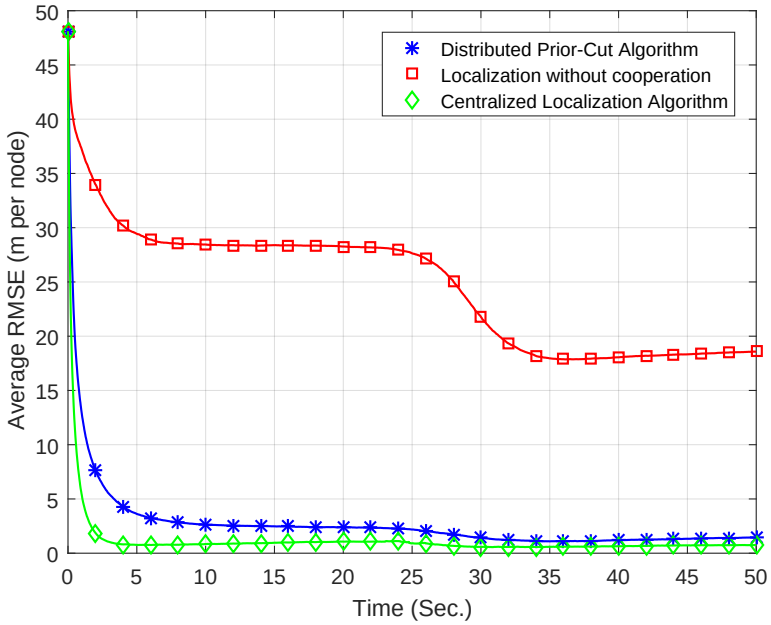


Figure 3.9: Comparisons for UAV localization in scanning task. The average RMSE of the distributed prior-cut algorithm reaches 3.5m/node at $t = 5$sec and 1.5m/node at $t = 30$sec, which is comparable to the centralized algorithm. For the localization without cooperation, the average RMSE cannot be smaller than 17m/node.

3.6.2 UAV Localization in Scanning Task

Consider 45 UAVs carrying out a scanning task for a region by hovering around the same point, where only 5 UAVs (anchor nodes) can access their own location timely and the other 40 UAVs are non-anchor nodes to be localized. The simulation period is $[0,\ 50]$.

3.6.2.1 Mobility model

The initial locations of these 45 UAVs are uniformly placed in a 500m $\times$ 500m $\times$ 500m region $[100,\ 600] \times [100,\ 600] \times [300,\ 800]$. Similar to Section 3.6.1.1, each UAV does not know the initial location, but we assume its guess on the initial location is within a 100m $\times$ 100m $\times$ 100m region centered at the true location. The mobility model of UAVs is govern by (3.1) and (3.2) with

$$f_i(x_i(t),t) = \begin{bmatrix} 0 & -\omega_i & 0 \\ \omega_i & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} x_i(t), \quad g_i(x_i(t)) = x_i(t), \quad \bar{Q}_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad i \in \mathcal{I}, \quad (3.50)$$

where $x_i(t), y_i(t) \in \mathbb{R}^3$. In (3.50), ω_i is the hover angular velocity for UAV i. For this problem, we assume $\omega_1 = \omega_3 = \ldots = \omega_{45} = 0.1$rad/s and $\omega_2 = \omega_4 = \ldots = \omega_{44} = -0.1$rad/s, in which

the positive angular velocity means the hover is counter-clock-wise from the bird's-eye view. One realization of the location trajectories is given in Fig. 3.8, where two non-anchor nodes and an anchor node are highlighted.

3.6.2.2 Measurement and communication models

The measurement model is the same as that in Section 3.6.1.2. For communication model, we still use a similar model in Section 3.6.1.3 (these parameters below can be found in Appendix B.3) with a small difference highlighted as follows: The transmit power is 0.5W. The path loss exponent is 2. The silence distance is 20m.

3.6.2.3 Results and comparisons

For the distributed prior-cut algorithm, we use the same parameters as that in Section 3.6.1.4, i.e., $N_l^{\max} = 5$ and $L_l^{\max} = 8$. For the prior prediction and posterior update, we still use the UKF. The comparison of the distributed prior-cut algorithm, the localization without cooperation, and the centralized algorithm averaged by 100 simulation runs is given in Fig. 3.9. We can see that the distributed prior-cut algorithm has a close performance to the centralized algorithm, and outperforms the localization without cooperation.

3.7 Summary

In this chapter, the mobile nodes' cooperative localization problem with asynchronous communications and measurements has been studied. We have modeled the node mobility by using the stochastic differential equation, and employed the continuous-discrete Bayesian filter to give the centralized localization algorithm which utilizes the global information. An important concepts AMS triple, prior cut, and cut gap, have been introduced in analyzing the estimation gap between the centralized algorithm and the algorithm with an AMS triple. We have strict proved that if the cut gap is small enough, i.e., the two cut parts are nearly independent, then the estimation gap can also be very small. With this important property, we have designed the prior-cut algorithm to solve the cooperative localization problem with asynchronous communications and measurements.

Chapter 4

Finite-Horizon Throughput Region for Multi-User Interference Channels

4.1 Introduction

In Chapter 2, the location-based interference prediction has been studied, and in Chapter 3, a cooperative localization method is proposed to refine the location information. These two chapters establish the foundation on understanding the short-term behavior of a communication system in MWNs. Specifically, the communication channel can be predicted by $H_{i,0}(t|s) = h_i\, g\big(\|\bar{y}_i(t|s)\|\big)$, which is a special case of the interference prediction (2.15). Since all the communication channels can be predicted in a real-time manner, the finite-horizon throughput region can be analyzed. Note that the finite-horizon throughput region reflects the fundamental limit on independent transmitter-receiver pairs for a short time window.

This chapter studies the throughput region of a MWN consisting of multiple transmitter-receiver pairs. Previously, the throughput region of such networks was characterized for an infinite time horizon. We aim to investigate the throughput region for transmissions over a finite time horizon. Unlike the infinite-horizon throughput region, which is simply the convex hull of the throughput region of one time slot, the finite-horizon throughput region is generally non-convex. Instead of directly characterizing all achievable rate-tuples in the finite-horizon throughput region, we propose a metric termed the rate margin, which not only determines whether any given rate-tuple is within the throughput region (i.e., achievable or unachievable), but also tells the amount of scaling that can be done to the given achievable (unachievable) rate-tuple such that the resulting rate-tuple is still within (brought back into) the throughput region. Furthermore, we derive an efficient algorithm to find the rate-achieving policy for any given rate-tuple in the finite-horizon throughput region.

This chapter is organized as follows. The system model and problem description are given in Section 4.2 and Section 4.3, respectively. In Section 4.4, an optimization problem is defined to solve rate margin and rate-achieving policy, and an efficient solution method for this problem is also proposed. The numerical examples are given in Section 4.5 to illustrate the effectiveness

of our approach and corroborate our analytical results. Finally, the concluding remarks of this chapter are given in Section 4.6.

4.2 System Model and Throughput Region

We consider N transmitter-receiver pairs in a wireless network, where Tx_n and Rx_n denote the transmitter and receiver of the n^{th} communication pair. The channel is memoryless, and the relationship between channel input $X_m \in \mathbb{R}$ (corresponding to Tx_m, and $m \in \{1,\ldots,N\} =: \mathcal{N}$) to output Y_n ($n \in \mathcal{N}$) is

$$Y_n = \sum_{m\in\mathcal{N}} \sqrt{H_{mn}} X_m + Z_n, \quad n \in \mathcal{N}, \tag{4.1}$$

where H_{mn} is the channel gain between Tx_m and Rx_n; and Z_n is Gaussian white noise with power W_n. When decoding, Rx_n treats the interference $\sum_{m\neq n} \sqrt{H_{mn}} X_m$ as noise. We can see that our channel is indeed a multi-user Gaussian interference channel, as shown in Fig. 4.1.

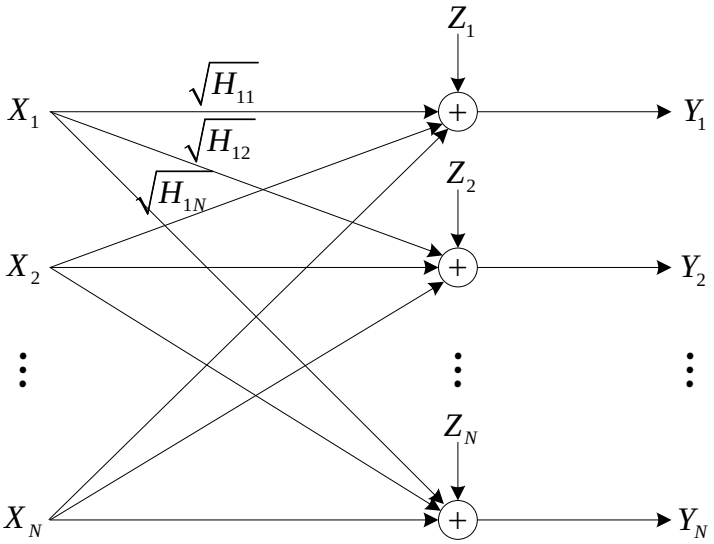


Figure 4.1: Channel model for N transmitter-receiver pairs.

The time is slotted and each time slot contains L channel uses for transmitting and receiving a codeword (i.e., the length of a codeword, or simply the blocklength, is L). We consider a finite time horizon of T time slots and assume that the channel gains H_{mn} remain constant over the T time slots. In each time slot, every transmitter-receiver pair chooses to transmit or not. That is, for time slot $t \in \{1,\ldots,T\} =: \mathcal{T}$, the transmitter Tx_n ($n \in \mathcal{N}$) chooses its transmit power $s_t^{(n)}$ from the transmit-power set $\mathcal{S}^{(n)}$, where 0 is included for representing no transmission. Since the number of available transmit power options in a practical communication system is usually finite (e.g., see the discrete power control in [81, 82]), we model $\mathcal{S}^{(n)}$ as a finite set. Furthermore, we label $s_t = \left[s_t^{(1)},\ldots,s_t^{(N)}\right]^{\mathrm{T}}$, and $\mathcal{S} := \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(N)}$. Hence, $s_t \in \mathcal{S}$, and we

call $\mathcal{S}$ the transmit-power-tuple set.

For time slot t, the SINR for each transmitter-receiver pair is determined by

$$\gamma_n(s_t) = \frac{H_{nn}s_t^{(n)}}{W_n + \sum_{m\neq n} H_{mn}s_t^{(m)}}, \quad n,m \in \mathcal{N}, \tag{4.2}$$

where the interference is treated as noise. Given the SINR $\gamma_n(s_t)$, blocklength L, and error probability ε, the maximum achievable rate for transmitter-receiver pair n is defined as

$$\mu_{\max}^{(n)}(\gamma_n(s_t), L, \varepsilon) = \frac{1}{L}\log_2 M^*(L,\varepsilon), \tag{4.3}$$

where $M^*(L,\varepsilon)$ represents the maximal code size as defined in [10, 83]. In our numerical results, we use the following accurate approximation of $\mu_{\max}^{(n)}(\gamma_n(s_t), L, \varepsilon)$ for Gaussian channels [10]:

$$\mu_{\max}^{(n)}(\gamma_n(s_t), L, \varepsilon) \approx \frac{1}{2}\log_2(1+\gamma_n(s_t)) - \sqrt{\frac{V}{L}}\tilde{Q}^{-1}(\varepsilon), \tag{4.4}$$

where $\tilde{Q}$ is the complementary Gaussian cumulative distribution function, and V is the channel dispersion

$$V = \frac{\log_2^2 e}{2}\left[1 - \frac{1}{(1+\gamma_n(s_t))^2}\right]. \tag{4.5}$$

We stress that all the analytical results in this chapter hold for both the generic expression in (4.3) and the explicit approximation in (4.4), where (4.4) is primarily used to obtain numerical results.

For transmitter-receiver pair n, in time slot t, any rate $\mu_t^{(n)} \in \left[0,\ \mu_{\max}^{(n)}(\gamma_n(s_t), L, \varepsilon)\right]$ is achievable, i.e., there exist some channel codes with rate $\mu_t^{(n)}$ such that the decoding error probability is bounded by ε. Under a given transmit-power-tuple s_t, all achievable rate-tuples for N transmitter-receiver pairs form the following set

$$\left\{\left[\mu_t^{(1)}, \ldots, \mu_t^{(N)}\right] : \mu_t^{(n)} \in \left[0,\ \mu_{\max}^{(n)}(\gamma_n(s_t), L, \varepsilon)\right], \quad n \in \mathcal{N}\right\}. \tag{4.6}$$

By defining $\mu_t = \left[\mu_t^{(1)}, \ldots, \mu_t^{(N)}\right]$ and

$$\mu_{\max}(s_t, L, \varepsilon) = \left[\mu_{\max}^{(1)}(\gamma_1(s_t), L, \varepsilon), \ldots, \mu_{\max}^{(N)}(\gamma_N(s_t), L, \varepsilon)\right], \tag{4.7}$$

equation (4.6) can be rewritten in the compact form

$$\left\{\mu_t :\ \mu_t \preceq \mu_{\max}(s_t, L, \varepsilon)\right\}. \tag{4.8}$$

Note that (4.8) contains all achievable rate-tuples in one time slot for one transmit-power-tuple

s_t. Then, the 1-slot throughput region is defined as the region of all achievable rate-tuples for all possible transmit-power-tuples, and the 1-slot throughput region for time slot t is

$$\Lambda_{[1],t} = \bigcup_{s_t \in \mathcal{S}} \{\mu_t \colon 0 \preceq \mu_t \preceq \mu_{\max}(s_t, L, \varepsilon)\}, \tag{4.9}$$

where $\mathcal{S}$ is the set of all possible transmit-power-tuples. Note that $\Lambda_{[1],t}$ are the same for all t, and thus, for simplicity, we label $\Lambda_{[1],1} = \cdots = \Lambda_{[1],T} = \Lambda_{[1]}$.

Similar to the one-slot throughput region, the finite-horizon throughput region for T time slots is defined as follows.

Definition 4.1 (Finite-Horizon Throughput Region)**.** The T-slot throughput region $\Lambda_{[T]}$ is the set of average rate-tuples that can be achieved in T time slots, i.e.,

$$\Lambda_{[T]} = \left\{ \mu_{[T]} \colon \mu_{[T]} = \frac{1}{T} \sum_{t=1}^{T} \mu_t, \quad \mu_t \in \Lambda_{[1]} \right\}. \tag{4.10}$$

For a clear illustration of finite-horizon throughput region, we define the weak Pareto frontier and Pareto frontier in Definition 4.2 which also plays an important role in the analytical results in this chapter.

Definition 4.2 (Weak Pareto Frontier and Pareto Frontier)**.** For a given set $\mathcal{A}$, the weak Pareto frontier is

$$\mathcal{B} = \{b \in \mathcal{A} \colon \{a \in \mathcal{A} \colon a \succ b\} = \emptyset\}, \tag{4.11}$$

and the Pareto Frontier is

$$\overline{\mathcal{B}} = \{b \in \mathcal{A} \colon \{a \in \mathcal{A} \colon a \succeq b\} = \{b\}\}. \tag{4.12}$$

It should be noted that $\overline{\mathcal{B}} \subseteq \mathcal{B}$.

With Definition 4.2, we define the weak Pareto frontier and Pareto frontier of $\Lambda_{[T]}$ as $\mathcal{M}_{[T]}$ and $\overline{\mathcal{M}}_{[T]}$, respectively. Additionally, we say the rate-tuples on the weak Pareto frontier are the boundary rate-tuples.

Three examples are given in Fig. 4.2 to illustrate the shape of finite-horizon throughput region. We consider two transmitter-receiver pairs, so the throughput regions are in two dimensions. The detailed network parameters are: Channel gains $H_{11} = H_{22} = 1$, $H_{12} = H_{21} = 0.3$, transmit-power sets $\mathcal{S}^{(1)} = \mathcal{S}^{(2)} = \{0, 3\}$, powers of white noises $W_1 = W_2 = 0.1$, blocklength $L = 100$, and error probability $\varepsilon = 0.001$. By using the maximum rate approximation in (4), the 1, 2, and 3-slot throughput regions are shown in (a), (b), and (c), respectively. We consider rate-tuples $\mu' = [0.3, 0.4]^{\mathrm{T}}$, $\mu'' = [1.08, 1.08]^{\mathrm{T}}$, and $\mu''' = [1.4, 0.6]^{\mathrm{T}}$. In each finite-horizon

throughput region, the pink circles compose the Pareto frontier, and the purple (thick) lines form the weak Pareto frontier, and the shaded area is the interior of a throughput region. Note that the finite-horizon throughput region includes both the interior and weak Pareto frontier (Pareto frontier is in the weak Pareto frontier as mentioned in Definition 2). For the dash-dotted lines, they are the Pareto frontier of the well-known infinite-horizon throughput region (see [4]), which is the convex hull of $\Lambda_{[1]}$.

Using the same network parameters, Fig. 4.2(a), Fig. 4.2(b), and Fig. 4.2(c) illustrate the throughput regions for $T = 1$, $T = 2$, and $T = 3$, respectively. We use the rate-tuples μ', μ'', and μ''' (whose values are given in the caption of Fig. 4.2) to compare the differences among finite-horizon throughput regions for different T:

- μ' is in $\Lambda_{[1]}$, $\Lambda_{[2]}$ and $\Lambda_{[3]}$.
- μ'' is in $\Lambda_{[2]}$, but not in $\Lambda_{[1]}$ or $\Lambda_{[3]}$.
- μ''' is in $\Lambda_{[3]}$, but not in $\Lambda_{[1]}$ or $\Lambda_{[2]}$.

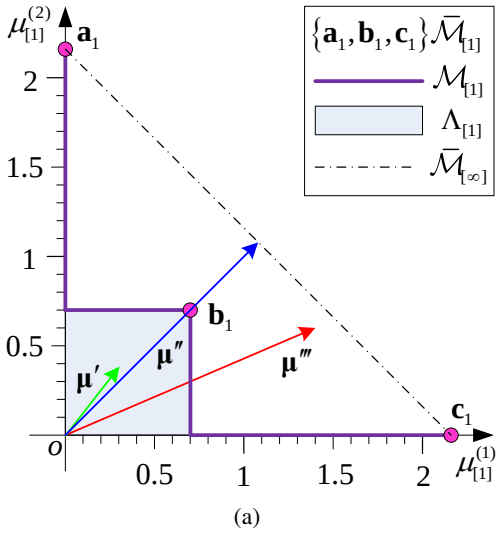
We take μ''' as an example. From the caption, $\mu''' = [1.4, 0.6]^{\mathrm{T}}$ and it can be achieved within 3 time slots by

$$\mu''' = \frac{1}{3}\left(\begin{bmatrix}2.1\\0\end{bmatrix} + \begin{bmatrix}2.1\\0\end{bmatrix} + \begin{bmatrix}0\\1.8\end{bmatrix}\right), \tag{4.13}$$

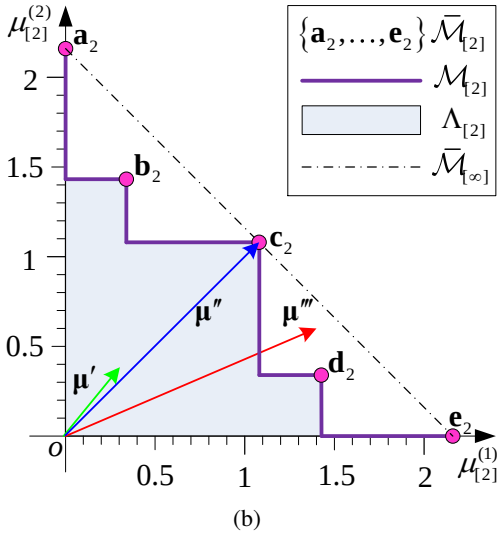
which means letting communication pair 1 transmitting at the rate of 2.1 in the first two time slots and communication pair 2 transmitting at the rate of 1.8 in the third time slot. Note that μ''' is not achievable within $T = 1$ or $T = 2$ time slots. Intuitively, if $T_1 < T_2$, the relationship between their throughput regions seems to be $\Lambda_{[T_1]} \subseteq \Lambda_{[T_2]}$ (e.g., $\Lambda_{[1]} \subseteq \Lambda_{[2]}$). However, this intuition turns out to be incorrect, as μ'' is in $\Lambda_{[2]}$ but not in $\Lambda_{[3]}$.

Remark 4.1 (Uncertain-Inclusion Property)**.** For the convenience of discussion, we label $\lim_{T\to\infty} \Lambda_{[T]}$ as $\Lambda_{[\infty]}$. It is easy to verify that $\Lambda_{[T]} \subseteq \Lambda_{[\infty]}$ holds for any finite T, since $\Lambda_{[\infty]}$ is a convex hull of $\Lambda_{[1]}$ (as shown in [51]). However, in general, if $T_1 < T_2$, the proposition $\Lambda_{[T_1]} \subseteq \Lambda_{[T_2]}$ does not always hold true. We call this property the uncertain-inclusion property of the finite-horizon throughput region. Specifically, it can be proved that if T_1 is an factor of T_2, then $\Lambda_{[T_1]} \subseteq \Lambda_{[T_2]}$; but if T_1 is not an factor of T_2, then $\Lambda_{[T_1]} \subseteq \Lambda_{[T_2]}$ does not hold in general, which highly depends on the structure of $\Lambda_{[1]}$.

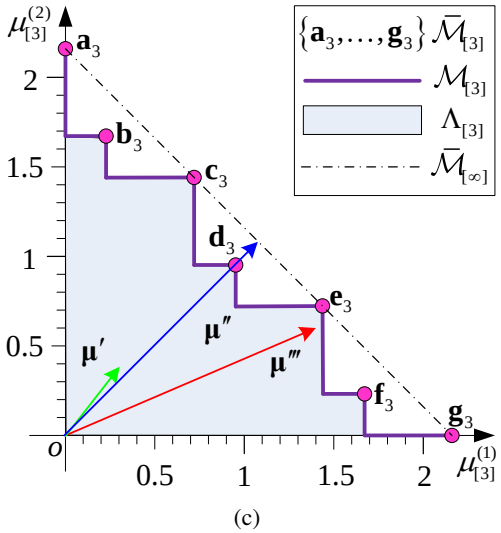
The uncertain-inclusion property prevents us from analyzing $\Lambda_{[T_2]}$ based on the information from $\Lambda_{[T_1]}$ in general, and therefore we cannot determine whether a rate-tuple achievable in T_1 slots is still achievable in T_2 slots.



(a)



(b)



(c)

Figure 4.2: Examples of throughput regions of two transmitter-receiver pairs.

4.3 Problem Description: Rate Margin and Rate-Achieving Policy

In this work, we propose an important metric to characterize $\Lambda_{[T]}$, termed the rate margin. The rate margin has three useful properties:

- The rate margin determines whether a given rate-tuple is achievable or not within T time slots.
- If a rate-tuple is achievable, the rate margin gives the headroom for scaling up the rate-tuple that remains achievable.
- Similarly, if a rate-tuple is unachievable, the rate margin tells exactly by what extent the rate-tuple needs to be scaled down to be achievable.

We also study the rate-achieving policy, which gives a method to achieve any given rate-tuple in $\Lambda_{[T]}$.

Now, we give the definition of the rate margin.

Definition 4.3 (Rate Margin)**.** For a 1-slot throughput region $\Lambda_{[1]}$ and T time slots, the rate margin $\delta_T(\cdot)\colon \overline{\mathbb{R}}_+^N \to \mathbb{R}_+ \bigcup \{\infty\}$ is a function of rate-tuple $\mu_{[T]}$ that

$$\delta_T(\mu_{[T]}) = \max_{\mu'_{[T]} \in \overline{\mathcal{M}}_{[T]}} \min_{n \in \mathcal{N}} \left\{ \frac{\mu_{[T]}^{\prime(n)}}{\mu_{[T]}^{(n)}} \right\}, \tag{4.14}$$

where $\mu_{[T]}^{(n)}$ and $\mu_{[T]}^{\prime(n)}$ are the n^{th} component of $\mu_{[T]}$ and $\mu'_{[T]}$, respectively. Note that the rate margin can be infinite.

The rate margin has several useful properties, which are given in Proposition 4.1, Proposition 4.2, Proposition 4.3, and Corollary 4.1. Firstly, the rate margin can be used to determine whether a given rate-tuple is achievable or not:

Proposition 4.1. $\forall \mu_{[T]} \in \Lambda_{[T]}$ if and only if $\delta_T(\mu_{[T]}) \geq 1$.

Proof: Necessity. $\forall \mu_{[T]} \in \Lambda_{[T]}$, there exists at least one $\mu'_{[T]} \in \overline{\mathcal{M}}_{[T]}$ such that

$$\min_{n \in \mathcal{N}} \left\{ \frac{\mu_{[T]}^{\prime(n)}}{\mu_{[T]}^{(n)}} \right\} \geq 1, \tag{4.15}$$

since it otherwise contradicts (4.12) in the Pareto frontier definition. Thus, $\delta_T(\mu_{[T]}) \geq 1$ in (4.14).

Sufficiency. If $\delta_T(\mu_{[T]}) \geq 1$, but we assume $\mu_{[T]} \notin \Lambda_{[T]}$, then there would be at least one component index n such that $\mu_{[T]}^{(n)} > \mu_{[T]}^{\prime(n)}$ ($\mu'_{[T]} \in \overline{\mathcal{M}}_{[T]}$). Thus, $\forall \mu'_{[T]} \in \overline{\mathcal{M}}_{[T]}$,

$$\min_{n\in\mathcal{N}} \left\{ \frac{\mu_{[T]}^{\prime(n)}}{\mu_{[T]}^{(n)}} \right\} < 1, \tag{4.16}$$

which implies $\delta_T(\mu_{[T]}) < 1$, and this contradicts $\delta_T(\mu_{[T]}) \geq 1$. Therefore, $\mu_{[T]} \in \Lambda_{[T]}$. ■

Another important property of rate margin is that it quantifies the extent of which an achievable rate-tuple can be linearly scaled-up while remaining achievable.

Proposition 4.2. $\forall \mu_{[T]} \in \Lambda_{[T]}$, the rate margin gives the maximum scalar ρ such that $\rho\mu_{[T]}$ is still an achievable rate-tuple, i.e.,

$$\max_{\rho\mu_{[T]}\in\Lambda_{[T]}} \rho = \delta_T(\mu_{[T]}) \geq 1. \tag{4.17}$$

Proof: For convenience, we label $\rho^* = \max_{\rho\mu_{[T]}\in\Lambda_{[T]}} \rho$ for the left-hand-side of (4.17). Let $\mu'_{[T]} = \rho^*\mu_{[T]}$, and we have $\mu'_{[T]} \in \Lambda_{[T]}$. In addition, by setting $\mu''_{[T]} = \delta_T(\mu_{[T]})\mu_{[T]}$, the proof starts as follows.

i). $\delta_T(\mu_{[T]}) \geq \rho^*$: assume $\delta_T(\mu_{[T]}) < \rho^*$, then $\delta_T(\mu''_{[T]}) < \delta_T(\mu'_{[T]}) \leq 1$. It implied $\delta_T(\mu''_{[T]}) < 1$, which contradicts to the following derivation

$$\delta_T(\mu''_{[T]}) = \max_{\mu'''_{[T]}\in\overline{\mathcal{M}}_{[T]}} \min_{n\in\mathcal{N}} \left\{ \frac{\mu_{[T]}^{\prime\prime\prime(n)}}{\mu_{[T]}^{\prime\prime(n)}} \right\} = \frac{1}{\delta_T(\mu_{[T]})} \max_{\mu'''_{[T]}\in\overline{\mathcal{M}}_{[T]}} \min_{n\in\mathcal{N}} \left\{ \frac{\mu_{[T]}^{\prime\prime\prime(n)}}{\mu_{[T]}^{\prime\prime(n)}} \right\} = 1. \tag{4.18}$$

Therefore, $\delta_T(\mu_{[T]}) \geq \rho^*$ holds.

ii). $\delta_T(\mu_{[T]}) \leq \rho^*$: assume $\delta_T(\mu_{[T]}) > \rho^*$, and the proof is similar to that in i) (equation (4.18) still holds).

To sum up, $\delta_T(\mu_{[T]}) = \rho^*$. According to Proposition 4.1, $\delta_T(\mu_{[T]}) \geq 1$ and (4.17) is satisfied. ■

With Proposition 4.2, we can also derive an important property that the rate margin quantifies the extent of which an unachievable rate-tuple should be linearly scaled down to become achievable.

Proposition 4.3. $\forall \mu_{[T]} \notin \Lambda_{[T]}$, the rate margin gives the minimum scalar r such that $\mu_{[T]}/r$ becomes an achievable rate-tuple, i.e.,

$$\min_{\mu_{[T]}/r\in\Lambda_{[T]}} r = \frac{1}{\delta_T(\mu_{[T]})} > 1. \tag{4.19}$$

Proof: Since the proof is similar to that in Proposition 4.2, we omit it here. ■

Last but not least, the rate margin is an indictor for those rate-tuples on the weak Pareto frontier, which is also very useful for results to be derived later.

Corollary 4.1. $\mu_{[T]} \in \mathcal{M}_{[T]}$ if and only if $\delta_T(\mu_{[T]}) = 1$.

Proof: Based on Proposition 4.2 and Proposition 4.3, the proof is straightforward. ■

Remark 4.2 (Properties of Rate Margin)**.** We use a numerical example to summarize the properties of rate margin. Specifically, we test three rate-tuples, i.e., $\mu'_{[3]}$, $\mu''_{[3]}$ and $\mu'''_{[3]}$, whose values are shown in the caption of Fig. 4.3. Using Proposition 4.1, we have $\delta_3(\mu'_{[3]}) \geq 1$, $\delta_3(\mu''_{[3]}) \geq 1$, and $\delta_3(\mu'''_{[3]}) < 1$, hence we know that $\mu'_{[3]} \in \Lambda_{[3]}$, $\mu''_{[3]} \in \Lambda_{[3]}$, and $\mu'''_{[3]} \notin \Lambda_{[3]}$. These conclusions are correct as shown in Fig. 4.3. Proposition 4.2 implies that $\mu'_{[3]}$ can be linearly scaled up by at most $\delta_3(\mu'_{[3]}) = 1.9046$ and remains achievable. Similarly, Proposition 4.3 tells that $\mu'''_{[3]}$ should be linearly scaled down by at least $\delta_3(\mu'''_{[3]}) = 0.6006$ and becomes achievable. By Corollary 4.1, $\delta_3(\mu''_{[3]}) = 1$ (hence $\mu''_{[3]} \in \mathcal{M}_{[3]}$), and there is no room for $\mu''_{[3]}$ to be linearly scaled up and remains achievable.

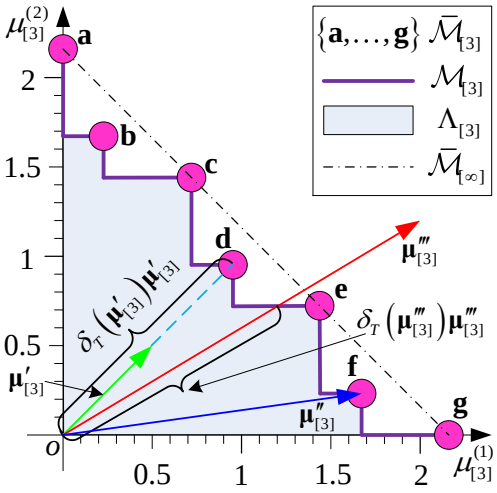


Figure 4.3: Illustration of rate margin. The network parameters are the same as those in Fig. 4.2(c), but the chosen rate-tuples are different: $\mu'_{[3]} = [0.5, 0.5]^{\mathrm{T}}$, $\mu''_{[3]} = [1.6729, 0.2316]^{\mathrm{T}}$, and $\mu'''_{[3]} = [2, 1.2]^{\mathrm{T}}$. $\mu'_{[3]}$ is in $\Lambda_{[3]} \setminus \mathcal{M}_{[3]}$ (i.e., the interior of $\Lambda_{[3]}$), and $\mu''_{[3]}$ is in $\mathcal{M}_{[3]}$, but $\mu'''_{[3]}$ is not in $\Lambda_{[3]}$. The rate margins are $\delta_3(\mu'_{[3]}) = 1.9046$, $\delta_3(\mu''_{[3]}) = 1$, and $\delta_3(\mu'''_{[3]}) = 0.6006$.

Now, we give the definition of rate-achieving policy.

Definition 4.4 (Rate-Achieving Policy)**.** For a given transmit-power-tuple set $\mathcal{S}$ and T time slots, $\forall \mu_{[T]} \in \Lambda_{[T]}$, the rate-achieving policy for $\mu_{[T]}$ is a sequence of rate-power pairs

$$\mathcal{P}_T = (\mu_t, s_t)_{t=1}^{T}, \quad s_t \in \mathcal{S}, \tag{4.20}$$

with the maximum rate constraints

$$\mu_t \preceq \mu_{\max}(s_t, L, \varepsilon), \tag{4.21}$$

such that $\mu_{[T]}$ can be achieved, i.e.,

$$\mu_{[T]} = \frac{1}{T}\sum_{t=1}^{T} \mu_t. \tag{4.22}$$

Equation (4.21) means that: for each transmitter-receiver pair, the transmission rate should not exceed the corresponding maximum rate.

Remark 4.3. Definition 4.4 tells that if $\mu_{[T]} \in \Lambda_{[T]}$ (or equivalently $\delta_T(\mu_{[T]}) \geq 1$, see Proposition 4.1), then there always exist some policies $\mathcal{P}_T$ described by (4.20), (4.21), and (4.22) to achieve $\mu_{[T]}$. To be more specific, a power sequence $(s_t)_{t=1}^T$ provides the maximum rates to support the rate-tuple $\mu_{[T]}$ (see (4.21)), which is eventually achieved by rate-tuple sequence $(\mu_t)_{t=1}^T$ (see (4.22)). The existence of $\mathcal{P}_T$ is exactly guaranteed by (4.10) in Definition 4.1 and (4.9) that: $\forall \mu_t \in \Lambda_{[1]}$, there at least exists one $\mu_{\max}(s_t, L, \varepsilon)$ such that $\mu_t \in \mu_{\max}(s_t, L, \varepsilon)$. Nevertheless, Definition 4.4 just only states the existence of $\mathcal{P}_T$ for an achievable rate-tuple, but how to find an efficient algorithm to find $\mathcal{P}_T$ is yet to be investigated.

After defining the rate margin and rate-achieving policy, we now focus on two key problems

- How to efficiently compute the rate margin $\delta_T(\mu_{[T]})$.
- If $\delta_T(\mu_{[T]}) \geq 1$, how to design a rate-achieving policy with high computational efficiency.

4.4 Derivation of Rate Margin and Rate-Achieving Policy

Although deriving the rate margin and rate-achieving policy are two different problems, we carefully formulate them into the following joint problem (see Problem 4.1) from the viewpoint of data transmission. Note that achieving a given average rate-tuple $\mu_{[T]}$ over T time slots is the same as transmitting $T\mu_{[T]}L$ amount of data from data queues within T time slots (recall that L is the blocklength).

Problem 4.1. For a given transmit-power-tuple set $\mathcal{S}$ and T time slots, we consider the follow-

ing problem:

$$\begin{array}{ll} \underset{(s_t)_{t=1}^{p},\, s_t\in\mathcal{S}}{\text{minimize}} & p+\max\limits_{n\in\mathcal{N}}\left\lceil \dfrac{Q_0^{(n)}}{\sum_{t=1}^{p}\mu_{\max}^{(n)}(\gamma_n(s_t),L,\varepsilon)L}\right\rceil \\ \text{subject to} & Q_t=(Q_{t-1}-\mu_{\max}(s_t,L,\varepsilon)L)^+, \\ & Q_0=T\mu_{[T]}L, \\ & Q_{p-1}\neq 0, \\ & Q_p=0, \end{array} \tag{4.23}$$

where $t\in\{1,\ldots,p\}$. The data queues are $Q_t=[Q_t^{(1)},\ldots,Q_t^{(N)}]^{\mathrm{T}}$, each $Q_t^{(n)}\in\overline{\mathbb{R}}_+$ is the length of the data queue for transmitter n after s_t is applied in time slot t ($t\in\{1,\ldots,p\}$). In addition, $Q_0=T\mu_{[T]}L$ is the initial lengths of queues before applying s_1, which means the given rate-tuple $\mu_{[T]}=[\mu_{[T]}^{(1)},\ldots,\mu_{[T]}^{(N)}]^{\mathrm{T}}$ at which transmission must take place in order to send a total of $T\mu_{[T]}L$ amount of data, and L is the blocklength. Note that p denotes the number of time slots for transmission (since $Q_{p-1}\neq 0$ and $Q_p=0$) and is a variable dependent on $(s_t)_{t=1}^{p}$. The optimal solution (not unique for $T>1$) is denoted as $(s_t^*)_{t=1}^{p^*}$. The corresponding data-queues sequence under optimal solution is $(Q_t^*)_{t=1}^{p^*}$. The optimal objective is

$$p^*+\max_{n\in\mathcal{N}}\left\lceil \frac{Q_0^{(n)}}{\sum_{t=1}^{p^*}\mu_{\max}^{(n)}(\gamma_n(s_t^*),L,\varepsilon)L}\right\rceil. \tag{4.24}$$

The following two lemmas explain the meaning of the first and second items in (4.24), respectively.

Lemma 4.1. p^* in (4.24) is the minimum number of time slots that clears the data queues Q_0.

Proof: By (4.23), p is the number of time slots that clears $Q_0^{(n)}$, which implies the following

$$\max_{n\in\mathcal{N}}\left\lceil \frac{Q_0^{(n)}}{\sum_{t=1}^{p}\mu_{\max}^{(n)}(\gamma_n(s_t),L,\varepsilon)L}\right\rceil\leq 1. \tag{4.25}$$

Then p^* in (4.24) is the optimal number of time slots to clear $Q_0^{(n)}$. This is because we can never find a $p'<p^*$ such that

$$p'+\max_{n\in\mathcal{N}}\left\lceil \frac{Q_0^{(n)}}{\sum_{t=1}^{p'}\mu_{\max}^{(n)}(\gamma_n(s_t'^*),L,\varepsilon)L}\right\rceil\geq p^*+\max_{n\in\mathcal{N}}\left\lceil \frac{Q_0^{(n)}}{\sum_{t=1}^{p^*}\mu_{\max}^{(n)}(\gamma_n(s_t^*),L,\varepsilon)L}\right\rceil \tag{4.26}$$

holds with (4.25) (since p' and p^* are integers). ■

The value of p^* tells the rate-achievability: If $p^*\leq T$, then it is possible to transmit $T\mu_{[T]}L$ amount of data within T slots. In other words, the rate-tuple $\mu_{[T]}$ is achievable within T slots.

If $p^* > T$, then it implies the rate-tuple $\mu_{[T]}$ is unachievable within T slots.

Lemma 4.2. The rate margin for $\mu_{[p^*]}$ in $\Lambda_{[p^*]}$ is the reciprocal of the second item in (4.24), i.e.,

$$\delta_{p^*}(\mu_{[p^*]}) = 1 \Bigg/ \max_{n\in\mathcal{N}} \left[\frac{Q_0^{(n)}}{\sum_{t=1}^{p^*} \mu_{\max}^{(n)}(\gamma_n(s_t^*), L, \varepsilon)L} \right]. \tag{4.27}$$

Proof: With the definition of the rate margin (see Definition 4.3) and replacing T with p^*, we can derive

$$\frac{1}{\delta_{p^*}(\mu_{[p^*]})} \overset{(a)}{=} \min_{\mu'_{[p^*]} \in \overline{\mathcal{M}}_{[p^*]}} \max_{n\in\mathcal{N}} \left\{ \frac{\mu_{[p^*]}^{(n)}}{\mu_{[p^*]}^{\prime(n)}} \right\} = \min_{\mu'_{[p^*]} \in \overline{\mathcal{M}}_{[p^*]}} \max_{n\in\mathcal{N}} \left\{ \frac{p^*\mu_{[p^*]}^{(n)}L}{p^*\mu_{[p^*]}^{\prime(n)}L} \right\}, \tag{4.28}$$

where (a) follows from that the max (min) of $\mu_{[p^*]}^{\prime(n)}/\mu_{[p^*]}^{(n)}$ is the min (max) of $\mu_{[p^*]}^{(n)}/\mu_{[p^*]}^{\prime(n)}$. Since $p^*\mu_{[p^*]}^{(n)}L = Q_0^{(n)}$, there exists $(s_t)_{t=1}^{p^*}$ such that

$$p^*\mu_{[p^*]}^{\prime(n)}L = \sum_{t=1}^{p^*} \mu_t^{\prime(n)}L \overset{(b)}{=} \sum_{t=1}^{p^*} \mu_{\max}^{(n)}(\gamma_n(s_t), L, \varepsilon)L, \tag{4.29}$$

where (b) holds for $\mu'_{[p^*]} \in \overline{\mathcal{M}}_{[p^*]}$. Thus, we rewrite (4.28) as

$$\begin{aligned} \frac{1}{\delta_{p^*}(\mu_{[p^*]})} &= \min_{(s_t)_{t=1}^{p^*}} \max_{n\in\mathcal{N}} \left[\frac{Q_0^{(n)}}{\sum_{t=1}^{p^*} \mu_{\max}^{(n)}(\gamma_n(s_t), L, \varepsilon)L} \right] \\ &\overset{(c)}{=} \max_{n\in\mathcal{N}} \left[\frac{Q_0^{(n)}}{\sum_{t=1}^{p^*} \mu_{\max}^{(n)}(\gamma_n(s_t^*), L, \varepsilon)L} \right], \end{aligned} \tag{4.30}$$

where (c) follows from Lemma 4.1 and the objective in (4.23). Therefore, (4.27) holds. ■

Note that the reciprocal of the second item in (4.24) is the rate margin for p^* time slots. However, we want to derive the rate margin for T time slots (p^* is not necessarily equal to T). In the rest of this section, we discuss how to derive the rate margin for any given T time slots (see Section 4.4.1) based on the optimal objective of Problem 4.1. Furthermore, by the optimal solution of Problem 4.1 the rate-achieving policy is derived (see Section 4.4.2). Finally, an efficient solution method of Problem 4.1 is given in Section 4.4.3.

4.4.1 Deriving Rate Margin

This subsection proposes a method to derive the rate margin by iteratively solving Problem 4.1.

Firstly, if we find $p^* = T$ after solving Problem 4.1, then the rate margin can be derived directly from Lemma 4.2, since $\delta_T(\mu_{[T]}) = \delta_{p^*}(\mu_{[p^*]})$ in this case.

If $p^* \neq T$, we can use an iteration strategy to derive rate margin $\delta_T(\mu_{[T]})$. To distinguish p^* (and Q_0) in different iterations, we label p_k^* (and $Q_{0,k}$) as the p^* (and Q_0) for the k^{th} iteration. Now, the main idea of our iteration strategy is given as follows: based on the information from Lemma 4.2, we linearly scale the initial condition $Q_{0,k}$ in Problem 4.1 for each iteration $k \in \{1,\ldots,K\}$, until $p_K^* = T$, in which case, $\delta_T(\mu_{[T]}) = \delta_{p_K^*}(\mu_{[p_K^*]})$. As such, the rate margin can be finally determined recursively in a finite number of steps whenever K is finite. The iteration strategy is given in Algorithm 4.1, and proved in Theorem 4.1.

Algorithm 4.1 Deriving Rate Margin

Require: T: the number of time slots; N: the number of transmitter-receiver pairs; $\mu_{[T]}$: the given average rate-tuple; $\mathcal{S}$: the transmit-power-tuple set; L: the blocklength; ε: the error probability.

Ensure: $\delta_T(\mu_{[T]})$: the rate margin.

Initialization: $k=1$; $Q_{0,k} = T\mu_{[T]}L$; $p_k^* = 0$; flag $= 0$; $\varepsilon = 10^{-7}$ {**comments:** ε is the precision for rate-tuple calculation}.
while $p_k^* \neq T$ **do**
  Solve Problem 4.1 with $Q_0 = Q_{0,k}$, and derive $\delta_{p_k^*}(\mu_{[p_k^*],k})$ by Lemma 4.2;
  if $p_k^* < T$ **then**
    $Q_{0,k+1} = Q_{0,k}\delta_{p_k^*}(\mu_{[p_k^*],k})\lfloor T/p_k^* \rfloor + R_k\rho\mu_{[T]}$, where $R_k := T \bmod p_k^*$, and $\rho = \delta_1(\mu_{[T]})$; flag $=$ 1;
    if $\lfloor T/p_k^* \rfloor == 1$ and $\rho == 0$ **then**
      $Q_{0,k+1} = Q_{0,k} + \varepsilon T$;
    end if
  else if $p_k^* > T$ and flag $== -1$ **then**
    $Q_{0,k+1} = 0$; $p_k^* = T$ {**comments:** condition for ending the loop};
  else if $p_k^* > T$ and flag $== 0$ **then**
    $Q_{0,k+1} = T\max\{\rho,\varepsilon\}\mu_{[T]}$, where $\rho = \delta_1(\mu_{[T]})$; flag $= -1$;
  else if $p_k^* > T$ and flag $== 1$ **then**
    $Q_{0,k+1} = Q_{0,k} - \varepsilon T$; $p_k^* = T$;
  else
    $Q_{0,k+1} = Q_{0,k}\delta_{p_k^*}(\mu_{[p_k^*],k})$; {**comments:** $p_k^* = T$}
  end if
  $K = k$; $k = k+1$; {**comments:** K is the total iteration number.}
end while
return $\delta_T(\mu_{[T]}) = Q_{0,K+1}^{(1)}/Q_{0,1}^{(1)}$.

Theorem 4.1 (Calculation of Rate Margin)**.** For $\mu_{[T]} \succ 0$,[1] the rate margin can be obtained by Algorithm 4.1 involving a finite number of K iterations, upper bounded by

$$K \leq \begin{cases} T - p_1^* + 1 & p_1^* < T, \\ 1 & p_1^* = T, \\ T - p_2^* + 2 & p_1^* > T. \end{cases} \tag{4.31}$$

[1] Algorithm 4.1 is valid for rate-tuple $\mu_{[T]}$ whose components are all greater than 0. This is without loss of generality because the zero components stand for zero transmissions, and hence we can remove these inactive transmitter-receiver pairs from the network model.

Proof: See Appendix C.1. ■

Remark 4.4. The inequality in (4.31) gives an upper bound on the number of iterations. For example, considering the case $p_1^* < T$, we have $K \leq T - p_1^* + 1$, which means that the rate margin requires solving Problem 4.1 at most $T - p_1^* + 1$ times.

Before closing this subsection, we give a useful corollary. The proof can be easily obtained by Proposition 4.1 and the proof of Theorem 4.1.

Corollary 4.2. The following three statements are equivalent: i) $\mu_{[T]} \in \Lambda_{[T]}$; ii) $\delta_T(\mu_{[T]}) \geq 1$; iii) $p_1^* \leq T$.

4.4.2 Deriving Rate-Achieving Policy

In this subsection, we derive a rate-achieving policy for any given achievable rate-tuple. It should be noted that our method is complete, i.e., for any given achievable rate-tuple, the corresponding rate-achieving policy can be obtained.

We present a rate-achieving policy for all rate-tuples in the T-slot throughput region as follows.

Theorem 4.2 (Rate-Achieving Policy for All Achievable Rates)**.** Given a transmit-power-tuple set $\mathcal{S}$ and a finite horizon of T time slots, then:

i) If $\mu_{[T]} \in \Lambda_{[T]}$, then $p^* \leq T$, and the rate-achieving policy is $\mathcal{P}_T = (\mu_t, s_t)_{t=1}^T$ with

$$(\mu_t, s_t) = \begin{cases} \left(\frac{Q_{t-1}^* - Q_t^*}{L}, s_t^*\right) & 1 \leq t \leq p^*, \\ (0,0) & p^* < t \leq T, \end{cases} \tag{4.32}$$

where $(s_t^*)_{t=1}^{p^*}$, is an optimal solution to Problem 4.1 and Q_t^* is the corresponding data queue vector in time slot t when applying the optimal solution.

ii) If $\mu_{[T]} \notin \Lambda_{[T]}$, then solving Problem 4.1 gives $p^* > T$.

Proof: See Appendix C.2. ■

4.4.3 Solution for Problem 4.1

In Section 4.4.1 and Section 4.4.2, all main results are based on the solution of Problem 4.1. Therefore, designing an efficient algorithm to solve this problem can directly improve the efficiency of deriving rate margin and rate-achieving policy. In this subsection, we discuss how to efficiently solve Problem 4.1.

To solve (4.23) in Problem 4.1, intuitively, we could use dynamic programming to search from $Q_p = 0$ to $Q_0 = T\mu_{[T]}L$ (backwards) or employ other uninformed search strategies [84].

However, in such searching methods, the complexity is $O((\#\mathcal{S})^{p^*})$, where $\mathcal{S}$ is the transmit-tuple set, and p^* is the minimum transmission time (see Lemma 4.1) which can be larger than T.

For example, if we start the search from $Q_0 = T\mu_{[T]}L$, for the first step, we will calculate all possible

$$Q_1 = (Q_0 - \mu_{\max}(s_t, L, \varepsilon)L)^+, \tag{4.33}$$

for all $s_1 \in \mathcal{S}$. Thus, the number of leaf nodes is $\#\mathcal{S}$ for the depth $t = 1$. Similarly, for every Q_1 in (4.33), we have $\#\mathcal{S}$ possible Q_2, and thus the leaf nodes for $t = 2$ is $(\#\mathcal{S})^2$. As such, the number of leaf nodes for depth $t = p^*$ (since the optimal transmission time is p^*, see Lemma 4.1, and we need compare all the objective functions in this depth) is $(\#\mathcal{S})^{p^*}$. Thus, the complexity of such searching methods is $O((\#\mathcal{S})^{p^*})$.

In this subsection, we use the following three steps to significantly improve the computational efficiency in solving Problem 4.1. The resulting complexity is $O(B^{\min\{p^*,T\}})$, where B (the effective branching factor[2]) is a much smaller number compared to $\#\mathcal{S}$, and p^* is reduced to $\min\{p^*, T\}$ for the case $p^* > T$.

For the convenience of applying our search algorithm, we modify the objective in (4.23) as

$$p - 1 + \max_{n\in\mathcal{N}} \left\lfloor \frac{Q_0^{(n)}}{\sum_{t=1}^{p} \mu_{\max}^{(n)}(\gamma_n(s_t), L, \varepsilon)L} \right\rfloor, \tag{4.34}$$

by adding -1 to the original objective. It is readily to see that this modification does not affect the optimal solution (i.e., the original and modified objectives have the same optimal solution).

Step 1: Firstly, we reduce the branching factor from $\#\mathcal{S}$ to $\#\overline{\mathcal{M}}_{[1]}$, which is given in Proposition 4.4.

Proposition 4.4. There exists a sequence $(s_t)_{t=1}^{p^*}$, where $\mu_{\max}(s_t, L, \varepsilon) \in \overline{\mathcal{M}}_{[1]}, t \in \{1, \ldots, p^*\}$, such that $(s_t)_{t=1}^{p^*}$ itself is an optimal solution of Problem 4.1.

Proof: Let $(\mathbf{s}_t^*)_{t=1}^{p^*}$ be any optimal solution of Problem 4.1, we have $Q_{p^*} = 0$, which implies

$$T\mu_{[T]}L \preceq \sum_{t=1}^{p^*} \mu_{\max}(s_t, L, \varepsilon)L. \tag{4.35}$$

Let $(s_t)_{t=1}^{p^*}$ be the sequence that $\mu_{\max}(s_t, L, \varepsilon) \in \overline{\mathcal{M}}_{[1]}, t \in \{1, \ldots, p^*\}$, and $\mu_{\max}(s_t^*, L, \varepsilon) \preceq$

[2]It is a very popular metric for characterizing the efficiency of a searching method, see Section 3.6.1 in [84].

$\mu_{\max}(s_t, L, \varepsilon)$. Thus, (4.35) can be rewritten as

$$T\mu_{[T]}L \preceq \sum_{t=1}^{p^*} \mu_{\max}(s_t^*, L, \varepsilon)L \preceq \sum_{t=1}^{p^*} \mu_{\max}(s_t, L, \varepsilon)L, \tag{4.36}$$

which implies $Q_{p^*} = 0$ when applying $(s_t)_{t=1}^{p^*}$. Therefore, $(s_t)_{t=1}^{p^*}$ is an optimal solution of Problem 4.1. ■

Remark 4.5. Proposition 4.4 tells that we only need to consider the transmit powers corresponding to the rate-tuple on the Pareto frontier of the 1-slot throughput region, instead of all possible transmit powers. Hence, the transmit-power-tuple set $\mathcal{S}$ in Problem 4.1 can be substituted by $\overline{\mathcal{S}}$, called the refined transmit-power-tuple set, such that $\mu_{\max}(s_t, L, \varepsilon) \in \overline{\mathcal{M}}_{[1]}$ holds for all $s_t \in \overline{\mathcal{S}}$. Therefore, the branching factor is $\#\mathcal{S} = \#\overline{\mathcal{M}}_{[1]}$.

Step 2: More importantly, A* search is employed to further improve the searching efficiency while maintaining the optimality for Problem 4.1. A brief description is given here on the application of A* search in solving Problem 4.1, while we refer the readers to Chapter 3.5.2 in [84] for a complete description of the A* search algorithm.

For the A* search (or any searching algorithm in general), a node is a fundamental concept. In our case, the node is $(Q_t, (s_i)_{i=1}^t)$, which depends on Q_t the state, and $(s_i)_{i=1}^t$ is the path to achieve this state from initial node $(Q_0, \emptyset)$. The A* search requires five components to be implemented:

- **Initial node.** The node for starting the search, which is $(Q_0, \emptyset)$.
- **Action space.** The set of actions that move from a node to all possible child nodes. In our case, the action space is $\overline{\mathcal{S}}$.
- **Goal.** The condition for stopping the search. In our case, the goal is $Q_p = 0$, or simply denoted as 0.
- **Step cost.** The step cost is the cost for each searching step. In Problem 4.1, it is

$$c_t = \begin{cases} 1 & t < p, \\ \max\limits_{n \in \mathcal{N}} \left\lceil \frac{Q_0^{(n)}}{\sum_{t=1}^{p} \mu_{\max}^{(n)}(\gamma_n(s_t), L, \varepsilon)L} \right\rceil & t = p. \end{cases} \tag{4.37}$$

- **Evaluation function.** It records the path cost (the summation of step cost) from the past and estimates the path cost in the future. To be more specific, for a given node $(Q_t, (s_i)_{i=1}^t)$, the evaluation function $F(\cdot, \cdot)$ is

$$F\left(Q_t, (s_i)_{i=1}^t\right) = G\left((s_i)_{i=1}^t\right) + E(Q_t), \tag{4.38}$$

where $G\left((s_i)_{i=1}^{t}\right)$ returns the path cost from initial node to node $(Q_t,(s_i)_{i=1}^{t})$ and $E(Q_t)$, called a heuristic function, estimates the path cost from $(Q_t,(s_i)_{i=1}^{t})$ to the goal 0. The A* search always expands the node with the smallest F.

It should be noted that the core of the A* search is to construct a function $E(\cdot)$ satisfying $E(Q_t) \leq E^*(Q_t)$ for every Q_t, where $E^*(Q_t)$ is the actual cost from Q_t to the goal 0. This constructed function is known as the admissible heuristic function in the artificial intelligence literature [84]. In this work, we propose the interference-free based heuristic function as follows

$$E^I(Q_t) = \max_{n\in\mathcal{N}} \frac{Q_t^{(n)}}{\mu_{\max}^{(n)}(\gamma_n'(s_{\max}^{(n)}),L,\varepsilon)L}, \tag{4.39}$$

where $t \in \{1,\ldots,p\}$, $s_{\max}^{(n)} = \max \mathcal{S}^{(n)}$, and

$$\gamma_n'(s_{\max}^{(n)}) = \frac{H_{nn}s_{\max}^{(n)}}{W_n}, \quad n \in \mathcal{N}. \tag{4.40}$$

We call this heuristic function interference-free based, since compared to (B.35), the expression (4.40) does not consider the interference from other transmitters. The following proposition shows that $E^I(\cdot)$ is admissible.

Proposition 4.5. Let the actual cost to reach the goal $Q_p = 0$ be

$$E^*(Q_t) := \begin{cases} p-1+\max\limits_{n\in\mathcal{N}}\left[\frac{Q_0^{(n)}}{\sum_{t=1}^{p}\mu_{\max}^{(n)}(\gamma_n(s_t),L,\varepsilon)L}\right]-t & t<p, \\ 0 & t=p, \end{cases} \tag{4.41}$$

where $t \in \{1,\ldots,p\}$. Then $E^I(Q_t) \leq E^*(Q_t)$ holds for every Q_t.

Proof: See Appendix C.3. ■

Remark 4.6. Based on Proposition 4.5, $E^I(Q_t)$ in (4.39) provides an A* search for Problem 4.1, which improves the computational efficiency and maintains the optimality. Note that the heuristic function in an A* search and the heuristic method in optimization are two totally different concepts: the latter is often suboptimal, while the former is always optimal once it is admissible. Thus, Proposition 4.5 indeed gives the optimality of our search algorithm. In terms of the computational efficiency, since an A* search reduces the number of nodes to be expanded, it avoids many redundant calculations, which improves the efficiency (see Section 4.5). We stress that the computational efficiency is high in the cases interference is strong or zero, because for these cases the optimal choice of node has a smaller $E^I(Q_t)$ than that in other nodes so that our A* search tends to have significantly fewer steps.

Step 3: Finally, we propose two pruning strategies to further improve the searching efficiency of the A* search:

- After a node is selected by the evaluation function (4.38), say $(Q_{t_1},(s_i)_{i=1}^{t_1})$, we will check whether the condition "$t_1 = T$ and $Q_{t_1} \neq 0$" holds. If this condition holds, then we delete this node from the fringe (or called open set, more details can be found in [84]). This is because the condition "$t_1 = T$ and $Q_{t_1} \neq 0$" corresponds to the node whose data queue has not been cleared in the T^{th} time slot, and there is no need to expand such a node. This consideration is reasonable, since: for the rate margin, Algorithm 4.1 does not need to know any exact value of p^* for $p^* > T$, i.e., any node with transmission time greater than T is not considered: and for the rate-achieving policy deriving, we just need to consider the nodes with transmission time not greater than T.
- After selecting a node $(Q_{t_1},(s_i)_{i=1}^{t_1})$ to expand, we delete those nodes with $t \geq t_1$ but with $(\mu_{\max}(s_i,L,\varepsilon))_{i=1}^{t} \preceq (\mu_{\max}(s_i,L,\varepsilon))_{i=1}^{t_1}$ in the fringe, since those nodes' child nodes are suboptimal.

To sum up, our algorithm for solving Problem 4.1 is given in Algorithm 4.2, where the A* search algorithm, with our pruning strategy, is

$$\text{A}^*\big(\text{initial node, action space, goal, step cost, evaluation function}\big). \tag{4.42}$$

We omit the details of the A* search here, since, other than the pruning strategy we already illustrated, the other parts of the A* search algorithm can be found in standard textbooks (e.g. [84]).

Algorithm 4.2 Solving Problem 4.1 with A* Search

Require: T number of time slots; N the number of transmitter-receiver pairs; $\mu_{[T]}$ the given average rate-tuple;
$\overline{\mathcal{S}}$ the constrained transmit-power-tuple set.

Ensure: $(s_t^*)_{t=1}^{p^*}$ the optimal solutions for Problem 4.1;
p^* and $\max\limits_{n\in\mathcal{N}}\left\{Q_0^{(n)} / \left[\sum_{t=1}^{p}\mu_{\max}^{(n)}(\gamma_n(s_t),L,\varepsilon)L\right]\right\}$ for the optimal objective in Problem 4.1.

1: $Q_0 = T\mu_{[T]}L$;

2: $\left[(s_t^*)_{t=1}^{p^*}, p^*, \max\limits_{n\in\mathcal{N}}\left\{Q_0^{(n)}/\sum_{t=1}^{p^*}\mu_{\max}^{(n)}(\gamma_n(s_t^*),L,\varepsilon)L\right\}\right] = \text{A}^*\left((Q_0,\emptyset),\overline{\mathcal{S}},0,c_t,F(\cdot)\right)$;

3: **return** $(s_t^*)_{t=1}^{p^*}$, p^* and $\max\limits_{n\in\mathcal{N}}\left\{Q_0^{(n)}/\sum_{t=1}^{p^*}\mu_{\max}^{(n)}(\gamma_n(s_t^*),L,\varepsilon)L\right\}$.

Remark 4.7 (Measuring the Searching Efficiency)**.** We propose the effective branching ratio (EBR) as the metric for evaluating the search efficiency of our solution method of Problem 4.1:

$$\text{EBR} = \frac{B}{\#\mathcal{S}}, \tag{4.43}$$

where B is the effective branching factor of our method, and the $|\mathcal{S}|$ is the branching factor of the original search tree (see the discussion at the beginning of this subsection). B is a metric on expanded nodes such that if the total number of expanded nodes is U, then

$$U = \sum_{t=1}^{p^*} B^t. \tag{4.44}$$

We can see that B increases with U, which means the smaller EBR is, the more efficient in our algorithm performs. We will use the proposed EBR in Section 4.5 to examine the searching efficiency.

4.5 Numerical Results

To corroborate our theoretical results, numerical results are presented. In this section, firstly, we provide two illustrative examples on achievable/unachievable rate-tuples, respectively: For the achievable rate-tuple, we give the rate-achieving policy and calculate the rate margin followed by the explanation of its meaning. For the unachievable rate, we calculate the rate margin and explain its meaning. Secondly, we conduct the Monte Carlo simulation to highlight the computational efficiency of our methods. Note that in this section, the maximum achievable rate-tuple for transmitter-receiver pair $n \in \mathcal{N}$ is calculated by (4.4).

Consider the transmission-rate design for a given network with $N = 3$ transmitter-receiver pairs within $T = 5$ time slots, each contains $L = 100$ channel uses. The following parameters are at hand (the corresponding units are normalized): The transmit-power sets of these 3 transmitter-receiver pairs are $\mathcal{S}^{(1)} = \mathcal{S}^{(2)} = \mathcal{S}^{(3)} = \{0,5\}$, each having an on-off structure. The power gains are $H_{11} = 0.8$, $H_{22} = 0.7$, $H_{33} = 0.9$, $H_{12} = H_{21} = 0.15$, $H_{13} = H_{31} = 0.25$ and $H_{23} = H_{32} = 0.3$. The noise powers are $W_1 = W_2 = W_3 = 0.1$.

Now consider whether the rate-tuple $\mu_{[5]} = [0.5, 0.5, 0.5]^{\mathrm{T}}$ can be achieved with error probability $\varepsilon = 0.001$. Using Theorem 4.2, $\mu_{[5]}$ can be achieved, and the rate-achieving policy is $\mathcal{P}_5 = (\mu_t, s_t)_{t=1}^5$, where $\mu_1 = [0, 2.2698, 0]^{\mathrm{T}}$, $\mu_2 = [0, 0, 2.4466]^{\mathrm{T}}$, $\mu_3 = [2.3636, 0, 0]^{\mathrm{T}}$, $\mu_4 = [0.1364, 0.2302, 0.0534]^{\mathrm{T}}$, $\mu_5 = 0$, and $s_1 = [0,5,0]^{\mathrm{T}}$, $s_2 = [0,0,5]^{\mathrm{T}}$, $s_3 = [5,0,0]^{\mathrm{T}}$, $s_4 = [5,5,5]^{\mathrm{T}}$, $s_5 = 0$. This result means that the required rate-tuple can be met in this network. Furthermore, transmission finished in 4 time slots, since in time slot 5, all transmitter-receiver pairs transmit nothing.

To maximally utilize the throughput region of this given network, we can linearly scale up $\mu_{[5]}$ so that each transmitter-receiver pair enjoys a rate increase without changing the fairness (i.e., the direction of rate-tuple). We employ Theorem 4.1 (details are shown in Algorithm 4.1) to compute the rate margin $\delta_5(\mu_{[5]}) = 1.2554$. Hence, the boundary rate-tuple is $\delta_5(\mu_{[5]})\mu_{[5]} = [0.6277, 0.6277, 0.6277]^{\mathrm{T}}$.

Secondly, we consider whether the rate-tuple $\mu'_{[5]} = [0.3, 1, 1]^{\mathrm{T}}$ can be achieved in this network. Unfortunately, by Theorem 4.2, the rate-tuple $\mu'_{[5]}$ is not achievable, since $\delta_5(\mu'_{[5]}) = 0.9079 < 1$. However, Theorem 4.1 says that in the case of not changing the parameters of the network, $\mu'_{[5]}$ should be at least linearly scaled down to $\delta_5(\mu'_{[5]}) = 0.9079$ of $\mu'_{[5]}$ to become achievable. Otherwise, the network should be redesigned (e.g., enlarge the maximum power of transmitter-receiver pairs).

To corroborate the efficiency of our methods, we conduct the Monte Carlo simulations. The average iteration number (AIN) and the average effective branching ratio (AEBR) for deriving rate margin are employed to measure the behaviors: The AIN represents on average how many iterations are required to derive the rate margin (see Theorem 4.1), and the AEBR reveals the searching efficiency for solving Problem 4.1 in every iteration.

The simulation parameters are given as follows. The number of transmitter-receiver pairs are $N = 3$, and the transmit-power sets are $\mathcal{S}^{(1)} = \mathcal{S}^{(2)} = \mathcal{S}^{(3)} = \{0, 1, 2\}$. The power gains are $H_{11} = H_{22} = H_{33} = 0.5$, and $H_{12} = H_{21} = H_{13} = H_{31} = H_{23} = H_{32} = 0.3$. The noise powers are $W_1 = W_2 = W_3 = 0.1$. Similar to two above examples, the blocklength is $L = 100$, and the error probability is $\varepsilon = 0.001$. Simulations are conducted for $T \in \{2, 3, 4, 5\}$, and for each T, we randomly and uniformly select 1000 different $\mu_{[T]}$ from $\Lambda_{[\infty]}$ to calculate the rate margin $\delta_T(\mu_{[T]})$. The results are shown in Table 4.1.

Table 4.1: Average Iteration Number and the Average Effective Branching Ratio

	$T = 2$	$T = 3$	$T = 4$	$T = 5$
AIN	1.802	1.844	2.163	2.627
AEBR	0.385	0.231	0.146	0.095

From Table 4.1, we can see that the AINs are reasonably small. For the searching efficiency in each iteration, the AEBRs are small and decrease with T. To give an intuitive illustration, we take $T = 5$ as an example: AIN equals 2.556 means that we need 2.556 iterations on average to derive the rate margin. AEBR is 0.095 implies that if we assume $p^* = T = 5$, the total number of nodes (except for the start node) of original search tree is $\sum_{t=1}^{5}(\#\mathcal{S})^t = \sum_{t=1}^{5} 27^t = 1.490 \times 10^7$, while, for our A* search, only $\sum_{t=1}^{5}(0.095 * 27)^t = 180$ number of nodes are expanded on average. It can be seen that our algorithm significantly improves the computational efficiency.

4.6 Summary

In this chapter, the finite-horizon throughput region for a multi-user interference channel in MWNs has been studied. We proposed rate margin as a metric to determine the achievability of any given rate-tuple and measure the ability to scale up (down) for any achievable (un-

achievable) rate-tuple so that the resulting rate-tuple is still within (brought back into) the finite-horizon throughput region. Also, we provided a complete algorithm for finding a rate-achieving policy for any achievable rate-tuple. Both the rate margin and the rate-achieving policy can be derived very efficiently by using a modified A* search algorithm, where the interference-free based heuristic function plays an important role.

This work represents a significant step towards understanding the network throughput region over a finite time horizon beyond the simplest one-time-slot scenario. It also demonstrates the fundamental differences in the throughput region between finite and infinite horizon.

Chapter 5

Conclusions

In this chapter, we first summarize the general conclusions drawn from the thesis, and then outline some future research directions arising from this work.

5.1 Thesis Conclusions

This thesis considers the communications in MWNs from a finite time-horizon perspective. Since the short-term effect is highly dependent on the node mobility in MWNs, we have proposed two mobility models in Chapter 2 and Chapter 3, respectively, which can describe or approximate a large class of mobilities in the real world. Additionally, in Chapter 2 the interference prediction has been provided and analyzed to effectively predict the time-varying interference; and in Chapter 3 the cooperative localization algorithm has been designed to reduce the uncertainty in node mobility. Finally, in Chapter 4 the throughput region for multi-user interference channels is studied to give the fundamental limit on the information transmission between multiple transmitter-receiver pairs in MWNs. The detailed contributions and directly related future works are given as follows.

Interference prediction: In Chapter 2, the interference prediction is studied to effectively predict any time-varying interference in MWNs. We have proposed the GLC model to describe or approximate a large class of mobilities in the real world. Based on the GLC model, we have studied the interference prediction problem in MWNs. The statistics of interference prediction with respect to a dynamic reference point have been analyzed. We have defined the CGPPF as a general framework to compute the mean and MGF, where some important closed-form expressions can be obtained. With expressing the CGPPF in series form, we have analyzed the limiting behavior of the statistics of the interference prediction, and given the necessary and sufficient conditions for when the node locations can be regarded as a Gaussian BPP when analyzing the statistics of interference prediction.

The presented work serves as the first step to develop more comprehensive results in interference predictions. Even though the CGPPF provides a general framework for calculating the statistics of interference prediction, the closed-form expressions exist only in some

special cases. It is possible to derive closed-form approximations for these statistics using a cumulant-based approach (e.g., similar to [85, 86]). Furthermore, for the limiting behavior of interference prediction with homogenous mobilities, it would be desirable to obtain a more direct link between the mobility model and the condition for the BPP approximations stated in Theorem 2.4. For example, we observed from our numerical results that if the matrix A of the mobility model is Lyapunov-stable [68], the condition for Gaussian BPP approximation holds, and vise versa. On the other hand, it would be interesting to extend the results on finite number of nodes to infinite number of nodes and study the condition under which the limiting behavior of the interference prediction converges to that from a PPP. Another interesting direction for future research is to generalize the assumption for $w_i(t)$ beyond Gaussian. Last but not least, the results on interference prediction obtained in this work can also been used to predict the outage probability at a future time instant.

Cooperative localization: In Chapter 3, we study the cooperative localization problem with asynchronous communications and measurements for the first time. We have proposed a more general mobility model by using non-linear stochastic differential equations, where the GLC model can be regarded as a special case of this model. An important concepts AMS triple, prior cut, and cut gap, have been introduced in analyzing the estimation gap between the proposed centralized algorithm (serves as the benchmark) and the algorithm with an AMS triple. We have strict proved that if the cut gap is small enough, i.e., the two cut parts are nearly independent, then the estimation gap can also be very small. With this important property, we have designed the prior-cut algorithm to solve the cooperative localization problem with asynchronous communications and measurements.

The presented work serves as the first step to develop the asynchronous localization problems. For future work, it is meaningful to analyze the property of the prior-cut algorithm, e.g., the stability and the convergence speed.[1] Also, it is interesting to combine the continuous-discrete Bayesian filter framework with the famous belief propagation algorithm.

Finite-horizon throughput region: In Chapter 4, the finite-horizon throughput region for multi-user interference channels in MWNs has been studied. We have proposed the rate margin as a metric to determine the achievability of any given rate-tuple and measure the ability to scale up (down) for any achievable (unachievable) rate-tuple so that the resulting rate-tuple is still within (brought back into) the finite-horizon throughput region. Furthermore, we have provided a complete algorithm for finding a rate-achieving policy for any achievable rate-tuple. Both the rate margin and the rate-achieving policy can be derived very efficiently by using a modified A* search algorithm, where the interference-free based heuristic function plays an important role. This work represents a significant step towards understanding the network

[1] The stability of Bayesian filtering (see [87, 88, 89, 90, 91, 92]) will play an important role in analyzing the stability of the prior-cut algorithm, and the convergence rate of Bayesian filtering (see [93]) will be important to the convergence rate of the prior-cut algorithm.

throughput region over a finite time horizon beyond the simplest one-time-slot scenario. It also demonstrates the fundamental differences in the throughput region between finite and infinite horizon.

More importantly, the presented work serves as the first step to develop more comprehensive results on finite-horizon throughput region in the future:

- The rate margin defined in this chapter resolves how to do rate-scaling when preserving fairness. If some of the transmitter-receiver pairs have more priority for scaling, then a generalization or different definitions of rate margin can be used to reflect the rate scalability from different design perspectives. For example, if a transmitter-receiver pair is predominant, then we need to consider the maximum scalability of one component corresponding to this communication pair, while keeping other components unchanged.

- It is worth trying to relax or remove the assumption of treating interference as noise. If one considers interference decoding (e.g., [94, 95]), the finite-horizon throughput region will be enlarged. It would be very interesting to consider interference decoding in the finite blocklength regime.

- Ultimately, it would be desirable to generalize the finite-horizon throughput region towards an information-theoretic setting which contains all possible coding/decoding strategies. One potential approach is the deterministic approximation approach (see [96, 97]) to derive an easy-to-compute approximated finite-horizon throughput region.

5.2 Future Research Directions

The finite-horizon communications in MWNs is important in meeting the requirements of low latency and high reliability. Future research problems on this topic include:

Advanced mobility models in MWNs: The mobility models proposed in Chapter 2 and Chapter 3 assume that the mobility parameters are known in advance. For example, we assume the mobile users follow the Brownian motion. But actually the mobile users may not perfectly satisfy this assumption, and the true model's parameters can be unknown in advance. For example, the true model can be Brownian motion with inertia (unknown in advance), but we wrongly take it as the standard Brownian motion. To deal with the modeling uncertainties, it is necessary to study the advanced mobility models in the future, and two promising ways are highlighted as follows:

- *Robust mobility model.* The modeling uncertainty is indeed the uncertainty in model parameters whose exact values are not known in advance. Nevertheless, we can specify the range of these parameters, then a group of location pattern processes (see Section 1.2.1) can be derived. If the difference among these location pattern processes is tolerable, then

the mobility model with such a parameter range is robust. This is inspired by the robust control [98, 99]. For example, a car's unknown velocity can be within a valid range, and sometimes the location pattern processes are with little difference for this velocity range in a short time window.

- *Adaptive mobility model.* In contrast to the robust mobility model which try to tolerate the modeling uncertainty, the adaptive mobility model concentrates on learning the parameters intelligently. This mobility model can attract more attention than the robust mobility model, since it links to the classic techniques in machine learning or even the popular deep neural network techniques.

Distributed finite-horizon communication designs in MWNs: The presented finite-horizon throughput region in Chapter 4 gives the fundamental limits on data transmissions of multiple transmitter-receiver pairs. Even though we provide a rate-achieving policy to achieve any achievable rate-tuple in any given finite-horizon throughput region, this method is in a centralized manner. Since sometimes there are no centers to schedule the communications, we need design the rate-achieving policy in a distributed manner.

One possible way is to modify the distributed rate-achieving policy of an infinite-horizon throughput region (e.g., [60]) such that it can be suitable for finite-horizon throughput region. However, the rate-achieving policies for infinite-horizon throughput regions rely on constructing Lyapunov functions which is an infinite time-horizon based method. Thus, there would be a big challenge in designing the rate-achieving policy directly from existing methods in infinite-horizon throughput regions.

One promising way is to use distributed optimization methods (see [100] for a good tutorial). More specifically, Problem 4.1 should be solved by distributed optimization methods. Nevertheless, the problem is not convex and the optimality cannot be guaranteed, which means the method cannot always find a rate-achieving policy even if it is in the finite-horizon throughput region. It will be very important and interesting to find the optimal solution to Problem 4.1 in a distributed manner when we design the distributed rate-achieving policies.

Other topics in MWNs: There are many other potential topics on short-term communications in MWNs. Firstly, the short-term communication is important for secure transmission, since decoding a short blocklength code highly relies on the short-term channel gain which is known to the receiver but hard to be timely estimated by the eavesdropper. Secondly, for the channel estimation in MWNs, the communication channel changes in time due to the node mobility. If short-term communications are considered, then the channel estimations have to be done frequently to make sure the channel information is not outdated. But if we combine the channel prediction with the channel estimation based on node mobility, the channel can be tracked in a real-time manner. Last but not least, some networked control systems, e.g., the UAV formations, swarms, cooperative tasks, are highly dependent on the short-term commu-

nications[2] in MWNs. This means many interdisciplinary topics between control theory and communication theory can be studied based on the short-term communications in MWNs.

[2]control systems are sensitive to system delays

Appendix A

A.1 Proof of Lemma 2.1

A more general proof can be found in [69]. For completeness, we provide a proof of Lemma 2.1 here:

The solution of (2.2) is

$$x_i(t) = e^{A_i(t-t_0)}x_i(t_0) + \int_{t_0}^{t} e^{A_i(t-\tau)}w_i(\tau)\mathrm{d}\tau \tag{A.1}$$

and then the mean of $x_i(t)$ can be derived as

$$\mathbb{E}[x_i(t)] = e^{A_i(t-t_0)}x_i(t_0) + \int_{t_0}^{t} e^{A_i(t-\tau)}\,\mathbb{E}[w_i(\tau)]\,\mathrm{d}\tau \overset{(a)}{=} e^{A_i(t-t_0)}x_i(t_0), \tag{A.2}$$

where (a) is established by $\mathbb{E}\big[w_i(\tau)\big] = 0$. Then we can get (2.7) from (2.3).

The covariance of $x_i(t)$ is given by

$$\begin{aligned}\mathrm{Cov}\big[x_i(t)\big] &\overset{(a)}{=} \mathrm{Cov}\big[e^{A_i(t-t_0)}x_i(t_0)\big] + \mathrm{Cov}\Big[\int_{t_0}^{t} e^{A_i(t-\tau)}w_i(\tau)\,\mathrm{d}\tau\Big] \\ &= 0 + \mathrm{Cov}\Big[\int_{t_0}^{t} e^{A_i(t-\tau)}w_i(\tau)\,\mathrm{d}\tau\Big],\end{aligned} \tag{A.3}$$

where (a) follows from the independence of $x_i(t_0)$ and $w_i(\tau)\,(\tau \in [t_0,t])$. Additionally,

$$\begin{aligned}&\mathrm{Cov}\Big[\int_{t_0}^{t} e^{A_i(t-\tau)}w_i(\tau)\,\mathrm{d}\tau\Big] \\ &= \mathrm{Cov}\Big(\int_{t_0}^{t} e^{A_i(t-\tau)}w_i(\tau)\,\mathrm{d}\tau, \int_{t_0}^{t} e^{A_i(t-\psi)}w_i(\psi)\,\mathrm{d}\psi\Big) \\ &= \int_{t_0}^{t}\int_{t_0}^{t} \mathrm{Cov}\big(e^{A_i(t-\tau)}w_i(\tau), e^{A_i(t-\psi)w_i(\psi)}\big)\,\mathrm{d}\tau\,\mathrm{d}\psi \\ &= \int_{t_0}^{t}\int_{t_0}^{t} e^{A_i(t-\tau)}\,\mathrm{Cov}\big(w_i(\tau), w_i(\psi)\big)e^{A_i^T(t-\psi)}\,\mathrm{d}\tau\,\mathrm{d}\psi \\ &\overset{(a)}{=} \Theta_{x_i}(t),\end{aligned}$$

where (a) follows the properties of GWN. Then (2.8) can be derived from (2.3).

A.2 Proof of Theorem 2.3

With the orthogonal transform $z = P^T y$ such that $P^T \mu = \eta = [\eta_a]_{d\times 1}$ and

$$P^T \Sigma^{-1} P = \mathrm{diag}\{1/\sigma_1^2, \ldots, 1/\sigma_d^2\} = \Lambda.$$

Then, equation (2.24) can be rewritten as

$$\begin{aligned} G[\nu] &= \int_{\mathbb{R}^d} \nu(\|y\|) \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(y-\mu)^T \Sigma^{-1}(y-\mu)} \,\mathrm{d}y \\ &= \int_{\mathbb{R}^d} \nu(\|z\|) \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(z-\eta)^T \Lambda(z-\eta)} \,\mathrm{d}z \\ &\overset{(a)}{=} \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \int_0^\infty \left[\nu(r) e^{-\frac{1}{2}(r\Phi-\eta)^T \Lambda(r\Phi-\eta)} r^{d-1} \right] \mathrm{d}r \int_\Theta V(\phi) \,\mathrm{d}\phi, \end{aligned} \tag{A.4}$$

where (a) follows a d-dimensional spherical transform. $\Phi = [\Phi_a]_{d\times 1}$ and $V(\phi)\mathrm{d}\phi$ are shown in (2.31) and (2.32), respectively. Labeling

$$\overline{\nu}(r, \Phi) = \int_0^\infty \nu(r)\, e^{-\frac{1}{2}(r\Phi-\eta)^T \Lambda(r\Phi-\eta)} r^{d-1} \,\mathrm{d}r, \tag{A.5}$$

we expand it into the Taylor series

$$\overline{\nu}(r, \Phi) = \sum_{n=0}^{\infty} \frac{(-1)^n}{2^n n!} \int_0^\infty \left\{ \nu(r) r^{d-1} \left[(r\Phi - \eta)^T \Lambda (r\Phi - \eta) \right]^n \right\} \mathrm{d}r. \tag{A.6}$$

By employing the multinomial theorem,

$$\begin{aligned} &\left[(r\Phi - \eta)^T \Lambda (r\Phi - \eta) \right]^n = \\ &\qquad \sum_{k_1+k_2+k_3=n} \left[\binom{n}{k_1, k_2, k_3} \left(\sum_{a=1}^d \frac{\Phi_a^2}{\sigma_a^2} \right)^{k_1} \left(\sum_{b=1}^d -\frac{2\Phi_b \eta_b}{\sigma_b^2} \right)^{k_2} \left(\sum_{q=1}^d \frac{\eta_q^2}{\sigma_q^2} \right)^{k_3} r^{2k_1+k_2} \right], \end{aligned} \tag{A.7}$$

where

$$\left(\sum_{a=1}^d \frac{\Phi_a^2}{\sigma_a^2} \right)^{k_1} = \sum_{k_1^{(1)}+\cdots+k_1^{(d)}=k_1} \left[\binom{k_1}{k_1^{(1)}, \ldots, k_1^{(d)}} \prod_{a=1}^d \left(\frac{\Phi_a}{\sigma_a} \right)^{2k_1} \right], \tag{A.8}$$

$$\left(\sum_{b=1}^d -\frac{2\Phi_b \eta_b}{\sigma_b^2} \right)^{k_2} = \sum_{k_2^{(1)}+\cdots+k_2^{(d)}=k_2} \left[\binom{k_2}{k_2^{(1)}, \ldots, k_2^{(d)}} \prod_{b=1}^d \left(-\frac{2\Phi_b \eta_b}{\sigma_b^2} \right)^{k_2} \right]. \tag{A.9}$$

Thus, (A.4) can be written as (2.27).

A.3 Integrations in CGPPF Series Form

A.3.1 Derivation for $\Psi[\nu]$ in (2.28)

In order to calculate the interference prediction mean, we set $\nu(r)=\frac{1}{\varepsilon+r^{\alpha}}$, thus $\Psi[\nu]$ is

$$\Psi[\nu]=\begin{cases}\frac{R^{c+1}H_2F_1\left(1,\frac{c+1}{\alpha},\frac{c+1+\alpha}{\alpha},-\frac{R^{\alpha}}{\varepsilon}\right)}{(1+c)\varepsilon} & \varepsilon>0\\ \frac{R^{c-\alpha+1}}{c+\alpha-1} & \varepsilon=0, c-\alpha+1>0,\end{cases} \tag{A.10}$$

where $c=2k_1+k_2+d-1$ and $H_2F_1(\cdot)$ is the hypergeometric function [101]. When $\varepsilon=0$, the condition $c-\alpha+1>0$ should be satisfied, this is because the singularity at 0.

To calculate the MGF of interference prediction, we set $\nu(r)=\left[\frac{m}{m-\beta\frac{1}{\varepsilon+r^{\alpha}}}\right]^{m}$, thus $\Psi[\nu]$ is

$$\begin{aligned}\Psi[\nu]=R^{c+1}\left(\varepsilon-\beta+R^{\alpha}\right)^{-m}\Bigg[&\frac{\varepsilon\left(\varepsilon-\beta+R^{\alpha}\right)}{(1+c)(\varepsilon-\beta)}\cdot\\ &H_2F_1\left(1,\frac{1+c+\alpha(1-m)}{\alpha},\frac{1+c+\alpha}{\alpha},\frac{R^{\alpha}}{\beta-\varepsilon}\right)+\\ &\frac{R^{\alpha}\left(1-\frac{R^{\alpha}}{\beta-\varepsilon}\right)^{m}H_2F_1\left(\frac{1+c+\alpha}{\alpha},m,\frac{1+c+2\alpha}{\alpha},\frac{R^{\alpha}}{\beta-\varepsilon}\right)}{1+c+\alpha}\Bigg],\end{aligned} \tag{A.11}$$

where $\varepsilon\neq\beta$.

A.3.2 Derivation for Ω in (2.29)

For $d=1$,

$$\Omega=\frac{(-2)^{k_2}\eta^{k_2+2k_3}}{\sigma^{2(k_1+k_2+k_3)}}. \tag{A.12}$$

In order to simplified the discussions for the cas $d=2$, we define two functions

$$I_{[0,\pi]}(m,n)=\begin{cases}\sum_{l=0}^{\frac{m}{2}}\binom{m/2}{l}(-1)^{l}\frac{\sqrt{\pi}\Gamma(\frac{2l+n+1}{2})}{\Gamma(\frac{2l+n+2}{2})} & m \text{ is even}, m>0, n\geq 0\\ 0 & m \text{ is odd}, m>0, n\geq 0\\ \frac{\sqrt{\pi}\Gamma(\frac{n+1}{2})}{\Gamma(\frac{n+2}{2})} & m=0, n\geq 0\end{cases} \tag{A.13}$$

and

$$I_{[0,2\pi]}(m,n)=\begin{cases}\Upsilon(m,n) & n \text{ is even}, m\geq 0, n>0\\ 0 & n \text{ is odd}, m\geq 0, n>0\\ \frac{\sqrt{\pi}\Gamma(\frac{n+1}{2})}{\Gamma(\frac{n+2}{2})} & m\geq 0, n=0\end{cases}, \tag{A.14}$$

where

$$\Upsilon(m,n)=\sum_{l=0}^{\frac{n}{2}}\binom{n/2}{l}(-1)^l\frac{\left[1+(-1)^{2l+m}\right]\sqrt{\pi}\Gamma(\frac{2l+m+1}{2})}{\Gamma(\frac{2l+m+2}{2})}.$$

Therefore, with $d=2$, we have

$$\Omega=\left[\sum_{i=1}^{2}\left(\frac{\eta_i}{\sigma_i}\right)^2\right]^{k_3}\sum_{\substack{k_1^{(1)}+k_1^{(2)}=k_1\\ k_2^{(1)}+k_2^{(2)}=k_2}}\binom{k_1}{k_1^{(1)},k_1^{(2)}}\binom{k_2}{k_2^{(1)},k_2^{(2)}}\Xi, \tag{A.15}$$

where

$$\begin{aligned}\Xi=&\left[\prod_{\substack{1\leq i\leq 2\\ 1\leq j\leq 2}}\left(\frac{1}{\sigma_i}\right)^{2k_1^{(i)}}\left(-2\frac{\eta_j}{\sigma_j^2}\right)^{k_2^{(j)}}\right]\cdot\\ &\left[I_{[0,2\pi]}\left(2k_1^{(1)}+k_2^{(1)},0\right)I_{[0,2\pi]}\left(2k_1^{(1)},k_2^{(2)}\right)\cdot\right.\\ &\left.I_{[0,2\pi]}\left(k_2^{(1)},2k_1^{(2)}\right)I_{[0,2\pi]}\left(0,2k_1^{(2)}+k_2^{(2)}\right)\right].\end{aligned} \tag{A.16}$$

A.4 Some Closed Forms for CGPPF

If condition (2.26) is satisfied, some closed-form expressions for $\mathbb{E}[I_i(t|s)]$ and $\mathbb{E}[I_i^2(t|s)]$ can be derived. Please refer to (2.33) for the definition of α and ε.

Firstly, if $\varepsilon=0$, which implies a singular path loss, we can derive the closed-form expressions for first and second order statistics as

$$\mathbb{E}\left[I_i(t|s)\right]=\int_0^\infty\frac{v_d r^{d-1}}{(2\pi)^{\frac{d}{2}}\sigma^d r^\alpha}e^{-\frac{r^2}{2\sigma^2}}\mathrm{d}r=\frac{v_d\Gamma\left[\frac{d-\alpha}{2}\right]}{2^{\frac{\alpha}{2}+1}\pi^{\frac{d}{2}}\sigma^\alpha} \tag{A.17}$$

$$\begin{aligned}\mathbb{E}\left[I_i^2(t|s)\right]&=\frac{m+1}{m}\int_0^\infty\frac{v_d r^{d-1}}{(2\pi)^{\frac{d}{2}}\sigma^d r^{2\alpha}}e^{-\frac{r^2}{2\sigma^2}}\mathrm{d}r\\ &=\frac{(m+1)v_d\Gamma\left[\frac{d-2\alpha}{2}\right]}{m2^{\alpha+1}\pi^{\frac{d}{2}}\sigma^{2\alpha}},\end{aligned} \tag{A.18}$$

when gamma function $\Gamma(\cdot)$ has finite value. v_d is the volume of the d-dimensional ball of radius 1, i.e.,

$$v_d = \frac{2\pi^{d/2}}{\Gamma(d/2)}. \tag{A.19}$$

However, if $\varepsilon > 0$, closed-forms are difficult to derive for general $\alpha > 0$. We just give the results for $\alpha = 2$ and $\alpha = 4$. When $\alpha = 2$

$$\mathbb{E}\left[I_i(t|s)\right] = \frac{v_d e^{\frac{\varepsilon}{2\sigma^2}} \varepsilon^{\frac{\alpha-2}{2}} \Gamma(d/2)\Gamma(\frac{2-d}{2}, \frac{\varepsilon}{2\sigma^2})}{2(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} \tag{A.20}$$

$$\begin{aligned}\mathbb{E}\left[I_i^2(t|s)\right] = &\frac{(m+1)v_d(d-4)\Gamma\left(\frac{d-4}{2}\right)}{16m(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}\varepsilon^2\sigma^4} \cdot \\ &\left\{2\sigma^2 e^{\frac{\varepsilon}{2\sigma^2}} \varepsilon^{\frac{d}{2}} \left[\varepsilon + \sigma^2(d-2)\right] \Gamma\left(\frac{4-d}{2}, \frac{\varepsilon}{2\sigma^2}\right) - 2^{\frac{d}{2}}\sigma^d\varepsilon^2\right\}.\end{aligned} \tag{A.21}$$

For $\alpha = 4$ and $d = 2$,

$$\mathbb{E}\left[I_i(t|s)\right] = \frac{2\mathrm{Ci}\left(\frac{\sqrt{\varepsilon}}{2\sigma^2}\right) \sin\frac{\sqrt{\varepsilon}}{2\sigma^2} + \cos\frac{\sqrt{\varepsilon}}{2\sigma^2}\left[\pi - 2\mathrm{Si}\left(\frac{\sqrt{\varepsilon}}{2\sigma^2}\right)\right]}{4\sqrt{\varepsilon}\sigma^2} \tag{A.22}$$

$$\mathbb{E}\left[I_i^2(t|s)\right] = \frac{(m+1)G_{1,3}^{3,1}\left(\begin{smallmatrix}1/2\\0,1/2,3/2\end{smallmatrix}\middle|\frac{\varepsilon}{16\sigma^4}\right)}{4m\varepsilon^{\frac{3}{2}}\sqrt{\pi}\sigma^2}, \tag{A.23}$$

where $\mathrm{Si}(\cdot)$ and $\mathrm{Ci}(\cdot)$ are sine/cosine integral function with the form

$$\mathrm{Si}(z) = \int_0^z \frac{\sin x}{x}\,\mathrm{d}x, \quad \text{and} \quad \mathrm{Ci}(z) = -\int_z^\infty \frac{\cos x}{x}\,\mathrm{d}x,$$

and $G_{p,q}^{m,n}\left(\begin{smallmatrix}a_1,\ldots,a_p\\b_1,\ldots,b_q\end{smallmatrix}|z\right)$ is the Meijer-G function [101].

Appendix B

B.1 Proof of Lemma 3.1

This proof is divided into two parts: In the first part, we prove the recursive equations [i.e., (3.11) in the prediction step and (3.12) in the update step] hold. As a result, $p(X(t_k^{\text{ms}})|Z[t_k^{\text{ms}}])$ can be derived sequentially from $t=0$. In the second part, we show that equation (3.10) holds.

i) In the prediction step, $p(X(t_k^{\text{ms}})|Z[t_{k-1}^{\text{ms}}],W[t_{k-1}^{\text{ms}}])$ can be written as

$$p(X(t_k^{\text{ms}})|Z[t_{k-1}^{\text{ms}}],W[t_{k-1}^{\text{ms}}]) = \int_{\mathcal{X}} p(X(t_k^{\text{ms}}),X(t_{k-1}^{\text{ms}})|Z[t_{k-1}^{\text{ms}}],W[t_{k-1}^{\text{ms}}])\mathrm{d}X(t_{k-1}^{\text{ms}}), \tag{B.1}$$

because the marginal distribution can be obtained by integration. Then, in equation (B.1), we write $p(X(t_k^{\text{ms}}),X(t_{k-1}^{\text{ms}})|Z[t_{k-1}^{\text{ms}}],W[t_{k-1}^{\text{ms}}])$ as

$$\begin{aligned} &p(X(t_k^{\text{ms}})|X(t_{k-1}^{\text{ms}}),Z[t_{k-1}^{\text{ms}}],W[t_{k-1}^{\text{ms}}])p(X(t_{k-1}^{\text{ms}})|Z[t_{k-1}^{\text{ms}}],W[t_{k-1}^{\text{ms}}]) \\ &\qquad \overset{(a)}{=} p(X(t_k^{\text{ms}})|X(t_{k-1}^{\text{ms}}))p(X(t_{k-1}^{\text{ms}})|Z[t_{k-1}^{\text{ms}}],W[t_{k-1}^{\text{ms}}]), \end{aligned} \tag{B.2}$$

where (a) holds with the Markov property of sequence $\{X(t_k^{\text{ms}})\}_{k\in\mathcal{K}}$. Combining (B.1) and (B.2), we have (3.11).

In the update step, we rewrite the posterior $p(X(t_k^{\text{ms}})|Z[t_k^{\text{ms}}],W[t_k^{\text{ms}}])$ by Bayes' rule:

$$p(X(t_k^{\text{ms}})|Z[t_k^{\text{ms}}],W[t_k^{\text{ms}}]) = \frac{p(X(t_k^{\text{ms}}),Z(t_k^{\text{ms}})|Z[t_{k-1}^{\text{ms}}],W[t_k^{\text{ms}}])}{\int_{\mathcal{X}} p(X(t_k^{\text{ms}}),Z(t_k^{\text{ms}})|Z[t_{k-1}^{\text{ms}}],W[t_k^{\text{ms}}])\mathrm{d}X(t_k^{\text{ms}})}, \tag{B.3}$$

where its probability kernel $p(X(t_k^{\text{ms}}),Z(t_k^{\text{ms}})|Z[t_{k-1}^{\text{ms}}],W[t_k^{\text{ms}}])$ has the following form

$$p(Z(t_k^{\text{ms}})|X(t_k^{\text{ms}}),Z[t_{k-1}^{\text{ms}}],W[t_k^{\text{ms}}])p(X(t_k^{\text{ms}})|Z[t_{k-1}^{\text{ms}}],W[t_k^{\text{ms}}]). \tag{B.4}$$

Since $Z(t_k^{\text{ms}})$ is independent of $Z[t_{k-1}^{\text{ms}}]$ and $W[t_{k-1}^{\text{ms}}]$ when $X(t_k^{\text{ms}})$ and $W(t_k^{\text{ms}})$ are given, the first item in (B.4) can be written as $p(Z(t_k^{\text{ms}})|X(t_k^{\text{ms}}),W(t_k^{\text{ms}}))$. For the second term in (B.4), it

can be written as $p(X(t_k^{\text{ms}})|Z[t_{k-1}^{\text{ms}}],W[t_{k-1}^{\text{ms}}])p(W[t_{k-1}^{\text{ms}}]|W(t_k^{\text{ms}}))$. Thus, (B.4) is rewritten as

$$p(Z(t_k^{\text{ms}})|X(t_k^{\text{ms}}),W(t_k^{\text{ms}}))p(X(t_k^{\text{ms}})|Z[t_{k-1}^{\text{ms}}],W[t_{k-1}^{\text{ms}}])p(W[t_{k-1}^{\text{ms}}]|W(t_k^{\text{ms}})). \tag{B.5}$$

We put (B.5) back into (B.3), and (3.12) is derived [note that the common term $p(W[t_{k-1}^{\text{ms}}]|W(t_k^{\text{ms}}))$ in the numerator and denominator can be cancelled out].

ii) For (3.10), the proof is similar to that in (3.11). Using $X(t_k^{\text{ms}})$ as the bridge, we have

$$\begin{aligned} p(X(t)|Z[t],W[t]) &= p(X(t)|Z[t_k^{\text{ms}}],W[t_k^{\text{ms}}]) \\ &= \int_{\mathcal{X}} p(X(t)|X(t_k^{\text{ms}}))p(X(t_k^{\text{ms}})|Z[t_k^{\text{ms}}],W[t_k^{\text{ms}}])\mathrm{d}X(t_k^{\text{ms}}). \end{aligned} \tag{B.6}$$

B.2 Proof of Theorem 3.1

We divide this proof into three parts: In the first two parts, we write $\mathbb{E}_{y_l(t)}[y_l(t)|Z[t_k^{\text{ms}}],W[t_k^{\text{ms}}]]$ and $\mathbb{E}^{\bar{X}(t)}_{y_l(t)}[y_l(t)|Z[t_{k-1}^{\text{ms}}],\bar{Z}(t_k^{\text{ms}}),W[t_{k-1}^{\text{ms}}],\bar{W}(t_k^{\text{ms}})]$ into two suitable forms, respectively, so that they are comparable. In the third part, we complete this proof by comparing those two comparable forms.

i) By (3.8) and (3.10), $\mathbb{E}_{y_l(t)}[y_l(t)|Z[t_k^{\text{ms}}],W[t_k^{\text{ms}}]]$ can be written as

$$\begin{aligned} &\int_{\mathcal{X}} g_l(x_l(t)) \left[\int_{\mathcal{X}} p(X(t)|X(t_k^{\text{ms}}))p(X(t_k^{\text{ms}})|Z[t_k^{\text{ms}}],W[t_k^{\text{ms}}])\mathrm{d}X(t_k^{\text{ms}})\right]\mathrm{d}X(t) \\ &= \int_{\mathcal{X}} \left[\int_{\mathcal{X}} g_l(x_l(t))p(X(t)|X(t_k^{\text{ms}}))\mathrm{d}X(t)\right] p(X(t_k^{\text{ms}})|Z[t_k^{\text{ms}}],W[t_k^{\text{ms}}])\mathrm{d}X(t_k^{\text{ms}}) \\ &= \int_{\mathcal{X}} \left[\int_{\mathcal{X}_l} g_l(x_l(t))p(x_l(t)|X(t_k^{\text{ms}}))\mathrm{d}x_l(t)\right] p(X(t_k^{\text{ms}})|Z[t_k^{\text{ms}}],W[t_k^{\text{ms}}])\mathrm{d}X(t_k^{\text{ms}}) \\ &\overset{(a)}{=} \int_{\mathcal{X}} \mathbb{E}_{x_l(t)}\left[g_l(x_l(t))|x_l(t_k^{\text{ms}})\right] p(X(t_k^{\text{ms}})|Z[t_k^{\text{ms}}],W[t_k^{\text{ms}}])\mathrm{d}X(t_k^{\text{ms}}), \end{aligned} \tag{B.7}$$

where (a) follows from (3.25). We write term $p(X(t_k^{\text{ms}})|Z[t_k^{\text{ms}}],W[t_k^{\text{ms}}])$ in (B.7) as (3.12). Letting $F(X(t_k^{\text{ms}})) = p(Z(t_k^{\text{ms}})|X(t_k^{\text{ms}}),W[t_k^{\text{ms}}])$ and $G_1(X(t_k^{\text{ms}})) = p(X(t_k^{\text{ms}})|Z[t_{k-1}^{\text{ms}}],W[t_k^{\text{ms}}])$, we can re-express (B.7) as

$$\begin{aligned} &\mathbb{E}_{y_l(t)}[y_l(t)|Z[t_k^{\text{ms}}],W[t_k^{\text{ms}}]] = \\ &\qquad \int_{\mathcal{X}} \mathbb{E}_{x_l(t)}\left[g_l(x_l(t))|x_l(t_k^{\text{ms}})\right] \frac{F(X(t_k^{\text{ms}}))G_1(X(t_k^{\text{ms}}))}{\int_{\mathcal{X}} F(X(t_k^{\text{ms}}))G_1(X(t_k^{\text{ms}}))\mathrm{d}X(t_k^{\text{ms}})}\mathrm{d}X(t_k^{\text{ms}}). \end{aligned} \tag{B.8}$$

ii) We rewrite the right-hand side of (3.13) as

$$\int_{\mathcal{X}_l} g_l(x_l(t))p(x_l(t)|Z[t_{k-1}^{\text{ms}}],\bar{Z}(t_k^{\text{ms}}),W[t_{k-1}^{\text{ms}}],\bar{W}(t_k^{\text{ms}}))\mathrm{d}\bar{X}(t), \tag{B.9}$$

where term $p(x_l(t)|Z[t_{k-1}^{\mathrm{ms}}],\bar{Z}(t_k^{\mathrm{ms}}),W[t_{k-1}^{\mathrm{ms}}],\bar{W}(t_k^{\mathrm{ms}}))$ is written as

$$\int_{\bar{\mathcal{X}}(t_k^{\mathrm{ms}})} p(x_l(t)|x_l(t_k^{\mathrm{ms}}))p(\bar{X}(t_k^{\mathrm{ms}})|Z[t_{k-1}^{\mathrm{ms}}],\bar{Z}(t_k^{\mathrm{ms}}),\bar{W}(t_k^{\mathrm{ms}}),W[t_{k-1}^{\mathrm{ms}}])\mathrm{d}\bar{X}(t_k^{\mathrm{ms}}). \tag{B.10}$$

Since $\int_{\check{\mathcal{X}}(t_k^{\mathrm{ms}})} p(\check{X}(t_k^{\mathrm{ms}})|\check{Z}[t_k^{\mathrm{ms}}],\check{W}[t_k^{\mathrm{ms}}])\mathrm{d}X(t_k^{\mathrm{ms}}) \setminus \bar{X}(t_k^{\mathrm{ms}}) = 1$, where we have $\check{\mathcal{X}}(t_k^{\mathrm{ms}}) = \mathcal{X} \setminus \bar{\mathcal{X}}(t_k^{\mathrm{ms}})$, $\check{X}(t_k^{\mathrm{ms}}) = X(t_k^{\mathrm{ms}}) \setminus \bar{X}(t_k^{\mathrm{ms}})$, $\check{Z}[t_k^{\mathrm{ms}}] = Z[t_k^{\mathrm{ms}}] \setminus \bar{Z}(t_k^{\mathrm{ms}})$, and $\check{W}[t_k^{\mathrm{ms}}] = W[t_k^{\mathrm{ms}}] \setminus \bar{W}(t_k^{\mathrm{ms}})$, equation (B.10) can be further rewritten as

$$\begin{aligned}\int_{\mathcal{X}} p(x_l(t)|x_l(t_k^{\mathrm{ms}}))p(\bar{X}(t_k^{\mathrm{ms}})|Z[t_{k-1}^{\mathrm{ms}}],\bar{Z}(t_k^{\mathrm{ms}}),\bar{W}(t_k^{\mathrm{ms}}),W[t_{k-1}^{\mathrm{ms}}])\times \\ p(\check{X}(t_k^{\mathrm{ms}})|\check{Z}[t_k^{\mathrm{ms}}],\check{W}[t_k^{\mathrm{ms}}])\mathrm{d}X(t_k^{\mathrm{ms}}).\end{aligned} \tag{B.11}$$

In (B.11), term $p(\bar{X}(t_k^{\mathrm{ms}})|Z[t_{k-1}^{\mathrm{ms}}],\bar{Z}(t_k^{\mathrm{ms}}),\bar{W}(t_k^{\mathrm{ms}}),W[t_{k-1}^{\mathrm{ms}}])$ can be written as [similar to (3.12)]

$$\frac{p(\bar{Z}(t_k^{\mathrm{ms}})|\bar{X}(t_k^{\mathrm{ms}}),\bar{W}(t_k^{\mathrm{ms}}))p(\bar{X}(t_k^{\mathrm{ms}})|Z[t_{k-1}^{\mathrm{ms}}],W[t_{k-1}^{\mathrm{ms}}])}{\int_{\bar{\mathcal{X}}(t_k^{\mathrm{ms}})} p(\bar{Z}(t_k^{\mathrm{ms}})|\bar{X}(t_k^{\mathrm{ms}}),\bar{W}(t_k^{\mathrm{ms}}))p(\bar{X}(t_k^{\mathrm{ms}})|Z[t_{k-1}^{\mathrm{ms}}],W[t_{k-1}^{\mathrm{ms}}])\mathrm{d}\bar{X}(t_k^{\mathrm{ms}})}. \tag{B.12}$$

Likewise, we can write term $p(\check{X}(t_k^{\mathrm{ms}})|\check{Z}[t_k^{\mathrm{ms}}],\check{W}[t_k^{\mathrm{ms}}])$ in the integral of (B.10) as

$$\frac{p(\check{Z}(t_k^{\mathrm{ms}})|\check{X}(t_k^{\mathrm{ms}}),\check{W}(t_k^{\mathrm{ms}}))p(\check{X}(t_k^{\mathrm{ms}})|Z[t_{k-1}^{\mathrm{ms}}],W[t_{k-1}^{\mathrm{ms}}])}{\int_{\check{\mathcal{X}}(t_k^{\mathrm{ms}})} p(\check{Z}(t_k^{\mathrm{ms}})|\check{X}(t_k^{\mathrm{ms}}),\check{W}(t_k^{\mathrm{ms}}))p(\check{X}(t_k^{\mathrm{ms}})|Z[t_{k-1}^{\mathrm{ms}}],W[t_{k-1}^{\mathrm{ms}}])\mathrm{d}\check{X}(t_k^{\mathrm{ms}})}, \tag{B.13}$$

where $\check{Z}(t_k^{\mathrm{ms}}) = Z(t_k^{\mathrm{ms}}) \setminus \bar{Z}(t_k^{\mathrm{ms}})$ and $\check{W}(t_k^{\mathrm{ms}}) = W(t_k^{\mathrm{ms}}) \setminus \bar{W}(t_k^{\mathrm{ms}})$. Note that

$$\begin{aligned}p(Z(t_k^{\mathrm{ms}})|X(t_k^{\mathrm{ms}}),W(t_k^{\mathrm{ms}})) &= p(\bar{Z}(t_k^{\mathrm{ms}}),\check{Z}(t_k^{\mathrm{ms}})|X(t_k^{\mathrm{ms}}),W(t_k^{\mathrm{ms}})) \\ &\overset{(b)}{=} p(\bar{Z}(t_k^{\mathrm{ms}})|X(t_k^{\mathrm{ms}}),W(t_k^{\mathrm{ms}}))p(\check{Z}(t_k^{\mathrm{ms}})|X(t_k^{\mathrm{ms}}),W(t_k^{\mathrm{ms}})) \\ &\overset{(c)}{=} p(\bar{Z}(t_k^{\mathrm{ms}})|X(t_k^{\mathrm{ms}}),W(t_k^{\mathrm{ms}}))p(\check{Z}(t_k^{\mathrm{ms}})|\check{X}(t_k^{\mathrm{ms}}),W(t_k^{\mathrm{ms}})),\end{aligned} \tag{B.14}$$

where (b) holds with the independence of the measurement noises in (3.4), and (c) follows from condition (3.18) in Definition 3.2. The product of the first items in the numerators of (B.13) and (B.14) is $p(Z(t_k^{\mathrm{ms}})|X(t_k^{\mathrm{ms}})) = F(X(t_k^{\mathrm{ms}}))$. Letting

$$G_2(X(t_k^{\mathrm{ms}})) = p(\bar{X}(t_k^{\mathrm{ms}})|Z[t_{k-1}^{\mathrm{ms}}])p(X(t_k^{\mathrm{ms}}) \setminus \bar{X}(t_k^{\mathrm{ms}})|Z[t_{k-1}^{\mathrm{ms}}]), \tag{B.15}$$

i.e., the product of the second items in the numerators of (B.13) and (B.14), we can write the estimation $\mathbb{E}_{y_l(t)}^{\bar{X}(t)}\left[y_l(t)|Z[t_{k-1}^{\mathrm{ms}}],\bar{Z}(t_k^{\mathrm{ms}}),W[t_{k-1}^{\mathrm{ms}}],\bar{W}(t_k^{\mathrm{ms}})\right]$ as

$$\begin{aligned}&\int_{\mathcal{X}_l} g_l(x_l(t))\left[\int_{\mathcal{X}} p(x_l(t)|x_l(t_k^{\mathrm{ms}}))FG_2(X(t_k^{\mathrm{ms}}))\mathrm{d}X(t_k^{\mathrm{ms}})\right]\mathrm{d}x_l(t) \\ &= \int_{\mathcal{X}} \mathbb{E}_{x_l(t)}\left[g_l(x_l(t))|x_l(t_k^{\mathrm{ms}})\right]FG_2(X(t_k^{\mathrm{ms}}))\mathrm{d}X(t_k^{\mathrm{ms}}),\end{aligned} \tag{B.16}$$

where

$$FG_2(X(t_k^{\mathrm{ms}})) := \frac{F(X(t_k^{\mathrm{ms}}))G_2(X(t_k^{\mathrm{ms}}))}{\int_{\mathcal{X}} F(X(t_k^{\mathrm{ms}}))G_2(X(t_k^{\mathrm{ms}}))\mathrm{d}X(t_k^{\mathrm{ms}})}. \tag{B.17}$$

We can see that (B.8) and (B.16) have a similar structure. Actually, the only difference comes from $G_1(X(t_k^{\mathrm{ms}}))$ and $G_2(X(t_k^{\mathrm{ms}}))$.

iii) Subtracting (B.16) from (B.8), we have

$$\begin{aligned}&\mathbb{E}_{y_l(t)}\left[y_l(t)|Z[t_k^{\mathrm{ms}}],W[t_k^{\mathrm{ms}}]\right] - \mathbb{E}^{\bar{X}(t)}_{y_l(t)}\left[y_l(t)|Z[t_{k-1}^{\mathrm{ms}}],\bar{Z}(t_k^{\mathrm{ms}}),W[t_{k-1}^{\mathrm{ms}}],\bar{W}(t_k^{\mathrm{ms}})\right] = \\ &\quad\int_{\mathcal{X}} \mathbb{E}_{x_l(t)}\left[g_l(x_l(t))|x_l(t_k^{\mathrm{ms}})\right]F(X(t_k^{\mathrm{ms}}))\frac{H_2G_1(X(t_k^{\mathrm{ms}})) - H_1G_2(X(t_k^{\mathrm{ms}}))}{H_1H_2}\mathrm{d}X(t_k^{\mathrm{ms}}),\end{aligned} \tag{B.18}$$

where

$$H_1 = \int_{\mathcal{X}} F(X(t_k^{\mathrm{ms}}))G_1(X(t_k^{\mathrm{ms}}))\mathrm{d}X(t_k^{\mathrm{ms}}), \quad H_2 = \int_{\mathcal{X}} F(X(t_k^{\mathrm{ms}}))G_2(X(t_k^{\mathrm{ms}}))\mathrm{d}X(t_k^{\mathrm{ms}}). \tag{B.19}$$

Note that $H_1, H_2 > 0$. We split (B.18) into two parts, i.e.,

$$\mathbb{E}_{y_l(t)}\left[y_l(t)|Z[t_k^{\mathrm{ms}}],W[t_k^{\mathrm{ms}}]\right] - \mathbb{E}^{\bar{X}(t)}_{y_l(t)}\left[y_l(t)|Z[t_{k-1}^{\mathrm{ms}}],\bar{Z}(t_k^{\mathrm{ms}}),W[t_{k-1}^{\mathrm{ms}}],\bar{W}(t_k^{\mathrm{ms}})\right] = A + B, \tag{B.20}$$

where

$$A = \int_{\mathcal{X}} \mathbb{E}_{x_l(t)}\left[g_l(x_l(t))|x_l(t_k^{\mathrm{ms}})\right]F(X(t_k^{\mathrm{ms}}))\frac{G_1(X(t_k^{\mathrm{ms}})) - G_2(X(t_k^{\mathrm{ms}}))}{H_1}\mathrm{d}X(t_k^{\mathrm{ms}}), \tag{B.21}$$

$$B = \frac{H_2 - H_1}{H_1H_2}\int_{\mathcal{X}} \mathbb{E}_{x_l(t)}\left[g_l(x_l(t))|x_l(t_k^{\mathrm{ms}})\right]F(X(t_k^{\mathrm{ms}}))G_1(X(t_k^{\mathrm{ms}}))\mathrm{d}X(t_k^{\mathrm{ms}}). \tag{B.22}$$

In the rest of this proof, we find the upper bounds for $\|A\|$ and $\|B\|$, respectively, so that the following can be upper bounded

$$\left\|\mathbb{E}_{y_l(t)}[y_l(t)|Z[t_k^{\mathrm{ms}}],W[t_k^{\mathrm{ms}}]] - \mathbb{E}^{\bar{X}(t)}_{y_l(t)}[y_l(t)|Z[t_{k-1}^{\mathrm{ms}}],\bar{Z}(t_k^{\mathrm{ms}}),W[t_{k-1}^{\mathrm{ms}}],\bar{W}(t_k^{\mathrm{ms}})]\right\| \le \|A\| + \|B\|. \tag{B.23}$$

For A, we have

$$\begin{aligned}\|A\| &= \left\|\int_{\mathcal{X}} \mathbb{E}_{x_l(t)}[g_l(x_l(t))|x_l(t_k^{\mathrm{ms}})]F(X(t_k^{\mathrm{ms}}))\frac{G_1(X(t_k^{\mathrm{ms}})) - G_2(X(t_k^{\mathrm{ms}}))}{H_1}\mathrm{d}X(t_k^{\mathrm{ms}})\right\| \\ &\overset{(d)}{\le} \int_{\mathcal{X}} \left\|\mathbb{E}_{x_l(t)}[g_l(x_l(t))|x_l(t_k^{\mathrm{ms}})])\right\| F(X(t_k^{\mathrm{ms}}))\left|\frac{G_1(X(t_k^{\mathrm{ms}})) - G_2(X(t_k^{\mathrm{ms}}))}{H_1}\right|\mathrm{d}X(t_k^{\mathrm{ms}}) \\ &= \left\|\left\|\mathbb{E}_{x_l(t)}[g_l(x_l(t))|x_l(t_k^{\mathrm{ms}})]\right\| F(X(t_k^{\mathrm{ms}}))\frac{G_1(X(t_k^{\mathrm{ms}})) - G_2(X(t_k^{\mathrm{ms}}))}{H_1}\right\|_1 \\ &\overset{(e)}{=} \left\|A_1(X(t_k^{\mathrm{ms}}))\frac{G_1(X(t_k^{\mathrm{ms}})) - G_2(X(t_k^{\mathrm{ms}}))}{H_1}\right\|_1,\end{aligned} \tag{B.24}$$

where (d) follows from the "triangle inequality"[1] and $F(X(t_k^{\text{ms}})) \geq 0$. For (e), it holds with $A_1(X(t_k^{\text{ms}})) = \left\| \mathbb{E}_{x_l(t)}[g_l(x_l(t))|x_l(t_k^{\text{ms}})] \right\| F(X(t_k^{\text{ms}}))$. With Hölder's inequality, (B.24) can be zoomed as

$$\|A\| \leq \|A_1(X(t_k^{\text{ms}}))\|_1 \, \|G_1(X(t_k^{\text{ms}})) - G_2(X(t_k^{\text{ms}}))\|_\infty. \tag{B.25}$$

From (3.23), we know that $\left\|A_1(X(t_k^{\text{ms}}))\right\|_1$ is bounded. Since the cut gap defined in (3.22) is not greater than δ, inequality (B.25) can be rewritten as

$$|A| \leq \|A_1(X(t_k^{\text{ms}}))\|_1 \, \delta. \tag{B.26}$$

For B, from (B.22), we have

$$\begin{aligned}
\|B\| &= \frac{|H_2 - H_1|}{H_2} \left\| \int_{\mathcal{X}} \mathbb{E}_{x_l(t)}[g_l(x_l(t))|x_l(t_k^{\text{ms}})] p(X(t_k^{\text{ms}})|Z[t_k^{\text{ms}}]) \mathrm{d}X(t_k^{\text{ms}}) \right\| \\
&\leq \frac{|H_2 - H_1|}{H_2} \int_{\mathcal{X}} \left\| \mathbb{E}_{x_l(t)}[g_l(x_l(t))|x_l(t_k^{\text{ms}})] \right\| p(X(t_k^{\text{ms}})|Z[t_k^{\text{ms}}]) \mathrm{d}X(t_k^{\text{ms}}) \\
&\overset{(f)}{=} \frac{|H_2 - H_1|}{H_2} \int_{\mathcal{X}} \left\| \int_{\mathcal{X}_l} g(x_l(t)) p(x_l(t)|x_l(t_k^{\text{ms}})) \mathrm{d}x_l(t) \right\| p(X(t_k^{\text{ms}})|Z[t_k^{\text{ms}}]) \mathrm{d}X(t_k^{\text{ms}}) \\
&\leq \frac{|H_2 - H_1|}{H_2} \int_{\mathcal{X}} \left[\int_{\mathcal{X}_l} \|g(x_l(t))\| \, p(x_l(t)|x_l(t_k^{\text{ms}})) \mathrm{d}x_l(t) \right] p(X(t_k^{\text{ms}})|Z[t_k^{\text{ms}}]) \mathrm{d}X(t_k^{\text{ms}}) \\
&\overset{(g)}{=} \frac{|H_2 - H_1|}{H_2} \int_{\mathcal{X}_l} \|g(x_l(t))\| \, p(X(t)|Z[t_k^{\text{ms}}]) \mathrm{d}x_l(t) \\
&\overset{(h)}{=} V \frac{|H_2 - H_1|}{H_2},
\end{aligned} \tag{B.27}$$

where (f) follows from (3.25), and (g) is based on the interchange of the order of integration. According to (3.9), we set

$$\int_{\mathcal{X}_l} \|g(x_l(t))\| \, p(X(t)|Z[t_k^{\text{ms}}]) \mathrm{d}x_l(t) = \int_{\mathcal{X}} \|g_l(x_l(t))\| p(X(t)|Z[t], W[t]) \mathrm{d}X(t) = V, \tag{B.28}$$

which leads to (h). According to (B.19), we rewrite term $|H_1 - H_2|$ in (B.27) as

$$\begin{aligned}
|H_1 - H_2| &= \left| \int_{\mathcal{X}} F(X(t_k^{\text{ms}})) \left[G_1(X(t_k^{\text{ms}})) - G_2(X(t_k^{\text{ms}})) \right] \mathrm{d}X(t_k^{\text{ms}}) \right| \\
&\overset{(i)}{\leq} \|F(X(t_k^{\text{ms}}))\|_1 \, \|G_1(X(t_k^{\text{ms}})) - G_2(X(t_k^{\text{ms}}))\|_\infty \\
&\leq \|F(X(t_k^{\text{ms}}))\|_1 \, \delta,
\end{aligned} \tag{B.29}$$

[1] To be more specific, we use the inequality $\| \int_{\mathcal{X}} f(x) \mathrm{d}x \| \leq \int_{\mathcal{X}} \|f(x)\| \mathrm{d}x$ which can be regarded as using the triangle inequality on infinite sum of infinitesimals. Since there is no name for this inequality, we use "triangle inequality" as its name.

where (i) follows from Hölder's inequality. Thus, (B.27) can be rewritten as

$$\|B\| \leq V \frac{\left\|F(X(t_k^{\text{ms}}))\right\|_1}{H_2} \delta. \tag{B.30}$$

Note that $\left\|F(X(t_k^{\text{ms}}))\right\|_1$ is bounded by (3.24). Combing (B.23), (B.24), and (B.30), we have

$$\begin{aligned}\left\|\mathbb{E}_{y_l(t)}[y_l(t)|Z[t_k^{\text{ms}}], W[t_k^{\text{ms}}]] - \mathbb{E}^{\bar{X}(t)}_{y_l(t)}[y_l(t)|Z[t_{k-1}^{\text{ms}}], \bar{Z}(t_k^{\text{ms}}), W[t_{k-1}^{\text{ms}}], \bar{W}(t_k^{\text{ms}})]\right\| \\ \leq \left[\|A_1(X(t_k^{\text{ms}}))\|_1 + V \frac{\left\|F(X(t_k^{\text{ms}}))\right\|_1}{H_2}\right] \delta.\end{aligned} \tag{B.31}$$

Therefore, $\forall \varepsilon > 0$, there exists a

$$\delta \leq \frac{H_2}{H_2 \left\|A_1(X(t_k^{\text{ms}}))\right\|_1 + V \left\|F(X(t_k^{\text{ms}}))\right\|_1} \varepsilon, \tag{B.32}$$

such that if the cut gap is bounded by δ, then the estimation gap defined in (3.14) is bounded by ε.

B.3 Communication Settings in Section 3.6.1.3

The transmit power of non-anchor node $l \in \mathcal{L}$ is $P_l^{\text{tr}}(t)$ with maximum value $P_{\max}^{\text{tr}} = 0.2\text{W}$, and the power of noise (from the receiver side) is $P_{\text{noise}} = -90\text{dBm}$. The relationship between the received and transmit powers at time $t \in [0,\ 50]$ is

$$P_{j,l}^{\text{rc}}(t) = \frac{h_{l,j}(t) P_l^{\text{tr}}(t)}{1 + D_{l,j}^{\alpha}(t)}, \tag{B.33}$$

where $h_{l,j}(t)$ is the fading gain with Nakagami-m model (see [64]) and $m = 2$. Notation $D_{l,j}(t)$ is the distance between nodes l and j, and α is the path loss exponent. In this problem, we assume $\alpha = 3$.

Note that there are 20 non-anchor nodes sharing the same bandwidth, and each broadcast can generate interference. To mitigate the interference, we use the CSMA technique. The threshold for detecting an idle channel is determined by the following equation

$$\theta = \frac{P_{\max}^{\text{tr}}}{1 + D_*^{\alpha}}, \tag{B.34}$$

where parameter D_* is the silence distance that if a node is outside the range covered by this distance, then the transmission interference can be neglected. We set $D_* = 2\text{m}$ in this problem, which means the node only regards the communication signals within 2 meters as interference. If the received power is lower than θ, then a node assumes this channel is idle for broadcasting,

and it uses the maximum transmit power to transmit (on-off power control), otherwise the node would wait for a random time follows the exponential distribution with mean $\mu = 0.05$s.

For message receiving, the SINR for node l receiving the message from node j is

$$\mathrm{SINR}_{l,j}(t) = \frac{P^{\mathrm{rc}}_{l,j}(t)}{\sum_{i \neq j} P^{\mathrm{rc}}_{l,i}(t) + P_{\mathrm{noise}}}. \tag{B.35}$$

The channel is complex Gaussian, and the Shannon's capacity is

$$C_{l,j}(t) = B \log_2 \left(1 + \mathrm{SINR}_{l,j}(t)\right), \tag{B.36}$$

where the bandwidth is $B = 40$MHz. The broadcasting rate for node $j \in \mathcal{L}$ is $R^{\mathrm{bc}}_j = 100\mathrm{KB/s} = 800\mathrm{Kb/s}$. If the average capacity for a broadcast message is lower than the broadcasting rate, then a transmission outage occurs. Specifically, an outage happens when

$$\frac{1}{t^{\mathrm{et}}_{j,k} - t^{\mathrm{st}}_{j,k}} \int_{t^{\mathrm{st}}_{j,k}}^{t^{\mathrm{et}}_{j,k}} C_{l,j}(t)\mathrm{d}t < R^{\mathrm{bc}}_j, \tag{B.37}$$

where $t^{\mathrm{st}}_{j,k}$ and $t^{\mathrm{et}}_{j,k}$ the start and end times, respectively, of broadcast message $m^{\mathrm{bc}}_{j,k}$ (see Fig. 3.2). The broadcasting time length $t^{\mathrm{et}}_{j,k} - t^{\mathrm{st}}_{j,k}$ depends on the packet size of $m^{\mathrm{bc}}_{j,k}$, i.e.,

$$t^{\mathrm{et}}_{j,k} - t^{\mathrm{st}}_{j,k} = \frac{\mathrm{size}(m^{\mathrm{bc}}_{j,k})}{R^{\mathrm{bc}}_j}. \tag{B.38}$$

The packet size is determined by the content in the broadcasting message, where the real numbers (e.g., for describing the priors, posteriors, and measurements) are stored in double precision numbers with 64bit size.

Appendix C

C.1 Proof of Theorem 4.1

Before starting the proof, we give a brief flow chart of Algorithm 4.1 in Fig. C.1. With this figure, we can clearly see the flow of Algorithm 4.1: the algorithm starts from **s** and ends at three possible terminals **b**, **d**, and **e** (more details can be found in the caption). We divide the proof into several cases according to Fig. C.1 which is shown as follows.

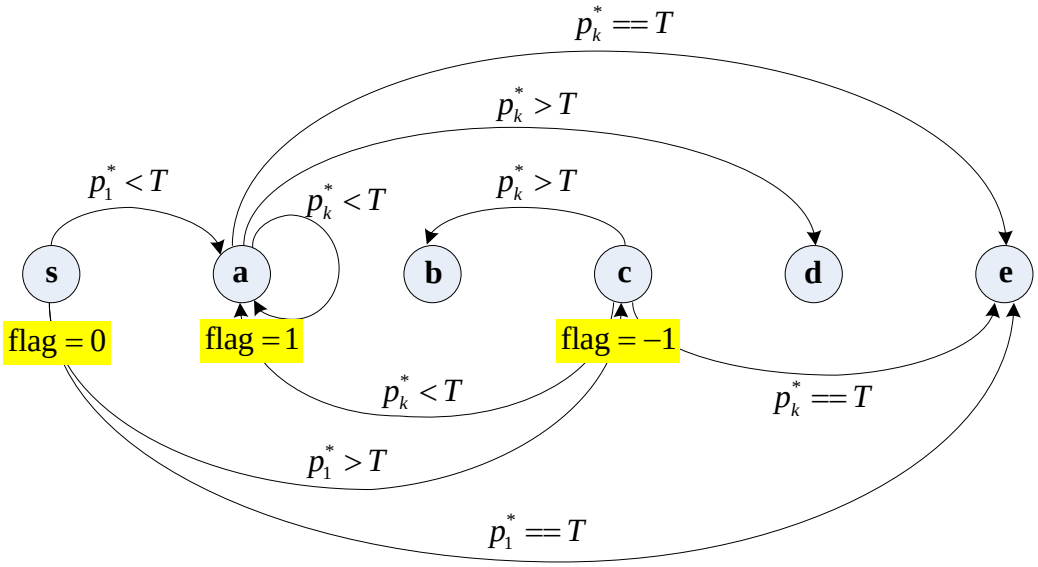


Figure C.1: A brief version of flow chart for Algorithm 4.1. The nodes with **a**, **b**, **c**, **d**, **e** represent Lines 5-8, Line 10, Line 12, Line 14, Line 16 in Algorithm 4.1, respectively, and the node with **s** stands for the starting point of Algorithm 4.1. For **s**, the value of flag is 0. After arriving at node **a**, the value of flag becomes 1. After arriving at node **c**, the value of flag becomes -1. There are three possible terminals corresponding to nodes **b**, **d**, and **e**.

1) $p_1^* = T$. In this case, the program directly goes from node **s** to **e**, and we can easily get $\delta_T(\mu_{[T]}) = Q_{0,k}^{(1)}/Q_{0,1}^{(1)} = Q_{0,K+1}^{(1)}/Q_{0,1}^{(1)} = \delta_T(\mu_{[T]})$ in Line 20, which means Algorithm 4.1 returns the correct result. Note that $K = 1$.

2) $p_1^* < T$. Initially, the program goes from **s** to **a**. Then (for $k = 2$), it has three possible destinations, i.e., nodes **a**, **d**, and **e**. However, the program cannot always stay in node **a**, and it must end either at **d** or **e**. This is because p_k^* at least increases by 1 for each time arriving at **a** (recall that **a** corresponds to Lines 5-8). For Line 5, if $\lfloor T/p_k^* \rfloor > 1$ or $\rho > 0$, then $\mathbf{Q}_{0,k+1} \succ \mathbf{Q}_{0,k}\delta_{p_k^*}(\mu_{\lfloor p_k^* \rfloor,k})$ (due to $\mu_{[T]} \succ 0$), which implies $\mathbf{Q}_{0,k+1}$ cannot be cleared within p_k^* time slots, and therefore $p_{k+1}^* \geq p_k^* + 1$. Similarly, if $\lfloor T/p_k^* \rfloor == 1$ and $\rho == 0$, Line 7 returns $\mathbf{Q}_{0,k+1} \succ \mathbf{Q}_{0,k}\delta_{p_k^*}(\mu_{\lfloor p_k^* \rfloor,k})$, and $p_{k+1}^* \geq p_k^* + 1$ still holds. Hence, the program must stop at node **d** or **e**, and the number of iterations is upper bounded by $K \leq T - p_1^* + 1$, i.e., at least goes to **e** (corresponding to $p_K^* = T$).

2-1) Ends at node **d**. From Line 5, we know that $p_k^* \leq T$ always holds, because the corresponding $\mathbf{Q}_{0,k+1}$ can always be cleared. Thus, if the program goes to node **d** ($p_k^* > T$), Line 7 must have run, i.e., the increment εT makes $p_k^* > T$, where $k = K$. This means that $\mathbf{Q}_{0,K-1}$ in the $(K-1)^{\text{th}}$ iteration corresponds to the maximum data queue can be cleared within T time slots. Then, subtracting the increment εT from the current data queue, we can derive $\mathbf{Q}_{0,K+1} = \mathbf{Q}_{0,K-1}$. Since $\mathbf{Q}_{0,K+1}$ is the maximum data queue that can be cleared within T time slots in the direction of $\mu_{[T]}$, we have $\mathbf{Q}_{0,K+1} = \mu_{[T]}\delta_T(\mu_{[T]})T$. Observe that $\mathbf{Q}_{0,1} = \mu_{[T]}T$, we have $\delta_T(\mu_{[T]}) = Q_{0,k}^{(1)}/Q_{0,1}^{(1)} = Q_{0,K+1}^{(1)}/Q_{0,1}^{(1)} = \delta_T(\mu_{[T]})$ in Line 20, which means Algorithm 4.1 returns the correct result.

2-2) Ends at node **e**. In this case, we directly have $\mathbf{Q}_{0,K+1} = \mu_{[T]}\delta_T(\mu_{[T]})T$, and Line 20 returns the correct rate margin similar to that in 2-1).

3) $p_1^* > T$. Initially, the program goes from **s** to **c**. Then (for $k = 2$), it has three possible destinations, i.e., nodes **a**, **b**, and **e**.

3-1) Goes to node **a**. This case is similar to 2-1): Algorithm 4.1 returns the rate margin correctly, and the iteration number is upper bounded by $K \leq T - p_2^* + 2$ (the program reaches node **a** for $k = 2$ rather than $k = 1$).

3-2) Ends at node **b**. In this case, the rate margin $\delta_T(\mu_{[T]})$ is zero, since in the last iteration $K-1$, the updated data queue is $\mathbf{Q}_{0,K-1+1} = \mathbf{Q}_{0,K} = \varepsilon T$ returned by Line 12, and the "smallest" rate-tuple $\varepsilon = \mathbf{Q}_{0,K}/T$ (i.e., its component reach the precision of calculation) in the direction of $\mu_{[T]}$ is not achievable.

3-3) Ends at node **e**. This case is similar to 2-2): Algorithm 4.1 returns the rate margin correctly, and the iteration number is upper bounded by $K \leq T - p_2^* + 2$.

C.2 Proof of Theorem 4.2

i) $\forall \mu_{[T]} \in \Lambda_{[T]}$, then the data queue can be cleared with some $p \leq T$, which implies $p^* \leq p \leq T$ holds. Based on $p^* \leq T$, we prove that (4.32) is exactly the rate-achieving policy for $\mu_{[T]}$.

By (4.32), the average rate over T slots is

$$\frac{1}{T}\sum_{t=1}^{T}\frac{\mathbf{Q}_{t-1}^{*}-\mathbf{Q}_{t}^{*}}{L}=\frac{\mathbf{Q}_0}{TL}=\frac{T\mu_{[T]}L}{TL}=\mu_{[T]}, \tag{C.1}$$

which means the rate is achieved by rate sequence $\left((\mathbf{Q}_{t-1}^{*}-\mathbf{Q}_{t}^{*})/L\right)_{t=1}^{p^*}$. Additionally, since the following holds for every $t\in\{1,\ldots,p^*\}$

$$\frac{\mathbf{Q}_{t-1}^{*}-\mathbf{Q}_{t}^{*}}{L}\preceq\mu_{\max}(\mathbf{s}_t,L,\varepsilon), \tag{C.2}$$

the maximum rate constraints (see Definition 4.4) are satisfied. Therefore, $\mu_{[T]}$ can be achieved by the policy $\mathcal{P}_T$.

ii) $\forall\mu_{[T]}\notin\Lambda_{[T]}$, it follows from Corollary 4.2 that $p^*>T$.

C.3 Proof of Proposition 4.5

$\forall\mathbf{Q}_t$, let $\bar{\mathbf{s}}_k=\left[\bar{s}_k^{(1)},\ldots,\bar{s}_k^{(n)}\right]$, $k\in\{t+1,\ldots,p\}$ be any possible action (transmit power) from $\mathbf{Q}_{k-1}$. We then divide $E^I(\mathbf{Q}_t)$ into two parts to prove the admissibility, i.e., $p-1-t$ and $\max\limits_{n\in\mathcal{N}}\left[Q_0^{(n)}/\sum_{t=1}^{p}\mu_{\max}^{(n)}(\gamma_n(\mathbf{s}_t),L,\varepsilon)L\right]$. For the first part, $\forall n\in\mathcal{N}$, we have

$$p-1-t=\sum_{k=t+1}^{p-1}1\geq\sum_{k=t+1}^{p-1}\frac{Q_{k-1}^{(n)}-Q_k^{(n)}}{\mu_{\max}^{(n)}(\gamma_n(\bar{\mathbf{s}}_k),L,\varepsilon)L}. \tag{C.3}$$

Additionally, since $\bar{s}_k^{(n)}\leq s_{\max}^{(n)}$, the following holds

$$\gamma_n(\bar{\mathbf{s}}_k)=\frac{h_{nn}\bar{s}_k^{(n)}}{W_n+\sum_{m\neq n}h_{mn}\bar{s}_k^{(m)}}\leq\frac{h_{nn}s_{\max}^{(n)}}{W_n}=\gamma_n'(s_{\max}^{(n)}). \tag{C.4}$$

Then, we have $\mu_{\max}^{(n)}(\gamma_n(\bar{\mathbf{s}}_k),L,\varepsilon)\leq\mu_{\max}^{(n)}(\gamma_n'(s_{\max}^{(n)}),L,\varepsilon)$ for all $n\in\mathcal{N}$. Thus, (C.3) can be further bounded as

$$p-1-t\geq\sum_{k=t+1}^{p-1}\frac{Q_{k-1}^{(n)}-Q_k^{(n)}}{\mu_{\max}^{(n)}(\gamma_n(\bar{\mathbf{s}}_k),L,\varepsilon)L}\geq\sum_{k=t+1}^{p-1}\frac{Q_{k-1}^{(n)}-Q_k^{(n)}}{\mu_{\max}^{(n)}(\gamma_n'(s_{\max}^{(n)}),L,\varepsilon)L}, \tag{C.5}$$

for all $n\in\mathcal{N}$, which implies

$$p-1-t\geq\max_{n\in\mathcal{N}}\sum_{k=t+1}^{p-1}\frac{Q_{k-1}^{(n)}-Q_k^{(n)}}{\mu_{\max}^{(n)}(\gamma_n'(s_{\max}^{(n)}),L,\varepsilon)L}=\max_{n\in\mathcal{N}}\left[\frac{Q_t^{(n)}-Q_{p-1}^{(n)}}{\mu_{\max}^{(n)}(\gamma_n'(s_{\max}^{(n)}),L,\varepsilon)L}\right]. \tag{C.6}$$

For the second part, we have for all $n \in \mathcal{N}$

$$\frac{Q_0^{(n)}}{\sum_{t=1}^{p} \mu_{\max}^{(n)}(\gamma_n(\mathbf{s}_t), L, \varepsilon)L} \overset{(a)}{\geq} \frac{Q_{p-1}^{(n)}}{\mu_{\max}^{(n)}(\gamma_n(\mathbf{s}_p), L, \varepsilon)L} \overset{(b)}{\geq} \frac{Q_{p-1}^{(n)}}{\mu_{\max}^{(n)}(\gamma_n'(s_{\max}^{(n)}), L, \varepsilon)L}, \tag{C.7}$$

where, for $Q_{p-1}^{(n)} \neq 0$, inequality (a) holds with $Q_0^{(n)} = \sum_{t=1}^{p}(Q_{t-1}^{(n)} - Q_t^{(n)})$ and $\sum_{t=1}^{p-1}(Q_{t-1}^{(n)} - Q_t^{(n)}) = \sum_{t=1}^{p-1} \mu_{\max}^{(n)}(\gamma_n(\mathbf{s}_t), L, \varepsilon)L$. For $Q_{p-1}^{(n)} = 0$, inequality (a) is satisfied by following that $Q_0^{(n)} / \sum_{t=1}^{p} \mu_{\max}^{(n)}(\gamma_n(\mathbf{s}_t), L, \varepsilon)L$ is nonnegative. Additionally, inequality (b) holds with $\gamma_n(\mathbf{s}_p) \leq \gamma_n'(s_{\max}^{(n)})$. From (C.7), we have

$$\max_{n \in \mathcal{N}} \left[\frac{Q_0^{(n)}}{\sum_{t=1}^{p} \mu_{\max}^{(n)}(\gamma_n(\mathbf{s}_t), L, \varepsilon)L} \right] \geq \max_{n \in \mathcal{N}} \left[\frac{Q_{p-1}^{(n)}}{\mu_{\max}^{(n)}(\gamma_n'(s_{\max}^{(n)}), L, \varepsilon)L} \right], \tag{C.8}$$

which, added by (C.6), implies $E^I(\mathbf{Q}_t) \leq E^*(\mathbf{Q}_t)$ holds.

Bibliography

[1] S. Mumtaz, K. M. S. Huq, M. I. Ashraf, J. Rodriguez, V. Monteiro, and C. Politis, "Cognitive vehicular communication for 5G," *IEEE Commun. Mag.*, vol. 53, no. 7, pp. 109–117, Jul. 2015. (cited on page 1)

[2] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, May 2016. (cited on page 1)

[3] J. Andrews, S. Buzzi, W. Choi, S. Hanly, A. Lozano, A. Soong, and J. Zhang, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014. (cited on page 1)

[4] M. Haenggi, "Local delay in static and highly mobile Poisson networks with ALOHA," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, Cap Town, South Africa, May 2010, pp. 1–5. (cited on pages 2, 3, 11, and 12)

[5] M. Haenggi, J. Andrews, F. Baccelli, O. Dousse, and M. Franceschetti, "Stochastic geometry and random graphs for the analysis and design of wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 7, pp. 1029–1046, Sept. 2009. (cited on pages 2, 11, and 12)

[6] Z. Gong and M. Haenggi, "Interference and outage in mobile random networks: Expectation, distribution, and correlation," *IEEE Trans. Mobile Comput.*, vol. 13, no. 2, pp. 337–349, 2014. (cited on pages 2, 3, 12, 13, 16, and 24)

[7] A. Simonetto and G. Leus, "Distributed maximum likelihood sensor network localization," *IEEE Trans. Signal Process.*, vol. 62, no. 6, pp. 1424–1437, Mar. 2014. (cited on page 3)

[8] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, vol. 97, no. 2, pp. 427–450, Feb. 2009. (cited on pages 3 and 14)

[9] A. El Gamal and Y.-H. Kim, *Network information theory*. Cambridge university press, 2011. (cited on page 3)

[10] Y. Polyanskiy, H. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010. (cited on pages 3, 10, and 75)

[11] M. Neely, E. Modiano, and C. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005. (cited on pages 4 and 15)

[12] M. Haenggi, *Stochastic Geometry for Wireless Networks*. Cambridge Univ., 2012. (cited on pages 4, 5, and 11)

[13] F. Baccelli and B. Blaszczyszyn, *Stochastic Geometry and Wireless Networks*. NOW: Foundations and Trends in Networking, 2009. (cited on pages 4 and 11)

[14] ——, *Stochastic Geometry and Wireless Networks-Volume II: Applications*. now Publishers Inc., 2009. (cited on page 4)

[15] T. S. Rappaport, *Wireless communications: principles and practice*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice Hall, 2002. (cited on page 6)

[16] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. New York, NY, USA: Academic Press, 1970. (cited on pages 9 and 52)

[17] B. Anderson and J. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1979. (cited on page 9)

[18] S. Särkkä, *Bayesian filtering and smoothing*. Cambridge University Press, 2013, vol. 3. (cited on pages 9, 14, and 52)

[19] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948. (cited on page 10)

[20] T. M. Cover and J. A. Thomas, *Elements of information theory*, 2nd ed. Hoboken, NJ: Wiley-Interscience, 2006. (cited on pages 10 and 25)

[21] M. Haenggi and R. K. Ganti, *Interference in large wireless networks*. NOW: Foundations and Trends in Networking, 2009. (cited on pages 11 and 12)

[22] R. Tanbourgi, H. Dhillon, J. Andrews, and F. Jondral, "Effect of spatial interference correlation on the performance of maximum ratio combining," *IEEE Trans. Wireless Commun.*, vol. 13, no. 6, pp. 3307–3316, Jun. 2014. (cited on pages 11 and 12)

[23] S. Srinivasa and M. Haenggi, "Distance distributions in finite uniformly random networks: Theory and applications," *IEEE Trans. Veh. Technol.*, vol. 59, no. 2, pp. 940–949, Feb. 2010. (cited on pages 11 and 12)

[24] J. Guo, S. Durrani, and X. Zhou, "Outage probability in arbitrarily-shaped finite wireless networks," *IEEE Trans. Commun.*, vol. 62, no. 2, pp. 699–712, Feb. 2014. (cited on pages 11 and 12)

[25] R. K. Ganti and M. Haenggi, "Spatial and temporal correlation of the interference in aloha ad hoc networks," *IEEE Commun. Lett.*, vol. 13, no. 9, pp. 631–633, Sept. 2009. (cited on pages 12 and 13)

[26] U. Schilcher, C. Bettstetter, and G. Brandner, "Temporal correlation of interference in wireless networks with rayleigh block fading," *IEEE Trans. Mobile Comput.*, vol. 11, no. 12, pp. 2109–2120, Dec. 2012. (cited on pages 12 and 13)

[27] M. Haenggi, "Diversity loss due to interference correlation," *IEEE Commun. Lett.*, vol. 16, no. 10, pp. 1600–1603, Oct. 2012. (cited on pages 12 and 13)

[28] A. Crismani, S. Toumpis, U. Schilcher, G. Brandner, and C. Bettstetter, "Cooperative relaying under spatially and temporally correlated interference," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4655–4669, Oct. 2015. (cited on pages 12 and 13)

[29] U. Schilcher, S. Toumpis, A. Crismani, G. Brandner, and C. Bettstetter, "How does interference dynamics influence packet delivery in cooperative relaying?" in *Proc. ACM/IEEE Int. Conf. Modeling, Anal. and Simulation of Wireless and Mobile Syst.* New York, NY, USA: ACM, Nov. 2013, pp. 347–354. (cited on pages 12 and 13)

[30] K. Gulati, R. Ganti, J. Andrews, B. Evans, and S. Srikanteswara, "Characterizing decentralized wireless networks with temporal correlation in the low outage regime," *IEEE Trans. Wireless Commun.*, vol. 11, no. 9, pp. 3112–3125, Sept. 2012. (cited on pages 12 and 13)

[31] Z. Gong and M. Haenggi, "The local delay in mobile poisson networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4766–4777, Sept. 2013. (cited on page 12)

[32] ——, "Temporal correlation of the interference in mobile random networks," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, Kyoto, Japan, Jun. 2011, pp. 1–5. (cited on pages 12 and 13)

[33] ——, "Mobility and fading: Two sides of the same coin," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Miami, FL, Dec. 2010, pp. 1–5. (cited on pages 12, 16, and 24)

[34] C. Bettstetter, G. Resta, and P. Santi, "The node distribution of the random waypoint mobility model for wireless ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 2, no. 3, pp. 257–269, 2003. (cited on page 12)

[35] F. Bai and A. Helmy, *A survey of mobility models*, ser. Wireless Ad-Hoc Networks. Los Angeles, CA, USA: University of Southern California, June 2004, vol. 206, ch. 1, pp. 1–30. (cited on pages 13, 16, 24, and 45)

[36] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, Sept. 2002. (cited on page 13)

[37] H. A. Loeliger, "An introduction to factor graphs," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 28–41, Jan. 2004. (cited on page 14)

[38] F. Meyer, O. Hlinka, H. Wymeersch, E. Riegler, and F. Hlawatsch, "Distributed localization and tracking of mobile networks including noncooperative objects," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 1, pp. 57–71, Mar. 2016. (cited on page 14)

[39] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 809–819, Apr. 2005. (cited on page 14)

[40] V. Savic and S. Zazo, "Reducing communication overhead for cooperative localization using nonparametric belief propagation," *IEEE Wireless Commun. Lett.*, vol. 1, no. 4, pp. 308–311, Aug. 2012. (cited on page 14)

[41] F. Meyer, O. Hlinka, and F. Hlawatsch, "Sigma point belief propagation," *IEEE Signal Process. Lett.*, vol. 21, no. 2, pp. 145–149, Feb. 2014. (cited on page 14)

[42] S. V. de Velde, G. T. F. de Abreu, and H. Steendam, "Improved censoring and nlos avoidance for wireless localization in dense networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 11, pp. 2302–2312, Nov. 2015. (cited on page 14)

[43] C. Pedersen, T. Pedersen, and B. H. Fleury, "A variational message passing algorithm for sensor self-localization in wireless networks," in *Proc. IEEE Int. Symp. Inf. Theory (ISITŠ11)*, Saint Petersburg, Russia, Aug. 2011, pp. 2158–2162. (cited on page 14)

[44] B. Çakmak, D. N. Urup, F. Meyer, T. Pedersen, B. H. Fleury, and F. Hlawatsch, "Cooperative localization for mobile networks: A distributed belief propagation – mean field message passing algorithm," *IEEE Signal Process. Lett.*, vol. 23, no. 6, pp. 828–832, Jun. 2016. (cited on page 14)

[45] B. Denis, J. B. Pierrot, and C. Abou-Rjeily, "Joint distributed synchronization and positioning in uwb ad hoc networks using toa," *IEEE Trans. Microw. Theory Techn.*, vol. 54, no. 4, pp. 1896–1911, Jun. 2006. (cited on page 14)

[46] W. Yuan, N. Wu, B. Etzlinger, H. Wang, and J. Kuang, "Cooperative joint localization and clock synchronization based on gaussian message passing in asynchronous wireless networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7258–7273, Sept. 2016. (cited on page 14)

[47] B. Etzlinger, F. Meyer, F. Hlawatsch, A. Springer, and H. Wymeersch, "Cooperative simultaneous localization and synchronization in mobile agent networks," *IEEE Trans. Signal Process.*, vol. 65, no. 14, pp. 3587 – 3602, Jul. 2017. (cited on page 14)

[48] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992. (cited on page 15)

[49] L. Tassiulas, "Dynamic link activation scheduling in multihop radio networks with fixed or changing connectivity," Ph.D. dissertation, University of Maryland, College Park, MD, USA, 1991. (cited on page 15)

[50] M. Neely and E. Modiano, "Capacity and delay tradeoffs for ad hoc mobile networks," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 1917–1937, Jun. 2005. (cited on page 15)

[51] L. Georgiadis, M. J. Neely, and L. Tassiulas, *Resource Allocation and Cross-Layer Control in Wireless Networks*. Found. Trends Netw., 2006. (cited on pages 15 and 77)

[52] X. Lin, N. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1452–1463, Aug. 2006. (cited on page 15)

[53] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Commun. Netw.*, vol. 3, no. 1, pp. 1–211, 2010. (cited on page 15)

[54] K. Kar, S. Sarkar, A. Ghavami, and X. Luo, "Delay guarantees for throughput-optimal wireless link scheduling," *IEEE Trans. Autom. Control*, vol. 57, no. 11, pp. 2906–2911, Nov. 2012. (cited on page 15)

[55] C. Boyaci and Y. Xia, "Delay analysis of the approximate maximum weight scheduling in wireless networks," in *Proc. of the 9th Int. Wireless Commun. and Mobile Computing Conf. (IWCMC)*, Jul. 2013, pp. 41–46. (cited on page 15)

[56] M. J. Neely, "Delay-based network utility maximization," *IEEE/ACM Trans. Netw.*, vol. 21, no. 1, pp. 41–54, Feb. 2013. (cited on page 15)

[57] D. Xue and E. Ekici, "Delay-guaranteed cross-layer scheduling in multihop wireless networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 6, pp. 1696–1707, Dec. 2013. (cited on page 15)

[58] L. B. Le, E. Modiano, and N. Shroff, "Optimal control of wireless networks with finite buffers," *IEEE/ACM Trans. Netw.*, vol. 20, no. 4, pp. 1316–1329, Aug. 2012. (cited on page 15)

[59] D. Xue and E. Ekici, "Power optimal control in multihop wireless networks with finite buffers," *IEEE Trans. Veh. Technol.*, vol. 62, no. 3, pp. 1329–1339, Mar. 2013. (cited on page 15)

[60] D. Xue, R. Murawski, and E. Ekici, "Capacity achieving distributed scheduling with finite buffers," *IEEE/ACM Trans. Netw.*, vol. 23, no. 2, pp. 519–532, Apr. 2015. (cited on pages 15 and 98)

[61] M. J. Neely, "Opportunistic scheduling with worst case delay guarantees in single and multi-hop networks," in *Proc. IEEE INFOCOM*, San Diego, CA, USA, Apr. 2011, pp. 1728–1736. (cited on page 15)

[62] C. W. Tan, M. Chiang, and R. Srikant, "Fast algorithms and performance bounds for sum rate maximization in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 3, pp. 706–719, Jun. 2013. (cited on page 15)

[63] P. Giaccone, E. Leonardi, and D. Shah, "Throughput region of finite-buffered networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 2, pp. 251–263, Feb. 2007. (cited on page 15)

[64] Y. Cong, X. Zhou, and R. A. Kennedy, "Interference prediction in mobile ad hoc networks with a general mobility model," *IEEE Trans. Wireless Commun.*, vol. 14, no. 8, pp. 4277–4290, Aug. 2015. (cited on pages 17, 45, and 112)

[65] Y. Cong, X. Zhou, B. Zhou, and V. Lau, "Cooperative localization in wireless networks for mobile nodes with asynchronous measurements and communications," Sept. 2017, to be submitted. (cited on page 18)

[66] Y. Cong, X. Zhou, and R. A. Kennedy, "Finite-horizon throughput region for wireless multi-user interference channels," *IEEE Trans. Wireless Commun.*, vol. 16, no. 1, pp. 634–646, Jan. 2017. (cited on page 19)

[67] ——, "Rate-achieving policy in finite-horizon throughput region for multi-user interference channels," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–6. (cited on page 19)

[68] C. T. Chen, *Linear System Theory and Design*, 3rd ed. Oxford Univ., 1999. (cited on pages 24 and 96)

[69] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *J. Basic Eng.*, vol. 83, no. 1, pp. 95–108, 1961. (cited on pages 25 and 101)

[70] L. Jetto, S. Longhi, and G. Venturini, "Development and experimental validation of an adaptive extended kalman filter for the localization of mobile robots," *IEEE Trans. Robot*, vol. 15, no. 2, pp. 219–229, Apr. 1999. (cited on page 25)

[71] M. Mammarella, G. Campa, M. Napolitano, M. Fravolini, Y. Gu, and M. Perhinschi, "Machine vision/gps integration using ekf for the uav aerial refueling problem," *IEEE Trans. Syst., Man, Cybern., A, Syst.,Humans*, vol. 38, no. 6, pp. 791–801, Nov. 2008. (cited on page 25)

[72] D. Ashbrook and T. Starner, "Using gps to learn significant locations and predict movement across multiple users," *Pers. Ubiquitous Comput.*, vol. 7, no. 5, pp. 275–286, Oct. 2003. (cited on page 34)

[73] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proc. 4th Annu. ACM/IEEE Int. Conf. Mobile Comput. and Netw.* New York, USA: ACM, 1998, pp. 85–97. (cited on page 34)

[74] H.-S. Tan and J. Huang, "DGPS-based vehicle-to-vehicle cooperative collision warning: Engineering feasibility viewpoints," *IEEE Trans. Intell.Transp. Syst.*, vol. 7, no. 4, pp. 415–428, Dec. 2006. (cited on page 34)

[75] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong, "On the levy-walk nature of human mobility," *IEEE/ACM Trans. Netw.*, vol. 19, no. 3, pp. 630–643, Jun. 2011. (cited on page 34)

[76] D. J. Higham., "An algorithmic introduction to numerical simulation of stochastic differential equations," *SIAM Rev.*, vol. 43, no. 3, pp. 525–546, Aug. 2001. (cited on page 52)

[77] I. Arasaratnam, S. Haykin, and T. R. Hurd, "Cubature kalman filtering for continuous-discrete systems: Theory and simulations," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 4977–4993, Oct. 2010. (cited on page 52)

[78] L. Murray and A. Storkey, "Particle smoothing in continuous time: A fast approach via density estimation," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 1017–1026, Mar. 2011. (cited on page 52)

[79] Y. Xia, Z. Deng, L. Li, and X. Geng, "A new continuous-discrete particle filter for continuous-discrete nonlinear systems," *Inform. Sci.*, vol. 242, pp. 64 – 75, Sept. 2013. (cited on page 52)

[80] J. G. Siek, L.-Q. Lee, and A. Lumsdaine, *The Boost Graph Library: User Guide and Reference Manual*. Reading, MA, USA: Addison-Wesley Professional, 2001. (cited on page 54)

[81] Y. Xing and R. Chandramouli, "Stochastic learning solution for distributed discrete power control game in wireless data networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 4, pp. 932–944, Aug. 2008. (cited on page 74)

[82] H. Zhang, L. Venturino, N. Prasad, P. Li, S. Rangarajan, and X. Wang, "Weighted sum-rate maximization in multi-cell networks via coordinated scheduling and discrete power control," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 6, pp. 1214–1224, Jun. 2011. (cited on page 74)

[83] V. Y. Tan, "Asymptotic estimates in information theory with non-vanishing error probabilities," *Found. Trends Commun. Inf. Theory*, vol. 11, no. 1-2, pp. 1–184, Sept. 2014. (cited on page 75)

[84] S. Russell, *Artificial intelligence: A modern approach*, 3rd ed. Prentice Hall, 2009. (cited on pages 86, 87, 88, 89, and 90)

[85] C. de Lima, M. Bennis, and M. Latva-aho, "Coordination mechanisms for self-organizing femtocells in two-tier coexistence scenarios," *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 2212–2223, Jun. 2012. (cited on page 96)

[86] ——, "Statistical analysis of self-organizing networks with biased cell association and interference avoidance," *IEEE Trans. Veh. Technol.*, vol. 62, no. 5, pp. 1950–1961, Jun. 2013. (cited on page 96)

[87] D. Ocone and E. Pardoux, "Asymptotic stability of the optimal filter with respect to its initial condition," *SIAM J. Control Optim.*, vol. 34, no. 1, pp. 226–243, 1996. (cited on page 96)

[88] R. van Handel, "Filtering, stability, and robustness," Ph.D. dissertation, California Institute of Technology, 2006. (cited on page 96)

[89] A. R. Liu, "Stochastic observability, reconstructibility, controllability, and reachability," Ph.D. dissertation, University of California, San Diego, USA, 2011. (cited on page 96)

[90] R. van Handel, "Observability and nonlinear filtering," *Probability Theory and Related Fields*, vol. 145, no. 1, pp. 35–74, 2009. (cited on page 96)

[91] K. Xiong, H. Zhang, and C. Chan, "Performance evaluation of ukf-based nonlinear filtering," *Automatica*, vol. 42, no. 2, pp. 261 – 270, Feb. 2006. (cited on page 96)

[92] K. Reif, S. Gunther, E. Yaz, and R. Unbehauen, "Stochastic stability of the discrete-time extended kalman filter," *IEEE Trans. Autom. Control*, vol. 44, no. 4, pp. 714–728, Apr 1999. (cited on page 96)

[93] R. Atar, "Exponential decay rate of the filterŠs dependence on the initial distribution," *The Oxford Handbook of Nonlinear Filtering*, pp. 299–318, 2011. (cited on page 96)

[94] H. Sato, "The capacity of the gaussian interference channel under strong interference," *IEEE Trans. Inf. Theory*, vol. 27, no. 6, pp. 786–788, Nov. 1981. (cited on page 97)

[95] S. Rini, E. Kurniawan, L. Ghaghanidze, and A. Goldsmith, "Energy efficient cooperative strategies for relay-assisted downlink cellular systems," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 11, pp. 2075–2089, Nov. 2014. (cited on page 97)

[96] A. S. Avestimehr, S. N. Diggavi, and D. N. C. Tse, "Wireless network information flow: A deterministic approach," *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 1872–1905, Apr. 2011. (cited on page 97)

[97] G. Bresler and D. Tse, "The two-user gaussian interference channel: a deterministic view," *Eur. Trans. Telecommun.*, vol. 19, no. 4, pp. 333–354, Jun. 2008. (cited on page 97)

[98] K. Zhou, J. C. Doyle, K. Glover *et al.*, *Robust and optimal control*. Prentice Hall New Jersey, 1996. (cited on page 98)

[99] Y. Cong and X. Zhou, "Event-trigger based robust-optimal control for energy harvesting transmitter," *IEEE Trans. Wireless Commun.*, vol. 16, no. 2, pp. 744–756, Feb. 2017. (cited on page 98)

[100] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011. (cited on page 98)

[101] H. Bateman, *Higher Transcendental Functions*. New York: McGrawHill, 1953, vol. 1. (cited on pages 103 and 105)